INSTRUCTIONS FOR RUNNING THE CODE

This project implements the generalized solver for a point robot using the Breadth First Search algorithm. It has a Node class having the attributes and methods useful for executing the BFS algorithm. Also, it keeps track of the parent node for backtracking after finding the goal state for the maze. It uses the deque class from collections module as the data structure to optimize the enqueue and dequeue operations. **NOTE: The obstacle space for the robot is defined using the half plane equations and it has 8 actions i.e up, down, left, right, upleft, upright, downleft and downright.** Please follow the following steps to run the code:

1. Run the file bfs.py to execute the code for the Maze generation and searching.
2. After running the above file, it will ask for the start and goal coordinates to generate the path. If you will enter any coordinate in the obstacle space of the maze, you will be prompted to enter them again.
3. The code prints all the visited nodes, and finally it backtracks the path to print the smallest path, also it prints the execution time.
4. For map generation and simulation of the algorithm, python's 'Pygame' module is used (NOTE: You may see some pixels overlapping with some of the obstacle edges, that is due to the line approximation).
5. To run the code you will require the following modules:
    a. time
    b. sys
    c. pygame
    d. copy
    e. collections
    f. numpy

6. The two test cases for the graph search are Start Node: (70,150), Goal Node: (260,180) and Start Node: (0,0), Goal Node: (400,300).
7. Github link for reference: https://github.com/savnani5/BFS-for-a-point-robot

PS: Feel free to contact me if you have any difficulties in running the code. Thank you.