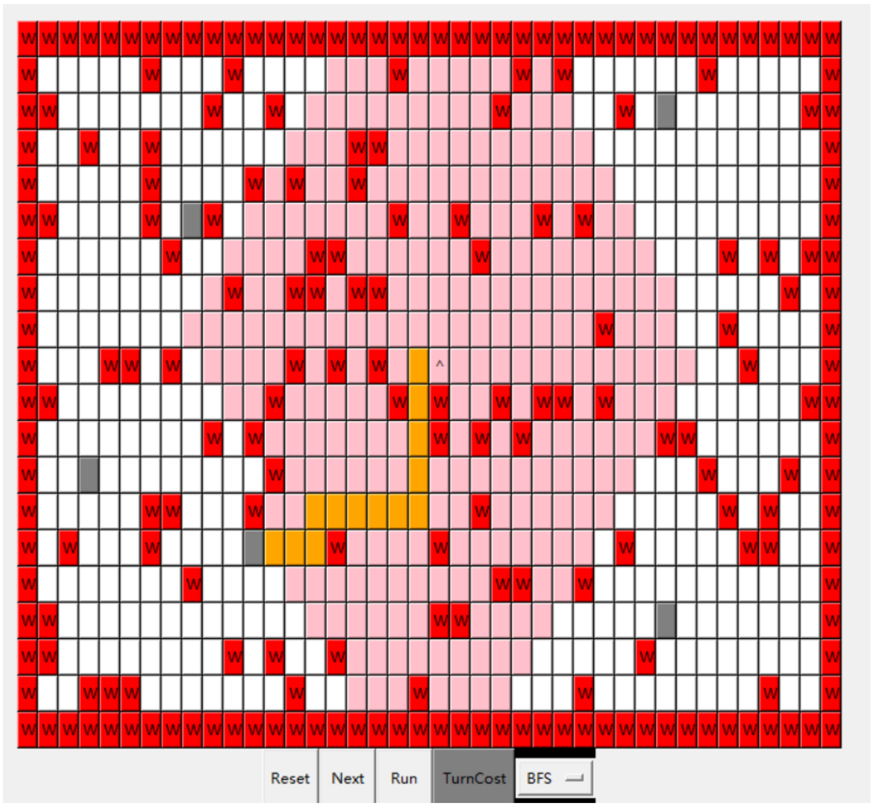
CMPT 310

Assignment 1 Report

Ruixuan Liu

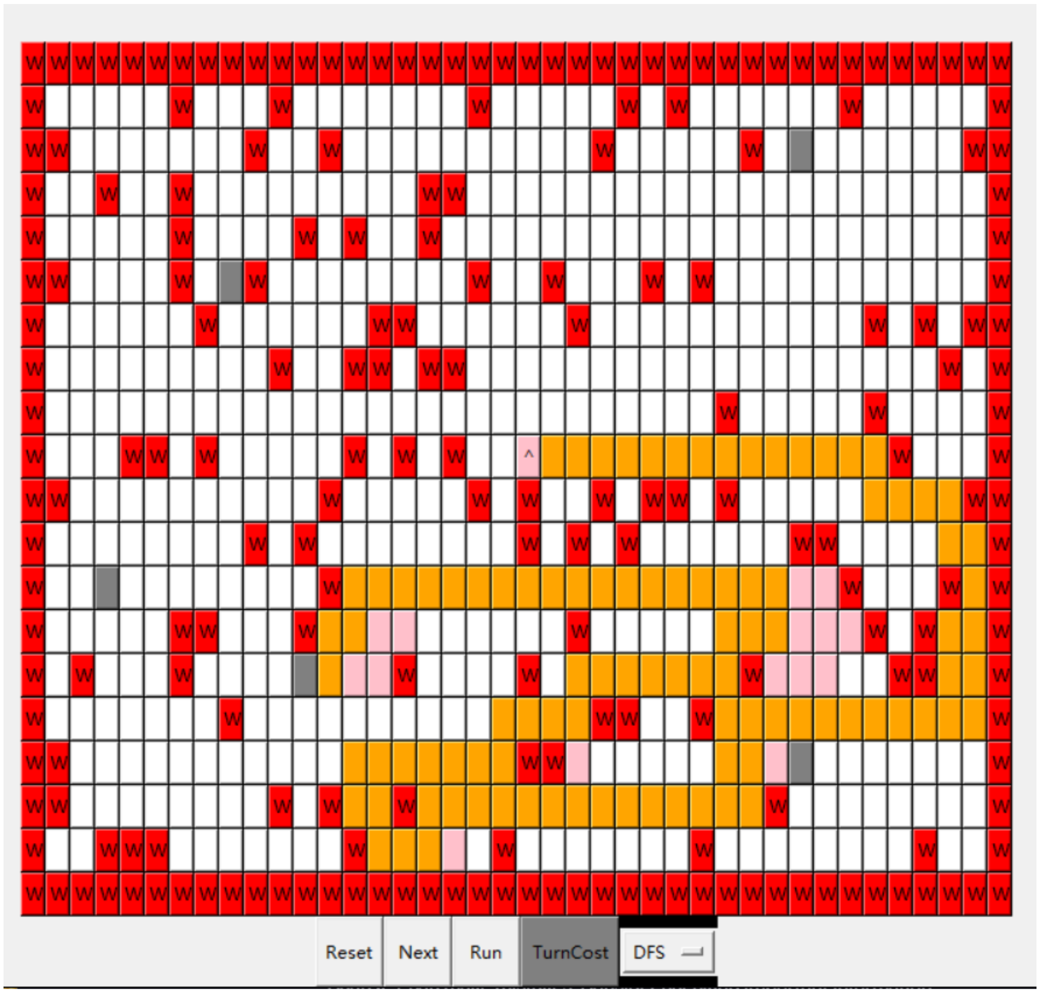
301383988

1. Breadth first graph search:
   We used a FIFO queue and traversed this queue infinitely until there were no elements in this queue. We check if the leftmost node in this queue is the target node, if not, we add that node to the explored list then continue to check the children of that node and add their children to the boundary.

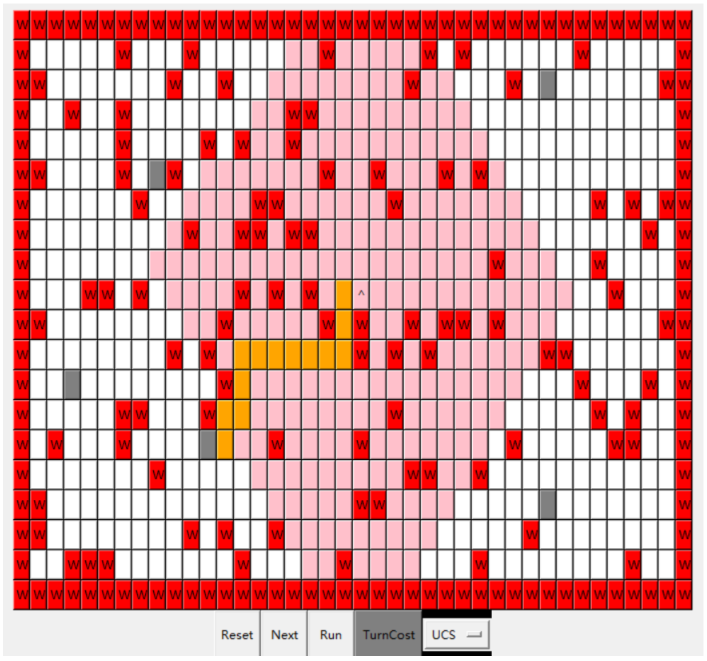
