Jonathan Pearce 260672004

October 31, 2017

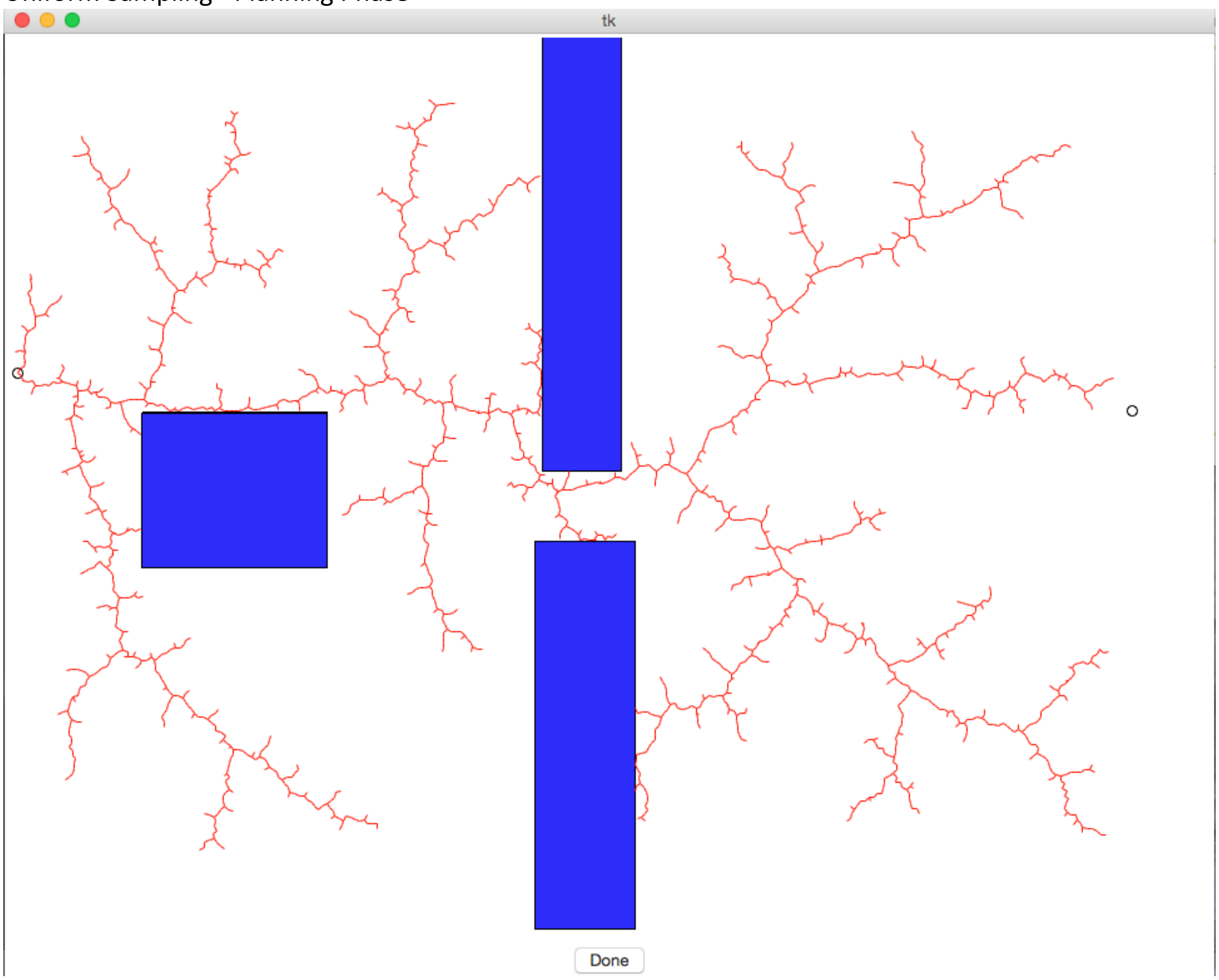
