COMP3106 A1 - Technical Document

Group members: Adrian Alexander (101150602), Ryan Lo (101117765)

Contributions:

- Both group members did approximately half of the work each. Adrian wrote the helper functions and contributed to the search algorithm, and Ryan made contributions to the search algorithm logic and iterating through the neighbours. Both split up the work for the technical document and worked together.

Please answer the following questions in the technical document. For all questions, explain why your answers are correct.

1. Briefly describe how your implementation works. Include information on any important algorithms, design decisions, data structures, etc. used in your implementation. [10 marks]

Our implementation starts with the node class. Each node contains its coordinates (x and y), costs (g, h, and f), a reference to its parent node (parent), and a bool indicating whether it has passed an obstacle or not (passed_obstacle).

The algorithm by first reading the grid and initializing the start and goal nodes. The start node's values are set and added to the frontier. The algorithm loops until the frontier is empty. In each iteration, the node with the lowest f cost is removed from the frontier and added to the explored list. If this node is the goal, the path is returned. Otherwise, the neighbours are checked. If a neighbouring node is an obstacle and the current node hasn't passed an obstacle yet, the neighbour's passed_obstacle value is set to true. If the new g cost to reach the neighbour is less than the current g value or it's not in the frontier yet, the neighbour's values are updated, and the neighbour is added to the frontier.

For design decisions, we decided to use the Manhattan distance as our heuristic because it works well for grid-based environments, where only horizontal and vertical movements are allowed.

2. What type of agent have you implemented (simple reflex agent, model-based reflex agent, goal-based agent, or utility-based agent)? [3 marks]

We have implemented a model-based reflex agent. The agent keeps a model of whether or not it has passed an obstacle yet. If it has not, it would be able to traverse an obstacle. If this node is the goal, the path is returned. If the cost to reach the neighbour is less than the current g value or it's not in the frontier yet, the neighbour's values are updated, and add neighbour to frontier.

3. Is the task environment: [7 marks]

a) **Fully or partially observable?**
   Fully observable
b) **Single or multiple agents?**
   Single agent
c) **Deterministic or stochastic?**
   Deterministic
d) **Episodic or sequential?**
   Sequential
e) **Static or dynamic?**
   Static
f) **Discrete or continuous?**
   Discrete
g) **Known or unknown?**
   Known

4. What heuristic did you use for A* search for this environment? Show that your heuristic is consistent. [8 marks]

We used the Manhattan distance for our heuristic. The formula for this is described as:

$$\left[h(n) = |x_1 - x_2| + |y_1 - y_2|\right]$$

To show that the Manhattan distance is consistent,

Let $h(x)$ be the manhattan distance from x to the goal
Let $pos(x)$ be the position of x
Consider node n and successor n'

$h(n) = |pos(n) - pos(goal)|$
$h(n') = |pos(n') - pos(goal)|$
$h(n) - h(n') = |pos(n) - pos(goal)| - |pos(n') - posM(goal)|$
$\leq |pos(n) - pos(n')|$
$\leq Distance(n, n')$
$\leq cost(n, n')$
Thus, $h(n) \leq cost(n, n') + h(n')$
Therefore, $h(goal) = |pos(goal) - pos(goal)| = 0$

5. Suggest a particular instance of this problem (i.e. grid) where A* search using your heuristic would find the optimal solution faster than uniform cost search. [4 marks]

A particular instance of this problem where A* search using our heuristic would find the optimal solution faster than uniform cost search would be when the path cost of every square is 1, then every direction that uniform cost search would take is equally likely thus A* search using heuristic would go in the direction where it would be closest to the goal instead.

6. Suggest a particular instance of this problem (i.e. grid) where a greedy heuristic search using your heuristic would not find the optimal solution. [4 marks]

A particular instance of this problem where a greedy heuristic search using our heuristic would not find the optimal solution would be when the goal is blocked by multiple obstacles. It would make the path take a really long path or might not find the solution at all if every node has traversed an obstacle and the goal is still blocked by an obstacle.

7. Consider a modification of this problem where the agent must traverse exactly one obstacle (for this question, assume all grids have at least one obstacle). Determine whether your heuristic is still consistent. [4 marks]

If the problem were to be modified and the agent must traverse exactly one obstacle then our heuristic is no longer consistent as our agent is allowed to traverse an obstacle per node. Our heuristic can be easily changed though to keep track of our agent traversing an obstacle instead of a node traversing an obstacle. This change would work with this new modification of this problem.