

Angel Hernandez



Program 1 Report

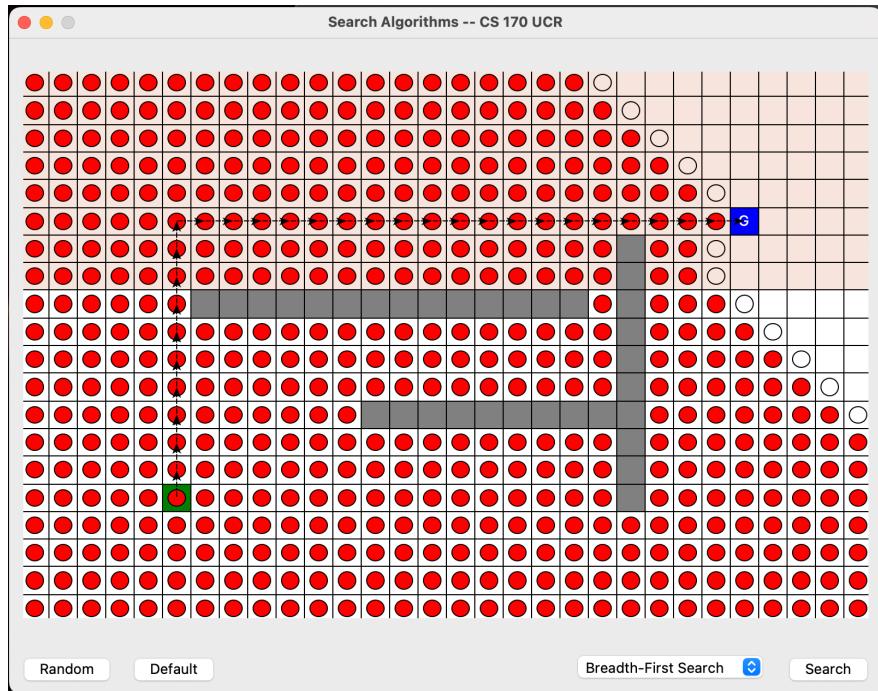
In this programming assignment we were asked to implement the following algorithms: Breadth-First Search, Depth-First Search, Uniform Cost Search, and A* Search to find the optimal path from start to goal, given a two-dimensional grid.

Results

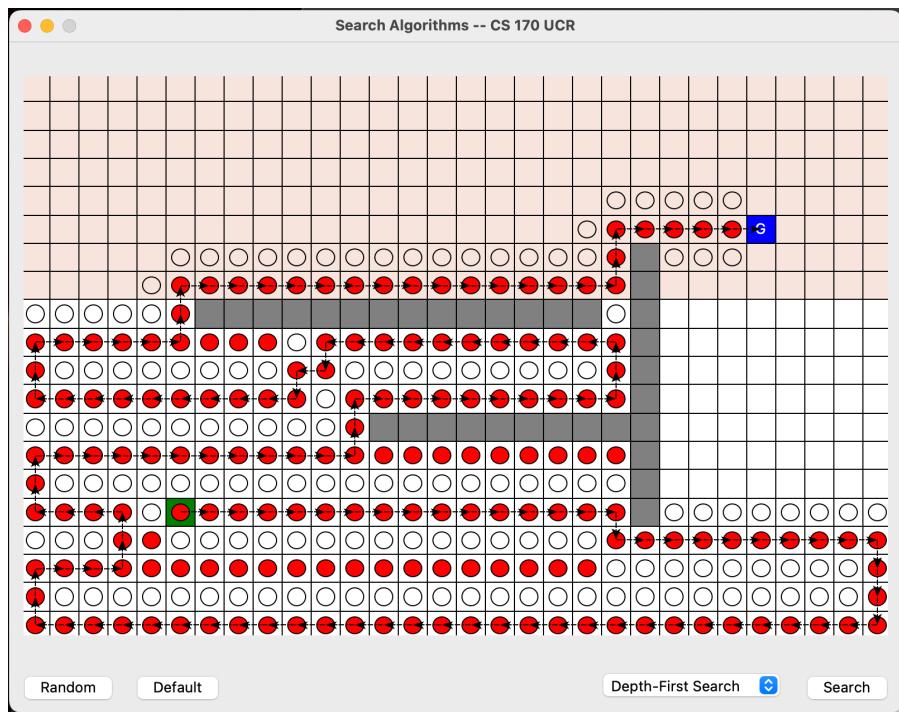
Path-Costs (from terminal output):

