|  |
| --- |
| **Keys To Insanity  Software Requirements Specification Version 1.0 9/28/2015  Keys To Insanity** |

**Table of Contents**

Contents

[**1. Introduction** 4](#_Toc431229511)

[**1.1 Purpose** 4](#_Toc431229512)

[**1.2 Scope** 4](#_Toc431229513)

[**1.3 Definitions, Acronyms, and Abbreviations** 4](#_Toc431229514)

[**1.4 References** 4](#_Toc431229515)

[**1.5 System Overview** 4](#_Toc431229516)

[**2. Functional Description** 4](#_Toc431229517)

[**2.1 Software Design** 4](#_Toc431229518)

[**2.1.1 UML Diagram** 4](#_Toc431229519)

[**2.1.2 Game Specification** 4](#_Toc431229520)

[**2.1.3 Asset specification** 5](#_Toc431229521)

[**2.1.3.1 Sprite Specification** 5](#_Toc431229522)

[**2.1.3.2 Music Specification** 5](#_Toc431229523)

[**2.1.3.3 Sound Effects Specification** 5](#_Toc431229524)

[**2.1.3.4 Title Screen and Menus Specification** 5](#_Toc431229525)

[**2.2 Game Design** 5](#_Toc431229526)

[**2.2.1 User Interfaces** 5](#_Toc431229527)

[**2.2.1.1 Title Specifications** 5](#_Toc431229528)

[**2.2.1.2 Menu Specifications** 5](#_Toc431229529)

[**2.2.2 Characters** 5](#_Toc431229530)

[**2.2.2.1 The Gentleman** 5](#_Toc431229531)

[**2.2.2.3 Nurses** 6](#_Toc431229532)

[**2.2.2.4 Security** 6](#_Toc431229533)

[**2.2.2.5 Cops** 6](#_Toc431229534)

[**2.2.2.6 Attack Dogs** 6](#_Toc431229535)

[**2.2.1.7 Rats** 6](#_Toc431229536)

[**2.2.3 Sound** 6](#_Toc431229537)

[**2.2.4 Story** 6](#_Toc431229538)

[**2.3 Division of Work** 6](#_Toc431229539)

[**3. System Requirements** 6](#_Toc431229540)

[**3.1 Hardware Requirements** 6](#_Toc431229541)

[**3.2 Software Requirements** 6](#_Toc431229542)

[**4. Interfaces** 6](#_Toc431229543)

[**4.1 Standalone Program** 6](#_Toc431229544)

[**4.2 Keys for Movement** 6](#_Toc431229545)

[**5. Performance** 7](#_Toc431229546)

[**6. Delivery** 7](#_Toc431229547)

[**7. Schedule** 7](#_Toc431229548)

[**8. Miscellaneous** 7](#_Toc431229549)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 9/28/2015 | 1.0 | Software Requirements Specification | Andrew D. Riehl |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Software Requirements Specification, is to lay out the design for our game, Keys to Insanity. This document will serve as a guide for our programming and design of the game. It will also provide a basis for further documentation.

## **1.2 Scope**

This document pertains to the development of our game, Keys to Insanity. It will enable a player to play a game on a Windows machine.

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| GB | Gigabyte |
| GUI | Graphic User Interface |
| PC | Personal Computer |
| PNG | Portable Network Graphics |
| UML | Unified Modelling Language |
|  |  |

## **1.4 References**

N/A

## **1.5 System Overview**

Our game, is a platformer game. This is where the player has to maneuver around and onto moving sprites that will enable the player to reach the end of a level. Specifically our game is a puzzle platform where the player must solve puzzles to complete a level. The player will be forced to maneuver around enemies. The player is not able to attack enemies. They will also receive damage if they come in contact with an enemy.

# **2. Functional Description**

## **2.1 Software Design**

### **2.1.1 UML Diagram**

### **2.1.2 Game Specification**

This game will create classes that implement a platformer game. The player will be able to navigate through a main screen to play the game, to load a saved game, and to delete saved games. When a player loads a level they will have to move amongst platforms and enemies to reach the end of the level; the player is not able to damage enemies, they have to avoid them to reach the end. There will be a health bar and an insanity bar. As time goes on inside the game, the insanity bar will increase. This will lead to the game getting harder, by generation of more enemies, and changing the input, to give a “drunk” feeling to the gameplay. The player will have to grab keys to reach the end of each level.

