SOFTWARE REQUIREMENTS SPECIFICATION (SRS) FOR

Team Name: Domino Squad

Class: 490/L

Instructor: Edmund Dantes

Revision History

Revision Letter	By	Change Description	Date
0/-	Domino Squad	Document creation	10/23/2022

Table of Contents

Table of Figures

List of Tables

1. INTRODUCTION

1.1 Scope

The scope of this Software Requirements Specification document is to list the requirements of Project Domino, a third-person horror-centric action adventure game, developed by Domino Squad for California State University of Northridge. These requirements include design requirements, graphics requirements, operating system requirements, etc.. This document also lists the verification methods of each requirement.

1.2 Product Value

Project Domino is a video game that takes the player on a story driven adventure that offers fun combat, exploration, puzzles, resource and inventory management, and more.

1.3 Intended Audience

The intended audience are fans of the dark fantasy/ medieval genre, as well as fans of action games (such as Souls-like games) and horror games (such as Resident Evil). The game is also aimed at retro game fans.

1.4 Intended Use

Project Domino is a game that's meant to be played and enjoyed as a single-player, story driven adventure. Players are meant to have fun fighting enemies, getting lost in exploration and getting enthralled in a thrilling narrative.

2. FUNCTIONAL REQUIREMENTS

Design Requirements (Func-001):

- **Func-001.1:** The game shall provide a controller that allows the player to move forward, backward, right and left, as well as fully control their camera.
- **Func-001.2:** The game shall provide the ability of interacting with some specific objects during the game.
- **Func-001.3:** The game shall provide a combat system that allows users to fight enemies.
- **Func-001.4:** The game shall provide the user the ability to keep track of their own health and stamina.
- **Func-001.5:** The game shall provide different kinds of enemies.
- **Func-001.6:** The game shall provide an inventory system that keeps track of the items the player found during their adventure.
- **Func-001.7:** The game shall provide different locations that give the player the ability to explore different areas of the level.
- **Func-001.8:** The game shall provide two multiple protagonists including the knight and the daughter.

System Requirements (Func-002):

- **Func-002.1:** All versions of the game shall provide system requirements both for minimum requirements and recommended system requirements as an industry standard.
- **Func-002.2:** All versions of the game shall provide the following requirements as minimum requirements for the game:

  Operating system: Windows 10/11 64-bit
  Processor: Quad-core Intel or AMD, 2.5 GHz or faster
  RAM: 8 GB
  Video Card/DirectX Version: DirectX 11 or DirectX 12 compatible graphics card
  Mouse and Keyboard devices or gamepad
- **Func-002.3:** All versions of the game shall provide the following requirements as recommended requirements for the game:

  Operating system: Windows 10/11 64-bit
  Processor: 6-core Intel or AMD, 2.5 GHz or faster
  RAM: 8 GB
  Video Card/DirectX Version: Nvidia GPU GeForce GTX 770 / AMD GPU Radeon R9 290
  Mouse and Keyboard devices or gamepad

Constraints (Func-003):

- **Func-003.1:** This project shall run only on windows O/S machines.

- **Func-003.2:** The game shall support playing multiple sessions.
- **Func-003.3:** The game shall have PS1 graphical design and feel.
- **Func-003.4:** The game shall be designed and developed using the combination of Unreal Engine 5 and Blueprints along with C++.
- **Func-003.5:** The game design shall assume that the user has an appropriate version of .NET framework installed and running on their system.
- **Func-003.6:** The game design shall assume the user has at least the minimum hardware requirements and enough space to install and run the game.
- **Func-003.7:** The game design shall assume that the user has access to appropriate peripherals like speakers, headsets, mouse and keyboard.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interface Requirements

Design Guides:

- Menu elements shall be centered or presented in a way that allows the player to achieve desired results quickly ie. starting a game, loading, etc
- In-game UI elements shall be positioned around the borders of the screen, leaving action space in the middle so as not to obstruct the gameplay flow.
- Player Status UI Elements are easy to discern and convey to players the necessary information that enable gameplay and decision making.

