Name: Theodoro Gasperin Terra Camargo
McGill ID: 260842764

Assignment 1 COMP 417

Question 2: Simple Point Robot

A) Uniform Sampling

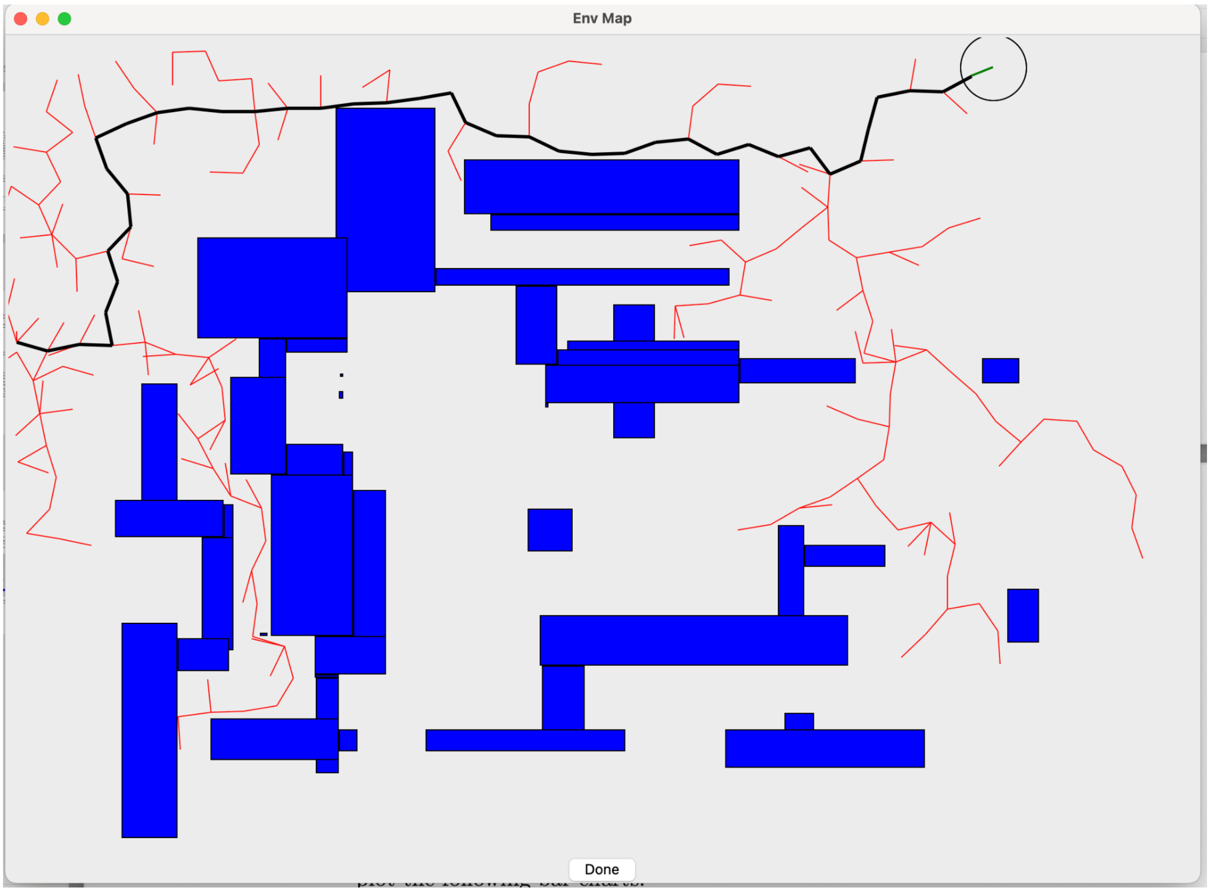

Figure 1: Point robot results of path search with RRT algorithm in the shot.png map.

