**Software Requirements Specification**

**Project *Luigi’s Adventure (Unofficial Name)***

**Created By: Audrina Smith, Brady Schiff, Edwin Dorwelo Jr.**

**1. Introduction**

Welcome to the Software Requirements Specification (SRS) for “Luigi’s Adventure,” an exciting platformer game designed to take players on a thrilling journey through immersive worlds filled with challenges, puzzles, and endless fun from the perspective of Luigi, Mario’s brother. This document serves as a vital reference for the development and realization of the Luigi’s Adventure project.

**1.1 Purpose**

The Luigi’s Adventure platformer game aims to provide an engaging and entertaining gaming experience to players of all ages. This SRS outlines the software requirements essential to create a captivating platformer adventure that offers unique gameplay, stunning visuals, and a user-friendly interface.

**1.2 Intended Audience**

Luigi’s Adventure is intended for players and gamers who seek an engaging and enjoyable platformer experience. The audience includes individuals of all ages and backgrounds who appreciate challenging gameplay, creative level design, and immersive worlds.

**1.3 Intended Use**

"Luigi’s Adventure" is designed to be a source of entertainment and engagement for players, offering a challenging and enjoyable platformer experience with intricate levels and dynamic environments. It addresses the issue of boredom and monotony in players' lives by providing an escape from daily routines and the opportunity to have fun, unwind, and explore exciting virtual worlds filled with adventure and discovery.

**1.4 Scope**

Luigi’s Adventure encompasses a captivating single-player platformer experience, with the primary focus on level design, character control, and interactive elements. This document defines the core functionality, gameplay features, and overall vision for the game. While we may explore additional features in the future, this SRS primarily addresses the essential components of the game.

**1.5 Objectives**

Our primary objective is to deliver an outstanding platformer game that challenges players’ skills, stimulates their creativity, and immerses them in a captivating and vibrant gaming world. We aim to provide an unforgettable adventure that appeals to a broad audience, offering hours of entertainment and excitement.

**2. Overall Description**

This section provides a comprehensive perspective on "Luigi’s Adventure.” Here, we explore the user needs that underpin its creation, the assumptions and dependencies shaping its development, and the driving forces compelling players to seek this gaming adventure. This section offers a holistic view of the project, serving as a guiding compass for our development journey and ensuring that "Luigi’s Adventure" emerges as an engaging and captivating platformer game that meets the expectations of our players and end-users.

**2.1 User Needs**

Some of the key factors to keep “Luigi’s Adventure” appealing to a wide range of audiences for the long term includes:

* Engaging Gameplay: Players need a platformer game that offers challenging, fun, rewarding gameplay experiences that keep them entertained and engaged.
* Intuitive Controls: Users require user-friendly and responsive controls that allow them to navigate the game’s environments and interact with various in-game elements with ease.
* Creative Level Design: Players expect levels that are thoughtfully designed with creative and dynamic obstacles.
* Visual Appeal: Users need visually appealing graphics and animations that enhance the game’s aesthetics and create an immersive gaming atmosphere.
* Accessibility: It’s important to meet the needs of a diverse audience by considering accessibility features, such as adjustable difficulty settings, colorblind modes, and controls that cater to different playstyles.

**2.2 Assumptions and Dependencies**

In this critical section, we address the key assumptions made during the planning and development of "Luigi’s Adventure." We also identify the external factors, systems, or services on which the game relies. By acknowledging these assumptions and dependencies, we aim to mitigate potential risks and uncertainties that could impact the project's progress. This section ensures that we maintain clarity and preparedness throughout the development journey, enabling "Luigi’s Adventure" to thrive as a smoothly functioning and successful platformer game.

Assumptions:

* Player Skill Levels: Assumption that players have a basic understanding of platformer game mechanics and controls, allowing for a certain level of gameplay complexity.
* Hardware and Software Compatibility: Assuming that players will have access to compatible hardware and software, ensuring the game’s performance and functionality on supported platforms.
* Player Engagement: Assuming that players will engage with the game as intended, completing levels and progressing through the gameplay experience.

