Software Implementation and Testing Document

For

Group 10

Version 3.0

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.

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. JSON was used for all saved data and serialization.

3. Execution-based Functional Testing (10 points)

With damage-dealing enemies fully implemented, the health system could now be tested properly. Testing ensured that damage is properly dealt to the player when the player is attacked by an enemy. In this iteration, the health system was slightly tweaked to use 6 half hearts instead of 6 full hearts. Enemies were also made to drop hearts when killed (as were pots, when destroyed). This was further tested with the new health system in various states.

With the addition of projectile objects, Testing was done to ensure that temporary entities properly despawned on collision with an object or after a certain amount of time. A collision matrix was edited to ensure that projectiles spawned by a type of entity can not affect the entity that spawned it.

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

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 have reused sprites over multiple enemies and across the map that was designed 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.