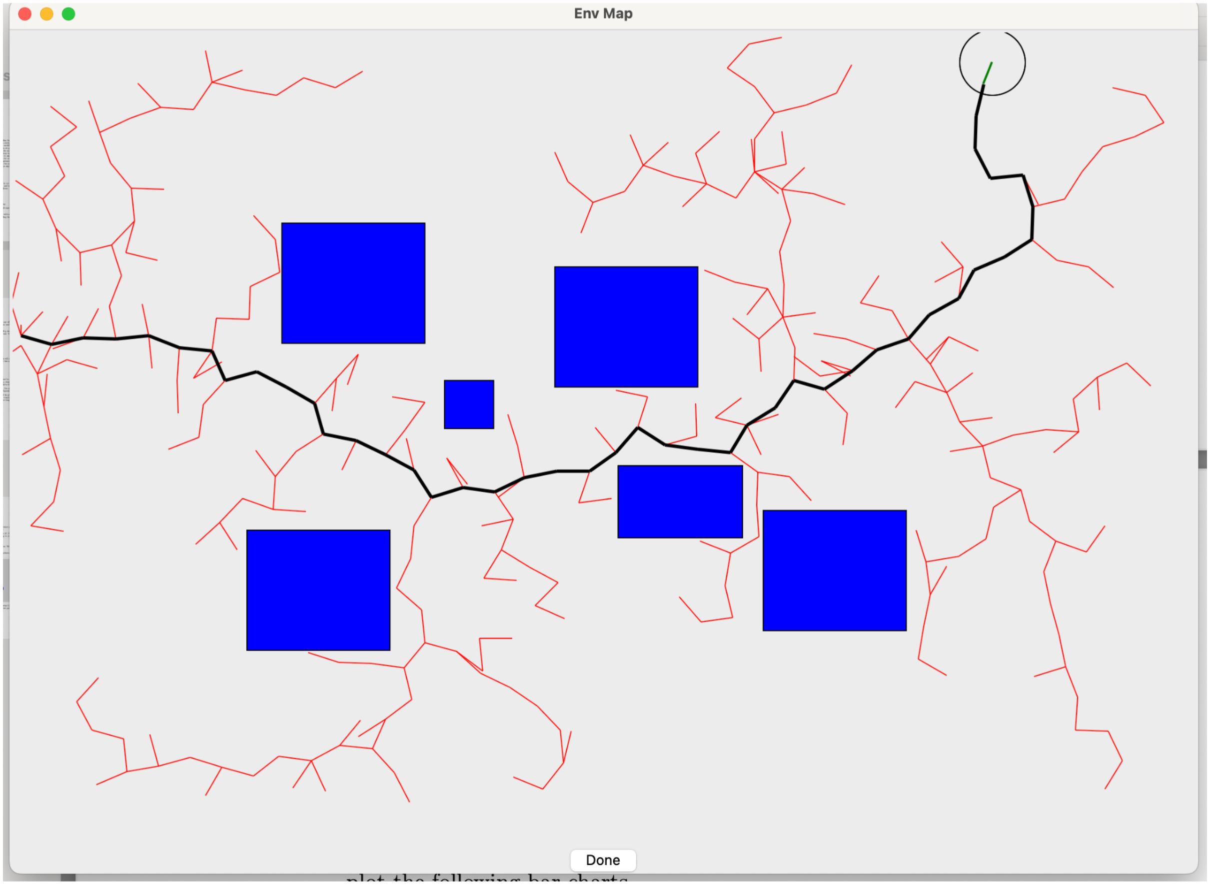

Figure 2: Point robot results of path search with RRT algorithm in the simple.png map.

I encountered difficulties in figuring out when a line hits or passes over a rectangle. But I eventually developed a way to check for all the conditions in my function lineHitsRect.