Style Guide:

- Follows the theme of the game. Heavy usage of wooden / metal / rust / period era textures to give appropriate atmosphere as well as immersion.

ExtInt-UI-001: The game shall have Main Menu with the following elements:

1. Start new game button
2. Continue Game
3. Load button
4. Level Selector button
5. Options button
6. Exit button

ExtInt-UI-002: The game shall have Load Menu with the following elements:

1. Load slots
2. Back

ExtInt-UI-003: The game shall have Level Selector Menu with the following elements:

1. Option to play a specific Act
2. Back

ExtInt-UI-004: The game shall have Option Menu with the following elements:

1. audio: Sound Effects Volume
2. audio: Music Volume

ExtInt-UI-005: The game shall have Pause Menu with the following elements:

1. Level Name
2. Unpause button
3. Options
4. Exit to Main Menu button
5. Exit to Desktop button

ExtInt-UI-006: The game shall have Player Status UI with the following elements:

1. Health Bar
2. Action Bar

3. Item Durability* TBD

ExtInt-UI-007: The game shall have 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

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: The games shall require appropriate graphical drivers and for other systems such as sounds.

ExtInt-SI-003: The game shall require appropriate Visual C++ Redistributable Installation

3.4 Communication Interface Requirements

ExtInt-CI-001: TBD

3.5 CSCI Internal Interface Requirements

ExtInt-IR-001: TBD

3.6 CSCI Internal Data Requirements

ExtInt-ID-001: TBD

4. NON FUNCTIONAL REQUIREMENTS

4.1 Security

TBD

4.2 Capacity

Describe the current and future storage needs of your software

The game itself will be around 10 GB and may get larger with updates and patches. For game development we are currently using 1 TB of storage (via Helix Core).

4.3 Compatibility

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

4.4 Reliability

Our game would only critically fail if it were to reference a nullptr causing a crash, which may happen once every couple months.

4.5 Scalability

TBD

4.6 Usability

The user must have Steam downloaded with an account. The user can then purchase the game on Steam, and it'll be added to their steam library, where they can install it. The game will also require them to install C++ redistributable and .NET framework.

4.7 Other

N/A

5. QUALIFICATION PROVISIONS

A – Analysis: Verifies requirements through the use of analytical data and simulation. Tests to see if requirements satisfy theoretical conditions. Used when testing through realistic conditions is not possible.

D – Demonstration: Verifies requirements through a presentation of the requirement's functionality in the software itself. Used when requirements require a visual representation to properly verify.

I – Inspection: Verifies requirements through checking the software's code itself. Used when analysis or demonstration are not possible.

T – Test: Verifies requirements through actually using the software itself and checking if it satisfies requirements.