   As can be seen from the figure, the breadth search explores outward following the agent's node and eventually finds the closest target node to the agent's node.

2. Depth first graph search:
   The biggest difference with breadth search is that instead of using a FIFO queue, a stack is used to achieve the same functionality.



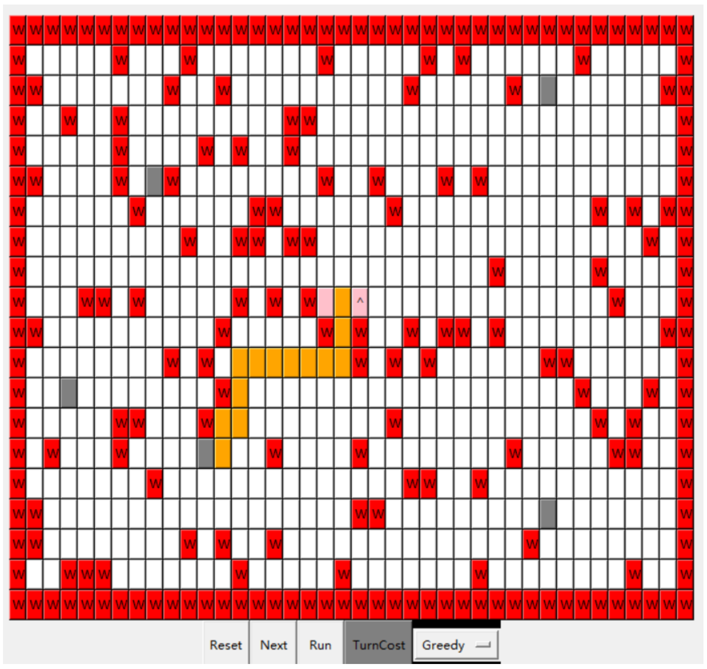
   For the same graph, depth search and breadth search yield the same results, but the number of steps taken is the largest.

