Video Game Project

Software Requirements Specification

Project Manager:
Michael Swedo

Assistant Project Manager:
Jonathan Cabe

Software Engineers:
Jonathan Robinson
Deanna Sulli
Michelle Irish
Joseph Bravo
Robert Rojas
Sally Phang

Server Team Assistant:
James Small

Revision 1.2
February 3, 2014

Contents

1. Introduction 3
    1.1 - Purpose 3
    1.2 - Scope of the Project 3
    1.3 - Definitions, Acronyms, and Abbreviations 3
    1.4 - References 4
    1.5 - Overview 4
2. Overall Description 4
    2.1 - Product Perspectives 4
        2.1.1 - System Interfaces 5
        2.1.2 - User Interfaces 5
        2.1.3 - Communication Interfaces 6
        2.1.4 - Memory 6
    2.2 - Use Cases 6
    2.3 - User Characteristics 6
    2.4 - Constraints 7
        2.4.1 - Operating Systems 7
        2.4.2 - Connections 7
        2.4.3 - Platforms 7
        2.4.4 - Content Creation 7
        2.4.5 - Accessibility 7
    2.5 - Assumptions and Dependencies 7
3. Specific Requirements 8
    3.1 - External Interface Requirements 8
    3.2 - User Interfaces 8
    3.3 - Hardware Interfaces 9
    3.4 - Functional Requirements 9
    3.5 - Performance Requirements 10
    3.6 - Design Constraints 10
    3.7 - Security Requirements 10
    3.8 - Document Approval 10

1. Introduction

1.1 - Purpose

This software requirements specification is intended to provide an outline and functional description of *Reignite*, a video game. The expected audience for this document is for the team that will be creating this game for reference as well as Dr. Concepcion for review and approval. The game designers for *Reignite* are Michael Swedo, Deanna Sulli, and Robert Rojas.

This game will be used as a project between CSUSB and Norco College for the purpose of instruction of Norco faculty and students as per the Title V grant in Summer 2014.

1.2 - Scope of the Project

Reignite, the video game that will be created as described in this document, is intended to be open for use by the public. This game is solely intended to be used for entertainment purposes only.

Specifically, the primary target platform for which this application will be run on will be the PC, however, this application is not limited to it if the client desires to meet other target platforms.

In the iteration planned for the final prototype in this course, the player will be able to explore several adventure game environments, play two rhythm game sections, and find the first character, Oliver. The primary features of the game will all be implemented, but the story and art assets will not be fully developed, and as such our scope is reflective of this. The prototype should feature anywhere between 8 and 20 minutes of gameplay, depending on level of experience of the user.

1.3 - Definitions, Acronyms, and Abbreviations

C# - A programming language used to program the logic within the application.

Client - For the extent of this project, this refers to Michael Swedo.

Github - A web-based hosting service for software development projects that use the Git revision control system.

MB - Shorthand for megabyte, a unit of measurement for digital information storage/transmission.

Note Chart - Visual representation of the pitch of music.

Puzzles - In general, tasks that players will have to take in order to progress in the story. These tasks will require a degree of thinking for success.

RAM - Random Access Memory. A form of computer data storage.

Splash Screen - Initial screen displayed when application is executed.

Target Platform - The hardware device that the program will operate on. Development takes this into account in order to ensure program compatibility.

Unity - A cross platform game engine with an integrated development environment.

XML - A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

1.4 - References

IEEE Std 830-1998 Software Requirements Specification
World Wide Web Consortium (W3C) Guidelines for Accessibility
Reignite Game Design Document

1.5 - Overview

The rest of the document will outline the overall description of the application and specific requirements. The overall description will describe the features such as interfaces, constraints that the program and user will have to adhere to, and detailed requirements of the actual implementation of the video game.

Attached to this document is an up-to-date version of our Game Design Document, which describes in further detail the specifications of the game.

