Super Ultra Mega Platformer

Group 13

Grahm Manley, Noah Loos, Xiang Ma, Eric Tran

Software Requirements Specification

Document

Version: 1.6 **Date: (4/7/2017)**

Table of Contents

1. Introduction **3**

1.1 Purpose *3*

1.2 Scope *3*

1.3 Definitions, Acronyms, and Abbreviations *3*

1.4 Overview

3

2. The Overall Description **3**

2.1 Product Perspective *3*
2.1.1 System Interfaces 3
2.1.2 User Interfaces 6
2.1.3 Hardware Interfaces 6
2.1.4 Software Interfaces 4
2.1.5 Communications Interfaces 4
2.1.6 Memory Constraints 4
2.1.7 Operations 4
2.1.8 Site Adaptation Requirements 4

2.2 Product Functions *4*

2.3 User Characteristics *4*

2.4 Constraints *4*

2.5 Assumptions and Dependencies *4*

3. Specific Requirements **5**

3.1 External interfaces *5*

3.2 Functions *5*

3.3 Performance Requirements *7*

3.4 Logical Database Requirements *7*

3.5 Design Constraints *7*

3.6 Software System Attributes *7*
3.6.1 Reliability 7
3.6.2 Availability 8
3.6.3 Security 8
3.6.4 Maintainability 8
3.6.5 Portability 8

4. Appendix **15**

1. Introduction

1.1 Purpose

The purpose of this document is to explain the requirements and features of the Super Ultra Mega Platformer.

1.2 Scope

The application, Super Ultra Mega Platformer, is defined as a 2D platformer build in the Unity® game development software which is applicable to desktop devices by using the Unity® Engine. The main objective for the user or player is to move from the beginning of the stage to the end of the stage. This is achieved by using standard movement controls and abilities chosen by the player to traverse a stage to a defined end of the stage. This allows the user to move on to the next stage and continue progressing through all available levels.

1.3 Definitions, Acronyms, and Abbreviations.

Platformer – A game in which the character has to jump from platform to platform to make it forward in the game.

Peer-to-peer – The network connection where 2 computers communicate directly with each other, rather than both computers connecting to a neutral server.

2D – Within this game, the 2D refers to the dimensions the user is able to move and interact in which would be in the x or y direction.

Player Character – The object within the game the user as the immediate control over.

Stage – A predefined course with a beginning and an end for which the user plays in.

Roster – A collection of abilities the user can attach to the player character

Glitch – Achieving a state of the game by unintended means.

Bug – Not being able to reach a state of a game by normal means.

FPS – Frames Per a Second

UI – User Interface

1.4 Overview

The rest of this document will contain a more in depth analysis of the system we plan to develop with an emphasis on the system functions in section 3.2. These functions will be organized according to the phases we plan to develop in. The document will also cover different aspects of the system we intend to develop like the interfaces, user characteristics system attributes.

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The system does not require system interfaces

2.1.2 User Interfaces

The user interface will be limited to the mouse, screen, and keyboard. The user will have to click through the menu and play the game using the keyboard and possibly the mouse.

2.1.3 Hardware Interface

The hardware necessary to play the game will be a computer, mouse, display, and keyboard.

2.1.4 Software Interfaces

The game should be able to be obtained on any desktop environment that the Unity® Engine supports.

2.1.5 Communications Interfaces

The computer will, not require any communications outside the application.

2.1.6 Memory Constraints

The game should be fairly low in memory usage, given that it is a 2D game with limited graphics. Due to this simplicity it should also take a low amount of ram to run.

2.1.7 Operations

The game should be comprised of only three different modes, The start menu, Stage Select, and In Stage modes.

2.1.8 Site Adaptation Requirements

No adaptations should be needed on the clients computer to play the game.

2.2 Product Functions

The application will serve simply as a platformer game on the Unity® platform.

2.3 User Characteristics

The user of this system will likely already be familiar with games of it’s type. The user does not require prior experience in games, but the game will be of moderate difficulty so as to keep more experienced game players interested. We do expect most of our players to be familiar with basic keyboard game controls. The age of the player should not be a problem.

2.4 Constraints

Since the game will developed on the Unity® platform it should run on computers compatible with the platform like listed in 3.6.5. The game should not have visible lag on computers even without dedicated graphics cards. It will not be developed with the mobile platform in mind and will be limited to computers.

2.5 Assumptions and Dependencies

This project assumes the developers have not made or will make $10k in revenue with this game, otherwise developers must pay a sum of cash to Unity® Engine.

Aside from that, an appropriate assumption is that the user as the proper amount of resources to play intended application. E.g. A display that support 60fps.

