Noah Breton 101232464

Marco Toito 101233884

Nicolas Hawrysh 101151961

1.

We have implemented a model-based agent, within the state we store the path, path cost, current grid, the location and the number of keys held.

The model is updated every move that is made, the path gets changed, the path cost is increased, the grid is changed if a door has been opened or a key is gathered, the location is updated with every move, and the number of keys will be updated if you get another key or open a door.

2.

a. Fully Observable

To find goal position and create the heuristic based on where the goal position is, it must be fully observable

b. Single Agent

There is only one pathfinder in the grid

c. Deterministic

Only agent's actions affect the state, the next state is determined by the current state and the pathfinder action

d. Sequential

The current action of the pathfinder affects the future actions

e. Static

The environment is not changing while the agent is deciding the next action

f. Discrete

The states are discrete, there are only up to four actions, so the actions are discrete, and there is no time constraint

g. Known

The agent knows how to move (traverse an edge)

3.

We used the Manhattan Distance for our A* Search

h = abs(current_x - goal_x) + abs(current_y - goal_y)

To show our heuristic is consistent:

i. h(n) <= cost(n,n') + h(n')

Consistency Proof:

Consider a node n and successor n'

h(n) = abs(n_x - goal_x) + abs(n_y - goal_y)

h(n') = abs(n'_x - goal_x) + abs(n'_y - goal_y)

h(n) - h(n') = (abs(n_x - goal_x) + abs(n_y - goal_y)) - (abs(n'_x - goal_x) + abs(n'_y - goal_y))

<= abs(n_x - n'_x) + abs(n_y - n'_y)

<= distance(n, n')

<= cost(n,n')

h(n) <= cost(n,n') + h(n')

aside

abs(n_x - goal_x) - abs(n_x' - goal_x) = abs(n_x - n'_x)

this shows that when calcuating the vector in the x direction, the difference between

n'_x to goal and n_x to the goal is the same as n_x to n'_x.

This case also applies to the y positions.

ii. h(goal) = 0

h(goal) = abs(goal_x - goal_x) + abs(goal_y - goal_y) = 0

4.

S	O	O	O	G
O	O	O	O	O

Our A* Search Implementation:

Optimal path: [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4)]

Optimal path cost: 4

States Explored: 5

Uniform Cost Search:

Optimal path: [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4)]

Optimal path cost: 4

States Explored: 9

Since the uniform cost search searches prioritizing the shortest path, it will explore multiple position on the second row but in our A* search it will never explore the second row due to the added heuristic, meaning the uniform cost search will take a longer amount of time (states explored in uniform cost > states explored in A*).

5.

O	O	O	O	S	O
O	D	D	D	D	O
O	O	G	O	O	O

Our A* Search Implementation:

Optimal path: [(0, 4), (0, 5), (1, 5), (2, 5), (2, 4), (2, 3), (2, 2)]

Optimal path cost: 6

States Explored: 10

Greedy Heuristic Search:

Optimal path: [(0, 4), (0, 3), (0, 2), (0, 1), (0, 0), (1, 0), (2, 0), (2, 1), (2, 2)]

Optimal path cost: 8

States Explored: 9

Using the Manhattan Distance as the heuristic, the greedy heuristic search would never go down the path to the right of the start because the heuristic is 5, and the heuristic for every value down the past to the left is less than 5. This means the heuristic will not go down the optimal path.

6.

Since we are using Manhattan Distance as our heuristic, keys and doors are not considered, we only store the distance from the current state to the goal state.

The Q3 Consistent heuristic proof is still valid as the new modification has no effect on the formula.

$h(n) <= cost(n,n') + h(n')$ or $h(goal) = 0$

Group assessment:

1. Each group member made significant contributions

2. Each group member made equal contributions

3. Code and questions were written in a group setting, with all group members present