B) Gaussian distribution

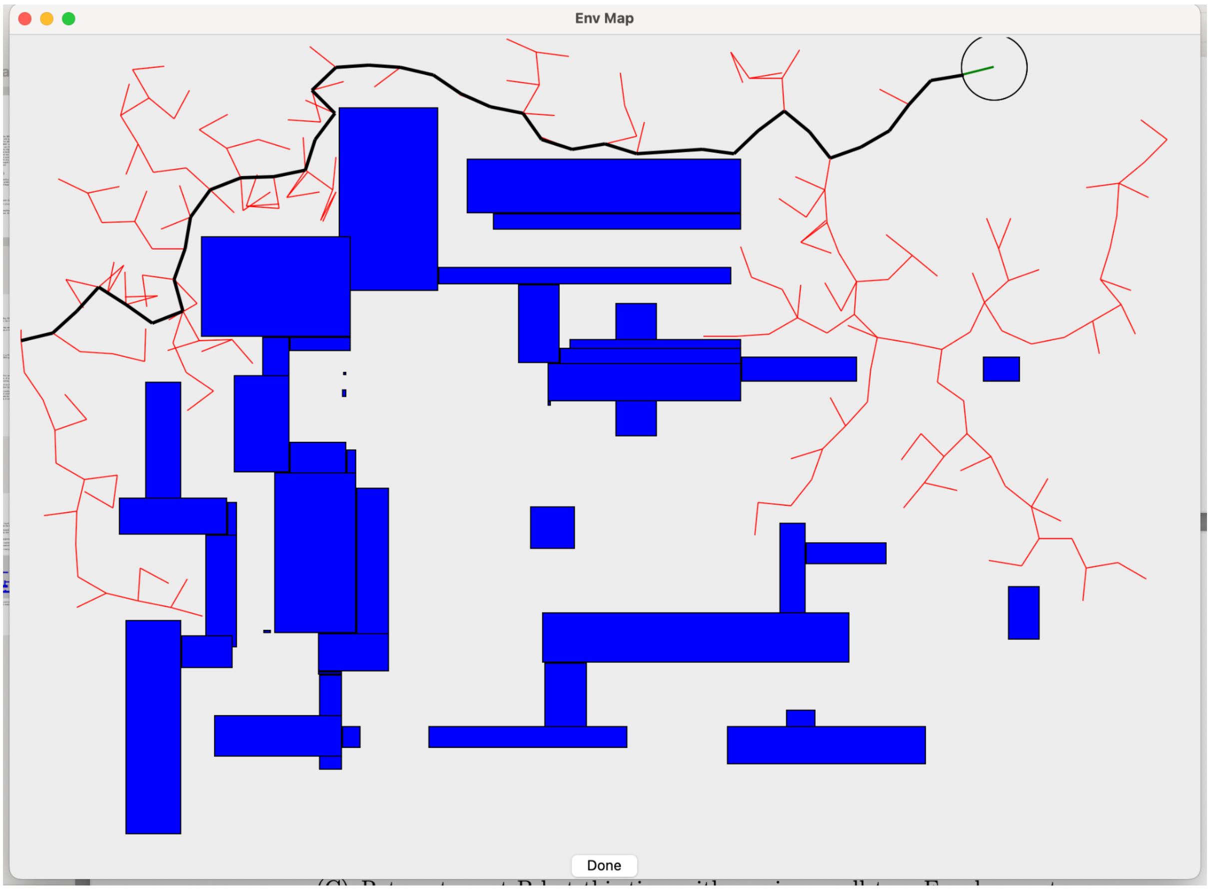

Figure 3: Gaussian distribution point robot results of path search with RRT algorithm in the shot.png map.