Table IV. Requirements Verification

SRS Req. ID	Paragraph Title	Verification Method
Func-001.1	The game shall provide a controller that allows the player to move forward, backward, right and left, as well as fully control their camera.	Demonstration
Func-001.2	The game shall provide the ability of interacting with some specific objects during the game.	Demonstration
Func-001.3	The game shall provide a combat system that allows users to fight enemies.	Test
Func-001.4	The game shall provide the user the ability to keep track of their own health and stamina.	Demonstration
Func-001.5	The game shall provide different kinds of enemies.	Demonstration
Func-001.6	The game shall provide an inventory system that keeps track of the items the player found during their adventure.	Test
Func-001.7	The game shall provide different locations that give the player the ability to explore different areas of the level.	Demonstration
Func-001.8	The game shall provide two multiple protagonists including the knight and the daughter.	
Func-002.1	All versions of the game shall provide system requirements both for minimum requirements and recommended system requirements as an industry standard.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
Func-002.2	All versions of the game shall provide the following requirements as minimum requirements for the game:	Test
Func-002.3	All versions of the game shall provide the following requirements as recommended requirements for the game:	Inspection
Func-003.1	This project shall run only on windows O/S machines.	Inspection
Func-003.2	The game shall support playing multiple sessions.	Test
Func-003.3	The game shall have PS1 graphical design and feel.	Demonstration
Func-003.4	The game shall be designed and developed using the combination of Unreal Engine 5 and Blueprints along with C++.	Inspection
Func-003.5	Func-003.5: The game design shall assume that the user has an appropriate version of .NET framework installed and running on their system.	Inspection
Func-003.6	The game design shall assume the user has at least the minimum hardware requirements and enough space to install and run the game.	Test
Func-003.7	The game design shall assume that the user has access to appropriate peripherals like speakers, headsets, mouse and keyboard.	Analysis
ExtInt-UI-001:	Main Menu	Demonstration
ExtInt-UI-002:	Load Menu:	Demonstration
ExtInt-UI-003:	Level Selector	Demonstration
ExtInt-UI-004:	Option Menu	Demonstration
ExtInt-UI-005:	Pause Menu	Demonstration
ExtInt-UI-006:	Player Status UI	Analysis
ExtInt-UI-007:	Player Inventory UI	Analysis
ExtInt-HI-001	The game shall run on Low to Medium Specs Desktop Personal Computers	Test
ExtInt-HI-002	The game shall require Initial Internet Connection for Downloading	Test

SRS Req. ID	Paragraph Title	Verification Method
ExtInt-SI-001	The game shall be developed for Windows 10/11 OS	Inspection
ExtInt-SI-002	The games shall require appropriate graphical drivers and for other systems such as sounds.	Inspection
ExtInt-SI-003	The game shall require appropriate Visual C++ Redistributable Installation	Inspection

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 Definitions:

6.1.1 Controller:

6.1.1.1 In video games, a controller references a scheme of input from a player, typically using a peripheral device such as a mouse+keyboard or console controller.

6.1.2 User/Player:

6.1.2.1 The person playing the game.

6.1.3 Camera:

6.1.3.1 In the context of a video game, camera refers to a virtual viewport of the virtually modeled world. This viewport may be moved and rotated to adjust the viewing orientation, allowing the player to view the world as they move through it.

6.1.4 Combat System:

6.1.4.1 A combat system includes multiple attacks, often differing depending on the weapon used, as well as defensive actions, and their corresponding animations. Combat systems can also include evasive actions as well as countering and timing mechanics.

6.1.5 Health and Stamina (Bar)

6.1.5.1 A concept commonly used in video games in order to set limitations on the player, to create challenges. Health, as the name suggests, is an indication of the player character’s bodily state. If the health drops to zero, the player dies and must face the consequences of death, be it reloading to a save point or being relocated. Stamina limits the player character’s actions and particularly the frequency that certain actions may be taken in. This prevents the player from, for example, running full speed indefinitely.

6.1.6 Inventory System:

6.1.6.1 An Inventory system in a video game typically provides an interface from which the player may access items that have been acquired while playing the game. This inventory may display and allow access to items carried by the player character, or in some cases items that are stored in a specific location, or may only be accessed from special locations.

6.1.7 Unreal Engine 5:

6.1.7.1 Unreal Engine 5 is the latest version of the industry leading non-proprietary game engine. Game engines provide a foundation an set of tools for building a video game. Common functionality including graphics, physics, user input, and 3D/2D environment organization is provided by a game engine in order to allow a development team to focus their efforts on aspects specific to the game they are developing.

6.1.7.2 Unreal Engine Blueprints:

6.1.7.2.1 Blueprints are a visual programming language that allows developers to create easily modifiable scripts.

6.1.8 Steam:

6.1.8.1 An online video game market, as well as a manager for the installation, use and maintenance of video games on a user's personal computer.

6.2 Acronyms and Abbreviations

Table VII. Acronyms and Abbreviations

Abbreviation	Full name
PS1	Playstation (The original playstation console)
UI	User Interface