Philippe Sayegh

260637812

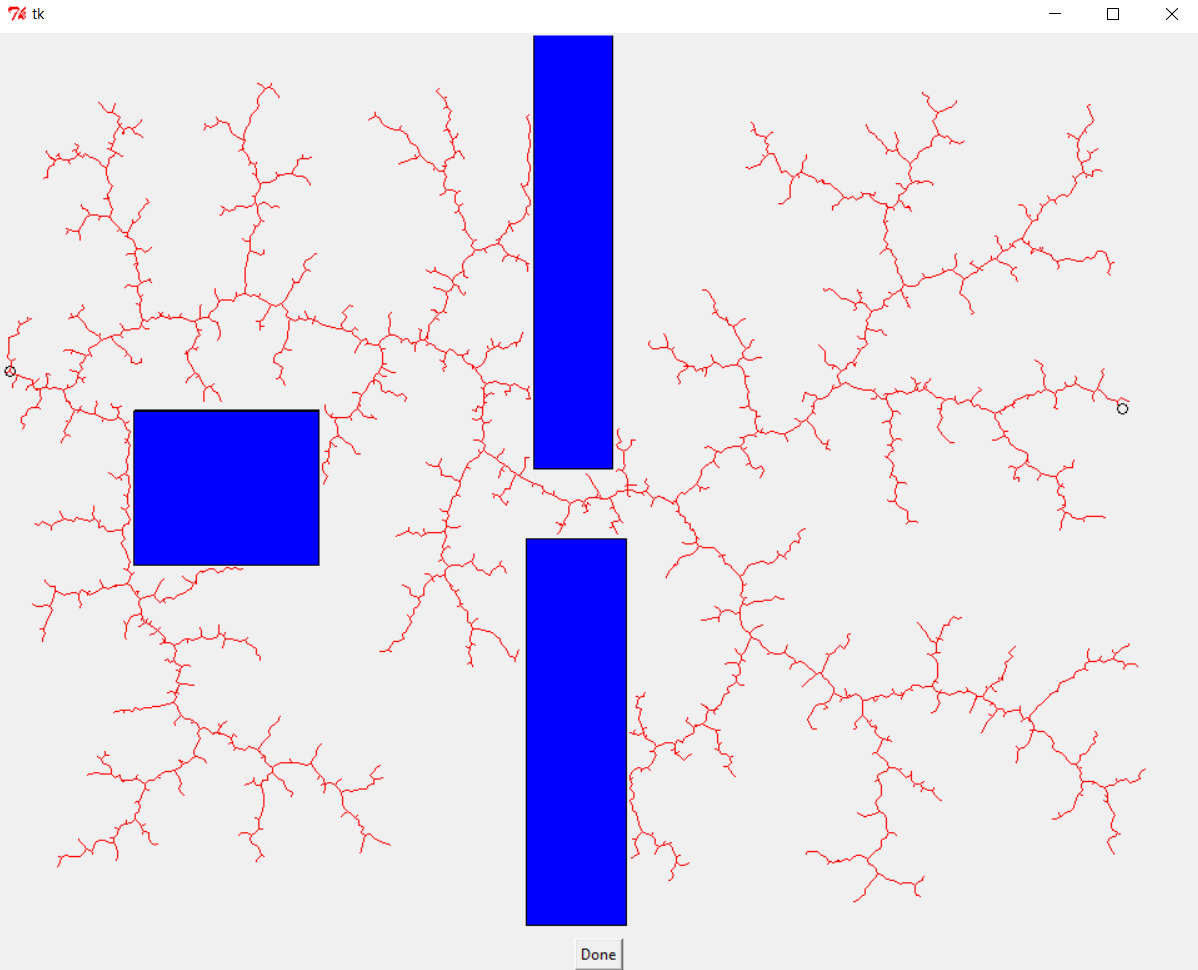
Comp 417 Assignment 2 – RRTs

Part 1

