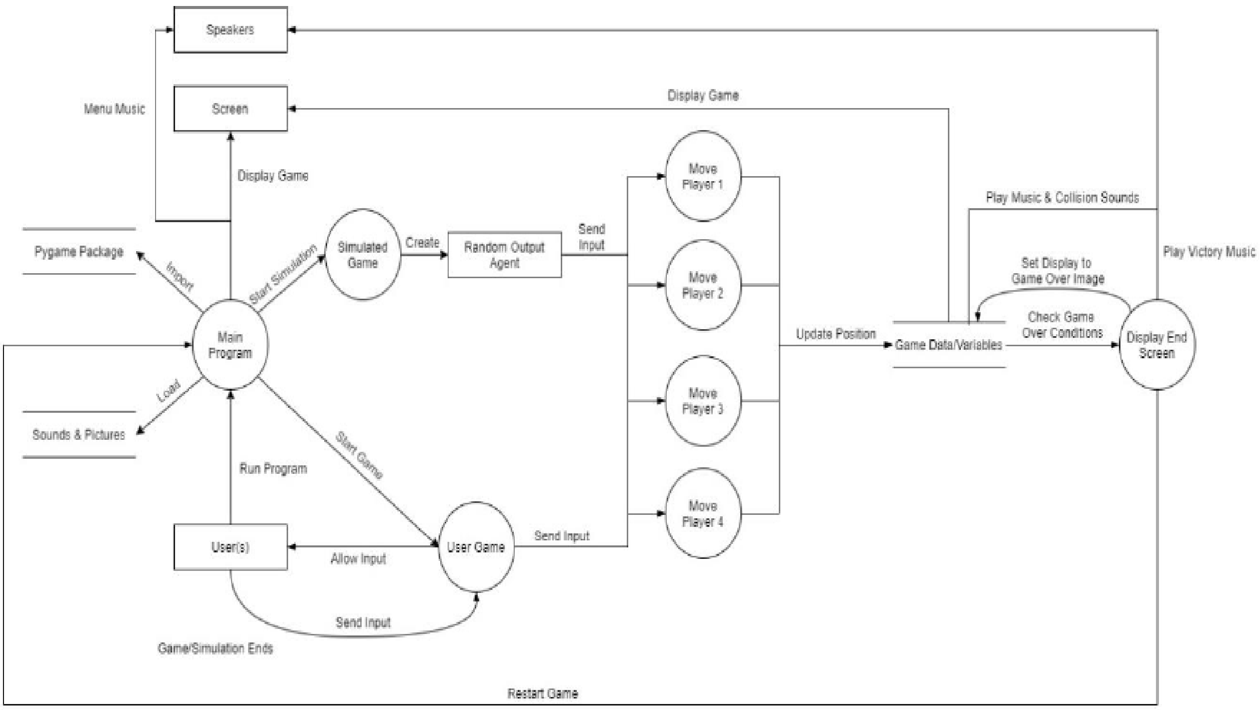
Programmer’s Guide

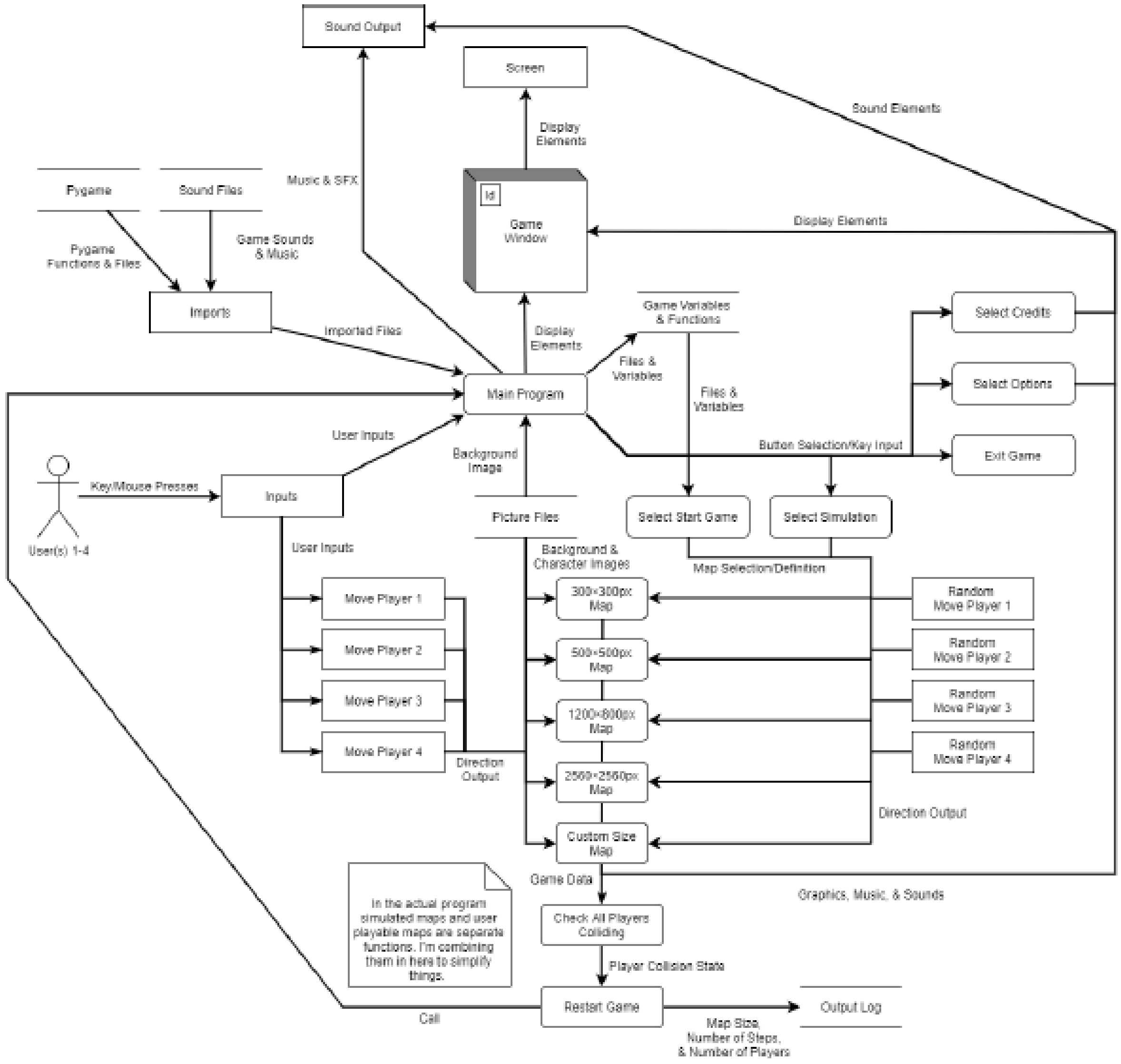
Section 1: Assumptions about the Programmer

To run the code, the programmer will need Python and Pygame installed, and they should maintain it with PyCharm or another Python IDE. This project is cross-platform and was written in Python. The code was produced using the PyCharm environment, and it was exchanged and stored on GitHub. The folder contains all graphics and sounds, which are free to use. The results will be saved in a pre-determined txt file in the folder. The data in the txt file will then be utilised to calculate the average, shortest, and longest run times. Students can then analyse the data to see how different options, such as player count and map sizes, 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. This can be done by:
   - Clicking 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 basic instruction on how to download and install the necessary software in order to launch the programme and subsequently the game. Please refer to the user manual for instructions on how to browse the menu and select different game modes.

Appendix A: Implementation Code

Menu.py:

For updating the x and y coordinates of the player, there are universal functions that can be utilised by all different modes, such as player1 movement, player2 movement, player 3 movement, and player4 movement. There is also a drawing function that may be used to display the player and map image in real time (100FPS). Collision detection functions are built into the movement function; when a player collides with a wall, the collision function is called. When each stimulation is finished, the function record () is called to save the stimulation data in a text file for further analysis. Each of the map's numerous modes has its own function, which may be accessed straight from the menu.

The player movement differs slightly from manual movement in terms of stimulus. Rather than reading key input, it chooses a movement for the player at random. It also has a collision detection feature built into it. Every mode uses the universal variables color, sound, frames per second, movement speed, and typeface. Furthermore, pygame is the framework that binds all the components together.

Analyze.py:

Based on the list of step data in the.txt file in the results folder, a function called data sorter is used to find the average, longest, and shortest path. The menu has two functions: one for analysing stimulation data and another for analysing user manual input data. The information is sorted and displayed according to the categories.