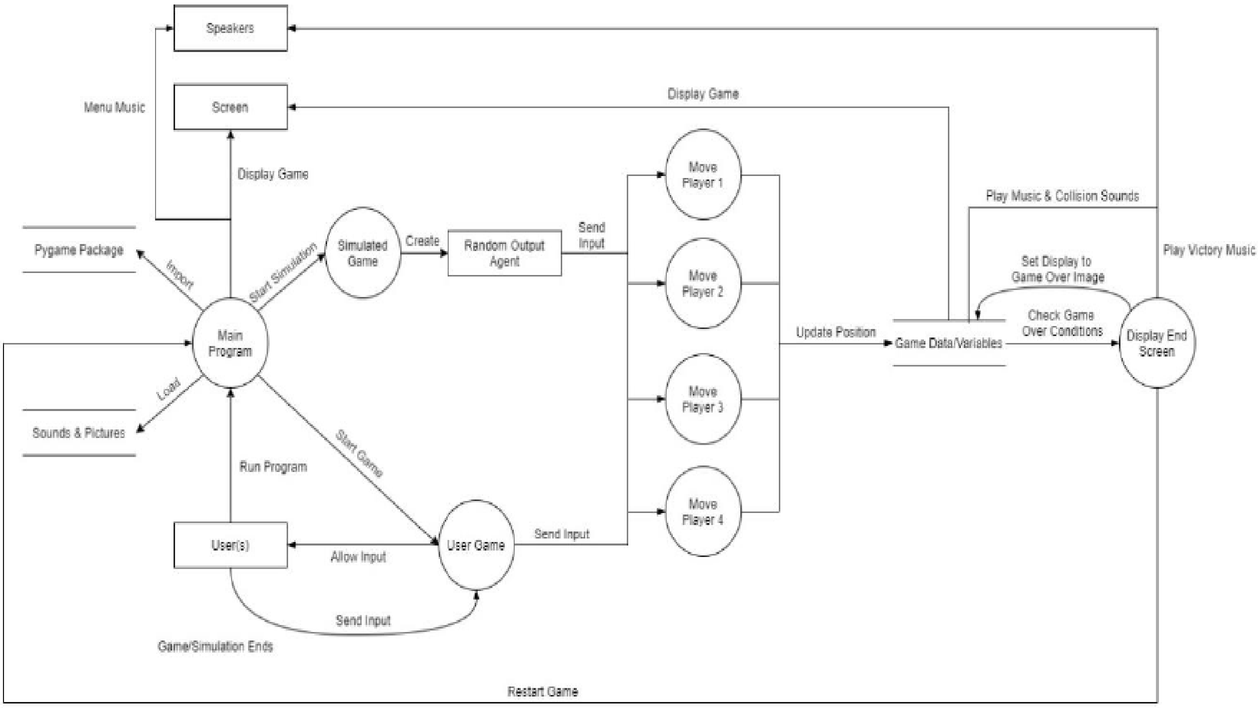
Programmer's Guide

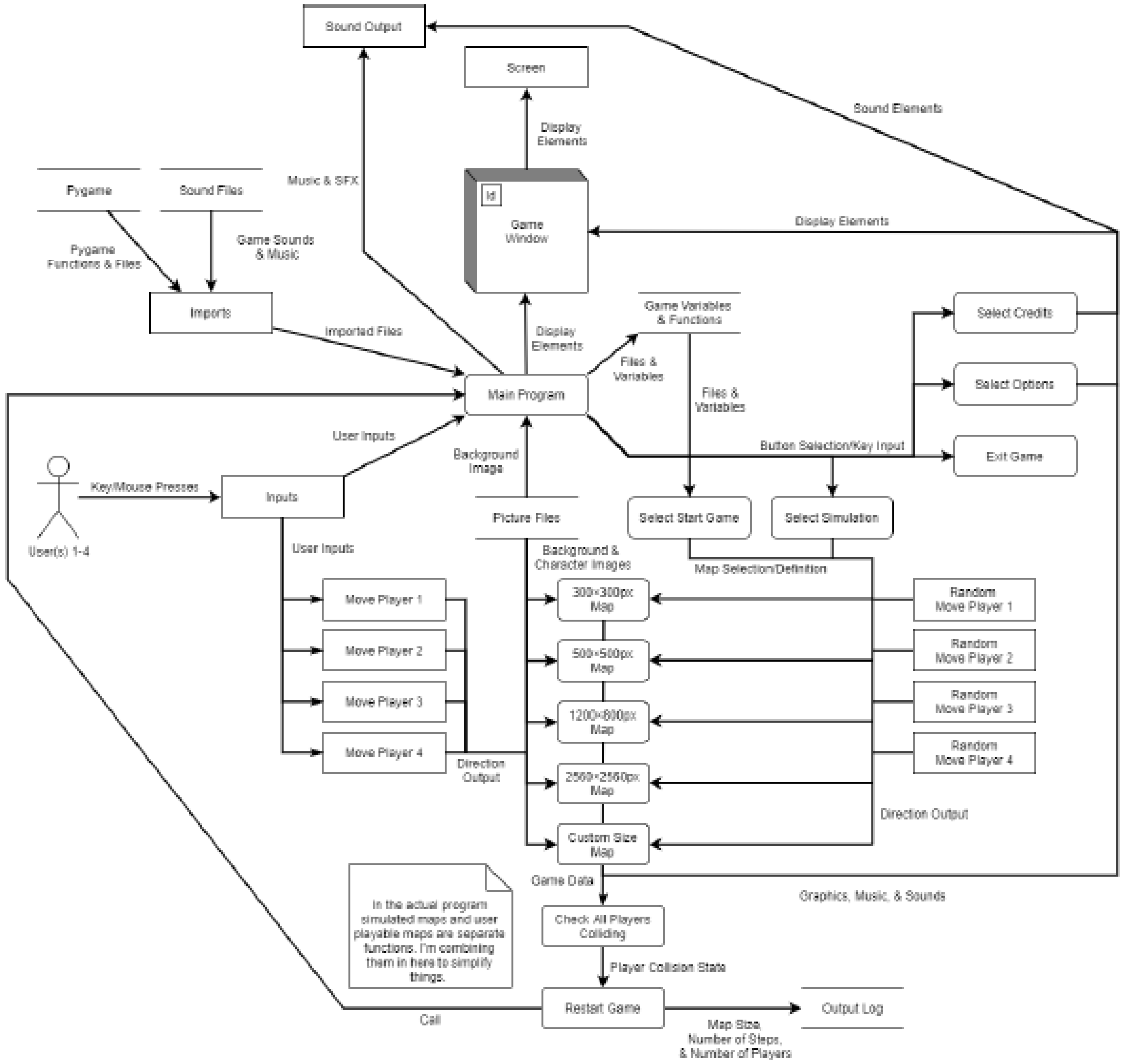
Section 1: Assumptions about the Programmer

To run the code, the programmer will need Python and Pygame installed, as well as PyCharm or another Python IDE to maintain it. This project was written in Python and is cross-platform. The code was written in the PyCharm environment and then shared and saved on GitHub. The visuals and sounds in the package are all free to use. The results will be saved in a folder in a pre-determined txt file. The average, shortest, and longest run times will be calculated using the data in the txt file. Students can then examine the data to understand how various parameters, such as player count and map size, affect the data and runtime.

Section 2: High Level Design



Section 3: More Detailed Designs



Section 4: Installation Instructions

The installation instructions are simple and straightforward, and they are written in a way that anyone without a technical experience can understand them. The following are the procedures to download and execute the programme:

1. The first step is to download Python onto your computer. For this:
    - Goto this link https://www.python.org/.
    - Click the download tab and select your computer's operating system.
    - Click the download link that says, "Latest Python 3 Release - Python 3.9.4". This will start your download.
    - After that simply follow instructions on for installing application.
    - For Window devices, you will be given the option to add Python to PATH. You will need to click the box associated with this option.
2. Then you must install a library to run this game: simply open command prompt and follow:

    For Windows: py -m pip install -U pygame –user

For Mac OS X: python3 -m pip install -U pygame --user