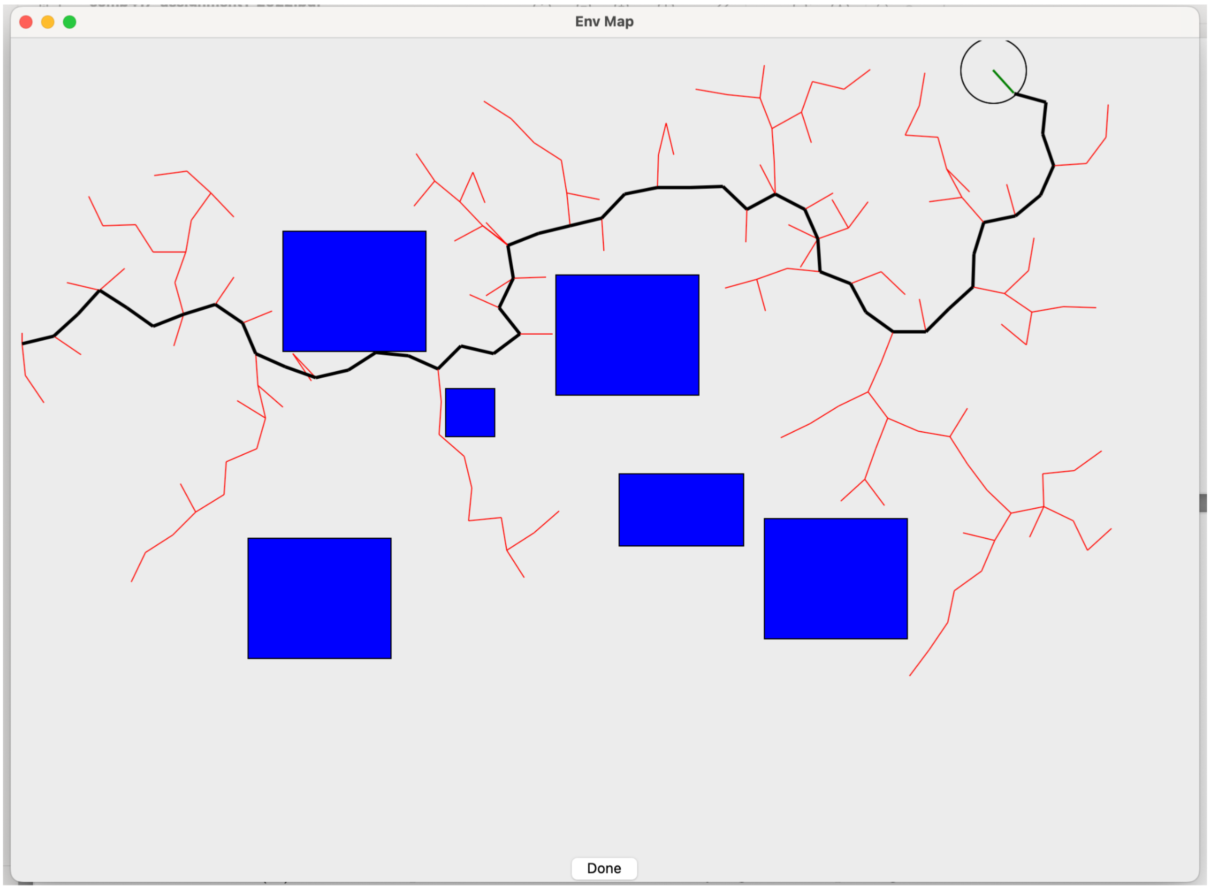

Figure 4: Gaussian distribution point robot results of path search with RRT algorithm in the simple.png map.

Using gaussian distribution does have an effect on the planner. Because the points are generated more centered at the goal, the RRT algorithm explores less minor branches and focuses on the branches toward the goal, even creating new branches on the tree, but achieving the goal state much quicker than using uniform distribution. Using the gaussian distribution, RRT algorithm achieved the target goal by creating a tree with 190 nodes in the simple map and 239 nodes in the shot.png, whereas in the uniform distribution it used 214 and 325 nodes for the respective maps. RRT was run using a step of 30 px.

C)

a)