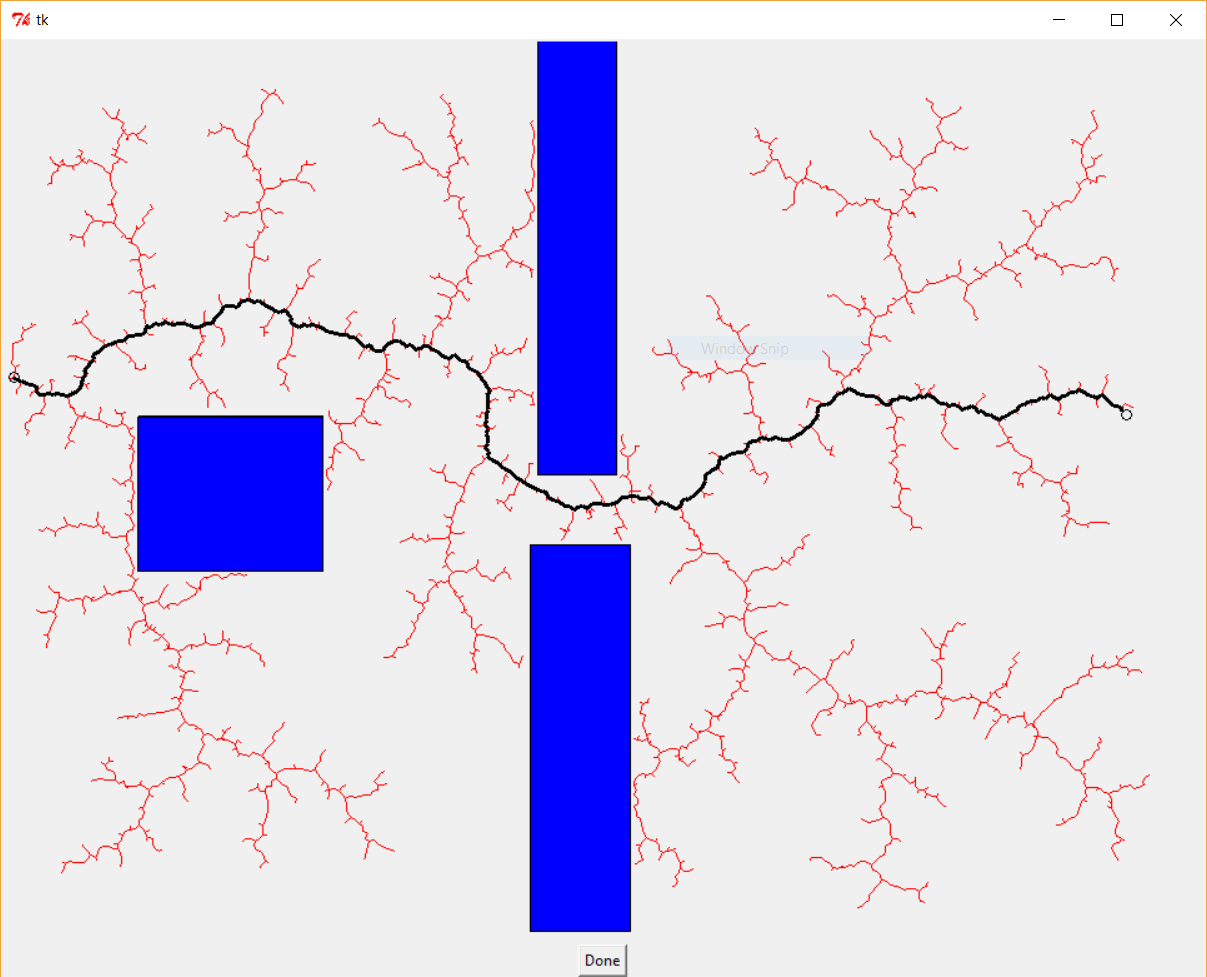
Images only (answers will follow)

1. Planning phase, then path.

(1a planning)

