**Software Implementation and Testing Document**

**For**

**Group 13**

Version 2.0

**Authors**:

Thomas Cano

Tyler Flaherty

Jack Hayes

Thomas Marconi

# Programming Languages (5 points)

We are using C# to write the scripts and object oriented components of the game (like classes for instance) due to its relatively low time to learn and its compatibility in Unity. We are using Unity as our game engine because we believe Unity offers the most versatility and utility for developers which grants you greater freedom to fully design and implement your vision for your game.

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

In terms of creating the game, we will just be using unity and C# for the reasons previously stated. We do not plan on adding any additional softwares as of right now but that may change if it is necessary. We are also using github to collaborate on the project and easily share code.

# Execution-based Functional Testing (10 points)

Tested the movement grid throughout its development, tested components such as highlighting of the block when hovering over it with mouse and got that working properly.

# Execution-based Non-Functional Testing (10 points)

Tested to see if the game successfully loaded elements for the game, such as the movement grid and characters of the scene. Tested to see if information would transfer correctly and update correctly from scene to scene, more specifically, would make sure the right units would go into our combat scene and would make sure the right unit got deleted depending on which one was defeated. The level1 scene and the battle scene were the scenes tested for this back and forth interaction.

# Non-Execution-based Testing (10 points)

We did not do any real non-execution-based testing over the course of our project. Because we made a game, most all features had to be implemented then checked to see if they worked by running the game.