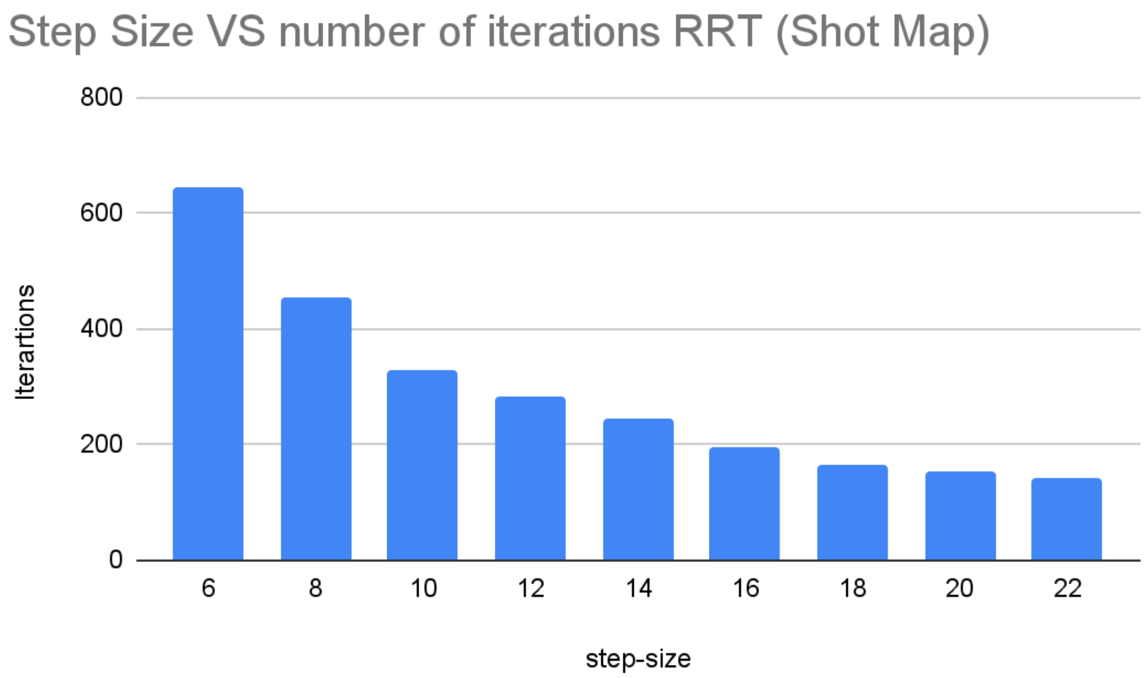

Figure 5: Step size vs number of iterations with RRT algorithm in the shot.png map.

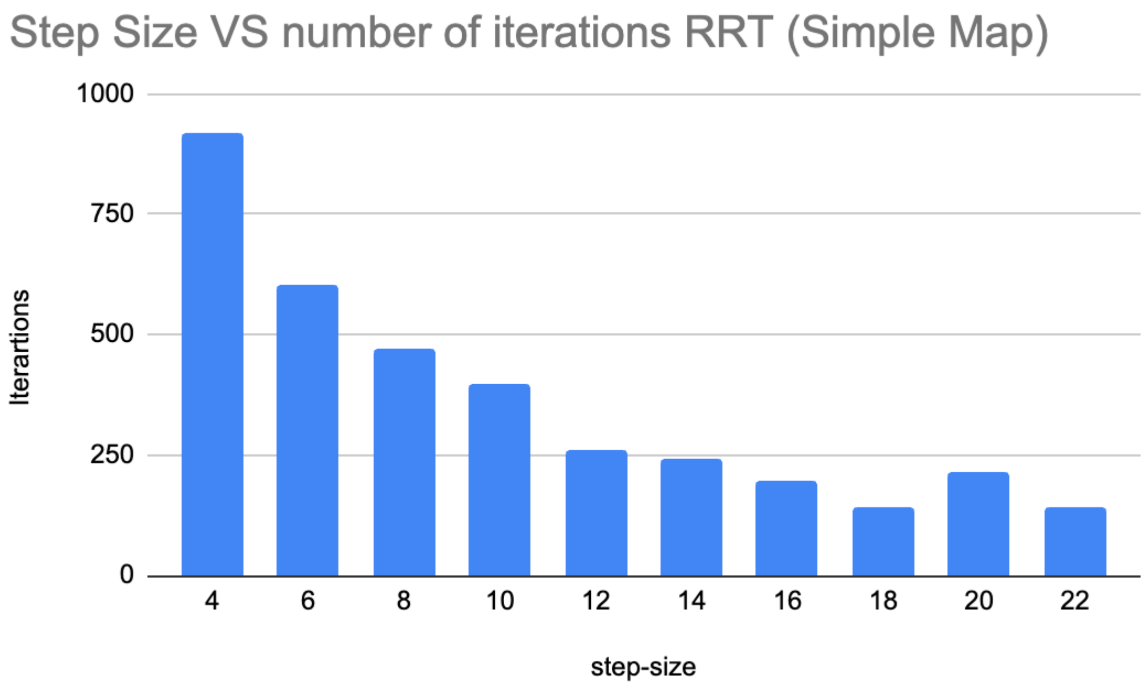

Figure 6: Step size vs number of iterations with RRT algorithm in the simple.png map.

