**Software Requirements Specification**

**<Graphical RPG Interface>**

Version 1.0 Draft - Prepared by Jacob Watkins

<Team 6>

Team Mates: Jacob Watkins, Alejandro Camacho, Micah Johnson, Justin Entz

Date: <01/19/2021>

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |

-------------------------------------------------------------------------------------------------------------------------------

**Table of Contents**

Table of Contents........................................................................................................................... ii

Revision History............................................................................................................................. ii

1. Introduction............................................................................................................................. 1

1.1 Purpose................................................................................................................................. 1

1.2 Document Conventions.......................................................................................................... 1

1.3 Intended Audience and Reading

Suggestions........................................................................... 1

1.4 Project Scope......................................................................................................................... 1

1.5 References............................................................................................................................. 1

2. Overall Description................................................................................................................. 2

2.1 Product Perspective................................................................................................................ 2

2.2 Product Features.................................................................................................................... 2

2.3 User Classes and

Characteristics............................................................................................. 2

2.4 Operating Environment.......................................................................................................... 2

2.5 Design and Implementation

Constraints................................................................................... 2

2.6 User Documentation............................................................................................................... 2

2.7 Assumptions and

Dependencies.............................................................................................. 3

3. System Features....................................................................................................................... 3

3.1 System Feature 1.................................................................................................................... 3

3.2 System Feature 2 (and so

on).................................................................................................. 4

4. External Interface Requirements........................................................................................... 4

4.1 User Interfaces....................................................................................................................... 4

4.2 Hardware Interfaces............................................................................................................... 4

4.3 Software Interfaces................................................................................................................. 4

4.4 Communications Interfaces..................................................................................................... 4

5. Other Nonfunctional Requirements...................................................................................... 5

5.1 Performance Requirements..................................................................................................... 5

5.2 Safety Requirements.............................................................................................................. 5

5.3 Security Requirements............................................................................................................ 5

5.4 Software Quality Attributes.................................................................................................... 5

6. Other Requirements................................................................................................................ 5

Appendix A: Glossary................................................................................................................... 5 Appendix B: Analysis Models...................................................................................................... 6 Appendix C: Issues List................................................................................................................ 6

**1. Introduction**

**1.1 Purpose**

Our team will be implementing a dungeon crawler RPG with a GUI. It will include unique customizations for Monster’s, NPC’s, Maps and Quests. We will also include a combat, status and quest system. In this dungeon crawler game, we will provide the user with an interactive 2D game, where the hero/user will have the ability to navigate through a world-map. The world will contain challenges for the user to complete in order to finish the game, examples consist of collecting Pillars of Object Orientated Programming.

**1.2 Document Conventions**

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

**1.3 Intended Audience and Reading Suggestions**

The audience it is intended for is people who like grid based RPGs such as Pokemon or the older Final Fantasy titles and want to be able to customize a campaign for playing.. The best way to read this document is straight down.

**1.4 Project Scope**

We will be creating a grid based map using Unity, C# and WPF forms along with creating a GUI for the user to customize maps, monsters, and NPC’s. We will create a visual representation of the player, NPC’s and Monsters that will be placed on the 2D map. We will also provide windows for inventory and quest management. Finally we will implement a turn based combat system to round out the RPG featureset.

**1.5 References**

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Ref for coding standards here[1]

**2. Overall Description**

**2.1 Product Perspective**

Our program builds on existing programs and games such as Heroes and Monsters and other grid-based games but with unique customizations. We will also expand on the ideas of other 2D grid-based games by giving users the ability to select monsters, weapons, and NPC’s. Our plan is to create a GUI with WPF and utilize dynamic map loading in order to allow for the customization we want to give the user access to.

**2.2 Product Features**

2D Maps:

Top down 2D maps on a grid with textures for floors, walls, player character, NPCs, and Items that will be displayed on the GUI.

Movement Systems:

Movement in the game will be controlled by a set of buttons in the GUI that will make the player character move up, down, left, right and will check to make sure that the player cannot move through walls.

Leveling Systems:

In the GUI there will be a displayed level/experience bar that will keep track of the players level. The player can level their character by completing quests, slaying monsters, or other forms of progress in the game. On leveling the player will be able to increase an attribute such as strength or increase damage dealt.

Combat Systems:

The player can engage with single or multiple monsters in turn-based combat. Who goes first may depend on the speed of the monster or player. The combat will continue until one side is victorious and grant an amount of experience if the player wins. The combat system will use attributes such as chance to block, min and max damage, attack speed, and hitpoints. The player will be able to use basic and special attacks.