2. Overall Description

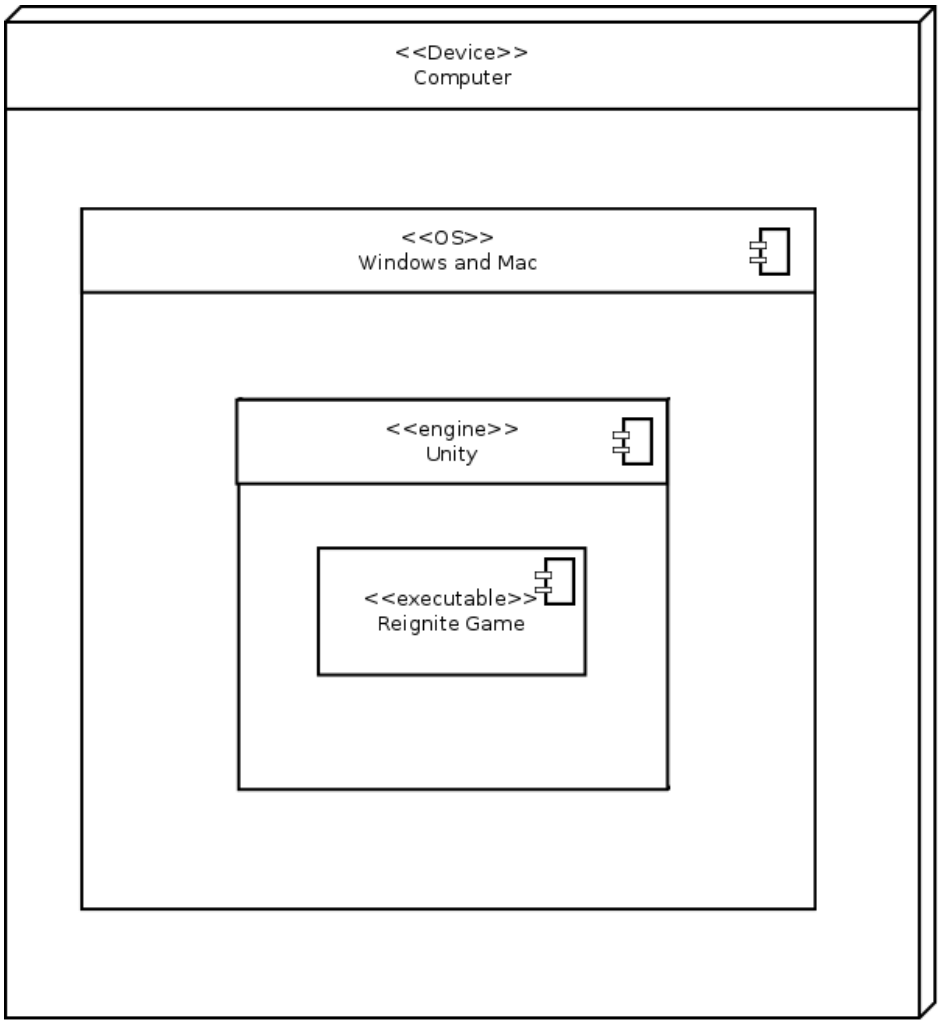
2.1 - Product Perspectives

The following features will be implemented in *Reignite*:

- For the adventure game aspects,
    - Movement in eight directions, controlled by standard WASD keys or a point-and-click interface, on a two-dimensional plane, with collision detection.
    - Interactivity with objects, including things like doors, notebooks, chalkboards, and other objects. When interactive items are hovered over, they will glow. When clicked, a speech bubble will appear, giving the player options: Observe or store in inventory.

    - An inventory system, items that are picked up will be stored here and can be used in conjunction with other objects in the environment to solve puzzles.
        - This will be implemented via a pop-up menu on one side of the screen which displays the items currently in the player's inventory, and allows the player to use these items on the environment.
- For rhythm game elements,
    - Control using a hybrid mouse and keyboard control scheme, where vertical mouse movement controls the current pitch and a keyboard button press (or multiple) follows the rhythm of a playing song.
    - A multifaceted scoring system based on both Mouse and Keyboard accuracy will be required.
        - Visual rewards (animations / graphics) will be provided during play as a reward for doing well.
    - Creation of a scrolling note chart, which will need to be synchronized with the playing music.
- For menus,
    - A Main Menu with options revolving around creating a new game, loading from a saved file, system settings / options like resolution and graphical quality.
    - An in-game menu for the inventory system and for saving the game.
- A save and load system.

2.1.1 - System Interfaces



2.1.2 - User Interfaces

The User will interface with the menu of the game upon program opening, which will contain such items as “New Game,” “Continue,” “Practice”,“Options,” and “Exit to Desktop.”

The purpose of “New Game” is to allow a new user to start the game from the first chapter of the adventure game.

The purpose of “Continue” allows a returning user to resume the last saved game.

The purpose of “Options” allows the user to change the preferences of the game at anytime such items as “resolution, quality, and volume.

The purpose of “Practice” allows the user to practice the rhythm game freely.

The purpose of “Exit to Desktop” allows a user to leave the game when they decide to.

Once the game has been started, the user will interface with the game through keyboard and mouse input, for movement, interaction with objects and characters, and with menus for their Inventory, Key Items, Options, and to return to the menu or exit the program.

