**Software Implementation and Testing Document**

**For**

**Group <24>**

Version 3.0

**Authors**:

Garett A

Daniel B

Cameron H

Jared U

# Programming Languages

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

Python - used for the totality of the current state of the project, specifically we use the pygame API to control all game aspects, and all regular pygame functionality to manipulate gameplay. We chose Python because it’s a powerful language commonly used today in the industry, and want to become better at it.

# Platforms, APIs, Databases, and other technologies used

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

Pygame - Essentially for the entire project since it’s a game

SqLite - For the High Scores

# Execution-based Functional Testing

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

*The functional requirements of our game are basically what make the game the game. Since we were producing a video game it was all very visual, especially testing in things we added in. Since we’re building a video game, most of the functional requirements were things you could see on screen. When adding in new features we either saw the results directly on screen and saw the issues to fix, printed out statements in the code to test, or used debuggers to track states of game objects while running through it. Again, functional testing was super easy since all the functional requirements made up the video game.*

# Execution-based Non-Functional Testing

*We did not have to test for non-functional requirements as our only non-functional requirement was to have a computer with Python installed.*

# Non-Execution-based Testing

*We did not perform non-execution-based testing.*