b)

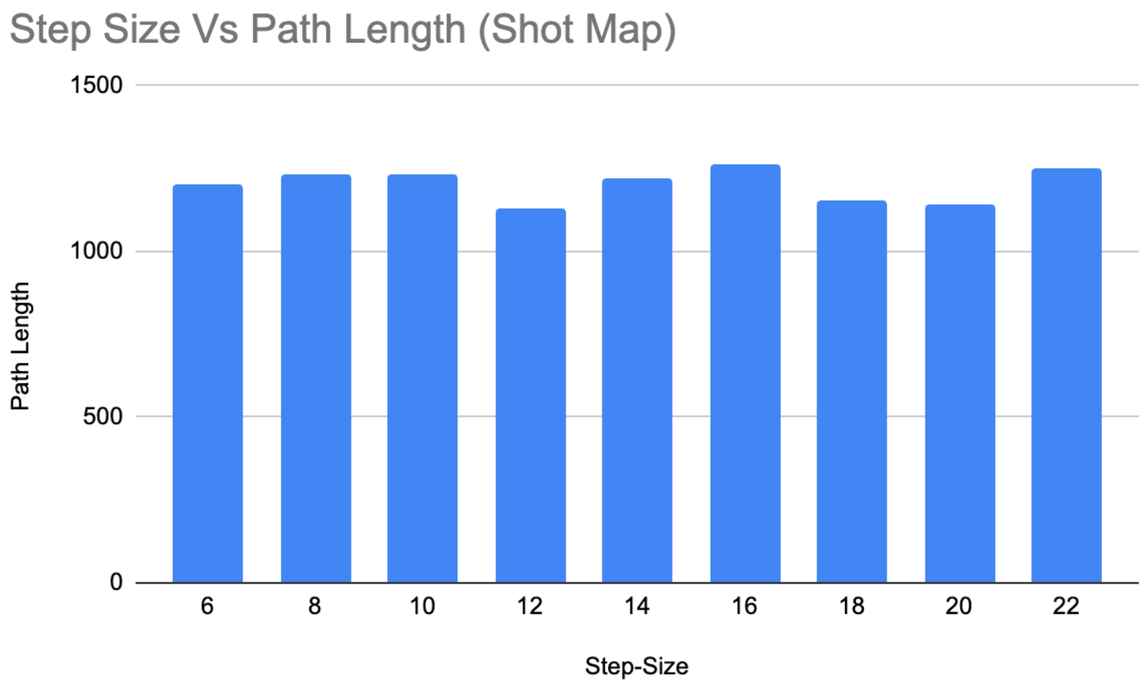

Figure 6: Step size vs path length with RRT algorithm in the shot.png map.

