**Software Implementation and Testing Document**

**For**

**Group 2**

Version 1.0

**Authors**:

Grant Leatherman

Miles Brosz

Joey Riley

Caroline Mangrum

Kendall Fretwell

# Programming Languages (5 points)

We are primarily using python for this project. So far, we don’t expect to use any other languages unless we finish early and think of any new features that might require SQL or HTML or something like that. We chose python because it seemed to be the easiest way to build a game using pygame and pymunk without having to learn something new and complex like unity or unreal engine.

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

We are using pygame to create the window for our game as well as sockets to support multiplayer game hosting/joining. We also plan to use pymunk later to implement the physics of the game and tkinter to implement a drawing function to build terrain. So far, we’ve just focused on setting up the very basics of the game window and the functions to create/connect to a game instance.

# Execution-based Functional Testing (10 points)

We used vs community and vs code to open the window to make sure that the main menu buttons were correctly placed, the buttons brought the user to a password input screen, the user can use arrow keys to move the stand-in player character, and that the network sockets can connect to each other.

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

Since the code is pretty simple so far, this was done in concurrence with the functional testing, because it was pretty much guaranteed that if the code ran, it ran smoothly as well.

# Non-Execution-based Testing (10 points)

We didn’t really do much of this because again the code is pretty simple for now. Every time we made a change we could easily just run it quickly to see whether it worked or not.