The Battle for Sugar Rush

A 2D Zelda Style Game

CS 4398
Software Engineering Project
Spring 2023

Group 7
Sam Randall
Dallas Gere
Victor Ekpenyong

Table Of Contents

I. Introduction

1) Purpose

This document serves as a detailed description for *The Battle for Sugar Rush*, A 2D Zelda style game. The contents of this software requirements specifications documents include the many functional, non-functional, and overall features of our software. All group members have reviewed the contents.

2) Scope

The scope of this project includes the programming language of Python. Object Oriented Programming principles will be used in order to implement classes and objects which will work together for the game to function. Multiple Python modules will be imported for use within this project such as *pygame*. *Pixel-boy itch.io* will be used to create our sprites for the game to display within the layout.

3) Intended Audience

The intended audience for this document is an individual or collective interested in the intricacies which go into the development of "Our Game Title", A 2D Zelda style game. As far as the intended audience for the game itself, there is no necessary constraint on who we are targeting. Anyone who has an interest in playing and experiencing our game falls within the intended audience.

4) Definitions

AI	Artificial Intelligence
Sprite(s)	A computer graphic which may be moved on-screen and otherwise manipulated as a single entity
NPC	Non-playable character
Player	The person playing the game
Map/Maps	The area where the player can move around
Inventory Item	Item that a player can store long term
Inventory	Space for player to store items
visible_sprites	Group of sprites that will be drawn

obstacle_sprites	Sprites that a player can collide with

5) Overview / Document Conventions

Following this introduction section, the rest of the document goes more in depth about specifications regarding this project. The roman numerals represent the parent sections while the numbered sections represent the subsections of main sections. After reviewing this document, the reader should have a better understanding about the requirements and details regarding our game.

II. Overall Description

1) Document Perspective

The software specification document describes the functional and nonfunctional requirements of a game application for a computer such as mac or windows. For functional requirements, the document describes a basic path taken by a user, and then describes how the requirements will be fulfilled in the software. The document also describes some nonfunctional yet essential qualities about the game.

2) Document Functions

This document is intended to be a basic outline of the software and the basic guide and flow for an end user.

3) Synopsis

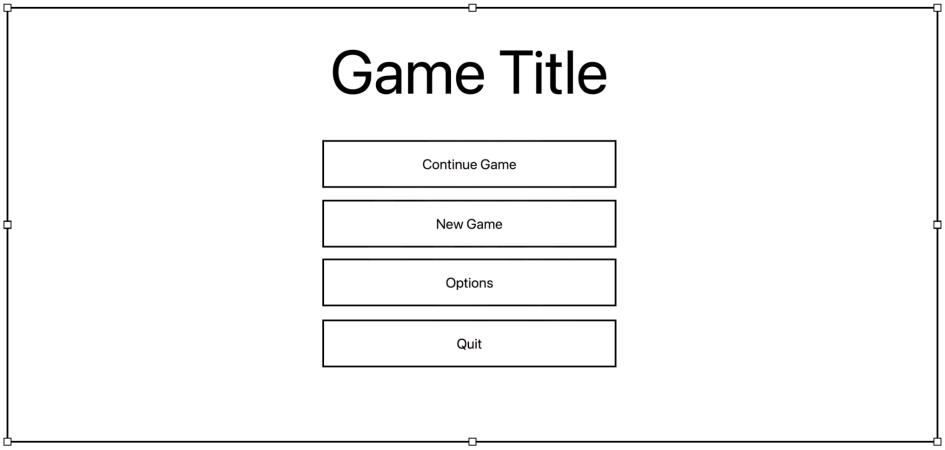
Evil Lord Azazel has stolen the entire supply of jelly beans from the Zamor Town residents. Heroic warrior of Zamor, Iroas Zelé, must defeat Azazel and return the jelly beans to the town of Zamor.

4) The Object of the Game

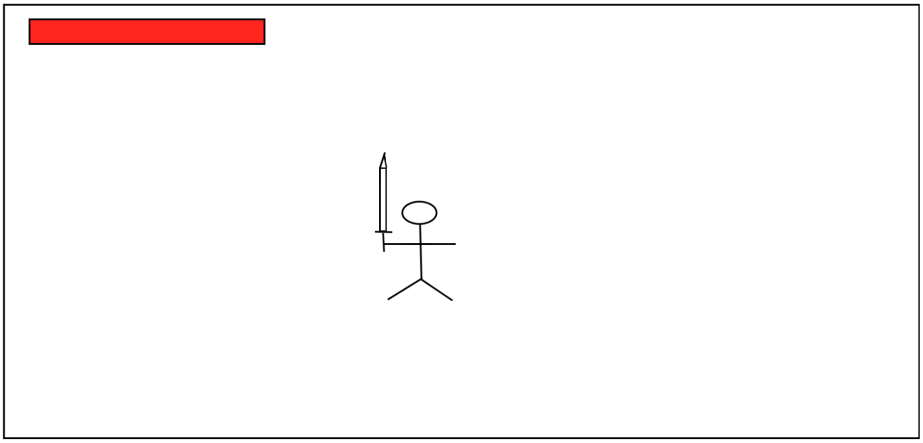
The player, Iroas, will defeat enemies and traverse several different levels. Each level will contain its own objective for the player to complete. Completing a level will unlock the next level. The player will need to progress through each level and defeat the final boss Azazel in order to return the jelly beans and beat the game.

