Lab 2 help:

Part A:

This is how the bot actually moves:

top.do({’visit’:[’o109’,’storage’,’o109’,’o103’]})

Chart, line chart

Description automatically generated

Start playing with ”walls”.

Change the slope of the first wall. It worked

env = Rob\_env({((20, 20), (35, -5)), ((70, -5), (70, 25))})

This code ”mirrors” the actual first wall and causes the robot to crash.

Chart, line chart

Description automatically generated

The reason it crashed is, the robot only checks one of its sides, in

this case its the right side and determines to turn according to that but doesn’t follow the wall. So, when the wall is also on the left side, it still makes the

wrong turn.

Another solution:

to make it crash should be making a more difficult barrier to dodge. To accomplish that I modified the walls making them form a 90o obstacle:

Graphical user interface, text, application

Description automatically generated

This results on the agent going in circles for a couple of seconds until trying a new path and successfully reaching the goal:

Chart, line chart

Description automatically generated

Another solution:

Graphical user interface

Description automatically generated with low confidence

Part B:

The following code implements a robot trap. Write a controller that can escape the “trap” and get to the goal.

Text

Description automatically generated

To solve this exercise, first I execute the program to observe how the agent behaves.

Chart, box and whisker chart

Description automatically generated

As the graph shows, the agent tries to dodge the obstacle unsuccessfully and crashes. So the obvious thing to do is adding two new points in the controller so the agent do not enter the trap.

Text

Description automatically generated

Shape

Description automatically generated