Unity Project

Software Requirements Specification

Members: Anthony Jamora, Jonny Nabors, Thomas Burke

Draft 1

*CSC 4420*

*Unity Group*

*2015*

[1 Introduction 3](#_Toc411465380)

[1.1 Purpose 3](#_Toc411465381)

[1.2 Problem Description 3](#_Toc411465382)

[1.3 Project Scope 3](#_Toc411465383)

[1.4 Glossary 3](#_Toc411465384)

[2 General Description 3](#_Toc411465385)

[2.1 Product Perspective 4](#_Toc411465386)

[2.2 Product Functions 4](#_Toc411465387)

[2.3 Main Features 4](#_Toc411465388)

[2.4 Additional Features 6](#_Toc411465389)

[2.5 Users/Perspectives 6](#_Toc411465390)

[2.6 Constraints 6](#_Toc411465391)

[3 Game Features 7](#_Toc411465392)

[3.1 Functional Requirements 7](#_Toc411465393)

[3.2 Non-Functional Requirements 7](#_Toc411465394)

[4 References 7](#_Toc411465395)

[5 Document Revision History 7](#_Toc411465396)

[6 Appendix 7](#_Toc411465397)

1. Introduction
   1. Purpose

The purpose of this requirements document is to offer a comprehensive description of the Unity Game Capstone Project. It will cover each intended feature as well as go over the client’s specifications for completion. The document’s envisioned use is to be a contract between the Capstone team and the client that acts as a reference for developing the finished product. This document will also cover software, hardware, and the user interface.

* 1. Problem Description

The problem which gives rise to this project is that there exists a team of students from the College of Creative Studies whom are really great designers, but lack solid programming skills and experience. The Unity project is the collaboration between the artists of the College of Creative Studies and the software engineers of Wayne State University.

* 1. Project Scope

The goal of the project is to create a small, fully playable video game known as a demo that is playable on personal computers with Windows 7 or Windows 8 operating systems. This demo will exhibit the various gameplay elements that the full-fledged game will contain. Required game elements include: One island environment designed to be an introduction tutorial, a user-controlled camera, appropriate and correctly assigned character animations and sounds, a combat system, and a functional user interface.

* 1. Glossary

**Example Term or Acronym** – Example Description

**Demo** - Playable demonstration of core gameplay mechanics

**PC** – Personal computer

**UI** – User interface

**HUD** – Heads-Up-Display

# General Description

The following section of the document will provide a high-level overview of all aspects of the game as well as covering its basic functioning parts. It will also cover constraints both business and technical as well as define the expected user base or players of the game.

* 1. Product Perspective

The Unity Capstone Game will be a standalone, offline video game that will run as an installed program on a user’s personal computer. This game will allow the user to interact with the game environment acting within a set of constraints listed further on in the document.

Because video games are played by all different manners of individuals and because players play video games the way they think they should be played, the demo will guide the player through the basic fundamentals of gameplay while becoming more challenging as the game continues.

* 1. Product Functions

The following list of functions gives a short overview of the essential functions and features of the Unity Capstone game demonstration. These features are critical to the flow of gameplay. The Main features are what make up the heart of the game’s play system while all additional features exist to add new functionality to the game and will only be added if the schedule allows.

* 1. Main Features

1. Start/Main Menu
2. Allows Player to enter or exit game.
3. Accessible at start of the game or during gameplay.
4. Gives user ability to stop gameplay if needed.
5. Player-Controlled Camera
6. Robust camera to give user ability to view the world around them.
7. Able to be moved around based on preference or requirement.
8. Can toggle 3rd person camera angle and a “look” mode.
9. Player-Controlled Character
10. On-screen representation of user.
11. Gives the player an avatar to interact with the game world.
12. Player has full control of characters actions.
13. Able to use character to attack or block enemy attacks.

