Stephen Brewster

Lab 1 Reflection

Created in python2.7

Simplification Choices

A\* simplification

Unfortunately I did not have time to find a working approach to simplifying the path within the A\* algorithm. I simply ran out of time before I could find a method that worked. The approach I tried first that failed due to being far too inefficient was to generate all the valid successors in a 3 meter area. This “should have” worked by sacrificing a high branching factor for a low search depth. Unfortunately this approach was riddled with buggy problems and I ran out of time trying to get it working. I still think it should work to provide optimal distances between waypoints while still providing an optimal result, but it may require preprocessing of the map to speed up the successor generation routine.

I also attempted to simplify the path using the heuristic to favor paths with fewer turns. I added a penalty to the cost of nodes that cause the robot to turn. This seemed to help a little (in some special cases where it was moving through doorways) but I think it ultimately failed because I used a ‘constant’ penalty and I couldn’t find the right function to use.

My next approach was going to be to constantly derive the path on each node expansion and use the post processing algorithm in some way to maintain the path as it is built rather than derive the path afterwards. I didn’t like this approach because it didn’t seem very efficient or work with the algorithms existing functions (get\_successors and heuristic). I thought it made more sense to tweak the successors and heuristic to achieve the result, but I failed to get either approach to work properly.

Note: The code for the failed successor generation routine is commented out and still visible, though I reverted the working code for submission

Post-processing simplification

I was able to get the post-processing simplification to correctly break a path down to better sized chunks of 1-3 meters. I created a turn detection method to find turns in the path and mark them as waypoints. A counter ensured that straight paths were broken down to the correct size.

I also spent some time toying with the idea of using an optimization algorithm to post process the path and minimize the turns over the entire path but I did not have time to explore this.

Performance

No simplification

Starting X: 352

Starting Y: 252

Ending X: 3851

Ending Y: 1156

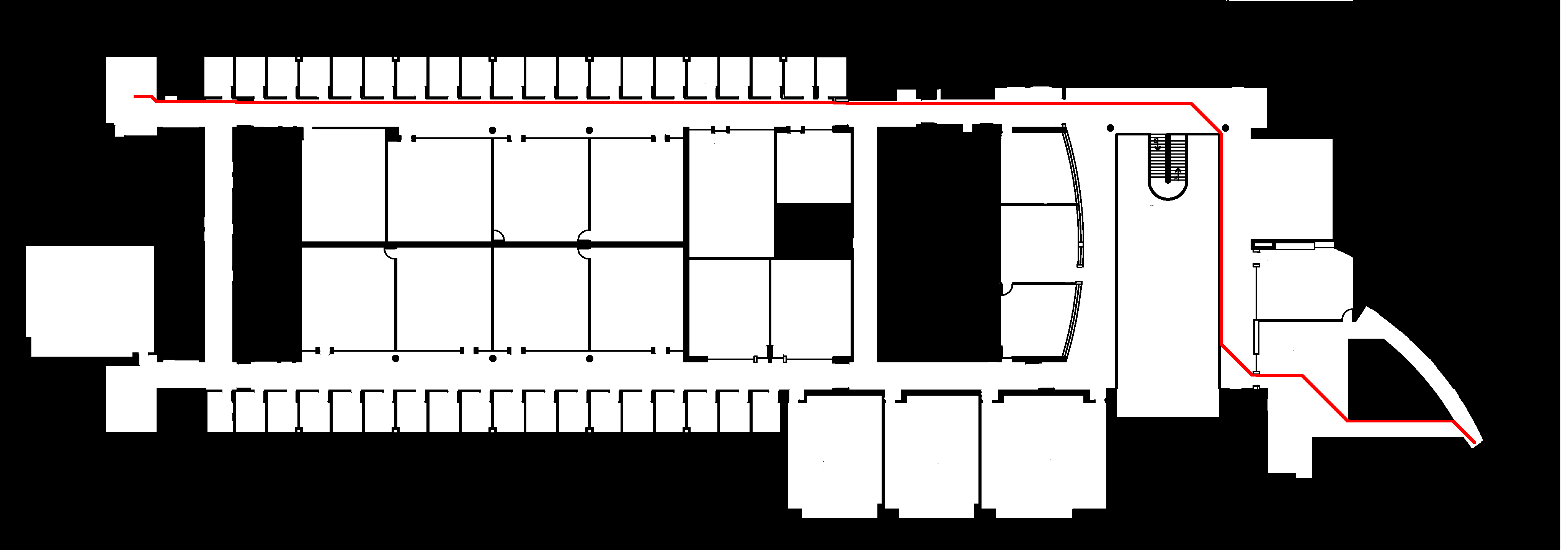
Time: 35.9960000515

Waypoints produced: **4050**

Distance Traveled (m): 146

Nodes Visited: 1065337

Print path (y/n)



