Software Implementation and Testing Document

For

Group 5

Version 2.0

Authors:

Aiden Allen | nickname: Alice B

Stefano Sanidas

William Couture

Rafael Cardoso

Aidan McGill

1. Programming Languages (5 points)

C# is used in the Unity scripting system and with any scripts we create. For example, we had to create a C# script to write code that allows the scenes or levels in our project to switch between one another. The reason why we chose C# is because we had no choice since Unity exclusively uses the language for its scripting.

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

- ☐ **Unity** - The game engine framework
- ☐ **Visual Studio** - Code editor paired with in the unity game engine
- ☐ **Cinemachine** - Unity package used for smooth camera tracking
- ☐ **Input Manager** - Unity package used for managing player input
- ☐ **TextMeshPro** - Unity package used for displaying text
- ☐ **Github Desktop** - Git-based software for version control.

3. Execution-based Functional Testing (10 points)

Towers:

For testing the functionality of towers, the prefabs were hand placed into the level to test the behavior when enemies come in range and compare with expected behaviors. Placing the towers directly into the scene allowed us to test the functionality of the towers before fully implementing the placing mechanism of the towers.

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

In order to test the non-functional requirements to ensure the performance and stability of the application we have implemented features such as a Frame Rate counter. The purpose of the FPS counter is to ensure consistent performance throughout the application. Our team has also implemented a coordinate system, in order to track the players location so we may ensure that the player is within bounds, as well as ensuring visual alignment and collision tracking.

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

After certain features were implemented such as the enemy pathing, player movement, and npc dialogue system, we did a code walkthrough. This ensured we were all aware of how the features work and how they were implemented in the case someone else needed to edit the associated code.