6.S078 ASSIGNMENT 1

WALTER MENENDEZ

Approach

Robot representation. At first, one of the problems I had was just wrapping my head around how to represent the robot's internal points. I didn't want to use a raw pixel representation as I needed to discretize its location somehow. At the same time, using the discretized location for plotting seemed too constrained for actually plotting the graph. I ended up keeping track of an initial geometry that I offset using a discretized initial location as offset.

Planning. My plan was as follows: I have the robot formulate a plan, and if every possible configuration of its vertices along that path would not collide with any obstacle, then we go ahead and move to that space. That's how I perform "along the path" collision checking. However, I don't bother with this check if the target itself is in an obstacle.

If I can't move to that space, I have the robot back up into its most recently known "safe" space, the last valid location it was able to be in, and I replan from this state, blacklisting the space I tried to enter.

There's one obvious deficiency: when I check to make sure that my target space, and every possible configuration of the robot's vertices, can traverse the space, I have to ask every obstacle if I can enter that space.

Performance

I worked with three test environments: one simple convex robot and one obstacle, one robot and two obstacles, and a multi-shaped robot with two obstacles.

For each test case, the robot was positioned at the origin. The obstacles were positioned in such a way that the robot was going to collide with it on its initial planning. The lengths of paths reported here then are the lengths of the final path that was traversed.

For heuristic exploration, my admissible heuristic was Euclidean distance. A quick inadmissible heuristic was then computing the square of that distance, as square roots are a costly computation.

	1 robot, 1 obstacle	1 robot, 2 obstacles	2 shape robot, 2 obstacles
BFS	54	72	73
DFS	402	432	432
Dijkstra	42	72	69
A*, admissible	42	72	69
A*, inadmissible	42	72	69

For times (reported in seconds):

	1 robot, 1 obstacle	1 robot, 2 obstacles	2 shape robot, 2 obstacles
BFS	0.023262	0.022169	0.022739
DFS	0.014919	0.01482	0.015284
Dijkstra	0.61939	0.634076	0.667952
A*, admissible	0.00186	0.00175	0.001743
A*, inadmissible	0.001652	0.001692	0.001706

Path printing

In each set of figures, starting from the top left: BFS, DFS, Dijkstra's, and then A* with the admissible heuristic.