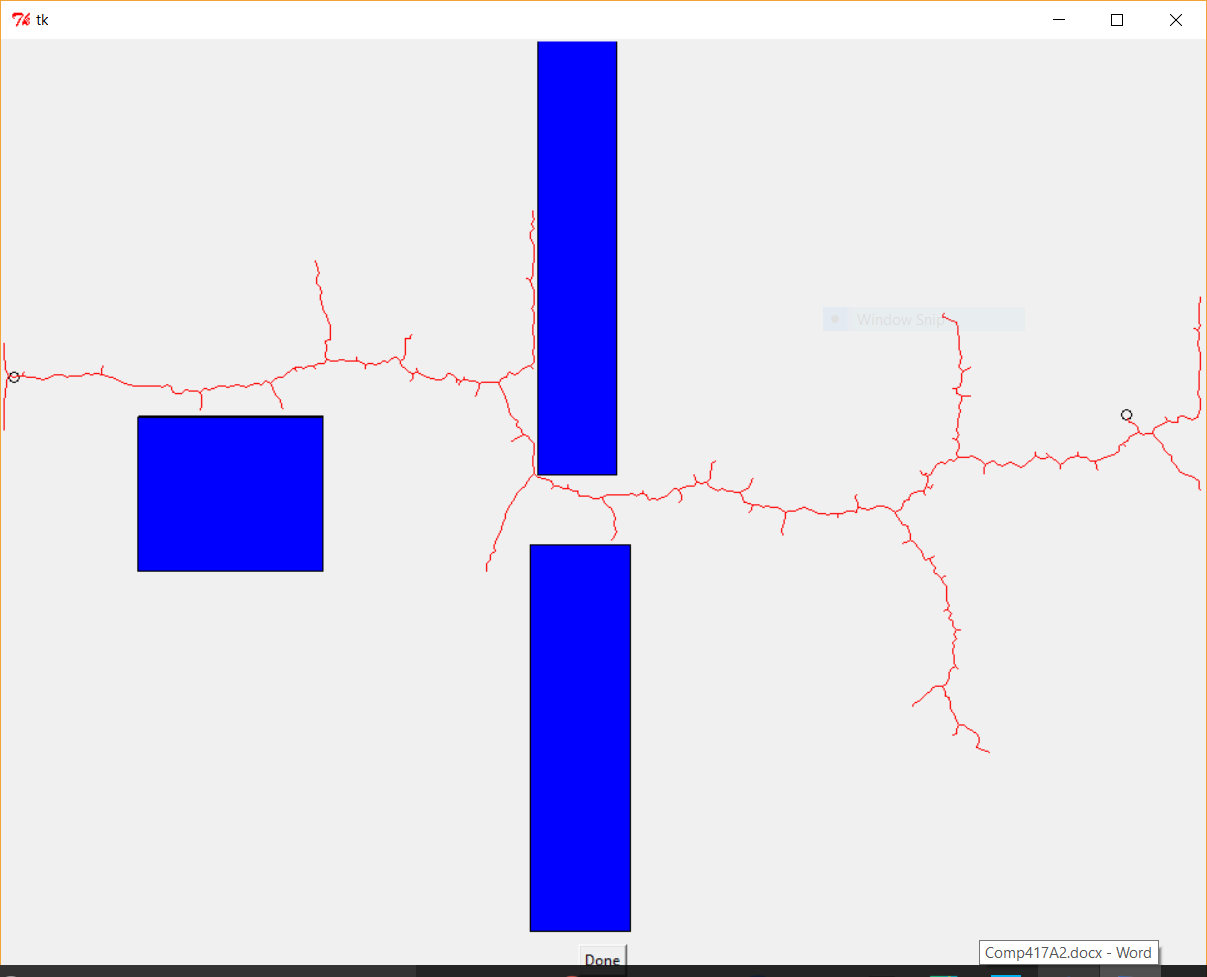


(1a path)