Dependencies:

* Third-Party Libraries: Dependence on third-party libraries or plugins for specific functionalities, such as physics engines, audio libraries, or asset management tools.
* External Art Assets: Dependencies on external art sources for creating character sprites, level backgrounds, and other visual assets as the main character originates from an already established universe.
* Sound Design Services: Dependence on sounds similar to what is found in already existing Mario games of the established intellectual property.
* Licensing: The game is based on the widely popular Mario franchise created by Nintendo, so external licensing rights for using their intellectual property is required.

**3. System Features and Requirements**

**3.1 Functional Requirements**

Character Control:

* Define Luigi's movement and controls, including walking, running, jumping, crouching, and special abilities.
* Implement responsive and precise character physics for a satisfying player experience.

Level Design:

* Create diverse and challenging levels with unique obstacles, enemies, and interactive elements.
* Define level layouts, themes, and progression from easy to difficult stages.

Boss Battles:

* Create epic boss battles with unique attack patterns and weaknesses, offering challenging and exciting encounters.

Power-Ups and Items:

* Implement power-ups like Super Mushroom, Fire Flower, and Super Star, each with unique effects and duration.
* Include collectible items such as coins, 1-Up mushrooms, and keys for level completion.

Enemies and AI:

* Design various enemy types with distinct behaviors and attack patterns.
* Develop enemy AI to provide engaging and dynamic encounters with Luigi.

Game Mechanics:

* Include platformer-specific mechanics like wall jumping, wall sliding, and ground pounding.
* Implement interactive objects such as moving platforms, switches, and warp pipes.

User Interface (UI):

* Design a user-friendly and visually appealing UI, including menus, score displays, and heads-up displays (HUD).
* Include in-game messages, notifications, and player feedback.

Sound and Music:

* Incorporate iconic Mario universe music and sound effects to enhance the game's atmosphere.
* Include background music that complements the level themes and events.

Save and Load System:

* Implement a save system that allows players to save their progress and load saved games.
* Enable players to continue their adventure from where they left off.

Accessibility Features:

* Consider adding features like adjustable difficulty settings, colorblind modes, and controller customization to accommodate a broad range of players.

Player Progression:

* Include a system to track player progress, achievements, and unlockable content as incentives for continued play.

Camera System:

* Create a camera system that adapts to the gameplay, providing players with the best view of the action while maintaining a clear and enjoyable experience.

Integration of Mario Universe Elements:

* Integrate iconic Mario universe elements, such as Goombas, Koopa Troopas, and the Mushroom Kingdom, to provide an authentic experience for fans of the series.

Load Times and Performance:

* Optimize the game to minimize load times and ensure smooth performance on various platforms.

**3.2 External Interface Requirements**

* Operating Systems: Specify the supported operating systems, such as Windows, macOS, and Linux, to ensure compatibility with Pygame.
* Sound and Music Libraries: Specify the libraries and tools used for sound effects and background music in Pygame, including any third-party audio libraries integrated into the game.
* Graphic Tools: Define the graphics software or tools used for creating visual assets that are compatible with Pygame

**3.3 Nonfunctional Requirements**

Performance:

* Response Time: The game should provide responsive controls with minimal input lag.
* Framerate: Achieve a consistent and smooth framerate for an enjoyable gaming experience.

Compatibility:

* Ensure compatibility with a wide range of devices, screen resolutions, and hardware configurations.
* Support for various operating systems and platforms, including Windows, macOS, Linux, and mobile devices.

Reliability:

* The game should be stable and not crash or hang during gameplay.
* Implement automated testing and error handling to minimize bugs and glitches.

Accessibility:

* Ensure the game is accessible to players with disabilities by including features like subtitles, customizable controls, and colorblind modes.

Usability:

* Design an intuitive and user-friendly interface for players of all skill levels.
* Provide clear tutorials or onboarding for new players.

Sound Design:

* Pay attention to sound design and mixing for a balanced audio experience, considering the auditory comfort of players.