LAB 4: Path Planning (Robot)

Due: 10/31 at 3:00 pm

The purpose of lab 4 is to develop your own pathfinding capabilities for Cozmo. In this second part you will take your RRT implementation from the first part and use it to help Cozmo navigate its physical environment. We have given you an updated copy of `rrt.py` which includes a framework for running Cozmo alongside the visualizer, you can either copy your RRT functions into it or copy the Cozmo functions and new main method into your previous file. We have also made some small bugfixes in `cmap.py` and `gui.py`, so you should update those files. For this assignment, you should complete the following:

`CozmoPlanning`: This method is executed by the RobotThread and should contain all your Cozmo behavior. The goals for this behavior are to identify a target cube and use RRT to find a path to a specific face, follow the path found by RRT, and replan to avoid any obstacle cubes that are added during navigation. As the behavior is running the map should be updated to reflect any new cubes that Cozmo sees, though we will not be moving them once they are placed so you don't need to account for that. Obstacle cubes can be added at any point during the demo, however.

Note that you will need to implement a way to retrieve the path from RRT yourself as part of path following. If the target cube is not visible from Cozmo's starting location, Cozmo should navigate to the arena center to look for it and navigate to it once it is seen. To run Cozmo's behavior simply execute the updated `rrt.py` file.

You cannot use `go_to_pose` in this lab as it would largely defeat the purpose, however you are not required to use `drive_wheels`. During our testing we found that `drive_wheels` did not update Cozmo's pose like the other drive methods do, which can lead to issues detecting cube positions correctly. To help identify specific cubes the snippet `robot.world.light_cubes[cozmo.objects.LightCube#Id].object_id` can be used to lookup matching object ids, where # is the number of a cube (1-3, it can be found in one of the hollow areas next to the battery screw).

Evaluation: Your robot behavior will be demoed in class, and your RRT implementation will be autograded based on your submission as described in the first part. The rubric is as follows:

RRT implementation, autograded with 6 maps, 10 points per solved map	60 pts
The robot follows the path found by RRT	15 pts
The robot identifies a target cube and navigates to a specific face	10 pts
The robot replans to avoid obstacle cubes	10 pts
The map is updated to reflect newly seen cubes	5 pts

Submission: By 3pm on Tuesday, October 31st 2017, submit only your final `rrt.py` on T-square. Make sure your code is entirely contained within this file. If you relied significantly on any external resources to complete the lab, please reference them in the submission comments.