Comp 417

Professor Dudek

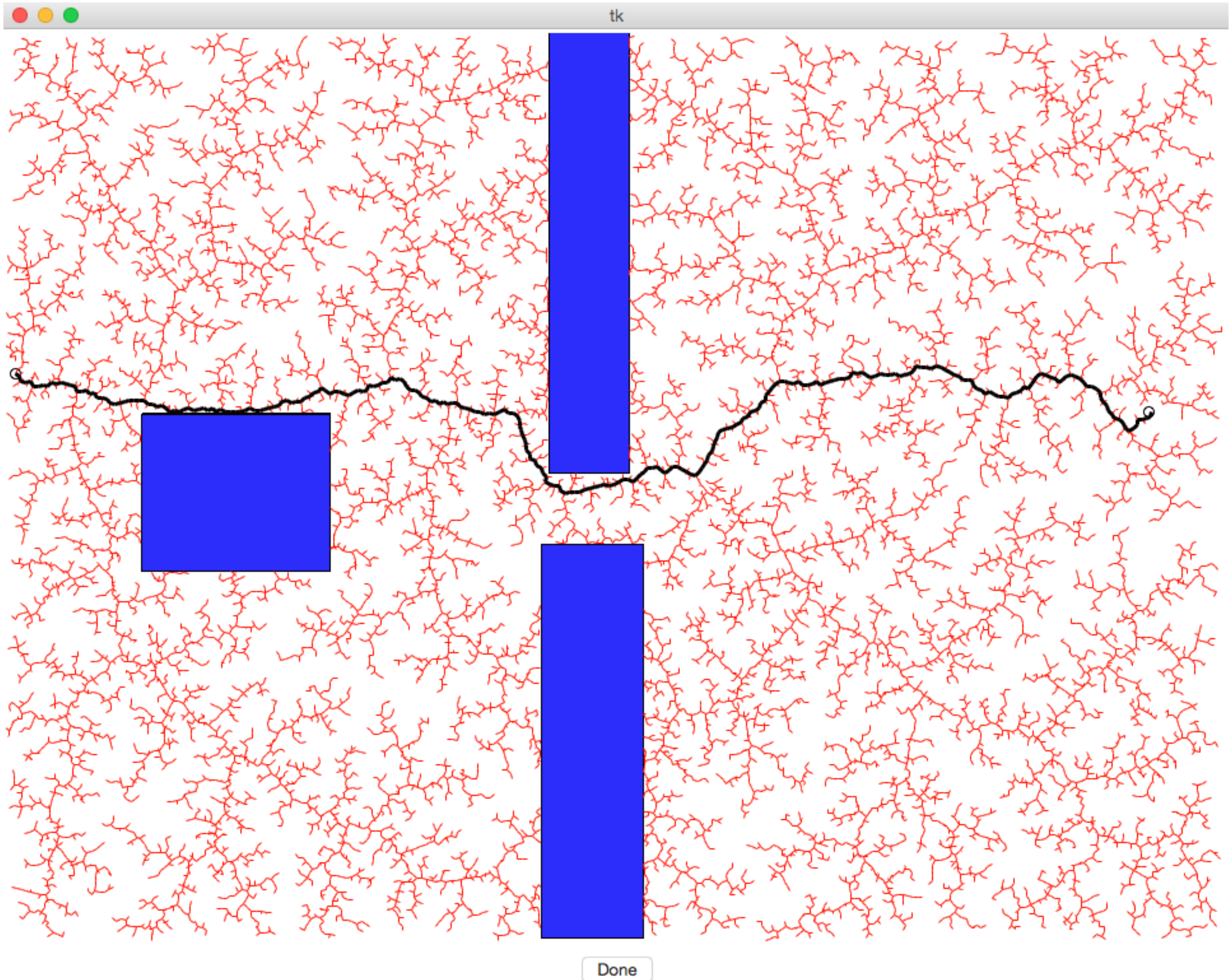
Assignment 2

Part 1)

