SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

FOR

Team Name: RoguePixel Games

Class: COMP490

Instructor: Edmund Dantes

# 

# Revision History

| Revision Letter | Primary Author | Description of Changes | Date completed |
| --- | --- | --- | --- |
| 0/- | All Team Members | Original Document | 10/08/23 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Page

1. INTRODUCTION \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

1.1 Scope \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

1.2 Product Value \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

1.3 Intended Audience \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

1.4 Intended Use \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

2. FUNCTIONAL REQUIREMENTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2

3. EXTERNAL INTERFACE REQUIREMENTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

3.1 User Interface Requirements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

3.2 Hardware Interface Requirements\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

3.3 Software Interface Requirements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

3.4 Communication Interface Requirements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

4. NON FUNCTIONAL REQUIREMENTS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

4.1 Security \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

4.2 Capacity \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

4.3 Compatibility \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

4.4 Reliability \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

4.5 Scalability \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

4.6 Usability \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

4.7 Other \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

5. QUALIFICATION PROVISIONS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

6. NOTES \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

6.1 Acronyms and Abbreviations \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

**Table of Figures**

Page

No table of figures entries found.

**List of Tables**

Page

TABLE I. REQUIREMENTS VERIFICATION\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

TABLE II. ACRONYMS AND ABBREVIATIONS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

**1. INTRODUCTION**

1.1 Scope

Section 1 will be designated as the SCOPE, which should briefly describe other

applicability of the document. The SCOPE should not be a summary of the contents of

the document and should not be confused with the FOREWORD, RATIONALE, etc. All

documents MUST have SCOPE.

Sample text:

This Software Requirements Specification (SRS) defines the requirements for the XY

Project Software (XYSW) for the XY Project (XYP) developed by Northrop Grumman

Defense Systems (NGDS) located in Northridge, CA (NRDG).

1.2 Product Value

Describe how the audience will find value in the product

1.3 Intended Audience

1.4 Intended Use

Describe how the audience will find value in the product

**1.1 Scope**

This Software Requirement Specification (SRS) outlines the specific requirements and design considerations for Project Horseman, a 2D Halloween-themed gaming experience. Beyond merely providing a blueprint for gameplay mechanics and narrative elements, this document also encompasses design specifications, and potential user interface (UI) and user experience (UX) pathways. Developed by the RoguePixel Games team of the California State University Northridge located in Northridge, CA. This game aims to merge the rich cultural backdrop of spooky folklore with contemporary gaming techniques, ensuring an engaging and atmospheric playthrough.

**1.2 Product Value**

Project Horseman offers players a unique blend of narratives and challenging gameplay. Set against a backdrop of horror landscapes, it taps into the nostalgia of classic 2D games while introducing innovative mechanics.

**1.3 Intended Audience**

Our primary audience encompasses gamers who have a liking for adventure and mystery. This includes individuals who appreciate a balanced blend of story-driven quests and skill-based challenges.

**1.4 Intended Use**

The game serves as an entertaining medium for players to immerse themselves in a virtual ghostly world. Beyond mere entertainment, Project Horseman also provides an opportunity for players to solve in-game puzzles, communicate with other characters, and escort someone, fostering individual skill enhancement.

**2. FUNCTIONAL REQUIREMENTS**

List the design requirements, graphics requirements, operating system requirements, and

constraints of the product.

Provide a Requirement Tag ID number like Func-001, etc

FUNC\_SRS\_001 – The program shall have a drop down menu displaying all the classes

(example)

FUNC\_SRS\_002

FUNC\_SRS\_003..

**FUNC\_SRS\_001** – The game shall feature a main menu interface allowing players to navigate to "Play", "Settings", "Load Game", "Quit" and "About" sections.

**FUNC\_SRS\_002** – Within "Settings", players should be able to adjust audio levels, screen resolution, and control configurations.

**FUNC\_SRS\_003** – The game shall support single-player mode.

**FUNC\_SRS\_004** – Project Horseman will incorporate a save and load system, allowing players to continue from their last checkpoint.

**FUNC\_SRS\_005** – The game's design shall include interactive NPCs (Non-Playable Characters) that provide quests, and trade items.

**FUNC\_SRS\_006** – The program shall have a character customization interface where players can modify their avatar's appearance and equip items.

**FUNC\_SRS\_007** – Project Horseman will have an inventory system that players can access to view collected items, equip gear, and use consumables.