5) Layout

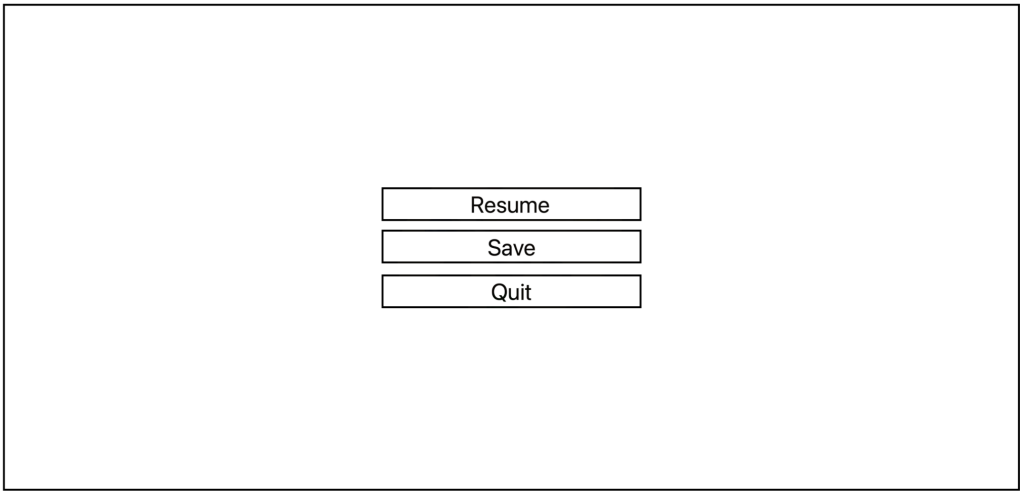
The starting Screen will have basic options on how to proceed.



The screen layout will include a player health bar which is overlaid on the level.



The pause screen will allow players to pause the game, save, and return to the title screen.



6) Game Flow
The player will start with a sword, a base amount of hp, a base amount of attack power, and a base amount of defense. By playing the game and progressing the player will be able to upgrade each stat/sword. A maximum of each stat will be imposed.

7) Game Controls / Functionalities
The basic controls include up, down, left, right, and attack. Players will navigate the map and have the ability to attack enemies. More controls can be added for better interaction. A pause button will allow the player to pause the game, as well as level up, perform inventory manipulation, and exit the game.

8) Level Structure / Difficulty
Levels should increase in difficulty by each successive level becoming harder. This should also take into account the fact that players level up stats.

9) Characters / Enemies
The main character is Iroas. The main boss is Azazel. There will also be several lesser enemies.

III. Functional Requirements

1) How to Play the Game
Users will need to download the game onto a computer, either windows or mac, and run the executable file (exe or dmg).

2) Save Files
Games and save files will be stored locally on the user's machine. This will save progress and user items.

3) Checkpoints
The Checkpoints will save a player's progress once a level is completed and when a checkpoint inside of a level has been reached.

4) Game Navigation
The Game Navigation will first be through a menu that allows players to set up, resume, and pause the game. Then the users will be able to play the game. We also want the player to navigate through the map using their keyboard.

5) User Login
The User Login will be so that the players can login to different accounts and be able to play the game at different stages, provided they have enough disk space.

IV. Non-Functional Requirements

1) Reliability
It could be a good idea for the game to save periodically, that way in the event a crash or game quitting unexpectedly, then the user won't have to worry as much about the game being lost

2) Robustness
If there are any errors in our game then we want there to be a good message displayed to the user about what went wrong and what to do next. We want the instructions to be clear and the users to know that it was the games fault and not their own fault. Also it would be good for the game to be able to save state as soon as an error arises, just in case the user has to restart for any reason.

3) Performance
Ideally we want the game to be able to run on any user's machine while other tasks are running as well. We do not want our game to take all the memory from the user. Also we do not want our game to take up too much storage as the player collects more items

4) Maintainability
We want our game to be easily fixable in the event of bugs and other problems. The game should be adaptable and have the ability to build on to so that we can keep development at a steady rate and so that we have the ability to make the game not get boring.

5) Usability
We want our game mechanics to be intuitive and designed in a way that someone who has never played a game before can play our game being taught the bare minimum. Also we want it to be so good that anyone who is experienced at playing games needs no introduction to how to play.

V. Design/Implementation Constraints

1) Standards Compliance
We want our game to only make changes to the game files and nowhere else on the user's machine. This will help to make the game more secure and so that only the game could be affected by any malicious software.

2) Development Constraints
We have to think about if the players are able to run python and specifically pygame on their machines without any other installs. We also have to do some more testing to see what some minimum machine requirements are so that the players will know if the game is optimized for their machine. We also have to figure out how to create a build for both windows and mac and how to make sure that the game is optimized for both machines

VI. Appendix

1) References

- ☐ American National Standards Institute, Institute of Electrical and Electronics Engineers, & IEEE Computer Society. Software Engineering Technical Committee. (1984). An American national standard IEEE guide to software requirements specifications.
- ☐ GettingStarted - pygame Wiki. (n.d.). https://www.pygame.org/wiki/GettingStarted
- ☐ Martin, R. C. (2003). Agile software development: Principles, patterns, and practices. Pearson.
- ☐ Ninja adventure - Asset pack. (n.d.). itch.io. https://pixel-boy.itch.io/ninja-adventure-asset-pack
- ☐ Pygame Docs - https://www.pygame.org/docs/