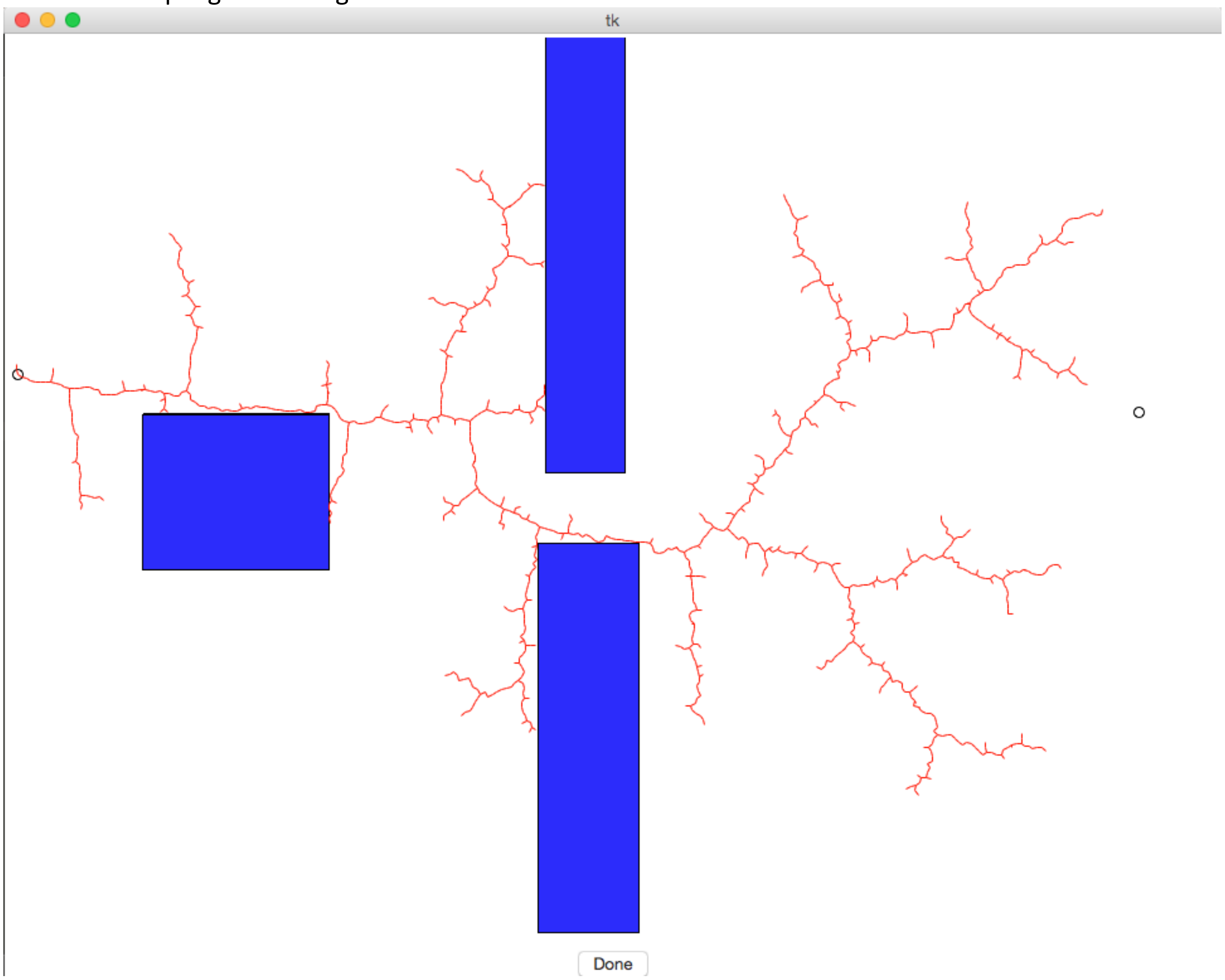
Uniform Sampling - Planning Phase



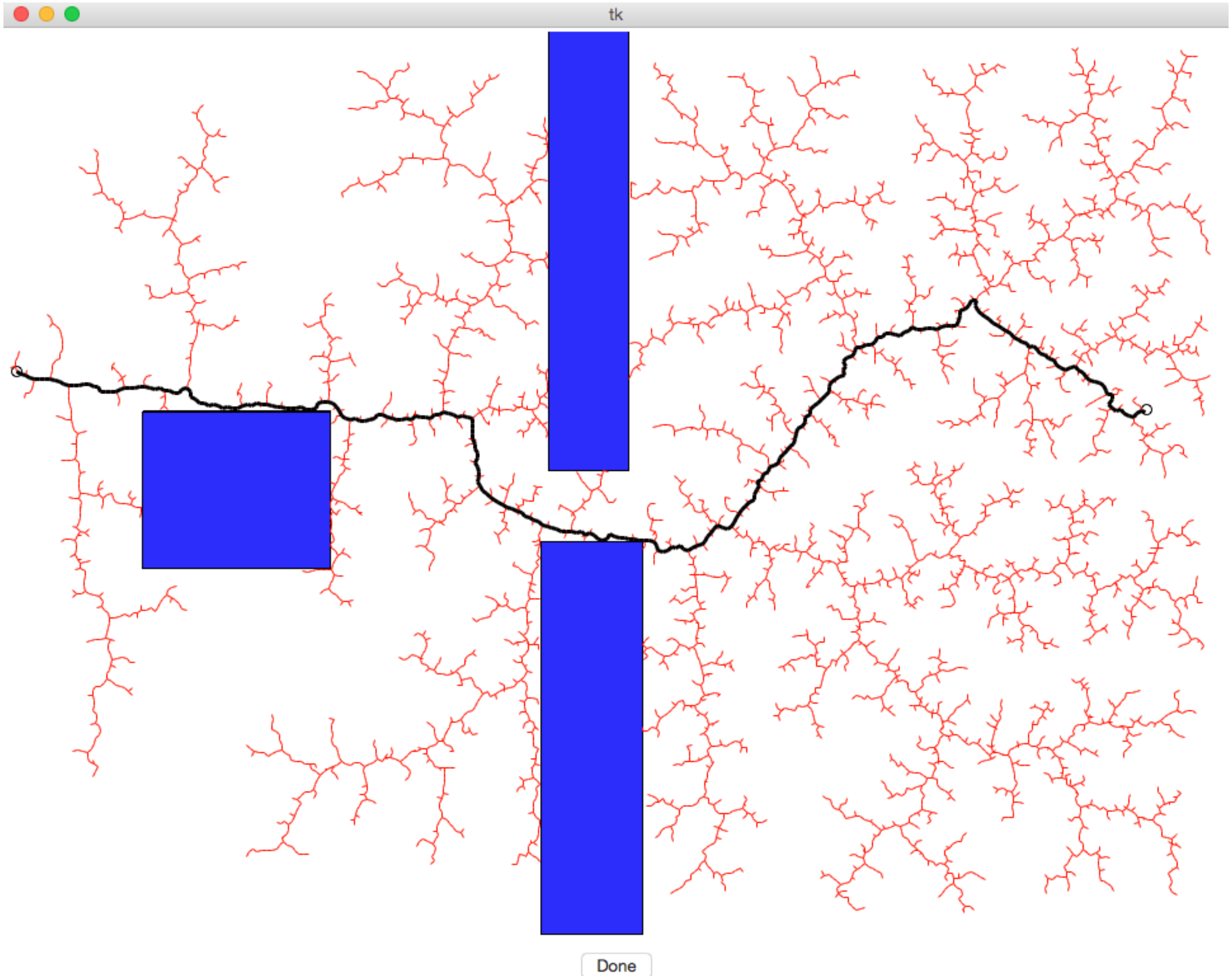
Uniform Sampling - Final Path



Gaussian Sampling – Planning Phase



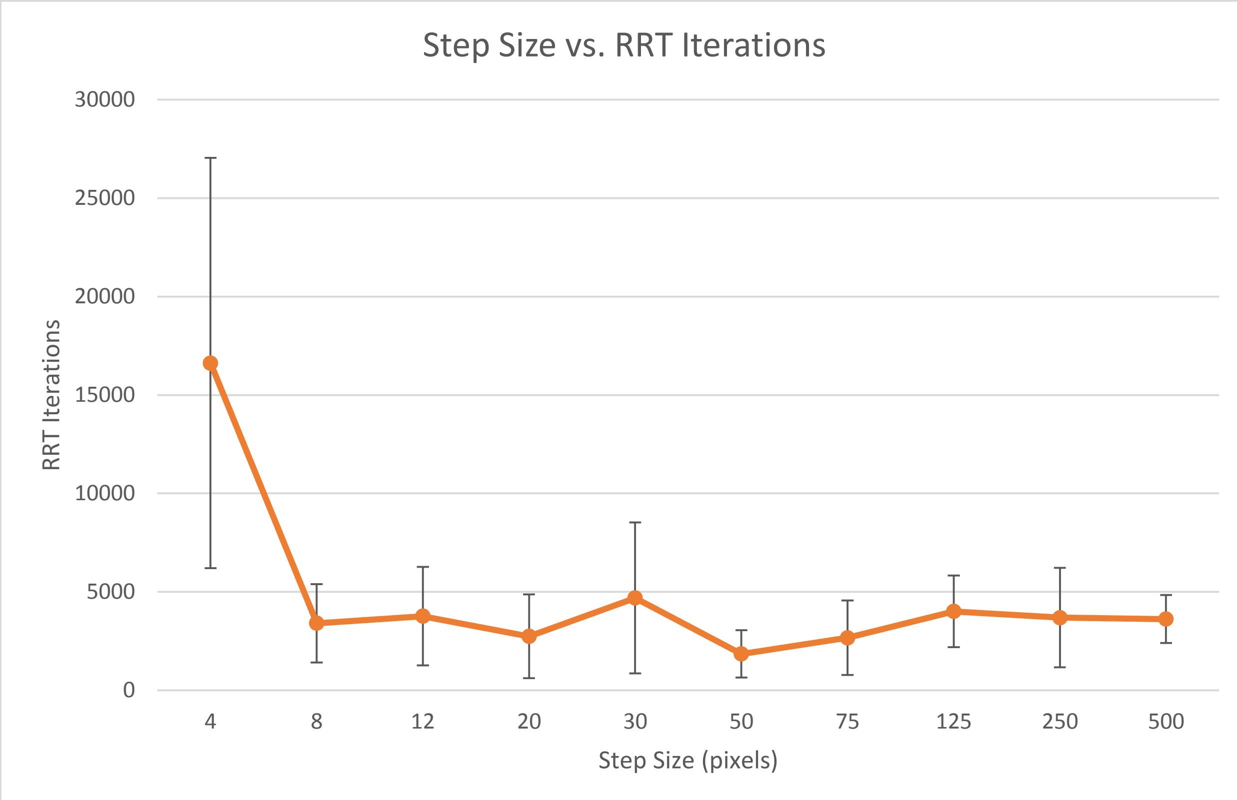
Gaussian Sampling – Final Path



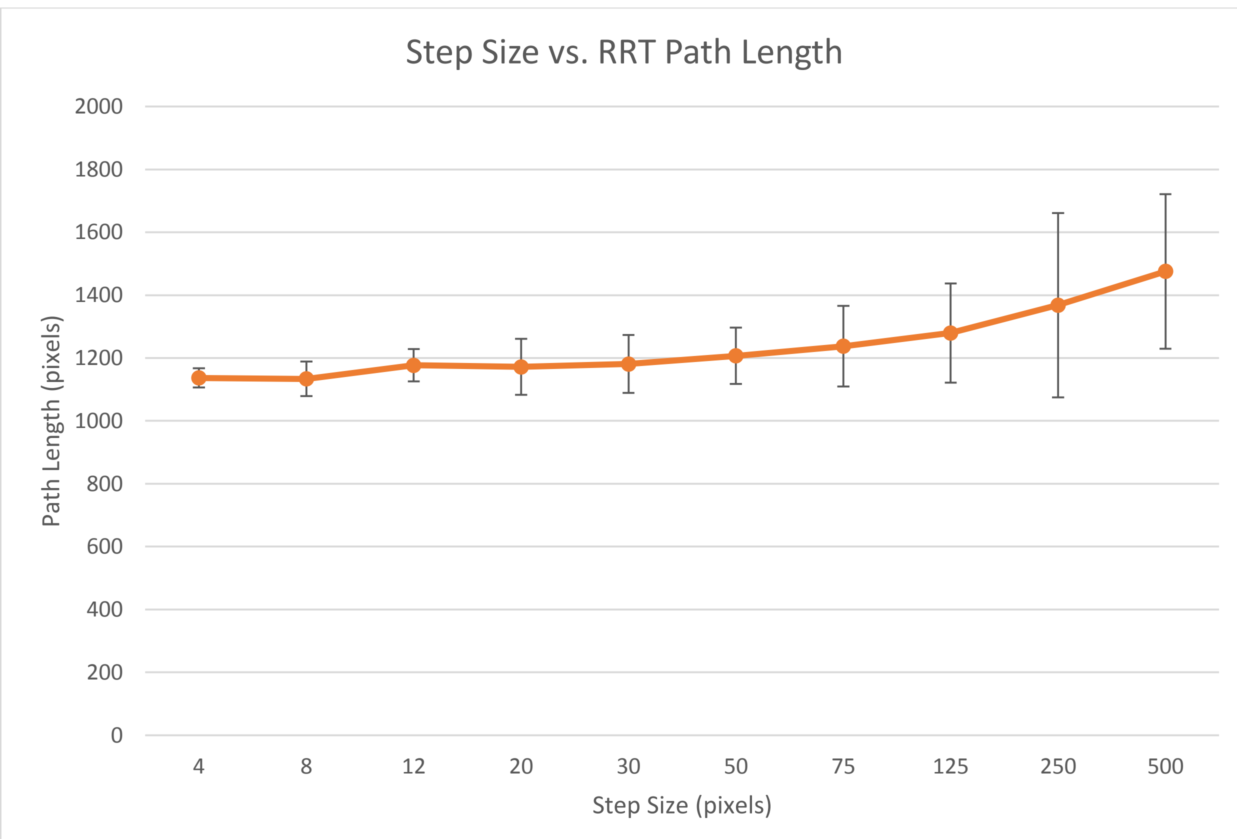

1b Discussion)

As expected when using uniform sampling to expand the RRT the tree covers the area uniformly, i.e. after a certain amount of time the algorithm explores every piece of free space equally. The Gaussian Sampling however did a better job leading the RRT towards the target (as expected since the distribution of our sampling points centered around the target). In the Planning phase image for the Gaussian sampling it is clear that the RRT is pulled towards the gap between the rectangles since there is much less vertical expansion by the RRT before going through this gap. Since our sampling standard deviation was relatively high the RRT appears to behave more similar to the Uniform RRT after passing through the gap (i.e. the right section of the world is explored somewhat uniformly). Clearly Gaussian sampling is more effective and efficient for the World 1 environment

Charts



Step Size vs. RRT Iterations
RRT Iterations
30000
25000
20000
15000
10000
5000
0
4
8
12
20
30
50
75
125
250
500
Step Size (pixels)

Step Size vs. RRT Path Length
Path Length (pixels)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
4
8
12
20
30
50
75
125
250
500
Step Size (pixels)

Note: for Part c, I fixed the radius of the target area to 10 to prevent larger step sizes getting a larger target area.

