Group 11 SRS Document

February 8, 2019

Contents

List of Tables

List of Figures

1 Introduction

This document provides the specifications for the software titled **Project Food** (this name is temporary and may change once a decision has been made among the development team). Henceforth, this project shall be referred to as Project Food for the entirety of this document.

This document is split into four major sections. The first (and current) section provides an introduction to the software. Section two elaborates on the overall description of the software. The third and fourth sections contain the functional and non-functional requirements of the product, respectively.

1.1 Purpose

The purpose of this document is to present a functional description of Project Food. The document shall elaborate on the overall description, functional requirements, and non-functional requirements of the software. This document is intended for developers, stakeholders, and users of Project Food.

1.2 Scope

The primary product produced will be a desktop based application, namely Project Food. The purpose of the application is to provide entertainment to users by means of a 2D video game. Inspired by the popular game "The Binding of Isaac", users will indulge in a classic rogue-like video game experience by controlling a character, attacking enemies, and collecting items all while making decisions to visit unique procedurally-generated levels. In addition, users will have the ability to choose whether or not they wish to combine collected items, allowing them to upgrade features pertaining to their player. Upgrades will allow users to progress through the increasingly difficult levels, preparing them for various final boss fights. The application is intended to provide entertainment for all of its users. The application will be free to download from an application store, thereby maximizing the number of players using the application. The main goal of Project Food is to provide users a challenging, immersive, and friendly mobile video game form of entertainment with the classic rogue-like experience.

1.3 Definitions, Acronyms, and Abbreviations

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

Basic Attack: The primary combat method of the *Player*.

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).

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

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

Enemy: Hostile game pieces that will attack the *Player*.

Entities: Objects that the player can interact with.

Ingredients: The material items that are consumed during crafting.

Inventory: The space that allows a player to store items, funds, and equipment.

Inventory Key: A user input that corresponds to the opening the *Player's* Inventory.

Levels: A series of stages of incrementing difficulty that the player traverses.

Map: A network of interconnected rooms that form the available playing space for the player.

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).

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

Procedural Generation: A method of creating data algorithmically as opposed to manually.

Rogue-like: A subgenre of role-playing video game.

Skills: Abilities that a player can acquire from playing the game that improve the combat effectiveness of the player.

Spawn Rooms: The starting room that the player spawns in when entering a map for the first time.

Special Attack: The secondary attack method of the player that has different effects based on the *Player's Skills* **Special Attack Key**: A user input that corresponds to the *Player* using a *Special Attack*.

Transition Area: An area that loads the next *Map* when the *Player* enters it.

1.4   References

N/A

1.5   Overview

This document is organized into four major sections. Section one (the current section) provided an introduction of Project Food. The remaining three sections of this document are described in detail below.

The second section of this document contains the overall description of Project Food. This section will elaborate on the general factors that affect the product and its requirements. This will include the product perspective, the product functions, the user characteristics, constraints, and any relevant assumptions/dependencies.

The third section of this document contains all of the software functional requirements pertaining to Project Food. These requirements provide the purpose for the products existence as they must meet the clients requirements.

The fourth section of this document contains a list of the non-functional software requirements pertaining to Project Food. These requirements were created by the development team with the client in mind.

2   Overall Description

2.1   Product Perspective

This product is a 2-dimensional video game similar to many games on the market such as Binding of Isaac, Soul Knight, Pokémon, and more. Our video game fits in the adventure category for players. The product is independent and totally self-contained, it does not require any other extra programs or internet.

Figure 1: Block Diagram

2.2   Product Functions

Player can start a new game	Player can enter a new room	Player can pick up an item
Player can craft an item	Player can enter a new level	Player can die
Player can start a transaction with the shop	Player can move the character	Player can fight other enemies

2.3   User Characteristics

The education level required would be the completion of middle school. The user must have experience with playing platformer/dungeon crawling games and should know the overall objective of the game. Technical expertise is required for the use of the *players* movement and action controls and volume adjustment.

2.4   Constraints

Time will limit the development of the game as we only have a couple of months before its completion. University course load, midterms, and labs will also be a limiting factor.

2.5   Assumptions and Dependencies

N/A

2.6   Apportioning of Requirements

N/A

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 inputed *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 inputed *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 wants to perform a *Special Attack*.

i. The user inputs the *Special Attack Key* without prompt from the system.

ii. The *Player* responds by performing a *Special Attack* in the cardinal direction the *Player* is facing.

iii. *Special Attacks* that hit collidable *Entities* that are not *Enemies* will be destroyed on those *Entities*.

BE1.10 The user wants to open the *Player's Inventory*.

      i. The user inputs the *Inventory Key* without prompt from the system.

     ii. The *Player's* inventory screen is rendered for the user.

    iii. The user can press the *Inventory Key* again to close the *Player's* inventory screen.