iv. Heads-Up Display

1. On-screen, always visible, representation of player information.
2. Gives player ability to always know their character’s remaining health.
3. Visual representation of actual in-game items.
4. Enemy with Artificial Intelligence
5. Natural enemy of main player character.
6. Built with robust artificial intelligence to challenge player.
7. Seeks to find and destroy the player character.
8. Non-Player Character
9. Exists in game to give player guidance and knowledge of surrounding world.
10. Player can find these characters and “talk” to them
11. Serve a multitude of purposes such as progressing game, or just to add to environment.
12. Living Environment
13. Playable area to consist of one island surrounded by water that the player cannot leave.
14. It will contain stairs and a ladder that are usable by the player to explore all predefined accessible areas of the island.
15. Fixed areas in game where enemies appear from and retreat to.
16. Contains one discoverable and operable treasure chest of potions used to regain health
17. Ambient soundtrack that changes upon beginning of combat with enemies
18. Will have one puzzle the character can solve that unlocks more of the island to explore.
19. Has a Day & Night cycle to emulate the rising and setting of the sun.
    1. Additional Features
20. Save/Load Option
21. Allows player to save the current game progress of the character to their local machine.
22. The player can then load these saved settings at a different time to pick up where they had previously left off.
23. Essence System
24. Upon killing enemies, players can absorb their “essence” to be used for purchasing upgrades and special abilities
25. Whenever an enemy is slain, these essences will be added to the player in the form of a numerical counter
26. The player can redeem these essences at the non-playable character to buy upgrades to various statistics of the playable character, such as increased attack speed or reduced damage taken from enemies.
    1. Users/Perspectives

Users of this demo are not limited by age, race, gender, or locality and is accessible by anyone with basic hand-eye coordination. The game may be utilized differently by some users. For example, for some it may act as a hobby, entertainment, or a stress reliever while others may dive deeper into the game to try and set the highest score or beat a personal record while playing in the game world.

* 1. Constraints

1. Business Constraints

In purchasing usable assets for the game the budget is set at and must not exceed $500 for the completed project. All menus and dialogue in game should be displayed in plain English.

1. Technical Constraints

The game is created to run on personal computers or laptops running Windows 7 or Windows 8 and will be optimized for the integrated graphical processing units commonly found in laptops. Additionally it must be developed and created within the Unity Engine.

Minimum Requirements:

-Intel HD Graphics 4400 Series

-4GB RAM

-Intel i3 Processor

-2 Gigabytes of Storage

The game should not drop below 20 frames per second on any computer that meets the minimum requirements.

1. Game Features

The finished product will be a playable demo demonstrating coordinated events, scripts, and textures running within the Unity engine. The demo of the game will be robust enough to not crash during regular play (except for unforeseen circumstances such as a sudden computer crash or power outage). The game itself will be able to handle user input and react accordingly on-screen via a keyboard and mouse.

* 1. Functional Requirements

This section contains the requirements and dependencies of the core functions of the game.

* + 1. Game Setup and Creation
       1. Functional Requirement 1.1

ID: **FR1**

Title: Install Game

Description: The player should be able to install the game on their PC using a downloaded or provided executable. This executable can either be a standalone installer or a zipped folder with the executable located inside of it as well as the game files.

Dependency: None

* + - 1. Functional Requirement 1.2

ID: **FR2**

Title: Logo/Loading Screen

Description: After launching the executable, the game will display a static screen with the logo of the company as well as a small animation to represent the loading of the game.

Dependency: FR1

* + - 1. Functional Requirement 1.3

ID: **FR3**

Title: Load Game Scripts

Description: During the Logo/Loading Screen and before the Main Menu, the game should load and attach all of the scripts to their relevant game objects.

Dependency: FR1, FR2

* + - 1. Functional Requirement 1.4

ID: **FR4**

Title: Main Menu Screen

Description: After the screen has loaded the player will be presented with the main menu. The main menu will give the user the ability to start a new game, exit the game, open the options menu, view the control scheme, and see the game’s credits.

Dependency: FR1, FR2, FR3

* + - 1. Functional Requirement 1.5

ID: **FR5**

Title: Options Screen

Description: From the main menu the player should be able to access the options menu, where the player will be able to change the resolution, adjust graphics video quality, and set a preferred game brightness.

Dependency: FR1, FR2, FR3, FR4

* + - 1. Functional Requirement 1.6

ID: **FR6**

Title: Change Resolution

Description: From the options menu the player will be able to select from a dropdown of 4 available resolutions to fit the one that best represents their current display device. The 4 resolutions are 640x360, 960x540, 1280x720, and 1920x1080.

