Software Implementation and Testing Document

For

Group 10

Version 2.0

Authors:

Reece Gabbett

Ryan Beck

Parker Stone

Timur Bickbau

Marcos Sivira

1. Programming Languages (5 points)

We have continued to use C# for scripting most of the game's functionalities, such as the map editor and health system.

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

We have continued to use the Unity game engine for game development, and Virtual Studio for writing scripts in C#. We have also continued to use GitHub as the central online database for hosting all of our code. We have temporarily used Legend of Zelda and Pokemon sprites as placeholders for ours which will be implemented next iteration.

3. Execution-based Functional Testing (10 points)

The map editor was tested throughout the development process and fixed appropriately along the way. We ensured that the editor could be used freely without any issues. A similar approach to testing was pursued for collision and hitboxes. We did not run into any significant issues in developing these processes (except for an issue with aspect ratios described in the progress report).

There was not a lot of testing to do with the health system as damage-dealing enemies have not yet been implemented. Currently, all that was tested was that the health bar correctly displays Lank's health for any initialized value between 0 and 6. Testing of the health system's actual responsiveness will be done when those damage-dealing enemies are implemented toward the beginning of the third iteration.

Testing was done continuously with the implementation of enemies. The enemy at first walked through objects such as buildings so that had to be fixed. Another big problem was the enemy would fly off screen when touched by Lank but that was also fixed through coding. Overall the implementation of enemies went rather smoothly.

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

We have not yet tested data encryption as there is not yet any saving functionality that we have implemented. The game has specifically been tested on Windows 10 with keyboard and mouse, but should also work on any Windows version since Windows 7. We will be reusing sprites over multiple enemies and across the map that we will design throughout the third iteration (using the map editor). The game seems to be reliable in its current state, as there are no known game-breaking bugs (except for an issue with aspect ratios mentioned in the progress report).

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

We have continued to actively test the game by downloading it from GitHub onto our personal computers and running it locally within Unity. This has been done consistently throughout the past month to ensure that the game is fully functional and bugless. Testing has involved ensuring that new implementations do not interfere with previous or concurrent updates by any developer.