3. Now you can unzip the src folder from the zip file.
4. Now open command prompt and type “python menu.py” to run the game.
5. If you want to analyse multiple games that are saved, then type “python analyze.py”.

This is a step-by-step guide on how to download and install the software required to run the programme and, in turn, the game. Instructions on how to navigate the menu and pick different game modes can be found in the user handbook.

Appendix A: Implementation Code

Menu.py:

There are universal functions that can be used by all different modes, such as player1 movement, player2 movement, player 3 movement, and player4 movement, to update the player's x and y coordinates. A drawing option is also available for displaying the player and map image in real time (100FPS). The movement function includes collision detection functions; when a player collides with a wall, the collision function is called. The function record () is called after each stimulation to save the stimulation data in a text file for subsequent analysis. Every one of the map's many modes has its own function, which may be accessed directly from the menu.

In terms of stimulation, player movement differs slightly from manual movement. It determines a movement for the player at random rather than reading key input. It also features a built-in collision detection feature. Color, sound, frames per second, movement speed, and typography are all used in every mode. Additionally, pygame is the framework that connects all of the components.

Analyze.py:

A function named data sorter is used to find the average, longest, and shortest path based on the list of step data in the.txt file in the results folder. The menu has two functions: one for analysing stimulation data and the other for analysing data from the user's manual. According to the categories, the data is sorted and displayed.