**FUNC\_SRS\_008** – The game shall be optimized for desktop operating systems including Windows 7-10, MacOS 10.12-11, and Linux distributions.

**FUNC\_SRS\_009** – The game will have adaptive background music and sound effects that respond to in-game events and player actions.

**FUNC\_SRS\_010** – Project Horseman shall feature a distinct map area. It will consist of challenges, enemies, and narratives.

**FUNC\_SRS\_011** – The program shall incorporate an achievements system, rewarding players for specific in-game milestones and accomplishments.

FUNC\_SRS\_012 - The system will be using libGDX and written in java.

FUNC\_SRS\_012 – Can the Graphics requirements include support for resolutions up to 4K?

?????????????????????????????

**3. EXTERNAL INTERFACE REQUIREMENTS**

Provide a Requirement Tag ID number like ExtInt-001, etc

3.1 User Interface Requirements

Describe the logic behind the interactions between the users and the software (screen

layouts, style guides, etc).

EXTINTF\_SRS\_001

3.2 Hardware Interface Requirements

List the supported devices the software is intended to run on, the network requirements

and the communication protocols to be used.

EXTINTF\_SRS\_002

3.3 Software Interface Requirements

Include the connections between your product and other software components, including

frontend/backend framework, libraries, etc.

EXTINTF\_SRS\_003

3.4 Communication Interface Requirements

List any requirements for the communication programs your product will use, like emails

or embedded forms.

EXTINTF\_SRS\_004

3.1 User Interface Requirements

**Design Guides:**

* Menu elements must be centrally located or presented in a manner that facilitates rapid access to desired functions, such as starting a game or loading content.
* During gameplay, in-game UI elements should be strategically positioned along the screen's periphery, ensuring that the central action space remains unobstructed to maintain smooth gameplay flow.
* Player Status UI elements should be designed for clarity, making it easy for players to understand and utilize the provided information to enhance their gameplay and decision-making abilities.

**Style Guide:**

The game will feature a pixelated art style inspired by classic Nintendo games, such as Pokémon and Legend of Zelda. It will utilize a 32-bit pixel art design, aiming to evoke nostalgia and pay homage to retro gaming aesthetics.

**EXTINTF\_SRS\_001:** The game shall have a Main Menu with the following elements:

1. Play button
2. Settings button
3. Load Game button
4. Quit button
5. About button

**EXTINTF\_SRS\_002:** The game shall have a Settings Menu with the following elements: ?????

1. Audio: Sound Effects Volume
2. Audio: Music Volume
3. Brightness: Increase
4. Brightness: Decrease

**EXTINTF\_SRS\_003:** The game shall have a Player Status UI with the Health Bar

**EXTINTF\_SRS\_004:** The game shall have a Player Inventory UI with the following elements:

1. Player’s inventory
2. Method to navigate said inventory
3. Selected Item name
4. Selected Item Description

**EXTINTF\_SRS\_005:** The game shall have a Character Customization Menu with the following elements:

1. Enter Name text box
2. Arrows to cycle through character select
3. Finish button

**EXTINTF\_SRS\_006: Dialogue**

1. Dialogue Box
2. Space bar to go to next dialogue

3.2 Hardware Interface Requirements

**EXTINT-HI-001:** The game shall run on Low to Medium Specs Desktop Personal Computers

Intended Minimum Hardware Requirements: AMD Ryzen 5 5500U, 8GB RAM, 256GB Storage, AMD Radeon 7 Graphics, DirectX 10 compatible.

**EXTINT-HI-002:** The game shall require Initial Internet Connection for Downloading 256KBPS or faster Internet connection

3.3 Software Interface Requirements

**ExtInt-SI-001:** The game shall be developed for Windows 10/11 OS

**ExtInt-SI-002:**

**ExtInt-SI-003:**

3.4 Communication Interface Requirements

**ExtInt-CI-001:**

3.5 CSCI Internal Interface Requirements

**ExtInt-IR-001:**

3.6 CSCI Internal Data Requirements

**ExtInt-ID-001:**

**4. NON FUNCTIONAL REQUIREMENTS**

4.1 Security

Include any privacy and data protection regulations that should be adhered to.

NONFUNC\_SRS\_001

4.2 Capacity

Describe the current and future storage needs of your software

NONFUNC\_SRS\_002

4.3 Compatibility

List the minimum hardware requirements for your software.