Dependency: FR1, FR2, FR3, FR4, FR5

* + - 1. Functional Requirement 1.7

ID: **FR7**

Title: Adjust Graphics Quality

Description: From the options menu the player will be able to select either Low, Medium, or High graphics qualities based on the player’s video card in their computer.

Dependency: FR1, FR2, FR3, FR4, FR5

* + - 1. Functional Requirement 1.8

ID: **FR8**

Title: Select Brightness Level

Description: From the options menu the player will be able to control the brightness level via a slider control to set a preferred brightness level to play the game in.

Dependency: FR1, FR2, FR3, FR4, FR5

* + - 1. Functional Requirement 1.9

ID: **FR8**

Title: Select Brightness Level

Description: From the options menu the player will be able to control the brightness level via a slider control to set a preferred brightness level to play the game in.

Dependency: FR1, FR2, FR3, FR4, FR5

* + - 1. Functional Requirement 1.10

ID: **FR10**

Title: View Control Scheme

Description: From the main menu the player will be able to select the option to view the control scheme of the game. This will bring up a visual representation of a keyboard that has clear indicators using arrows pointing to the keys on this keyboard showing what each control is. The player will not be able to change the control scheme, just view what the designated controls are.

Dependency: FR1, FR2, FR3, FR4

* + - 1. Functional Requirement 1.11

ID: **FR11**

Title: View Game Credits

Description: At the main menu screen the player will have the option to view the game’s credits. This menu selection will open up a new screen that will display the names and roles that each member of the team had. From the menu the player may return to the main menu.

Dependency: FR1, FR2, FR3, FR4

* + - 1. Functional Requirement 1.12

ID: **FR11**

Title: Exit Game

Description: The player should be exit the game from the start main menu or from the in-game menu. This will terminate the game process and will return the user to their desktop.

Dependency: FR1, FR2, FR3, FR4

* + 1. Gameplay and In-Game Interactions
       1. Functional Requirement 2.1

ID: **FR4**

Title: Generate Game World

Description: Upon the game launching the game should generate the fully functioning game world, the island. The island will contain the enemy spawn points, terrain, and objects that the player can interact with as well as music, the day/night cycle, and the treasure chest the character can open.

Dependency: FR1

* + - 1. Functional Requirement 2.1

ID: **FR4**

Title: Load Playable Island Object

Description: Once the game has loaded, the island will be generated using the game files that the player will spend their time in-game interacting with. This includes all textures and visual elements as the island consists of hundreds of individual visual elements.

Dependency: FR1

* + - 1. Functional Requirement 2.1

ID: **FR4**

Title: Attach Enemy Spawn Points

Description: After the island has loaded, the predefined enemy spawn points will attach themselves to the appropriate locations on the island using the game scripts and visual cues provided by the island’s layout.

* + - 1. Dependency: FR1 Functional Requirement 2.1

ID: **FR4**

Title: Begin Game Music & Audio

Description: Upon successful load of the island, the game will load the audio assets from the source files and related game scripts. This process will begin the playing of ambient sounds for background music as well as the specific triggers for the audio cues created by the player such as footsteps.

Dependency: FR1

* + - 1. Functional Requirement 2.1

ID: **FR4**

Title: Load Day/Night Cycle & Lighting

Description: Once the island has loaded, the game’s lighting system generate based off of the scripts. This includes the dozens of static directional, point, and spot lights that give the island the general lighting feel as well as the day/night cycle that will give the effect of a setting and rising sun in the game.

* + - 1. Dependency: FR1 Dependency: FR1 Functional Requirement 2.1

ID: **FR4**

Title: Initialize Treasure Chest

Description: Once the island has loaded, the treasure chest that the player can open to find a healing potion will be placed in the predefined location on the island. The player will collide with this object and will not be able to pass through it and will be able to open it by interacting with it using the keyboard.