Map Type	BFS	DFS	UCS	A*
Default	76	192	38	38
Random 1	21	151	19	19
Random 2	55	113	43	43

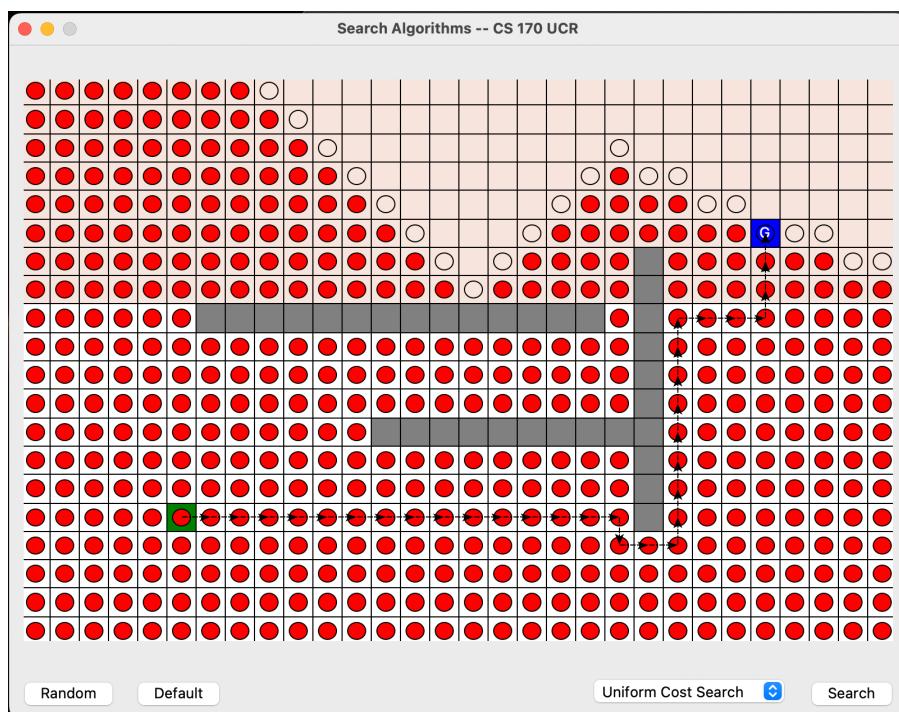
Default Maps:



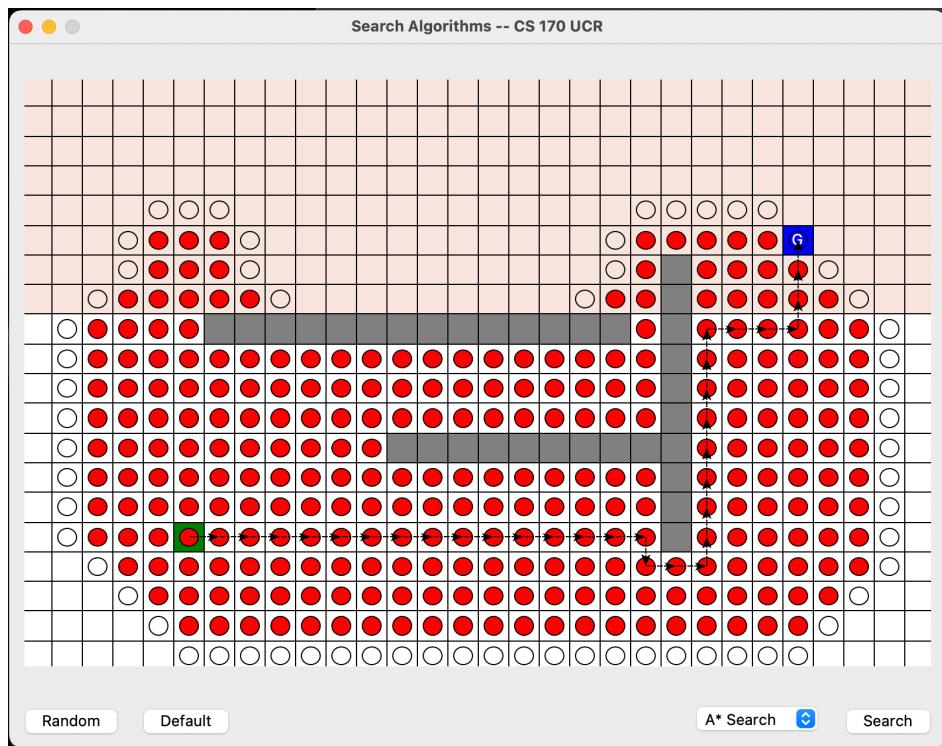
Breadth-First Search



Depth-First Search

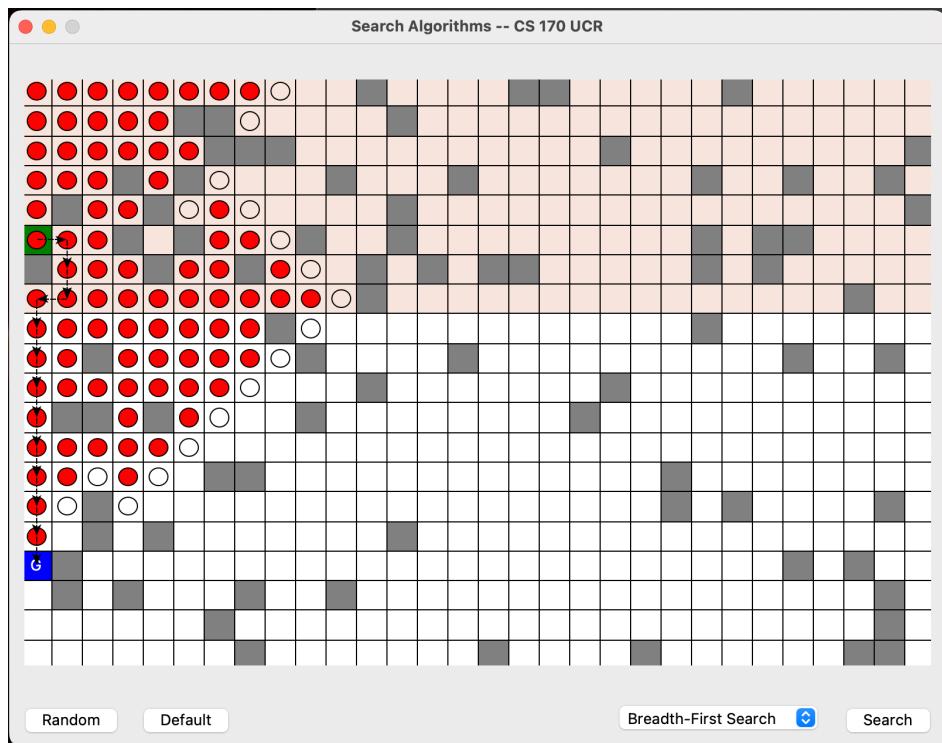