3. Uniform cost search:
   The best-first graph search algorithm is employed to prioritize the queue and then explore the graph by expanding the most promising nodes selected according to specified rules. Another difference is that during the boundary update, if a child node is already in the boundary but has a higher f-value, it is replaced with a new instance that has a lower f-value.



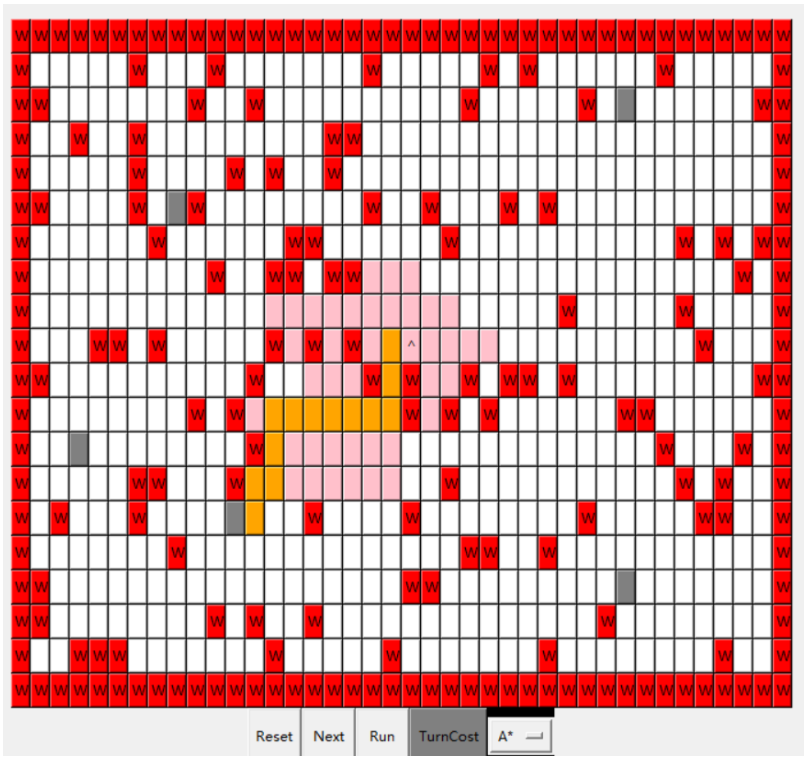
   It is still the same maze, and the optimal solution chooses a breadth first search.

4. Greedy search:
   Greedy best first graph search directly references best first graph search, which prioritizes the nodes based on a heuristic function that estimates the h(n) cost from the node to the target



As can be seen in the figure, the nodes explored by the algorithm are the least and the efficiency should be the highest.

5. A* search:
   The A* search algorithm finds the minimum cost path from a given initial node to the goal node. By using the actual cost from the starting node to the current node (g(n)) and the heuristically estimated cost from the current node to the goal node (h(n)), it combines the features of uniform cost search and greedy best-first search.



   The graph shows that the algorithm has more nodes explored than the greedy algorithm because of finding the path with minimum cost.

6. Turn cost effect:
   When the turning factor is 0.5, the impact on the UCS algorithm and the A* search algorithm is not that significant. But when the turn factor is 2, the A* search algorithm changes a lot and the paths all become long straight lines. The A* search algorithm explores fewer nodes in the directions that need to be turned.

7. Cost twice as much to climb up the vertical than moving horizontal:
   I would modify the turn cost as well as the g(n) function to directly modify the logic for horizontal and vertical movement. Simple and direct solution to the problem, but would require modifications to the core algorithm that could lead to different cost structures.