1c Discussion)

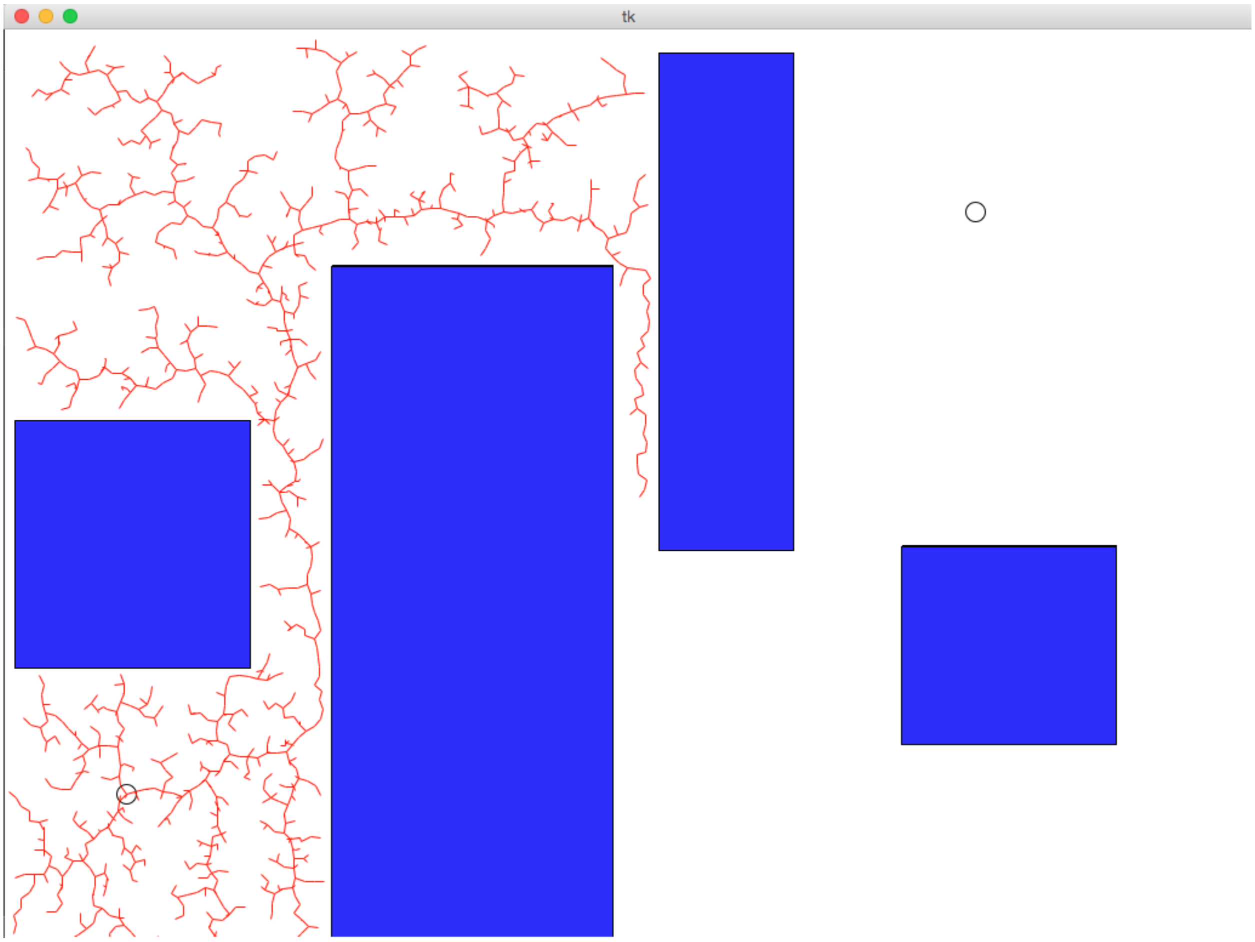
At first glance the iterations chart shows us that a step size of 4 is too small as it requires approximately 4 times as many iterations to find a solution compared to other step sizes, in fact the gap is so large that the error bars for step size of 4 and 8 do not overlap at all. Other than for step size 4, the number of iterations for the other 9 step sizes are somewhat similar with a clear minimum at step size 50. Further the step sizes of 8,12,20 and 30 have a considerably higher error (standard deviation) than the 5 largest step sizes, it’s possible that larger step sizes do create less variation in total RRT iterations however I feel that if you were to make the step size large enough this variation would begin to increase again. Alternatively, the difference in error between the last 5 and the other 4 step sizes may simply be a function of the World1 environment.

The Path length chart is very revealing. Average path length is proportional to step size, this is because paths of larger step sizes are going to be much more jagged and travel inefficiently towards the target, whereas small step sizes cannot effect the path direction as severely at any given iteration and thus the path is less jagged and more efficient at moving towards the target. Compounding on this downside to the larger step sizes is the fact that they also have much larger variation in the path length. The error for step size 4 is just over 30 pixels whereas the error for step size 500 is over 245 pixels. This chart shows that getting too greedy and picking a step size that is too large can have severe negative consequences.

If I had to choose a step size after this analysis, I would opt for a step size of 50. Step Size 50 had the minimum average number of iterations and tied the lowest variation in number of iterations with step size 500. Furthermore, the average path length of step size 50 is only marginally longer (less than 30 pixels) than the average path length for step size 12,20 and 30, with a similar variation in path length. I feel that a step size of 50 is the right mix of computational speed (number of iterations) and efficiency (path length).

