Software Implementation and Testing Document

For

Group 10

Version 1.0

Authors:

Reece Gabbett

Ryan Beck

Parker Stone

Timur Bickbau

Marcos Sivira

1. Programming Languages (5 points)

We have used only C# for all of our scripts as of now. We used C# when programming the movement and animations of Lank, as well as being used in our very early build of our game world maker.

2. Platforms, APIs, Databases, and other technologies used (5 points)

As of now, we have been using the Unity game engine with a Virtual Studio editor in order to program our game. As for the database where we are hosting the code, we have decided to use GitHub as it is extremely reliable and can be easily used with almost any editor.

3. Execution-based Functional Testing (10 points)

We tested out our movement, combat, and animation mechanics very easily due to Unity's testing environment. It allows the tester to jump right into the game to immediately test for bugs and issues. This made the functional testing extremely easy to do.

4. Execution-based Non-Functional Testing (10 points)

We have tested our scalability by testing what FPS the game can achieve on each of our personal computers. We have discovered that any number outside of the test 60 FPS can break the movement animations and make the game run poorly. We have not tested any sort of encryption to ensure data integrity due to how basic and simple our game environment currently is. As for usability, we have so far been testing our game using Windows 10 as well as a plugged-in keyboard and mouse. When thinking about reusability, we decided to plan on retexturing and reusing textures and animations for multiple NPCs, we tested this out by using the Legend of Zelda: Link to the Past Link sprite which ended up working. Finally, we tested out the reliability by simply playing the game: moving around and watching the animations to see if there were any visual/game-breaking bugs so far; we are not aware of any such bugs right now.

5. Non-Execution-based Testing (10 points)

We ran non-execution-based tests for our current game environment by pulling our repo into our personal PCs in order to run and test them in the Unity Editor. Since our game only contains basic 8-directional movement and walking animations, testing was not overly difficult. However, it was helpful as it helped us find a movement bug and combat bug fairly quickly. The execution-based testing was not drawn out due to the size of our game, but it has forced us to begin thinking about how we will test future features.