Software Implementation and Testing Document

For

Group <19>

Version 1.0

Authors:
Gabriel Clewis
Justin Taing
Brandon Embleton

1. Programming Languages (5 points)

C# - used in Unity for the entirety of the project
Unity only supports C# as of now so it was our only choice.

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

Unity is our main engine being used. We use a multitude of packages provided by Unity in some of the scripting.

3. Execution-based Functional Testing (10 points)

Functional Battle System – Tested the Fighting scene to properly see if the turns and health changes update properly in both the game and inspector views.
Working Overworld with movement – We implemented player movement and ran the unity scenes to test if player movement worked with the colliders used.

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

Simplicity – We tested the controls and made them easy to understand from a movement and fight-scene gameplay perspective.
Dynamic – We made the fight scene dynamic in that unit's can be added in and the turn system from BattleManager will react as needed.

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

Most of our code testing was dependent on actually running the Unity scenes and seeing if they functioned properly. Though, we checked each other's code to make sure it would run efficiently and correctly.