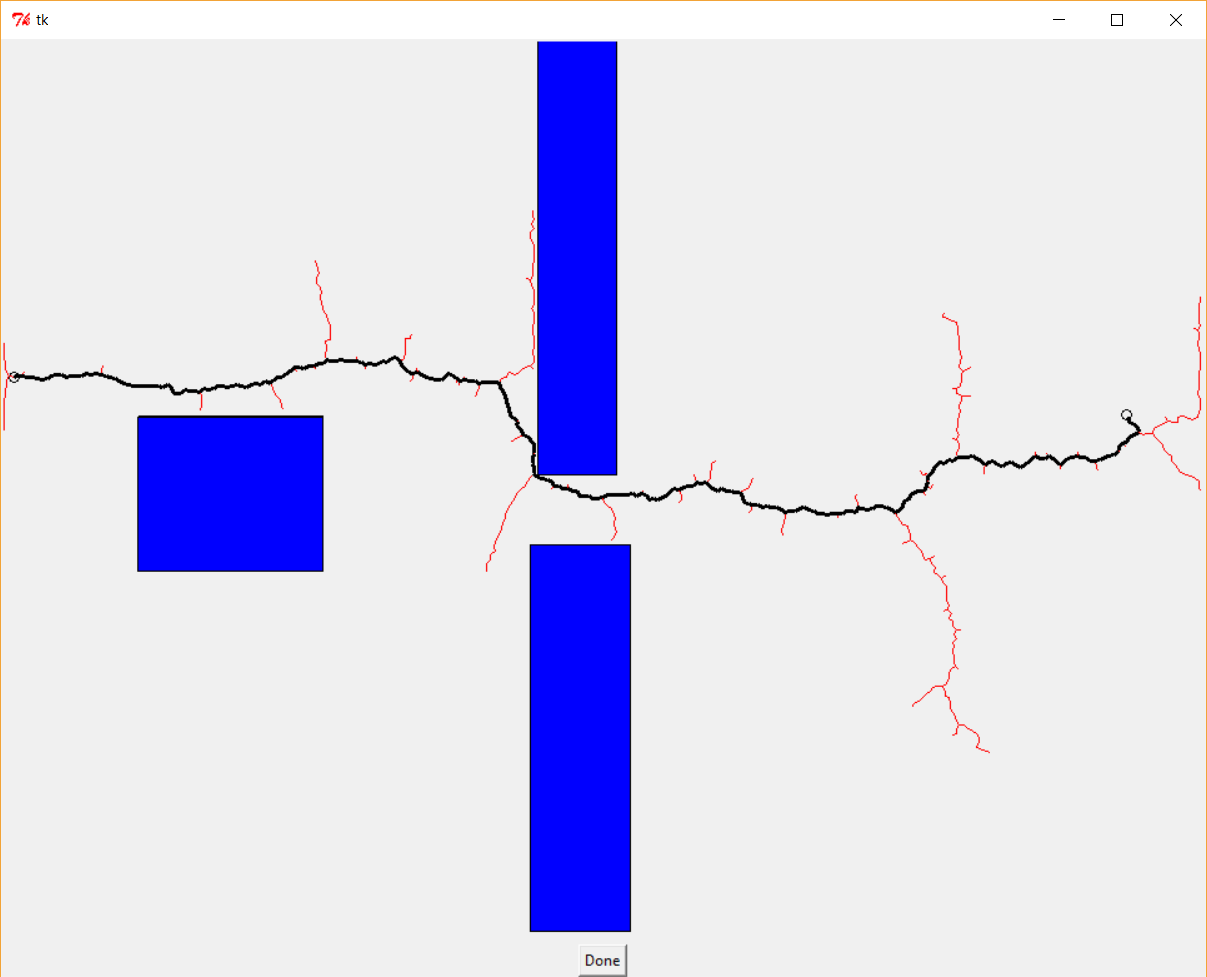


1. Planning phase, then path.

(1b planning)



(1b path)



