LAB 6: WAREHOUSE AUTOMATION

Due: December 5th, 11:59pm EST

In this lab, your robot will be tasked with automating a warehouse. This project consists of two phases - an exploration phase and task phase. First, the robot will have to map out its environment using a lot of the same algorithms you have employed throughout the class. Then, it will have to collect packages (landmarks) while avoiding obstacles. If an obstacle is hit, the run is terminated. In the first phase, your goal is to visit all of the landmarks within 4 minutes, while in the second phase, your goal is to pick up as many packages as possible within 2 minutes.

Phase I (exploration):

Map Layout:

The map is organized as an occupancy grid, where each cell in the grid is either labeled “free”, “obstacle”, or “landmark”. In addition, every cell in the map starts out as unexplored, and as grid cells enter the robot’s field of vision, the cells are detected by the robot’s sensors and should be marked as explored.

Localization:

Similar to project 3, markers can be examined by sensors, and a reading of their position and heading is acquired for these markers once they’re in the robot’s field of view. All 5 landmarks must be visited in order for the exploration phase to be awarded full credit.

Planning:

You will implement frontier-based exploration in **exploration.py** in order to traverse the map and visit all of the landmarks. Your frontiers are the unexplored cells in the map that are adjacent to the explored cells, and your algorithm should move to the centroid of the frontiers. You are given an implementation of RRT to use when navigating to your computed frontiers.

Motion Model:

Your warehouse robot is a differential drive robot, so you will have to set the speed of the left and the right wheel accordingly to achieve the desired velocity to reach different positions. You can tune a PID controller to adjust your motion updates if you’d like; make sure to extensively test on the given map JSON file.

Phase II (tasks):

Planning:
The locations of the markers from the previous phase will be provided to you, as well as their type (“L” for left-facing, “R” for right-facing, “U” for upward-facing, “D” for downward-facing). You must move to the marker location and orient yourself in the direction of the marker to pick it up. For example, if the marker has direction “U”, the robot must move to the location of the marker and face the top of the map to pick it up. Again, you can use the given RRT implementation to plan paths to markers. You may implement the functions in **tasks.py**.

Control:
In this phase, different kinds of noise will be injected into the robot wheel velocities such as drift, offset, and random walk noise. You will be implementing a PID controller like project 5 and tune for your differential drive robot’s motion model in **utils.py**. You will be given 4 maps to test on, and there will be 4 hidden maps of varying sizes that the autograder tests your implementation on.

To run your either phase, run **robot_gui.py** with the phase and desired map specified in the main method. Make sure to take advantage of functions in **utils.py**, and remember that your run for either phase will terminate if you collide with an obstacle.

Grading:

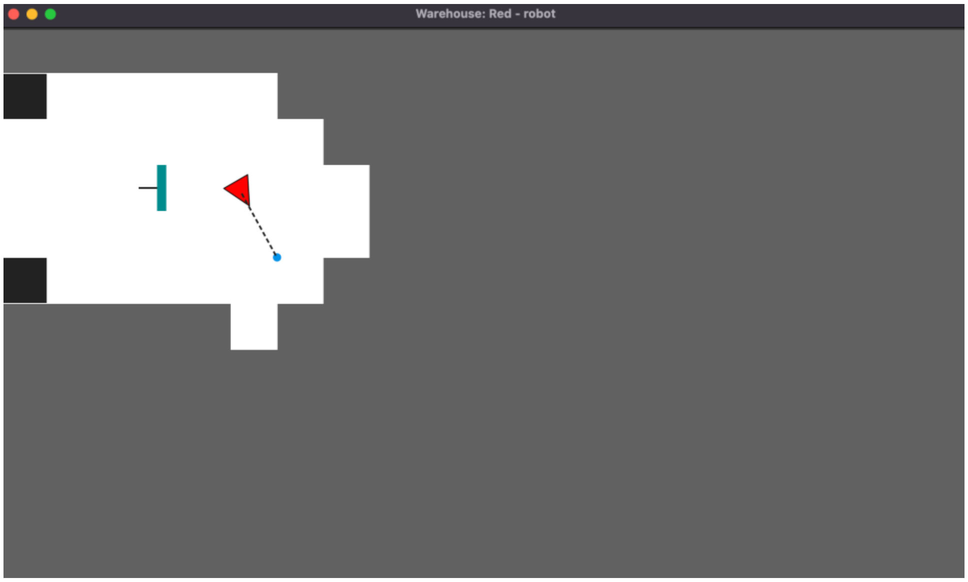
Grades will be assigned as follows (four cubes must be delivered for full credit):

Rubric	
Find all 5 markers in Phase 1	12.5 pts/map
Approach all 5 markers in Phase 2	12.5 pts/map

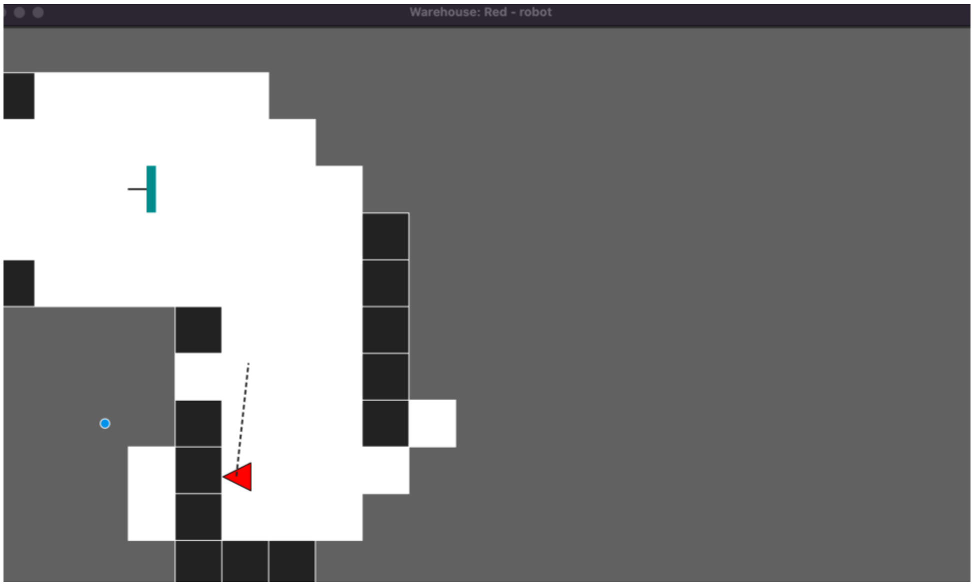
Submission: Submit your **exploration.py**, **tasks.py** and **utils.py** files to GradeScope. If you relied significantly on any external resources to complete the lab, please reference these in the submission comments.

Examples:

Robot should navigate towards frontier in Phase 1.



Robot can use RRT to get around obstacles in either phase.



Robot has to be facing in the direction of the marker in order to pick it up in Phase 2.