Uniform Cost Search

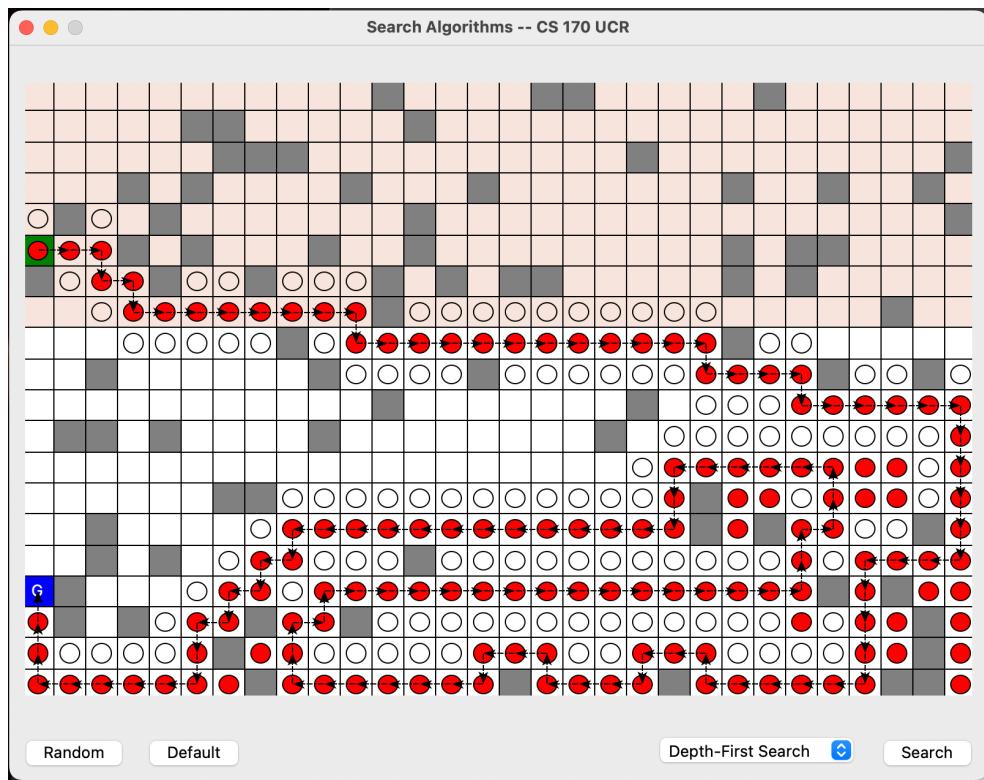


A Search*

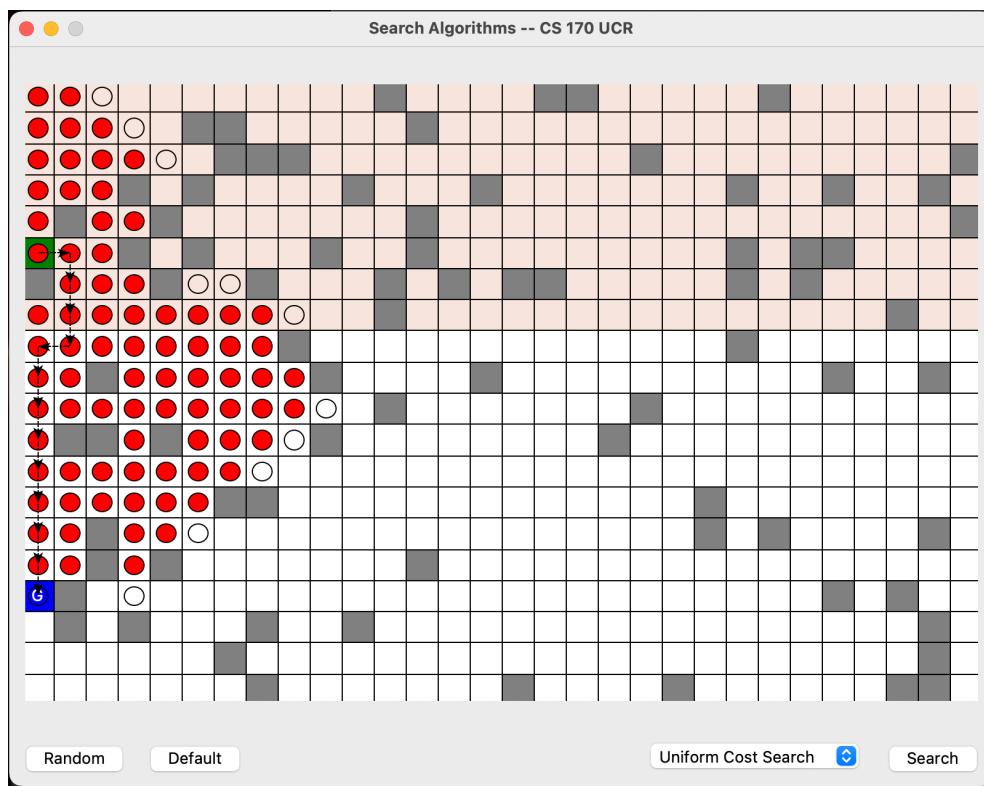
Random Map 1:



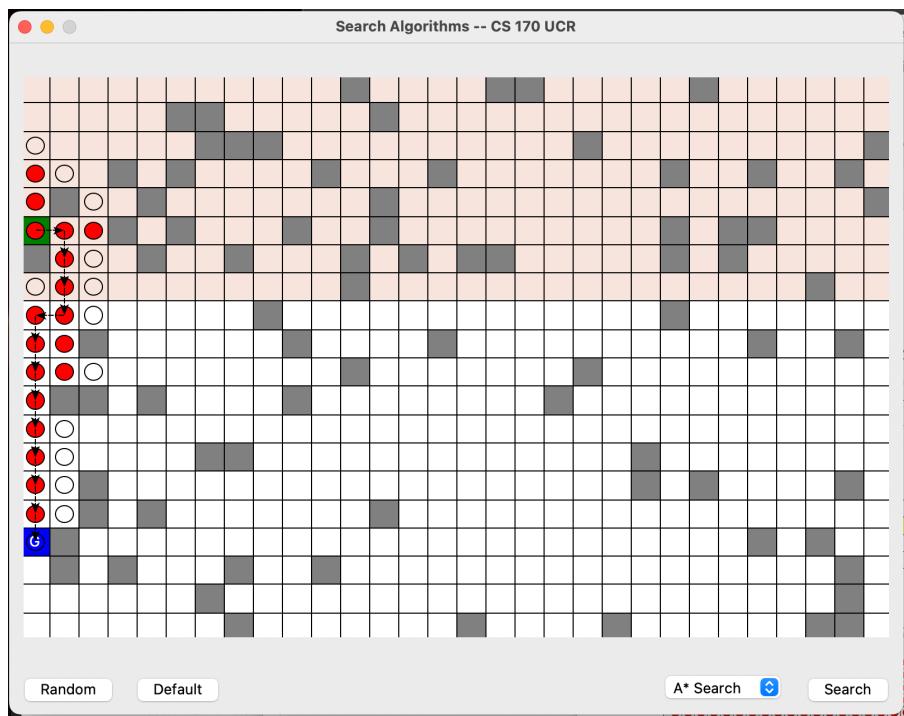
Breadth-First Search



Depth-First Search

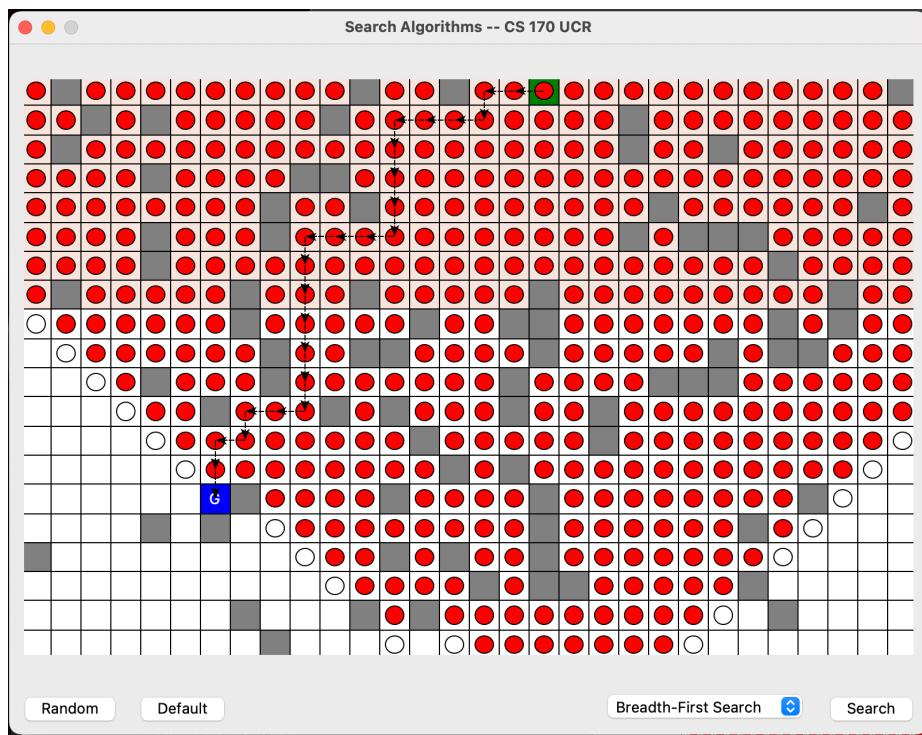