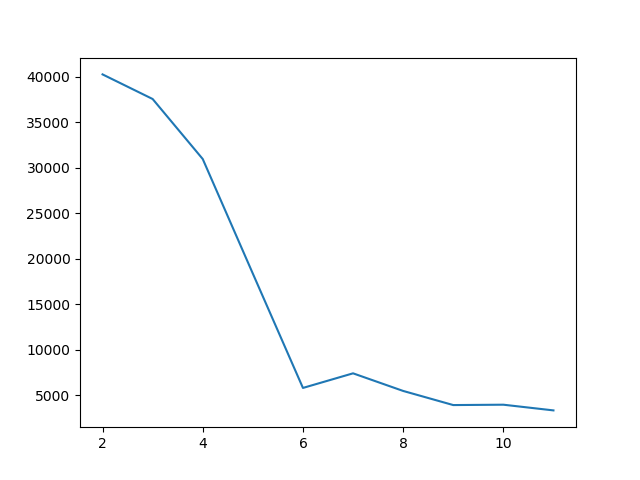
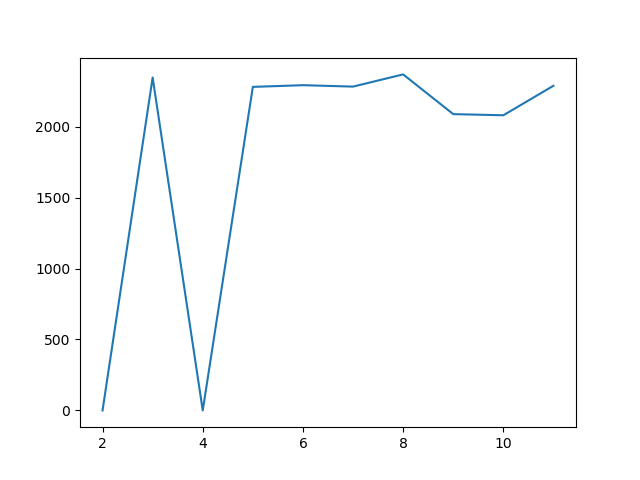
Questions and Discussion

b. The nature of the random distribution does influence the planner.

Using a gaussian distribution instead of a uniform one, the tree is biased to expand more towards the target location, whereas using a uniform distribution, the tree expands evenly throughout the free space (generally favoring large empty spaces, which is a property of RRTs, which also holds for the gaussian-sampled version).

As a second test, I changed the standard deviation values for the gaussian distribution, and noticed that smaller values of sigma x and y (increasing the scalar by which XMAX and YMAX are divided), and noticed that with smaller sigma values, the planner tended to expand much, much more in the region of the target location, resulting in a more efficient planner. It has to be said that as the sigmas get smaller and smaller, the planner has a tendency to spend more time stuck on walls standing between it and the goal location, as more random samples point in the direction of the goal.

c.



Step size (x axis) vs mean number of iterations required till target reached (y axis)

Step size (x axis) vs mean path length (y axis)

The first thing that becomes clearly observable through the right-hand graph is that with very small step sizes, the algorithm yields a shorter path.

The left-hand graph requires a bit more analysis. There are some obvious anomalies: for step sizes 2 and 4, the mean path length is 0. These zeroes represent the fact that in ten runs of 50,000 iterations, the RRT planner did not manage to reach the goal, when using a step size of either 2 or 4. Thus, one should interpret these zeroes as ‘DNF’ or ‘inf’. As such, the experiment for step sizes 2 and 4 was inconclusive. Even then, slightly larger step sizes seem to result in slightly shorter paths, on average.

It is worth mentioning that inconclusive runs were counted as having taken the maximum number of iterations, ie 50,000. This makes it so when plotting the mean number of iterations vs. the step size, the data remains relevant: these runs would have probably ended with the tree reaching the target, but after a large number of iterations.

Here is the numerical data from the python console:

step sizes: [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

mean path: [0, 2346, 0, 2280, 2292, 2282, 2368, 2088, 2080, 2288]

mean iterations [40262, 37558, 30963, 18373, 5814, 7419, 5484, 3932, 3972, 3346]

With the present data, the conclusion would be that a step size of around 10 to 11 leads to a more efficient RRT planner in this case. Having a bigger variety of step sizes would have been useful, however I was limited by computation time, which was already very long for 10 runs per each of the 10 sample step sizes.

As an hypothesis, I would say that as the step sizes increase, the efficiency of the planner would eventually deteriorate once again, as branches obstructed by obstacles would become increasingly frequent. Also, too big a step size would prevent the tree branches from making tighter turns, thus making the planner potentially “miss” the target.