COSC81/181 – Robotics – Fall 2019

Programming Assignment 3 (10% over the final grade)

The purpose of this assignment is to introduce you to path planning.

Instructions

Please read carefully the following tasks and write the program(s) accordingly. *You should do this assignment individually.*

The instructions on how to submit the project assignment is on Canvas, at the following link

https://canvas.dartmouth.edu/files/5616794/download?download_frd=1
Please look also at the general guidelines for writing code:

https://canvas.dartmouth.edu/files/5585034/download?download_frd=1

The Task

Assume that you get the map of the environment through a `nav_msgs/OccupancyGrid` (http://docs.ros.org/melodic/api/nav_msgs/html/msg/OccupancyGrid.html). Write a function that reads the occupancy grid from the related topic, takes as input the start and the goal (both of them in the reference frame associated to the occupancy grid, which can be read in the header), search for a path in the grid, and returns the sequence of poses in that reference frame that the robot should follow. For each pose in the sequence, publish a message `geometry_msgs/PoseStamped` (http://docs.ros.org/melodic/api/geometry_msgs/html/msg/PoseStamped.html) to a topic called `pose_sequence`. Undergraduate students will only send poses x and y, ignoring the orientation, while graduate students need to also include orientation of the robot at each pose that will minimize the rotations for the robot.

To test your code, you can run the following node that will load a map and will publish into a topic called `/map`

```
rosrun map_server map_server /opt/ros/kinetic/share/turtlebot_stage/maps/maze.yaml
```

You can see the results through rviz, by adding the related topics for visualization.

Comments

Please comment on the choice of the algorithm and in the evaluation show some tests together with the computation time. Include also some images showing some of the sequence of poses.

Since you'll do this project on your own, the "Allocation of Effort" part of your report should be very easy to write.

Evaluation

Your programs will be evaluated based on both their functionality and their coding style. In the notes for writing programs, you can find an informal style guide to help give you an idea of what is expected together with the coding style that you should follow.

ROS usage (5):

- Submitted file contains a well-formed ROS package.
- Package is named correctly.
- Package dependencies are correct.
- Package is configured correctly to build executable.

ROS Correctness (5):

- Becomes a ROS node correctly.
- Subscribes to correct topic and processes callbacks appropriately.
- Publishes to correct topic.
- Publishes messages of the correct type.

Task (50):

- Correct behavior and implementation.

Style (20):

- No duplication of executable code?
- No magic numbers?
- Names match functionality?
- Adequate comments?
- Comments match code?
- Consistent formatting?

Documentation (20):

- Report is complete and clear.
- Required sections exist under readily identifiable headings.
- Free of typos and grammatical errors.