UDACITY

‹ Return to "Flying Car" in the classroom

DISCUSS ON STUDENT HUB

3D Motion Planning

REVIEW

CODE REVIEW 8

HISTORY

Meets Specifications

You've done a **fantastic job** in this project!

Keep up the good work!

Writeup

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

Nicely done writing a comprehensive write-up including statements to explain how each rubric item was addressed.

Explain the Starter Code

The goal here is to understand the starter code. We've provided you with a functional yet super basic path planning implementation and in this step, your task is to explain how it works! Have a look at the code, particularly in the `plan_path()` method and functions provided in `planning_utils.py` and describe what's going on there. This need not be a lengthy essay, just a concise description of the functionality of the starter code.

Implementing Your Path Planning Algorithm

Here you should read the first line of the csv file, extract lat0 and lon0 as floating point values and use the `self.set_home_position()` method to set global home.

Nicely done setting the home position to the first line from the csv file!

Here as long as you successfully determine your local position relative to global home you'll be all set.

Good job retrieving the local position from the current global position!

This is another step in adding flexibility to the start location. As long as it works you're good to go!

This step is to add flexibility to the desired goal location. Should be able to choose any (lat, lon) within the map and have it rendered to a goal location on the grid.

🎉 The code has been modified to be flexible in setting the desired goal location. Any (lat, lon) position within the map will be rendered as the goal location on the grid.

Minimal requirement here is to modify the code in `planning_utils()` to update the A* implementation to include diagonal motions on the grid that have a cost of sqrt(2), but more creative solutions are welcome. In your writeup, explain the code you used to accomplish this step.

🎉 The code has been modified to include diagonal motions and the writeup does include an explanation on how this step has been accomplished.

For this step you can use a collinearity test or ray tracing method like Bresenham. The idea is simply to prune your path of unnecessary waypoints. In your writeup, explain the code you used to accomplish this step.

Awesome job using the collinearity check to prune unnecessary paths from your graph!

Executing the flight

At the moment there is some mismatch between the colliders map and actual buildings in the scene. To ensure success build in a 5+ m safety margin around obstacles. Try some different goal locations. Also try starting from a different point in the city. Your reviewer will also try some random locations so be sure to test your solution! There is no firm constraint or requirement on how accurately you land exactly on the goal location. Just so long as your planner functions as expected.

Superb job! The drone does successfully fly from the start position to the goal position without any issues!

DOWNLOAD PROJECT

8 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review