Software Requirements Specification

for

Undercave 2

Version 1.0 approved

Prepared by Ryan Floyd and Orion Mendes

Goonsquad Games

10/22/2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this project is to develop a game with various game systems to create a fun and interactive gameplay experience. Using inventory systems, simple combat systems, and a class and progression system we aim to give players the ability to replay this game to enhance their enjoyment of the game.

1.2 Document Conventions

In writing this document, we are adhering to the SRS IEEE template format to ensure that the information we have to convey is clear and concise. In this document, the requirements will each have their own priority.

1.3 Intended Audience and Reading Suggestions

This document is specifically aimed to convey the requirements of our project to our professor but also to catalog the various systems that will be put in place to create the game. This will be to ensure consistency between members working on the game as well as a reference for the future, should we choose to continue the development of the game to add in online multiplayer functionality or other systems that were not thought of at the time of planning.

1.4 Product Scope

Using C# we will be developing scripts for the game from the inventory system, to the movement and combat mechanics, to the Artificial Intelligence so that we can make enemies tough and engaging to fight against. The goal of our project is not simply in the software we create to help with our goals but to give us experience with the software engineering methodology of making games. Game making is a software engineering process but it abstracts

some of the models making it a great way to give us experience with various engineering methods.

2. Overall Description

2.1 Product Perspective

This product is somewhat of an entirely new self-contained project. The reasoning for the project being “somewhat” self-contained is due to the fact that our group is building off the idea of a previous project that had similar goals but was not as advanced or planned out as this project. We are viewing that previous project as a pseudo prototype, or rather as a proof of concept.

2.2 Product Functions

- Solid Gameplay
- Skill System
- Combat
- Inventory

2.3 User Classes and Characteristics

Mostly all users that use this product will be gamers. we are aiming for ages above 13+. Only technical expertise is the ability to understand modern gaming schemes and elements and how to unzip a file.

2.4 Operating Environment

Our software will be operating on the Windows 10 Operating system. We are building off of the Unity Engine, a free game engine that allows us to efficiently develop for Windows platforms free of charge. In addition, all of our physics simulations and game systems will be handled by the engine. The final product will be distributed through itch.io which is an online game distribution site.

2.5 Design and Implementation Constraints

As of now, the only constraint is time and developer skill level. all the tools needed to make this application are in use and no other tool is needed as of now or for future development. the major limiting factor is our skill level in developing the app which makes development time longer.

2.6 User Documentation

With this game, we will be creating a basic tutorial on how to download the game from itch.io and the overall goal of the game. This will include the game objective as well as the button input

maps so players are aware of the controls. The game can be played on a gamepad and keyboard and mouse so there will be controls schemes for both posted for the user to choose from.

2.7 Assumptions and Dependencies

Seeing as how we are developing in the Unity Engine, we run the risk of running into limitations with the engine or even bugs that have yet to have been fixed with the engine. This could possibly hinder our development time, or it could lead to the final product being faulty. We do not have access to the source code of the engine to fix any issues that we run into so we need to assume that engine issues are a possibility. Luckily, the Unity Engine is well maintained and is well supported by its developers and the community to help with any issues. We also need to ensure that any platform that we wish to expand to can support the Unity Engine to allow our product to run properly.

3. External Interface Requirements

3.1 User Interfaces

Our game will consist of user interfaces, not in the normal way that a normal program would but will have user-operated interfaces so that they may interact with the game and its systems. We will have a start menu that will include the options to start a new game, continue a previous save state, a help tab to give the players their main goal in the game, and give them information on the games combat, inventory, armor, and general systems, and an options tab to adjust the settings such as brightness, volume, and any other aspects we deem fit. We will also have in-game user interfaces that will dynamically update based on the game. This will include the inventory screen where players will acquire items and equip them and the player stats UI which will include the player's health, enemy information, equipment selection, time, and score.

3.2 Hardware Interfaces

Our game/project will not directly be handling the hardware interaction that is required for the game to run. This will be handled by the Unity Engine that we will be developing our game on. The engine was made beforehand by a third party and is freely distributed to allow developers to utilize its vast and fleshed-out systems to help increase project efficiency so that a custom game engine does not need to be created.

3.3 Software Interfaces

Our game will be interfacing with the Unity Engine and its subsequent systems to add functionality to our game. We are creating scripts for the game using the C# library and the Unity C# plugin for the Visual Studio IDE. The Unity C# plugin has a library of functions developed specifically for the Unity Engine to allow for better functionality and will allow us to create the proper scripts to properly utilize the software.

4. System Features

4.1 Combat

4.1.1 Description and Priority

High priority-9

Combat of this game is the main feature of this game so it needs to be fast pace and fun. The combat will be like most third-person games. But the difference is the skills-based leveling system.

4.1.2 Stimulus/Response Sequences

Left Click - Attack clicking left click multiple times creates an attack chain

1-9 numbers - uses abilities/skills to create interesting attack combos

Right Click - Blocks incoming attacks

4.1.3 Functional Requirements

- health System-need to be able to add remove health of characters in the game
    - Time needed: Done
- user input-needed to be to accept user input
    - Time needed: Done
- characters -need characters for the game to have a function combat system
    - Time needed: 1 week
- animations-need animations for the characters to do so the combat is fluid.
    - Time needed: Done

REQ-1: Attacking
REQ-2: Blocking
REQ-3: Skills

5. Other Nonfunctional Requirements

5.1 Performance Requirements

In order to make a smooth gameplay experience, we are targeting that the game will run at 60 frames per second. This is the general standard of modern games and at that frame refresh rate will allow us to adequately control the physics engine in Unity that is tied to the refresh rate.

5.2 Safety Requirements

Our general safety requirements are in regards to the user's health and the health of the hardware being used. The Unity Engine has been built and optimized to run on practically any Windows computer so we are confident that our game will follow the general safety guidelines of the engine's creators to not intentionally damage the hardware of any individual. In addition to that, we always recommend to players that they take periodic breaks from the game to avoid eye strain, fatigue, sleep deprivation, or any other serious medical effects that come from sitting in one place for a long period of time. In addition, we will have a warning at the beginning of the game to warn of any possible events in the game that may trigger an epileptic seizure to ensure the safety and preservation of our players.

5.3 Security Requirements

At this time we are not aiming to add in any security requirements simply because we are not making this game connect to any network and no vital information is being created or stored. In the future, we will begin to consider security requirements based if online multiplayer is implemented where we would need to consider network security options.

5.4 Software Quality Attributes

Characteristics for developers are reliability and availability. A lot of work is needed to be done to complete this project so being available and reliable is a very important characteristic that is needed.