Inventory Management System:

The player character’s inventory will be displayed in a small window. This will allow the player to interact with any item by clicking on it in the inventory to see details, equip a weapon or piece of armor, and use an item.

Armor management:

Quest System:

There will be an NPC(s) that gives quests to or ask help from the player. These quests might include item fetching, monster slaying, etc. There will be bonus rewards for these such as experience towards leveling.

Custom Campaign tools:

Map creation, Animation management, NPC scripting, Item scripting, Quest scripting, Tileset management.

Player customization menu:

Monster customization menu:

NPC customization menu:

**2.3 User Classes and Characteristics**

Player:

A player is any person that wants to play and experience the game. The player should be able to open the program and be presented with a GUI displaying the code in a nice style. The player then has the choice of loading into a game that they've already played, or starting a new game. The game will load any assets needed or created by the editor.

Editor:

The Editor will be able to create new maps, NPCs, and items through a GUI that will give straightforward and easy to follow instructions on how to add what you want to the game.

**2.4 Operating Environment**

The RPGI will operate on Windows 10. It may be compatible with systems running 8.1 or earlier, but the target is 10.

**2.5 Design and Implementation Constraints**

Implementation constraints include learning Unity, communicating with team over discord, developing the graphics for the game may be difficult as well depending on the level of customization needed.

The RPGI design, code, and maintenance documentation will adhere to our (coding standards doc ref here)[1]

The RPGI runtime will be rendered in Unity.

The scripts it uses will be written in WPF forms and a user-defined text editor.

**2.6 User Documentation**

Our team will deliver a text document with the games controls and instructions in it. Since the campaign will be user-created, they will have to implement instructions themselves, but the example campaign we design we include explanations.

**2.7 Assumptions and Dependencies**

Some assumed factors that may affect the project are the use of Unity in developing 2D worlds for players to traverse. Another factor is us building off of existing 2D map based games such as Heroes and Monsters.

**3. System Features**

**3.1 System Feature 1**

RPG creation tools and Runtime

3.1.1 Description and Priority

Runtime: 9

Mapping editor: 9

Art Manager: 9

Animation Manager: 6

Monster Manager: 9

NPC Manager: 7

Dialogue Manager: 2

3.1.2 Stimulus/Response Sequences

The user will be able to control the runtime using a combination of keyboard and mouse. The GUI should be able to query the player for the definitions of what art, map, object and scripting assets will be used for campaigns. The GUI will also define each of these individual objects with player input.

3.1.3 Functional Requirements

The program should be able to generate the scripts for the runtime that will not cause the runtime to crash.

The runtime, upon finding manually edited scripts that contain an error, should alert the user to the error and close.

**4. External Interface Requirements**

**4.1 User Interfaces**

**<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>**

**4.2 Hardware Interfaces**

Hardware should be simple enough, a standard personal PC running Windows 10 with a mouse and keyboard should have low requirements for running the runtime and tools.

**4.3 Software Interfaces**

The product is a closed ecosystem of its own programs.

**4.4 Communications Interfaces**

There will be no need to use email or communicate with other users because it’s a closed ecosystem.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

Good performance should be easy to achieve, but on an estimate an AMD Ryzen 3 2200G and a GTX 950 should be more than fine for running the runtime and creation tools.

**5.2 Safety Requirements**

The problem with creation tools is that they often end up deleting files that aren’t necessary and as such there is a possibility that we actually delete a user file unintentionally. It shouldn’t be too hard to create constraints so that this does not occur.

**5.3 Security Requirements**

User security is not a major concern. There will be no authentication for opening projects as the user will already be logged into their system.

**5.4 Software Quality Attributes**

The software should be entirely portable, and it shouldn’t leave any residual registry files on the system outside of the ones generated for save games by the runtime. The program will not be reliant on being installed in a certain folder or having a separate runtime installed (Outside of the standard DirectX runtime distributed by Microsoft).

**6. Other Requirements**

Appendix A: Glossary

RPG: Role-Playing Game

RPGI: Role Playing Game Interface

GUI: Graphical User Interface

NPC: Non-Player character

Quest: A task for the player to complete

Item: A representation of an object the player might pick up and use

Inventory: A set of items that the player has

Tileset: A computer generated map of cut up “tiles”, sourced from an image and cut up at a specific rate

2D: 2-Dimensional, usually meaning graphically