The A\* algorithm was written using the C++ programming language. It was based on various tutorials found online and put together from those tutorials. The basic concept behind it is that it will use a Heuristic value to determine the shortest path from the starting position to the ending position. Since the project asked that the robot be able to move in 8 cardinal directions (N, NE, E, etc..), the distance formula was used to help calculate this heuristic.

The way this was calculated was as follows: F = G + H where F is the total heuristic value attributed to a node on the graph, G is the value given to a square for its distance from the start position. H is the value for the square’s distance from the end position. Both distances were calculated using the distance formula.

The map itself was created using a double array to represent and x,y grid, with each node being represented by an x,y coordinate. I.E: The starting node was 0, 0, ending node 9, 9, and every node in the grid was a similarly represented value. To depict the starting position, it is depicted as a 5 on the grid. The ending coordinate was originally depicted as a 9, but when the path was created it was depicted as a 3 as was the rest of the path. Obstacles are depicted as a number 7.

Initially only the chairs in the V-Rep scene were depicted as obstacles (7). But then it was found that the algorithm would have the robot go through the chairs diagonally, which I believe was counter-productive to the intent of the project. So the diagonal squares on the opposite sides of the chairs were also depicted as unpassable obstacles so the algorithm would not take them into consideration.

In the end the map grid was printed out with the path the algorithm believed was the shortest path. This path is shown as number 3’s from the starting point and ends at the end point.