Software Implementation and Testing Document

For

Group <14>

Version 1.1

Authors:
Sergio Pantoja
Tyler Starling
Joseph Schmitt
Nathan Zhao
Talal Mahmoudi

1. **Programming Languages (5 points)**
   For our Unity project we are using the C# programming language. The reason for our choice is because Unity by default uses C# for its scripting. We use these scripts and attach them to game objects for gameplay functionality. Most of our members are familiar with C# and the learning curve wasn't as difficult as some other languages.

2. **Platforms, APIs, Databases, and other technologies used (5 points)**
   Our group used the Unity game engine for game design and Microsoft Visual Studio for programming the C# scripts

3. **Execution-based Functional Testing (10 points)**
   We tested the function for the highlight key upon press during gameplay by making sure the correct key lit up for the intended amount of time. Spawners are working but still need to link it to the keypress and make sure an error or correct hit sound is played while also incrementing the score accordingly which will be done in the next iteration. The main menu was tested by hitting play and exit in different runs while the game is fully loaded to make sure it goes to the game when play is hit and exits when in the main menu on the initial load up. The settings still needs polishing to actually change the settings which will be done by the next iteration.

4. **Execution-based Non-Functional Testing (10 points)**
   We were not able to perform any non-functional testing in this increment of our project. We are not able to determine lag at this point in development and have yet to determine security of the project

5. **Non-Execution-based Testing (10 points)**
   We looked over the scripts in Unity as a group during team meetings and made sure to edit them as a group if changes needed to be made