The graphical user interface for the in-game portions of *Reignite* will include several menus, one an inventory screen that opens upon the ‘i’ key being pressed, which brings up a bar at the bottom of the screen showing icons of all items the player has picked up, so that they may use these items on the environment without breaking game flow. This menu also has a button that will bring the player to the second of two GUI components in the game screens, the ‘pause’ screen, which can also be accessed using the ‘esc’ key. The ‘pause’ screen will include key items, options, save, load, and exit.

2.1.3 - Communication Interfaces

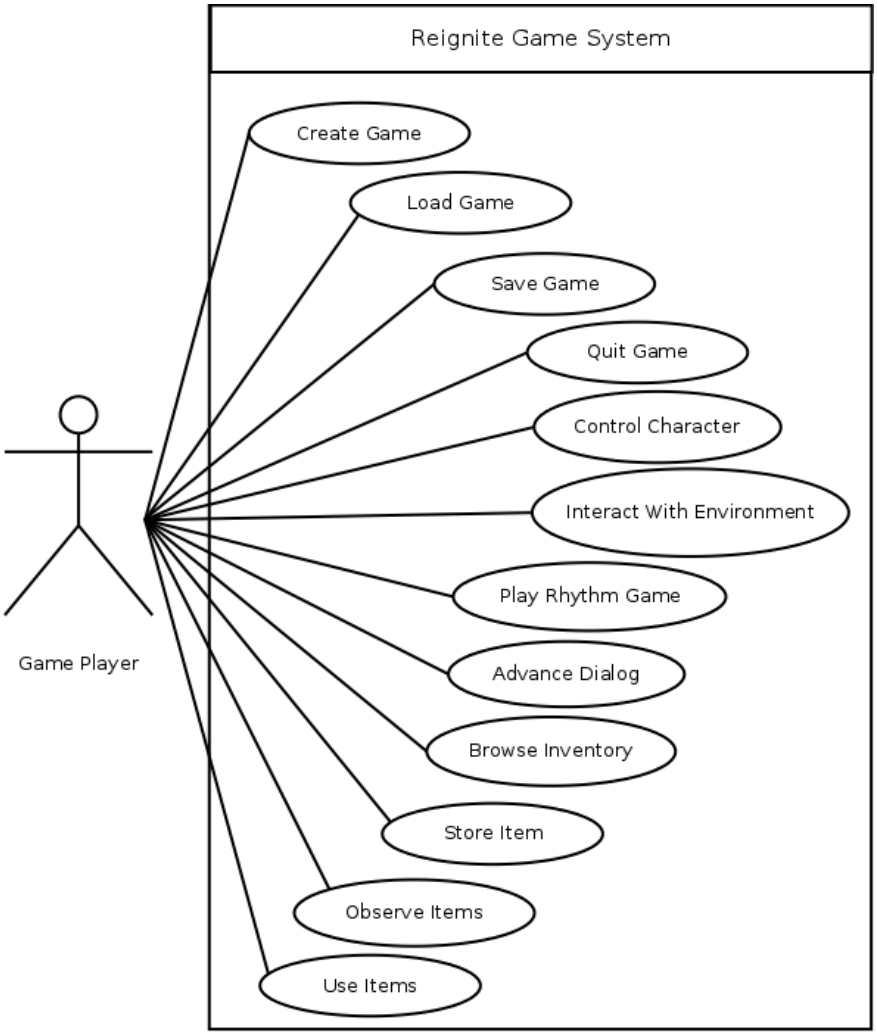
Because *Reignite* does not communicate with a server or any outside force, *Reignite* does not require any communication interfaces to be established.

However, an Internet connection from the user’s computer is required for download of the game, through whatever distribution method the client chooses to use for the released product of the game.

2.1.4 - Memory

We aim to keep the file size under 100MB, and RAM consumption to a minimum, around 50MB. Download of the game will only be possible on a standard internet connection.

2.2 - Use Cases



Use Case Name: Create New Game
Description: At the main menu upon program opening, when the user selects “New Game” the game will save to the first slot available.

Use Case Name: Load Game
Description: At the main menu upon program opening, when the user selects “Load Game” the program will load a previously saved game.

Use Case Name: Save Game
Description: Saves progress through the game.

Use Case Name: Quit Game
Description: The user can choose to quit the game at any time by pausing the game. When the “Pause” menu opens user can choose to “Exit” the game.

Use Case Name: Control Character
Description: For the exploration sections, control of the player using mouse input.

Use Case Name: Interact with Environment
Description: Interact with environment objects, use items from inventory in environment.