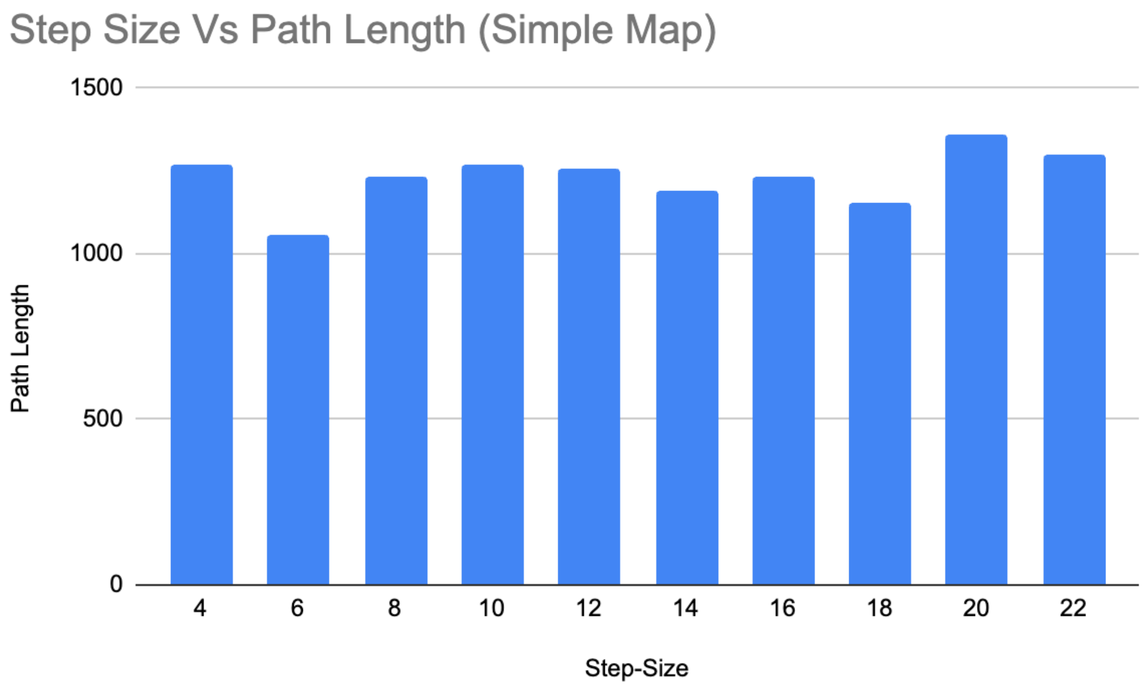

Figure 7: Step size vs path length with RRT algorithm in the simple.png map.

I propose that a step size of 18 px is a good choice for our point robot, as the size of the tree seems to decrease in a logarithmic way, and the path length does not seem to change according to the step size. Thus, I believe that a step size of 18 is a good choice because it

can circumnavigate the obstacle in the maps with certain ease and precision, whereas bigger steps start to lose precision and start to increase the path length.
For this question I changed my code so that every 15 steps the random point generated was the target, so that the program could run faster.

Question 3:

A) To implement the line robot, I modified the function newPoint of my code. This function, on the line robot, takes into consideration the angle, and the extra size of the line robot and returns a new point (x, y) if the robot was to go to the new point.

   I then checked if this point is valid in the RRT search function and if it is valid it will add the new point without the extra size of the robot to the graph.

B)

