COMP 417, Fall 2019

Assignment 1: Path Planning

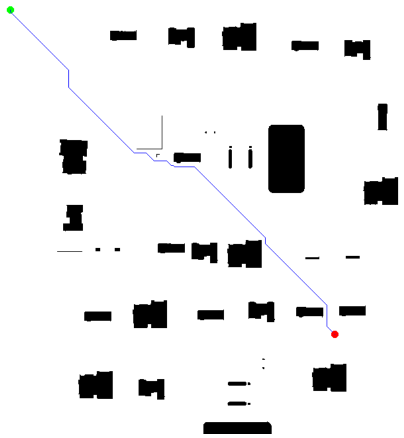
Out: Sept 24, 2019

Due: Oct 8, 2019 – 6pm

Question #1: Implement A* Planning

We have provided you Python code, called “astar_planner.py” that loads a 2D map in the form of a floorplan image. Given a start and a goal, this code now runs Dijkstra’s Algorithm using a simple implementation and finds a plan like the one shown below.

Your job in this question is to modify the algorithm to correctly run the A-star planner. We have left comments “#TODO” in each place where a modification is needed. You should read and understand all of the code (there’s not too much) but will only need to type a very few lines correctly to complete the question (so don’t over-think it!)



Steps to set up:

Obtain the sample code by running:

```
$ git clone
http://github.com/dmeger/COMP417_Fall2019.git
```

Run the sample code by running:

```
$ cd COMP417_Fall2019/assign1_comp417/python
$ python astar_planner.py ../worlds/map.pkl
```

Coding your new planner:

Now modify the code in the places that say #TODO. This is where you’ll need to do most of your thinking, make sure you understood the algorithm presented in class and dig into the code to figure out how to implement your changes. Run it with the same commands as above to test.

Report the results:

Save 3 images of paths produced by your planner. Keep the starting state the same, but set a different goal for each image, and make sure those cover the map well. Save the images as “astar_result[0|1|2]_yourname.png”. You will hand-in these images plus your modified code.

Question #2: Implement RRT Planning

In this case, we have given you the framework for an RRT planner, in “rrt_planner.py” which initially doesn’t grow the tree beyond the starting node. You must complete the implementation including the sampling, steering to new nodes and the overall algorithm structure, in order to build the RRT tree and find the plan. When completed, your code should show an image like the one to the right.

Setup, coding and reporting follow the same pattern as described for Q1 (only work with the rrt_planner.py code this time).