Use Case Name: Play Rhythm Game
Description: Once a user solve a puzzle, they will be able to play the rhythm game. For the rhythm game sections, vertical mouse control and (tentative) keyboard keys QWER. The mouse movement matches a pitch, while a keyboard key is pressed. This leads to two forms of accuracy, “Pitch” accuracy for the mouse input, and “Rhythm” accuracy for the button presses.

Use Case Name: Advance Dialog
Description: A dialog system for talking between characters will appear. When a character is speaking, a speech bubble appears near them with text.

Use Case Name: Browse Inventory
Description: During the exploration sections, the user has the option to open the menu to choose an item in their inventory , which can be opened by clicking on a button of the screen.

Use Case Name: Store Item
Description: The user has an option to store the item in their inventory which they have obtained in the exploration section.

2.3 - User Characteristics

A User, also known as a Player, is expected to be able to:

- Play through the exploration portions of the game,
- Play the rhythm game,
- Listen to the pieces and songs used as background music and for the rhythm game elements,
- Interact with objects and characters,
- Learn about the story and background of the game and characters.

Reignite is targeted to all video game players and persons interested in music. We seek to include all potential audiences who would be interested in the story and mechanics of the game, and will avoid using obscure terminology to persons who do not have experience with music performance. *Reignite* also aims to promote music education and could reach out to young teenagers.

2.4 - Constraints

2.4.1 - Operating Systems

Our target operating systems are Windows XP or later and Macintosh OSX 10.7+, and standard Linux distributions.

2.4.2 - Connections

Reignite does not communicate with a server or other outside network. The only connection required is an internet connection for download of the program.

2.4.3 - Platforms

Our target platforms are current Windows and Macintosh operating systems, but Unity's tools allow us to target more platforms in the future should the desire arise.

2.4.4 - Content Creation

No user input shall be able to modify or update the application's content. However, they will be able to create saved games, which will be stored as XML files in the game's root directory.

2.4.5 - Accessibility

Because our desired control scheme for the game relies on both Keyboard and Mouse input, these are required for use of the program. As such, we do not comply strictly with the W3C Guidelines for Accessibility.

2.5 - Assumptions and Dependencies

The *Reignite* project requires the following dependencies.

- Unity 4.3.1
- MonoDevelop C# Tools (Included in the Unity 4.3.1 Package)
- Windows XP or later or Macintosh 10.7+
- GitHub

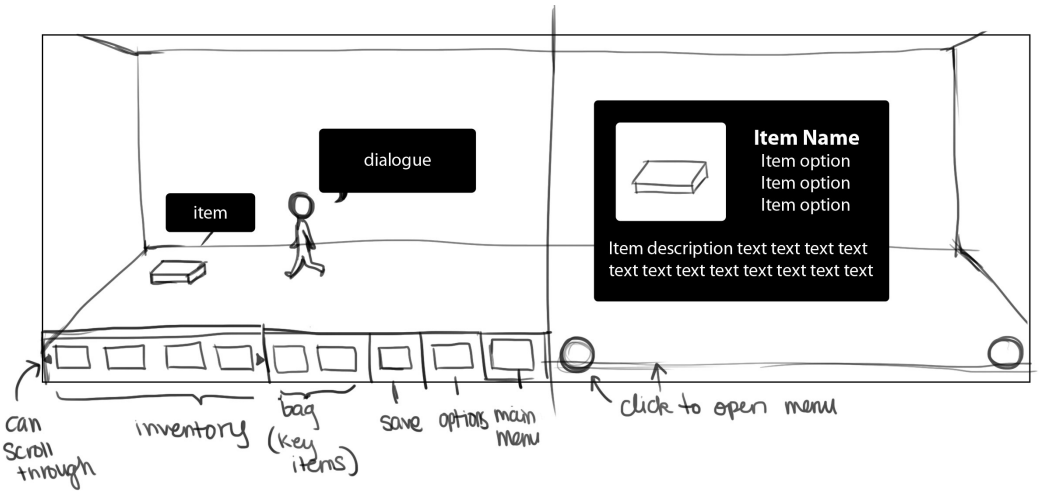
Due to the nature of the JB359 lab computers running Linux environments, these dependencies are the responsibility of the Project Team to acquire, configure, and maintain.