Part 2)

Planning Phase



Final Path

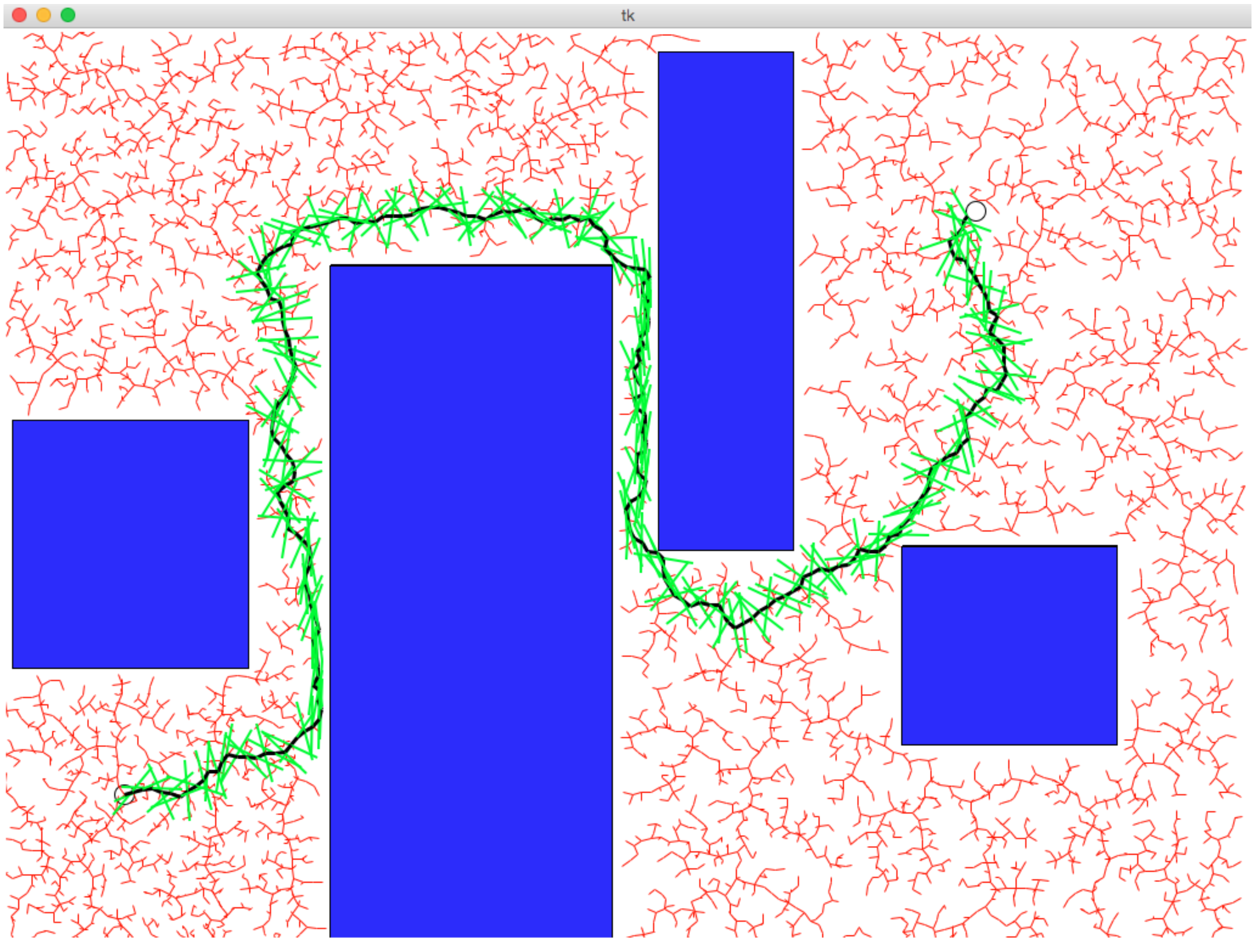
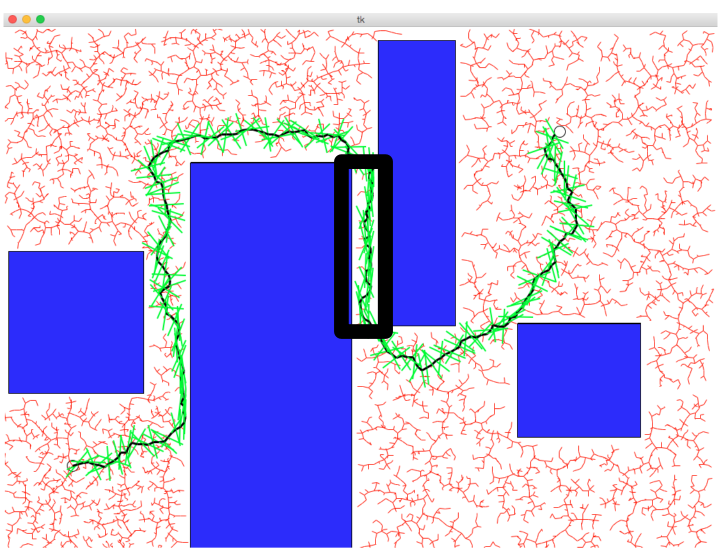
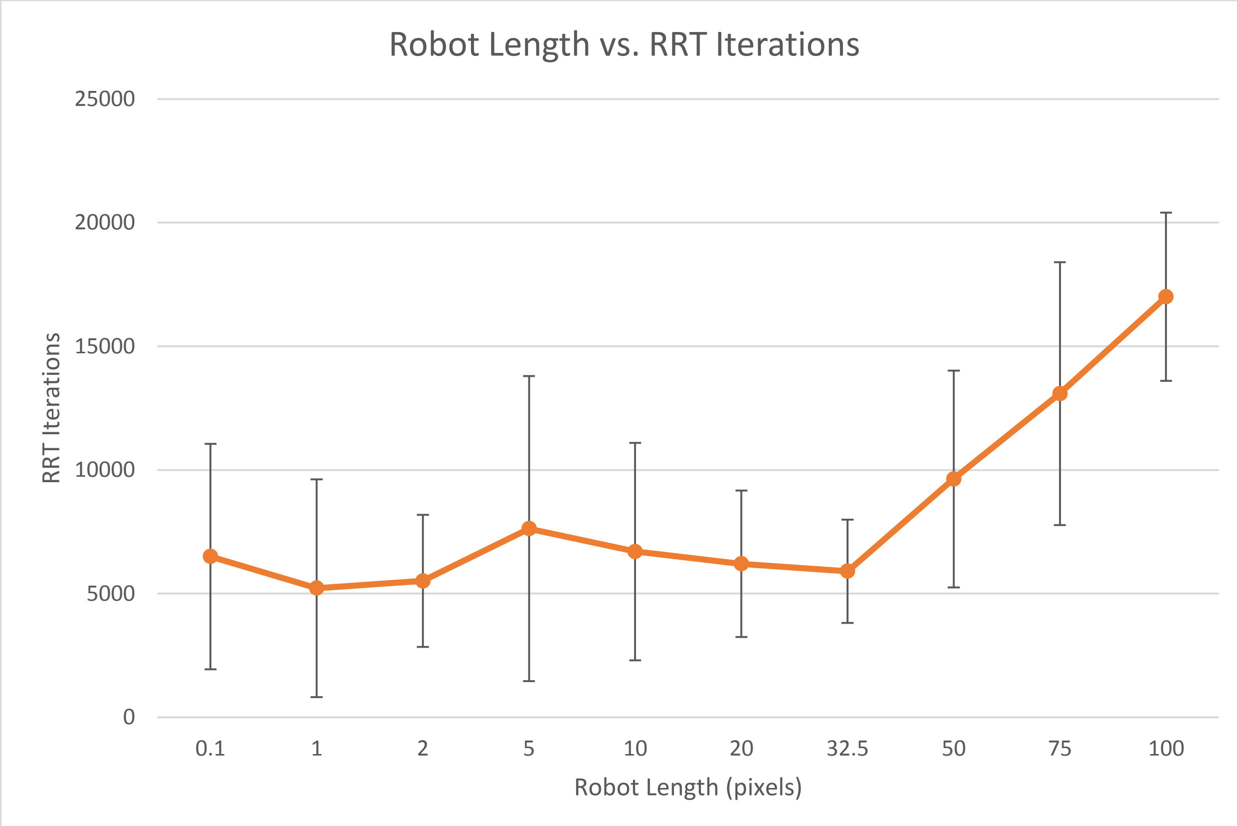


Image denoting vertical gap (referred to in discussion below)



Charts




2C Discussion)

The average number of iterations appear to be relatively constant from robot length 0.1 to robot length 32.5, but then experiences a sudden and constant linear increase for robot lengths 50,75 and 100. I believe the explanation to this interesting property is relativity simple and depends on the width of the gap in the middle of the world (I have highlighted the area I'm referring to in the image on the previous page). I measured this gap on my computer and it is 39 pixels wide. What this means is that for robot length 32.5 if our randomly generated point fell anywhere near the middle of that vertical gap then the angle of the robot would not matter since even if it was horizontal the 32.5 pixel wide horizontal line would fit in the 39-pixel wide gap. The reverse of this is the downside of robot lengths 50,75 and 100, no matter where a random point is placed in this vertical gap the angle of the robot must make the robot relatively vertical in order to fit through the gap (The final path image on the previous page shows this clearly where the length of that robot is 50). Furthermore, the robot length of 32.5 had the lowest variation in number of iterations. From this analysis a robot of length 32.5 would be optimal for World2.