READ ME

Given below are the instructions to execute the A* path planer algorithm for a differential robot (TurtleBot Waffle) in ROS2.

- In the map below is the obstacle space for the robot to operate in.
- The ‘Free Space’ is represented by the white space.
- The ‘Obstacle Space’ is represented by the red.
- The ‘Clearance Space’ is represented by the black.
- The map dimensions are 600*200 cm with (0,0) being in the bottom left corner.

Obstacle Space in cm

Instructions to execute the code:

1. Have all libraries downloaded and have ROS2 Humble installed.
2. Download the PROJECT3_WS to your computer from the github.
3. In your terminal enter the following commands:
    a. cd project3_ws
    b. colcon build --packages-select turtlebot3_project3
    c. source install/setup.bash
    d. ros2 launch turtlebot3_project3 competition_world.launch
4. After running the commands the Gazebo environment will open with the turtlebot in its initial position.
5. When you are ready to execute the path planer run these commands in another terminal.
    a. cd project3_ws
    b. colcon build --packages-select turtlebot3_project3
    c. source install/setup.bash
    d. ros2 run turtlebot3_project3 path_plan.py

6. After running the commands, the terminal will prompt you to enter the RPM1 and RPM2 values. We recommend entering RPM1=50 and RPM2 = 100.
7. Next, it will prompt the user to enter in the clearance value in cm. We recommend entering a value of 3 cm. After entering the map will begin to build.
8. After the map is built, the terminal will prompt the user to enter the x and y coordinates of the initial position and the angle of the bot. For the gazebo simulation, enter x = 50, y = 100, and orientation = 0.
9. Next enter your desired goal position on the other side of the map. Example: x = 570, y = 100.
10. After entering the values, the path planner will find the optimal pact and execute it in the gazebo simulation environment. This is how to execute the code.

Libraries used in the code:

- cv2
- heapq
- math
- numpy
- time
- rclpy
- rcply.node
- geometry_msgs.msg
- sys
- termios

Link to the GitHub Repository: https://github.com/calebmyersaz/Project3_Phase2_ENPM661