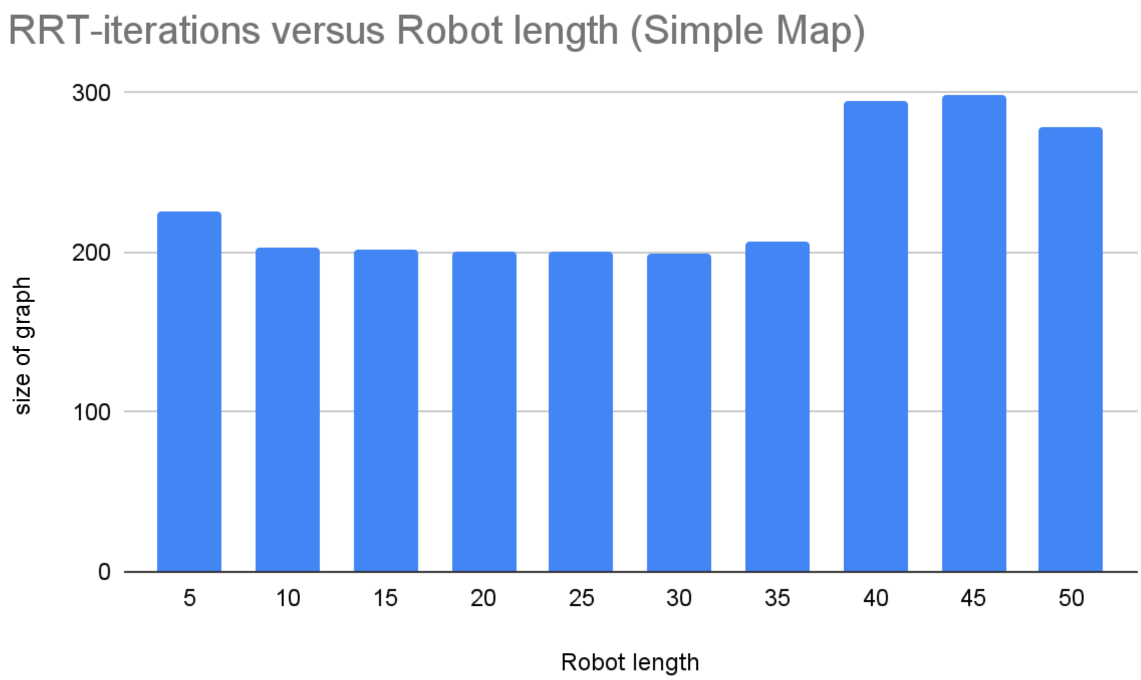

Figure 8: Robot length vs size of the graph with RRT algorithm in the simple.png map.

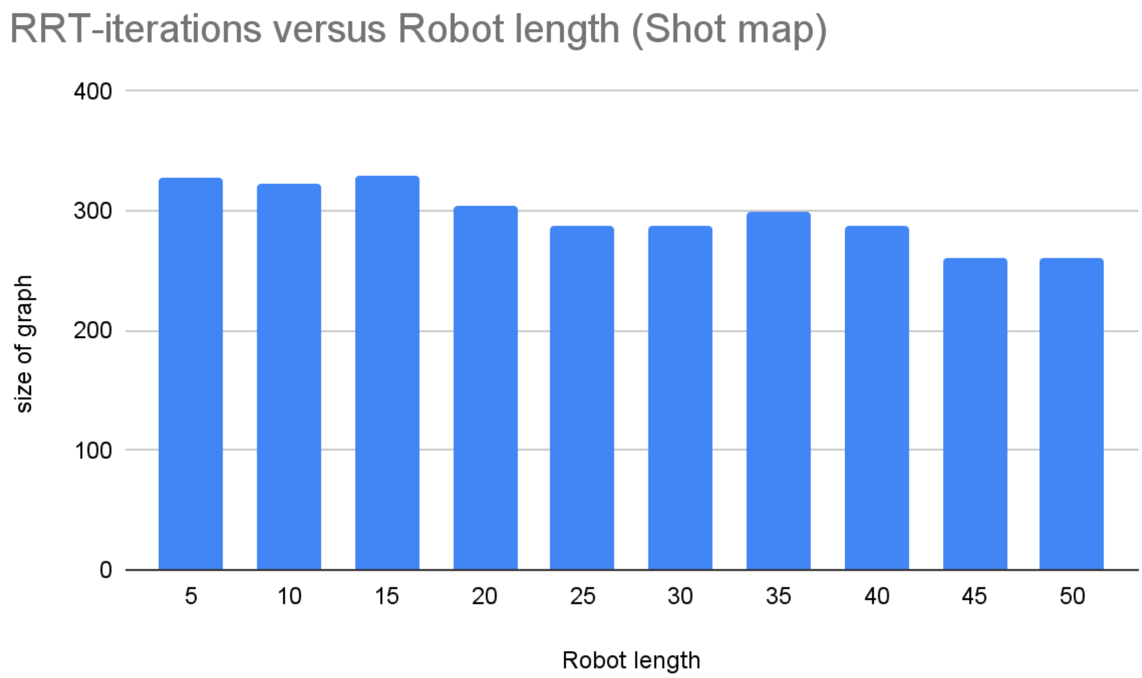

Figure 9: Robot length vs size of the graph with RRT algorithm in the shot.png map.

From both results, we can infer that the robot length has a minor effect on the number of nodes in the graph in the RRT algorithm. It seems that the number of nodes is more associated with the difficulty of the map as in the Simple map the number of nodes increased in length from 40 to 50px but decreased in the Shot map. It is evident that the bigger robot lengths will prevent the addition of new points in the graph that are too close to the obstacles as the robot would not be able to turn otherwise. Therefore it seems that a change in the robot length has the effect of distancing the graph from the obstacles, but it is not so clear its effect on the number of iterations of the rrt graph as the number of nodes does not show a clear pattern in our results.