BE1.11 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.

    iv. If the *Player* is hit by an *Enemy* attack, the *Player* will take damage.

     v. If an *Enemy* dies, it will drop an *Ingredient*.

BE1.12 The user initiates combat with a *Boss*.

      i. The *Player Agrovates* a *Boss* by attacking it.

     ii. *Basic Attacks* that hit a *Boss* will apply damage to those *Enemies*.

    iii. *Special Attacks* that hit *Enemies* will apply damage and other effects defined by the attack.

    iv. If the *Player* is hit by a *Boss's* attack, the *Player* will take damage.

     v. If the *Boss* is defeated, the *Player* will be allowed to transition to the next *Map*.

BE1.13 The user enters a *Map Transition*.

      i. The user must have defeated the *Boss* in the current *Map*.

     ii. The *Player* must enter a corresponding transition area to the *Map*.

    iii. The next *Map* will be gernerated and the *Player* will be placed in its *Spwan Room*.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

LF1. The game interface shall be easy and clear for users to operate.

LF2. The game must not contain any frightening scenes.

4.1.2 Style Requirements

LF3. The user should feel that the gameplay increases in difficulty as their game progress increases

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

UH1. Gameplay shall be easy for a person of 10 years of age or older to learn.

4.2.2 Personalization and Internationalization Requirements

N/A

4.2.3 Learning Requirements

UH2. It shall take a maximum time of 5 minutes to learn the basic operation

UH3. After a full read through of the provided instruction manual, the user should be able to use all in-game functionality with ease

4.2.4 Understandability and Politeness Requirements

UH4. All in game icons shall be taken from their common usage icons where applicable.

UH5. The appearance of all food enemies will be intuitive to their physical appearance

UH6. New terminology used in game shall come with a brief explanation of their meaning

UH7. Ambiguous language, such as words with conflicting alternate meanings, used in in-game decisions shall have their meaning clarified

4.2.5 Accessibility Requirements

UH8. The game navigation shall be based on the standard PC control scheme of using WASD mapping, and the mouse.

UH9. All in game elements will have high contrast with surrounding elements to aid colour blind users

UH10. All fonts used for in-game text shall be deemed accessible

UH11. All information communicated via sound shall have an accompanying box of text transcribing the information

UH12. The user should be able to mute or reduce the volume of in-game sound effects and music

UH13. The user should be provided with an epilepsy warning in advance of scenes with flashing lights or rapidly changing colours

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

PR1. The response time shall be within 1 ms.

PR2. Setting up a new game shall take no more than 5 seconds.

4.3.2 Safety-Critical Requirements

N/A

4.3.3 Precision or Accuracy Requirements

N/A

4.3.4 Reliability and Availability Requirements

PR3. The game shall be able to run 24 hours per day, 365 days per year.

4.3.5 Robustness or Fault-Tolerance Requirements

PR4. The game shall not be affected, such as information loss, due to any internet issue.

4.3.6 Capacity Requirements

PR5. The game shall store achievement data for up to one player

4.3.7 Scalability or Extensibility Requirements

N/A

4.3.8 Longevity Requirements

N/A

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

N/A

4.4.2 Requirements for Interfacing with Adjacent Systems

N/A

4.4.3 Productization Requirements

N/A

4.4.4 Release Requirements

OE1. Maintenance releases that fix an error within the game shall be available as soon as they are produced

OE2. Feature releases will be comprised of numerous features and will be released with a maximum monthly frequency

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

MS1. Levels should be modifiable after the product's release

MS2. Floors should be modifiable after the product's release

MS3. The ability to add new levels (and floors) after the product's release should be available

MS4. The variety of food should be modifiable after the product's release.

4.5.2 Supportability Requirements

MS5. The game shall provide the user with a comprehensive set of instructions

MS6. The game shall provide the user with a set of Frequently Asked Questions

MS7. The game shall provide the user with a detailed explanation of the functionality of each item that may appear.

4.5.3 Adaptability Requirements

MS8. The game is expected to run under Windows and Linux operating system

4.6 Security Requirements

4.6.1 Access Requirements

N/A

4.6.2 Integrity Requirements

N/A

4.6.3 Privacy Requirements

N/A

4.6.4 Audit Requirements

N/A

4.6.5 Immunity Requirements

N/A

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

CP1. The game shall not include any features that could be considered offensive in any of our market countries.

4.7.2 Political Requirements

1. N/A

4.8 Legal Requirements

4.8.1 Compliance Requirements

LR1. All game asset shall be compliant with the Web Content Accessibility Guidelines

4.8.2 Standards Requirements

N/A

5 Division of Labour

Andrew Lucentini - Section 1:

Rafen Hussain - Section 2:

Alex Lo - Section 3 (be1-be6 inclusive):

Lucas Zacharewicz - Section 3 (be7-be13 inclusive):

Chen Shuying - Section 4 (in collaboration with Alexie McDonald):

Alexie McDonald - Section 4 (in collaboration with Chen Shuying):