ITCS 3146 Spring 2016 – Assignment 1

Instructor: Bojan Cukic

The Pathfinder – Documentation

Darshesh Patel

The algorithm I used was the one specified in the assignment document. The algorithm is called the Right-Wall algorithm. The point of the algorithm is to mainly check and make sure that there is always a ‘wall’ to your right; ‘1’ being walls and ‘0’s being the open spaces that you can traverse.

Being that I used the Right-Wall algorithm, there are 3 cases per facing direction. The first thing I check is that is there is an open location to your right, if so then move to the right. If there is a wall to your right, then check if there is an open location in front of you, if so then move forward. Finally, if there is a wall to the right of you and in-front of you then turn left and stay put. I found out that for the last case where I turn left It doesn’t matter if I move forward after turning left. It only needs to turn left and loop again.

The main data structure used is an array. I used it because it was given in the initial code from Moodle. Basically the array is initialized to be 100x100. After scanning in the m by n from the user, the array is then initialized to only use those metrics.

Now down to the meat of my code and how it works. After scanning the maze from the file to the array, I first work on obtaining the entrance and exit locations, then I call a function to determine the direction that the entrance is facing. The logic of the function is to just take in the entrance row and column variables as arguments and return a char variable that represents the direction the entrance is facing. After that I set my entrance location to a new set of variables called current row and column. After that I use a while loop to keep navigating through the array until I find my exit location. I have 2 break statements that will leave the loop; one being that if my current location reaches my entrance point, I break out of the loop and determine that no path can be found. The other break is if my current coordinates equal my exit coordinates, then I also break out because I have then successfully navigated through the array. After breaking out of my loop and displaying if there is a path or not, I just print out the array after it has been traversed, the locations visited are now marked by ‘X’, as specified in the assignment.