Uniform Cost Search

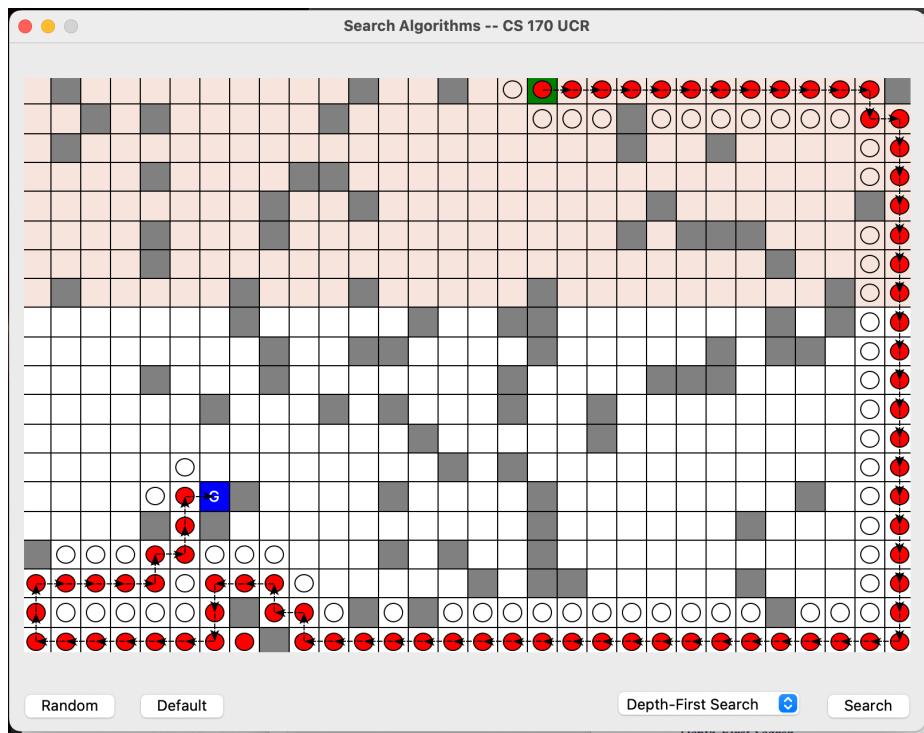


A Search*

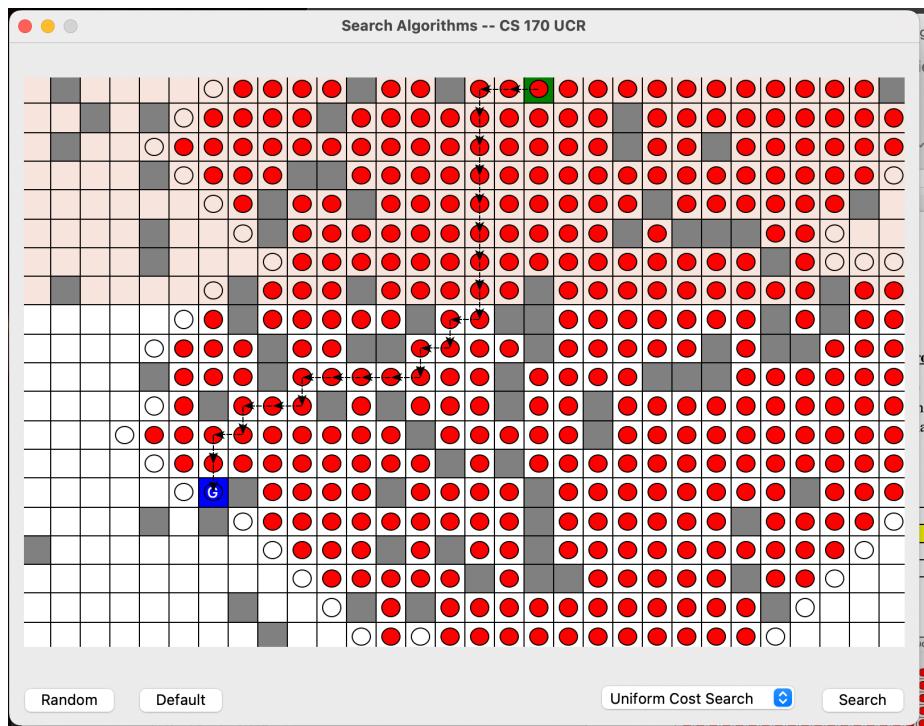
Random Map 2:



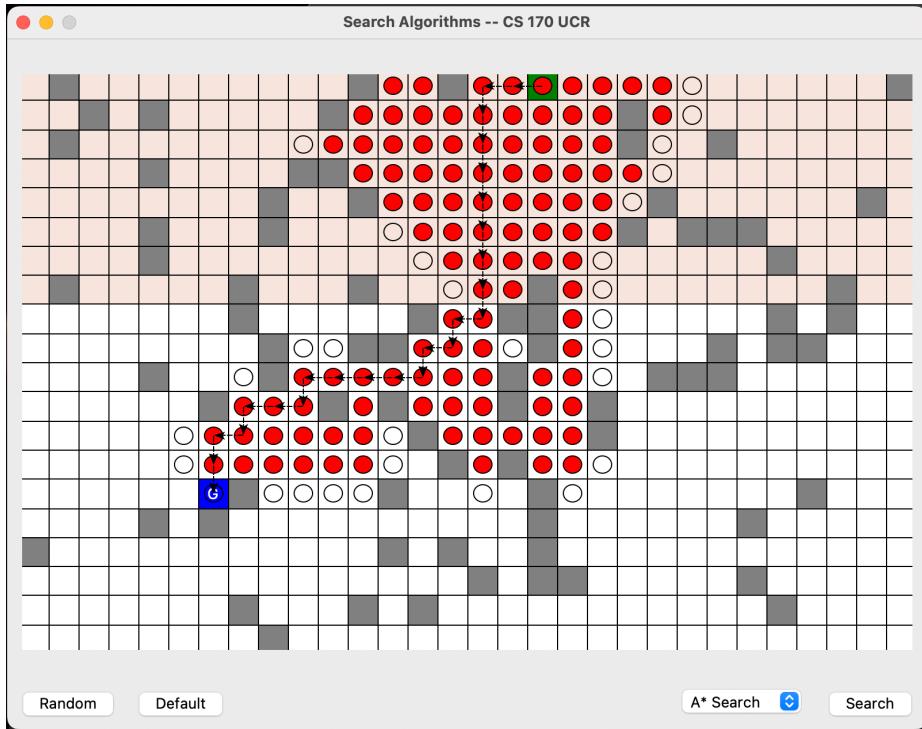
Breadth-First Search



Depth-First Search



Uniform Cost Search



A Search*

Observations:

