

Software Implementation and Testing Document

For

Group 2

Version 2.0

Authors:

Grant Leatherman

Miles Brosz

Joey Riley

Caroline Mangrum

Kendall Fretwell

1. Programming Languages (5 points)

We are still 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.

2. 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 created the main menu and game hosting/joining implementation with simple squares for player characters.

3. 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 an ip/port input screen, and once the user hosts/joins a game, they can move their player character around and see the other player's character moving around.

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

We still haven't needed to do any of this testing because when everything runs, it runs smoothly. There's no lag or performance issues.

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

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