Software Implementation and Testing Document

For

Group 6

Version 2.0

Authors:

Riley Arunakul

Gabriella Estevez

Braden Ford

Kevin Foughty

Mattan Pelah

1. Programming Languages (5 points)

C# scripts for editing GameObjects in the Unity Game Engine editor. We're using C# as it's the programming language supported by Unity.

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

Since we are developing a video game from scratch we currently aren't using any software platforms or databases for our project. It's likely in the future we may use a database for player data or for a leaderboard. However, we are using an API called, "Unity Scripting API." Unity Scripting API is why we are using C# for our project as it allows direct integration with GameObjects in Unity. This API also contains many subclasses that allow tweaking specific aspects of the game engine to our advantage. Therefore this API has been implemented in every asset of our project thus far.

3. Execution-based Functional Testing (10 points)

To test the functional requirements, we created multiple test levels that allowed us to test individual aspects of the game that we worked on. For example, while Ella was working on the character's movement, she created a sample level that showcased the movement capabilities of said character.

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

Most of the non-functional requirements are subjective (i.e. the game should be fun) so we are unable to test those at this stage in development. However, non-functional requirements that were objective (i.e. the game does not crash) were tested the same way the functional requirements were tested.

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

Non-execution base testing was done through the use of the meetings that took place throughout the increment.