### **2.1.3 Asset specification**

#### **2.1.3.1 Sprite Specification**

Our sprites will be 16 by 8 pixel size. We will be using the .png format. We intend to fit more on the screen so that each level can be more complicated. We expect to have seven sprite pictures for the main character. Moving forward with two pictures for animation, likewise for backwards. There will be a picture for jumping, pushing, and moving up/down ladders. Each of our enemies will only have four sprite pictures, two for going forward animation, and likewise for moving backwards.

#### **2.1.3.2 Music Specification**

Our music will be in the wave format. We will have about a five minute song that will be looped throughout the levels. We will have about a one minute song that will loop throughout a final boss battle.

#### **2.1.3.3 Sound Effects Specification**

Our sound effects will be in the wave format. They will be made by our own team.

#### **2.1.3.4 Title Screen and Menus Specification**

Our menus and title screens will be .png files that will be loaded in as the player accesses them.

## **2.2 Game Design**

### **2.2.1 User Interfaces**

#### **2.2.1.1 Title Specifications**

We will have a title that will be displayed when the game launches. It will have our games name, and our own names on it.

#### **2.2.1.2 Menu Specifications**

We will have a main menu that will run after the title screen. It will provide links to play a new game, saved games, and an about file. Each will lead to a new menu screen. The new game will create a new game in an available slot, allow the player to give himself a name, and start the first level. The saved game menu will allow the player to start from his last checkpoint. The about file will have our names and give general help, and how to play the game.

### **2.2.2 Characters**

#### **2.2.2.1 The Gentleman**

The Gentleman is the character that the player will play. He will be controlled using WASD, and the space bar. The Gentleman will encounter a talking Top Hat. The Top Hat will guide him throughout his journey, and help him to become sane again. The Gentleman is a middle aged man, who wears a suit, has a cane, and has a talking top hat on his head. He escapes the mental asylum, and has to maneuver throughout the city, to reach his bank vault. In reality this is all happening inside of his head, and the player’s choice at the end of the game will decide whether or not his is cured of his insanity.

***2.2.2.2 The Doctor***

The Doctor is the main boss for our game. The Gentleman will have to evade him to complete levels and to win the game. The Doctor will be controlled by an a\* pathfinding algorithm. He has been working at the hospital for nearly ten years. The Doctor chases after The Gentleman, because he has accidently killed more patients than he is allowed to. If he does not chase after The Gentleman he will be fired and lose his psychiatric license.

#### **2.2.2.3 Nurses**

The Nurses are minions of The Doctor. They have all worked for him for the past 10 years. They will spread out into the city to find The Gentleman. They will be controlled by a AI that will only move then a few pixels at a time, only ensuring that they will not fall off a cliff.

#### **2.2.2.4 Security**

S

#### **2.2.2.5 Cops**

#### **2.2.2.6 Attack Dogs**

#### **2.2.1.7 Rats**

#### **2.2.3 Sound**

We will be creating our own sound effects. For instance we will create a hurting noise for when the main character loses health. Our dogs will also shout bark, bark. This will aid the player in seeing that the events of the game are not occurring in real life, but that they are occurring inside of the Gentleman’s mind. Doing this will allow us to add a personal touch to our game, it will also help to lighten the mood of the rest of the game.

### **2.2.4 Story**

## **2.3 Division of Work**

Nolan Kramer is our lead programmer, he will coordinate the implementation of our game. Andrew Riehl, will lead the work on sound, and write all documentation. Lucas Skinkler will handle our artwork, and our story line. All members of the team will take part in programming, testing, and provide input on all aspects of our game.

# **3. System Requirements**

## **3.1 Hardware Requirements**

This game will be required to run on a classroom PC, and our own computers. The game will have to run on 2GB of memory, and a single processor.

## **3.2 Software Requirements**

The game will have to be an executable file that can run on Windows Seven, and above operating systems.

# **4. Interfaces**

## **4.1 Standalone Program**

The program will run standalone, not interacting with a network in any manner.

## **4.2 Keys for Movement**

The escape key will be used exit menus and levels. WASD will be used for character movement and navigating menus. The space bar will be used for jumping. Finally, the mouse will also be used for menu navigation.

# **5. Performance**

Our game must start within five seconds. All of our assets must load within 10 seconds.

# **6. Delivery**

Our game will be delivered as an executable file that will be able to run on a classroom computer. It must also be able to run on our own computers.

# **7. Schedule**

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 10/5/2015 | Milestone 1 | * Basic movement functionality, * Some gameplay implemented. |
| 10/21/2015 | Milestone 2 | * Some art assets complete * gameplay functionality 90% complete * AI implemented, * scoring incorporated, * GUI complete * Title screen and pause menus designed and implemented |
| 11/23/2015 | Milestone 3 | * Game complete * Delivery |

# **8. Miscellaneous**

N/A