NONFUNC\_SRS\_003

4.4 Reliability

Calculate what the critical failure time of your product would be under normal usage.

NONFUNC\_SRS\_004

4.5 Scalability

Calculate the highest workloads under which your software will still perform as expected.

NONFUNC\_SRS\_005

4.6 Usability

Describe how easy it should be for end-users to use your software.

NONFUNC\_SRS\_006

4.7 Other

List any additional non-functional requirements.

Not Applicable or N/A

**4. NON FUNCTIONAL REQUIREMENTS**

4.1 Security

**NONFUNC\_SRS\_001:** The game can only be downloaded from secure applications, such as Steam and Itch.io

4.2 Capacity

**NONFUNC\_SRS\_002:**

4.3 Compatibility

**NONFUNC\_SRS\_003:** The software will specify minimum hardware requirements for different platforms (Windows, MacOS, Linux) to ensure that it runs smoothly. These requirements are detailed in the "3.2 Hardware Interface Requirements" section of the project plan.

4.4 Reliability

**NONFUNC\_SRS\_004:**

NONFUNC\_SRS\_004

4.5 Scalability

**NONFUNC\_SRS\_005:** The game will have graphics options to scale with low or high-end hardware. Such as increasing or decreasing the game’s resolution or increasing or decreasing the level of detail.

4.6 Usability

**NONFUNC\_SRS\_006**: The software should be user-friendly, with an intuitive user interface and clear instructions. It should be easy for players of various experience levels to navigate, control the game character, interact with NPCs, and manage their inventory. Usability testing will be conducted to ensure the software meets these criteria. Steam availability TBD.

4.7 Other

List any additional non-functional requirements.

Not Applicable or N/A

**5. QUALIFICATION PROVISIONS**

This section identifies the qualification method(s) used to verify each requirement. This

information is generally provided in a table, with an example shown below. Definitions of

the qualification methods are usually provided before the table.

Sample text:

Qualification in this specification is interpreted as requirement verification. The following

are the base definitions for the verification methods.

A – Analysis: Use of analytical data or simulations under defined conditions to show

theoretical compliance. Used where testing to realistic conditions cannot be achieved or

is not cost-effective. Analysis (including simulation) may be used when such means

establish that the appropriate requirement, specification, or derived requirement is met by

the proposed solution. Examples include the reduction, interpretation or extrapolation of

test data.

D – Demonstration: A qualitative exhibition of functional performance, usually

accomplished with no or minimal instrumentation. Demonstration (a set of test activities

with stimuli selected by the developer) may be used to show that the CSCI, or a part of

the CSCI, response to stimuli is suitable (e.g. observation of fin deployment, etc.).

Demonstration may be appropriate when requirements or specifications are given in

statistical terms (e.g. mean time to repair, etc.).

I – Inspection: The examination of the CSCI code against applicable documentation to

confirm compliance with requirements. Inspection is used to verify properties best

determined by examination and observation.

T – Test: An action by which the operability, supportability, or performance capability of

the CSCI, or a part of the CSCI, is verified when subjected to controlled conditions that

are real or simulated. These verifications often use special test equipment or

instrumentation to obtain very accurate quantitative data for analysis.

An example verification table is provided below, but may be replaced by tables auto-

generated from the requirements management tool, if used

**Table I. Requirements Verification**

| **SRS Req.**  **ID** | **Paragraph Title** | **Verification**  **Method** |
| --- | --- | --- |
| FUNC\_SRS  \_001 | Actual requirement (SHALLS) | Demonstration |
| NONFUNC\_  SRS\_001 | Actual requirement (SHALLS) | Test |

**6. NOTES**

This section contains any general information that aids in the understanding of this SRS.

At a minimum, it should include an alphabetical listing of all acronyms, abbreviations, and

their meanings as used in this SRS and a list of any terms and definitions needed to

understand this SRS. Create subparagraphs 6.x as needed. Note that the list of acronyms

and abbreviations are specific to this SRS and should not contain program-wide terms

that are not used within this document.

The sample text below illustrates additional information along with acronyms and

abbreviations. Note that acronyms and abbreviations should follow any other types of

notes in this section:

6.1 Acronyms and Abbreviations

**Table II. Acronyms and Abbreviations**

| **Abbreviation** | **Full name** |
| --- | --- |
|  |  |