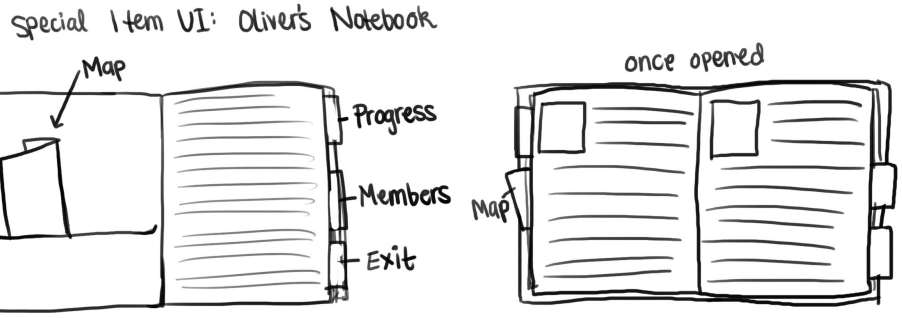
3. Specific Requirements

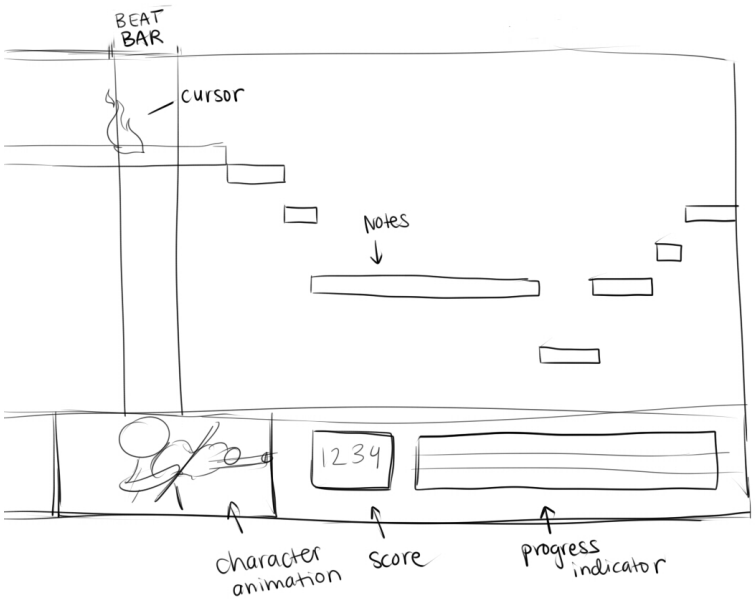
3.1 - External Interface Requirements

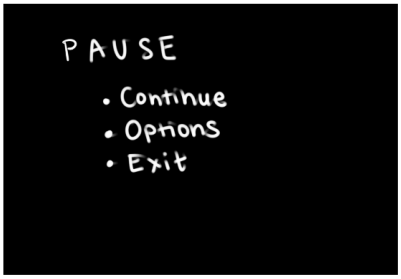
3.2 - User Interfaces

Following is a rough sketch of what our user interface will look like in-game, with menus.









3.3 - Hardware Interfaces

A personal computer running Windows XP or later, Macintosh 10.7+, or a standard Linux distribution will be required to run the application.

3.4 - Functional Requirements

- Control of character through keyboard input and mouse input,
- Interaction with in-game objects,
- Interaction with characters,
- A dialog system for talking between characters,
- A story system for monitoring progression through the story,
- Saving and loading of progress,
- Puzzles that unlock triggers which allow progression through the adventure sections,
- A rhythm game that the user can interact with which includes:

    - A scoring system for win/loss conditions,
    - A hybrid keyboard and mouse control scheme,
    - Visual and auditory feedback for success/failure,
    - Synchronization between audio and visual elements.
- A main menu, to be seen on opening the program and on exiting the main game,
- An in-game menu that includes an inventory, key items, options, save/load features, and returning to the main menu,
- Character and environmental animation,
- Lighting and particle effects where appropriate,
- A loading / splash screen for transition between scenes.

3.5 - Performance Requirements

- Cannot be on a loading screen for longer than 30 seconds without an indication, and even with an indication, cannot be on a loading screen for longer than 60 seconds.
- Less than 80 draw calls.
- Input, for both sections of the game, is responsive within 50 milliseconds.
- Audio and visuals for the rhythm game must be synchronized.

3.6 - Design Constraints

Design for the game will be built around a windowed mode in the 16:9 aspect ratio, specifically targeting 1280x720 as our primary resolution. However, the game needs to maintain its graphical fidelity at all resolutions, at different aspect ratios (4:3, 16:10) and resolutions. As such, letterboxing and cropping will occur, and art assets must be scalable.

3.7 - Security Requirements

Due to *Reignite* not communicating with outside servers, security precautions in the standard sense are not necessary. However, we do want our save files to be secure and not easily editable, as well as any/all data for the game are encrypted so they cannot be modified by the user.

3.8 - Document Approval

This document must be approved by:

- Michael Swedo, Project Manager and Lead Designer
- Dr. Arturo Concepcion, CEO