Post-processing simplification

Starting X: 352

Starting Y: 252

Ending X: 3851

Ending Y: 1156

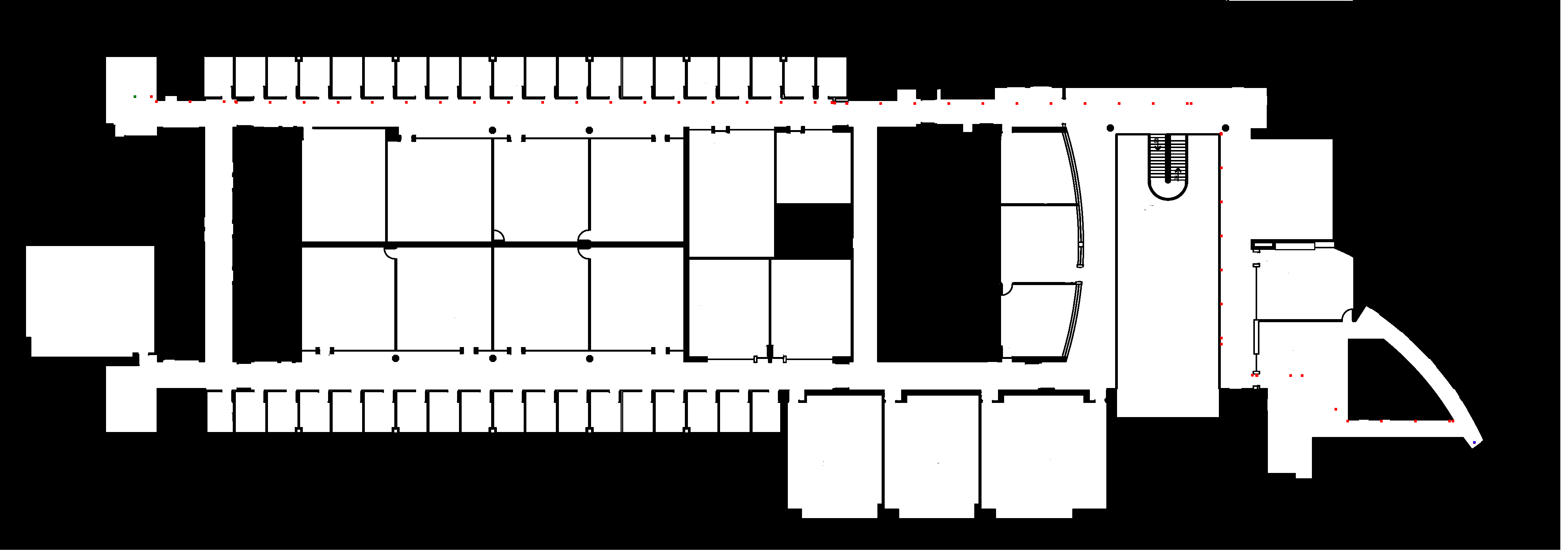
Time: 36.7219998837

Waypoints produced: **63**

Distance Traveled (m): 146

Nodes Visited: 1065337

Print path (y/n)



Search simplification

Starting X: 352

Starting Y: 252

Ending X: 3851

Ending Y: 1156

Time: 115.611000061

Waypoints produced: 4050

Distance Traveled (m): 147

Nodes Visited: 1855376

Print path (y/n)

This simplification obviously was not working correctly, it produces the same map as the simple search but takes longer because it’s also calculating the turn penalty for the heuristic. I left this in because it was the only thing that actually seemed to produce some results (in special cases).

Analysis

The post-processing simplification did not add much time to the search (about 800 milliseconds) since it was a fairly trivial function working in O(n) time. I attempted to make this function into a sort of Hill climbing optimization but lacked the time needed. I would have liked to optimize for minimum turns instead of a simple breakdown of the path.

Search simplification proved difficult as I mentioned, it seemed that any heavy processing in the search needed to accomplish this resulted in running times that were not reasonable. If I had more time I would have tested my approaches on smaller maps to limit the search space.

Final Note: I have had bronchitis for the past week and I should have asked for an extension on this due to a severe lack of time to work on this, but I thought it would be easier than it turned out. I feel I am very close to completing it, but needed to ask you a few more questions and do some more experimentation.