Dependency: FR1

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Generate Player Character

Description: The game should create a playable character that the user can control via keyboard/mouse and interact with the environment that exists within the game world. This includes generating the player’s sword, shield, and visual attributes & assets such as clothes or armor.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Load Player Model

Description: Upon initial creation of the playable character, the game will provide and attach the model that will be the in-game representation of the player. This includes the collision agent that prevents the player character from falling through the world or walking through any game objects as well as the three-dimensional model with the armor and clothes attached to it. The player character and his armor, hair, facial features, and anything else not in his hands are all one cohesive object.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Attach Control Scheme to Player Character

Description: When the request to generate the player has made, the game will attach the game scripts to the player object that link the control scheme to the player object. This allows the player to control the character via the mouse and keyboard.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Load Sword & Shield

Description: After the Control Scheme has been attached to the player character, the game will load the assets & scripts for the sword and shield that the player character wields in the game. This includes not only the visual representation but also the animations related to it.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Player Movement

Description: Once the Control Scheme and items have been attached to the player character, the player will be able to move in a 360° directional range of movement. The player character will also collide with any objects he may encounter which will stop the player from moving on contact.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Player Jumping

Description: In addition to the player being able to move freely on land or any surfaces, the player can also trigger the jump action by pressing the Space Bar on the keyboard. This will propel the player character in an upward motion with an attached animation the visually represent jumping. The player can use this to help their character navigate over small terrain or to reach higher up areas on the map.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Player Attacking

Description: The player will have the ability to make attacks against enemy characters using his sword. These attacks will be displayed visually on the screen using an animation with the player character’s sword and will award damage points to the enemy on collision.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.2

ID: **FR5**

Title: Player Climbing

Description: The player will not be limited to only using his jump to get to higher locations. There will also be an action available to the playable character that allows him to climb ladders to reach higher points in the game. This event will trigger on collision between the player and the ladder object.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.3

ID: **FR6**

Title: Generate Enemy Characters

Description: The game should generate the enemy objects, with specific spawn points, that the player must face in the game world. This includes the audio/visual cues & elements as well as the enemy’s artificial intelligence.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.4

ID: **FR7**

Title: Generate Non-Playable Character

Description: The game should generate the non-playable character model in the game world that the player can interact with.

Dependency: FR1, FR4

* + - 1. Functional Requirement 2.5

ID: **FR8**

Title: Generate Heads-Up-Display

Description: The screen should show the player character’s health & stamina as well as a depiction of the current items the player is holding.

Dependency: FR1, FR4, FR5

* + - 1. Functional Requirement 2.6

ID: **FR9**

Title: Give User Control of Playable Character

Description: The game should map the default control scheme to the playable character. This means that the playable character will be able to move, attack, defend, or interact with the world following the player’s input.

Dependency: FR1, FR4, FR5

* 1. Non-Functional Requirements

This section contains the non-functional requirements for the game.

* + 1. Game Startup
       1. Non-Functional Requirement 1.1

ID: **NFR1**

Title: Main Screen Load Time

Description: Upon loading the game, the time required by the system to launch the main menu should be less than one minute. Splash screens will be used to mask the loading.

Dependency: FR1

* + - 1. Non-Functional Requirement 1.2

ID: **NFR2**

Title: Game Load Time

Description: After pressing ‘Start’ on the main screen, the time to take the player into the game world should be less than three minutes.

Dependency: FR1, FR2

* + 1. In-Game Mechanics
       1. Non-Functional Requirement 2.1

ID: **NFR3**

Title: Character Reaction from Damage Sources

Description: During gameplay if the character comes into physical contact with a harmful or destructive force the player should notice the consequences of these actions in no more than three seconds.

Dependency: FR1, FR4, FR5, FR8

* + - 1. Non-Functional Requirement 2.2

ID: **NFR4**

Title: Character Death

Description: Upon a player’s health pool reaching zero, which will happen when a player interacts with harmful events, the player character should “die” within three seconds of the health bar in the HUD reaching zero and the game should notify the user that the game has ended and return the user back to the main menu.

Dependency: FR1, FR4, FR5

* + - 1. Non-Functional Requirement 2.3

ID: **NFR5**

Title: Triggered In-Game Events

Description: The game features many in-game events that are triggered upon the player entering a predefined zone or range of a certain object or enemy. Upon triggering the entrance or exit of these zones the game should be able to react to these triggers in no more than three seconds.

Dependency: FR1, FR4, FR5, FR6, FR7, FR9

1. References

|  |  |  |  |
| --- | --- | --- | --- |
| Doc Number |  | Doc Version | Doc Name & Location |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Changes |
|  |  |  |  |
|  |  |  |  |

1. Appendix

Material including referenced documentation the web or elsewhere, as well as alternative designs or items/ideas for future improvements.