FIGURE 1. Test Case 1

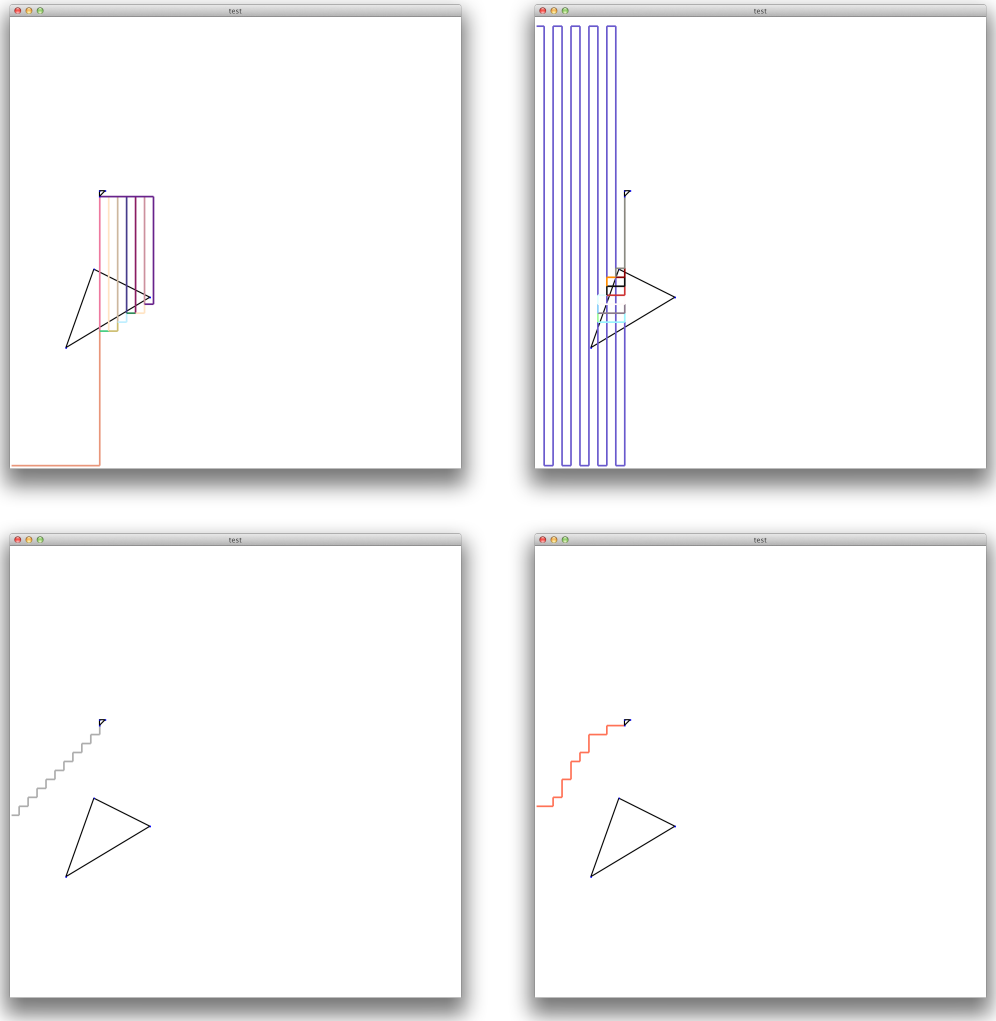

FIGURE 2. Test Case 2

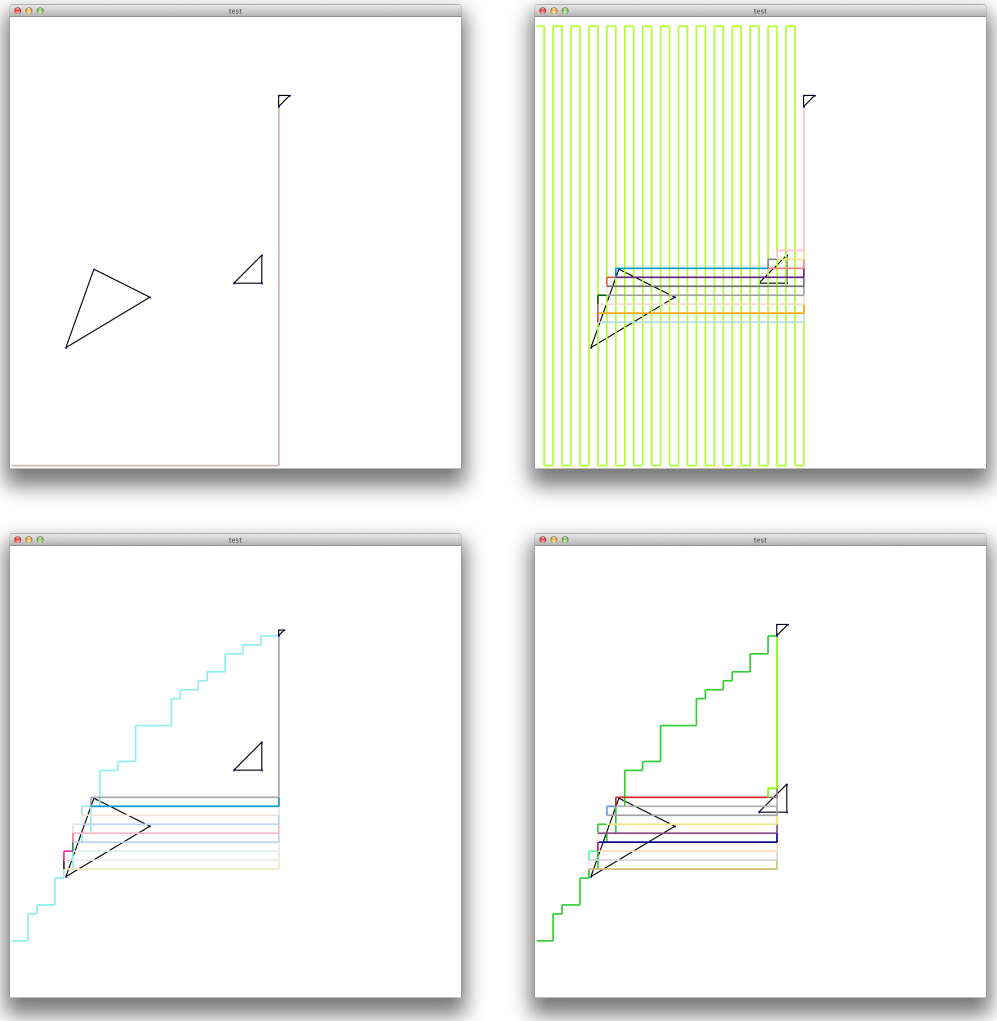

FIGURE 3. Test Case 3