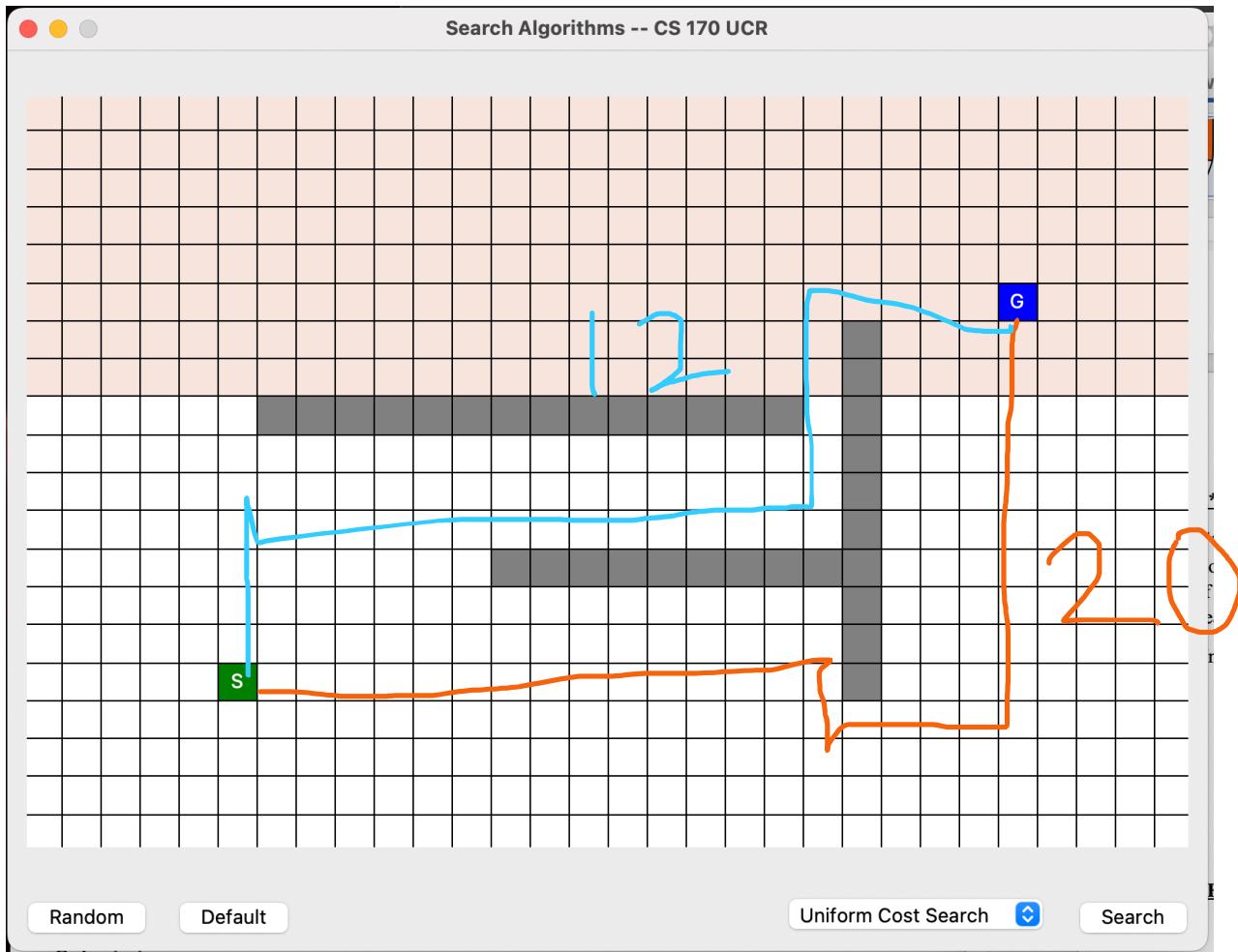
Upon running the algorithms on these three maps I noticed several things. One of the immediate things I observed was the fact that DFS was the costliest out of the four algorithms. When I ran the DFS algorithm I noticed that the nodes expanded (red dots) would extend all the way towards the end of the direction it was facing. This is to be expected because DFS requires for the expanded node to extend all the way to its depth until the goal is reached, hence its path follows direction as well. For BFS I noticed the expanding node would open up in a circular-like direction, opening layer by layer from the starting node, this is expected because the expanding node is supposed to open until the goal is reached, its expected path follows the direction in which the node expands in terms of layers.

Something interesting that stood out between UCS and A* was the fact that they had the exact same path-cost, but the way in which they reached it was vastly different. For UCS when I ran the algorithm the nodes would expand all together, thus allow the path to be as it is and being quickly. On the other hand, when running A* given the fact that we apply a heuristic, notably that of the goals location, we can allow the expanding node to move towards the goal. While it requires the same amount of path-cost, the finding of the path was a lot quicker than UCS.

A* on Uniform Cost Grid

If there was a uniform cost grid, say every cost is 1, then A* will find the shortest path there is. However, if there was a nonuniform cost grid, the algorithm will find the least costly path regardless of length. This is because if we have a uniform cost grid, we can ignore the path-cost and simply head straight to the goal.

Uniform  Nonuniform 



BFS vs. UCS on Uniform Cost Grid

BFS functions by checking each layer of nodes to find the minimum number of steps until the goal is reached while UCS works by looking for the least expensive path-cost. If we apply a uniform cost grid and UCS must find the least expensive path but if all path-cost=1 then the

algorithm will instead find the quickest path or the path with the least minimum number of steps. They should generate the same path.