3. Specific Requirements

3.1 External Interfaces

The application should not require external interfaces to function.

3.2 Functions

3.2.1 Load in

3.2.1.1 Start Menu mode

3.2.1.1 .1 The user upon launching the application, will be presented with a main menu

3.2.1.1.2 The user can access different game stages or exit the application

3.2.1.2.3 Stage Select mode

3.2.1.2.1 The user will be presented with a selection of stages and depictions of the stages. Stages at first will be built to teach the player how to use abilities. Later on Stages will be designed to challenge's the user by making stages difficult in how the user executes their actions

3.2.1.2.2 The user is able to access stages and play them by selecting them to execute the stage and move to Ability Select at 3.2.1.4. In addition, the user can exit back to the start menu

3.2.1.2.3 In phase 2, the user starts off with the first stage and is allowed to go to any stage they finish.

3.2.1.3 In Stage mode

3.2.1.3.1 User perform movement actions with their player character, and can attempt to complete the stage and overcome obstacles within that stage.

3.2.1.3.2 The user upon moving will have screen pan with them following the player character which changes the view of the stage, allowing the user to see new parts of the stage and reacting accordingly to perform an action to continue progressing.

3.2.1.3.3 Once the user reaches the defined end of the stage, the user is taken out of "In Stage" mode and into the Stage Select at 3.2.1.3 with the next stage available.

3.2.1.3.4 The Player Character

3.2.1.3.4.1 The player character is depicted with a 2d sprite that in phase 2 is able to change colors and could be animated in the future depending.

3.2.1.3.4.2 The user can move in any direction based on keys bounds to the movement.
3.2.1.3.4.3 The user is able to switch between abilities as defined in "Player Abilities" that are bound to a separate set of keys.

3.2.1.3.4.3 For certain abilities an alternate key is used by the user to activate the ability.

3.2.1.3.4.4 Player Abilities

Wall Jump – User is able to use the jump key along a wall to propel their character off of walls.

Double Jump – User is able to jump twice before landing on the jump.

Blink – User is able to move their character at an extremely high speed in any direction, but stops upon impacting a surface or a specified distance.

Wall Climb – User is able to control their character up walls for a certain duration.

Bounce - User's character is now able to

Sprint – User's character is able to move moderately faster.

Dash – User's move quickly on the ground.

Phase 2 or future additions

Time dilation – User's can slow down the stage and move at normal speeds.

Float – User's character has the attribute to glide through the air.

Grapple - User's character releases a line that attaches to surfaces and then the user can use movement keys to direct their motion.

Heavy – User's character has the attribute of being able to fall faster and trigger obstacles.

Shrink – User's character is smaller.

3.2.1.3.5 Obstacles

Spike - Upon contact with this object the character will die and be sent back to the beginning

Pitfall - If the character falls down and is sent back to the beginning

Laser Beam – Causes player to die and restart stage if collided with.

Phase 2 or future traps

Projectiles – These are moving obstacles that can kill the player

Portals – This is an area where if entered the character will be transported to a different location in the stage.

Null-field - The character is only able to use the currently chosen ability.

Ability nullifier - Upon entering this field the character is no longer able to use its special abilities.

3.2.1.4 Ability Select

3.2.1.4.1 User can pick 4 abilities and map them to keys.
3.2.1.4.2 User can reset choice of abilities.
3.2.1.4.3 User can exit ability select to stage select.

3.3 Performance Requirements

The game should be able to run at 60 frames per second without any noticeable lag. Launching the application should not take more than a minute to arrive at the main menu. Map loading should not take longer than a minute for all maps.

3.4 Logical Database Requirements

The game will save progress for the player without the need of a database.

3.5 Design Constraints

This application might be constrained by the Unity® Engine for which the application is being developed on.

3.6 Software System Attributes

3.6.1 Reliability

Reliability will need to be a focus for the application because the introduction of different characters with different abilities will mean any uses of those characters will have to be foreseen so that the game knows how to react to each use. The system should be able to accommodate all the different uses of the players to avoid bugs and errors and allow the player to play from beginning to end without experiencing bugs, glitches, and errors.

3.6.2 Availability

The application should be available at all times since it will be an offline game. It will only need to be downloaded and should be available on that computer indefinitely after that point.

3.6.3 Security

During the agile phase, preventing data mining of the game might be relevant, but otherwise not much security would be needed.

3.6.4 Maintainability

The game should require minimal maintenance after release, but should allow for the possibility of game improvements. The game will be easily modified because the Unity® Engine offers an organizational structure to add new features without interfering too heavily with the base game.

3.6.5 Portability

According to Unity® content developed with Unity® can run on most OS operating systems, including Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.

4. Appendix