CMPT 310

Assignment 1-Report

Weiling Zheng, weilingz@sfu.ca, 301294091

In the assignment, we need to find a path from the start cell to the goal cell. I use the A* search to find the path.

Firstly, I need to create a header file (SearchCell.h) with the structure of each cell including the x coordinate, y coordinate, cost value(G), heuristic function(H) and the total value(F). For the heuristic function, I use the Manhattan method, where you calculate the total number of squares moved horizontally and vertically to reach the target square from the current square, ignoring diagonal movement, and ignoring any obstacles that may be in the way.

Manhatten method:

H(n) = abs(start cell's x coordinate - goal cell's x coordinate) +
abs(start cell's y coordinate - goal cell's y coordinate)

This means that the cell can only expand in the vertical line and horizontal line.

<e.g>

```
  *
* A *
  *
```

A is the cell that I want to expand and '*' is the only way that it can expand.

Secondly, I create a PathFinding class for how to find a lowest F value path from the start cell to the goal cell. I need a open list for add the reachable or walkable squares adjacent to the cell that the start cell can go through, ignoring the obstacles. And I also need a visited list that I do not need to look again. For adding and deleting efficiently, I use the <vector> library and create a Vector class. Then I use the recursive method to continue finding the lowest value path.

This is the output of my program:
==========Output==========

```
--------------------------------------------------
--------------Assignment 1 Part A-----------------
--------------------------------------------------

This is an 18*18 grid with co-ordinates from 0 to 17.
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . ----------. .
. . . . . | . .
. . . . . .
. . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
Please enter the start cell(x coordinate and y coordinate):
0 0
Please enter the goal cell(x coordinate and y coordinate):
The shortest path is labelled by '* '
. . . . . *
. . . . . *
. . . . . *
. . . . . *
. . . . . ----------. *
. . . . . | . *
. . . . . *
. . . . . *
. . . . . | . . . . . *
. . . . . | . . . . . *
. . . . . | . . . . . *
. . . . . | . . . . . *
. . . . . *
. . . . . *
. . . . . *
. . . . . *
* * * * * * * * * * * * * * * * * *
The shortest path is :
(0,0) - (1,0) - (2,0) - (3,0) - (4,0) - (5,0) - (6,0) - (7,0) - (8,0) - (9,0) - (10,0) - (11,0) - (12,0) - (13,0) - (14,0) - (15,0) - (16,0) -
(17,0) - (17,1) - (17,2) - (17,3) - (17,4) - (17,5) - (17,6) - (17,7) - (17,8) - (17,9) - (17,10) - (17,11) - (17,12) - (17,13) - (17,14) -
(17,15) - (17,16) - (17,17) -
The length of the shortest path is 34
The total number of nodes is 312 which is places on the frontier
--------------------------------------------------
--------------Assignment 1 Part B-----------------
--------------------------------------------------

This is an 18*18 grid with co-ordinates from 0 to 17.
With A, B, C and D landmarks
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . ----------. .
. . . . . A . . . . . B . . | . .
. . . . . .
. . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . | . . . . . .
. . . . . C . . | . . . . D . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
Please enter the start cell(x coordinate and y coordinate):
0 0
Please enter the goal cell(x coordinate and y coordinate):
17 17
```

```
The shortest path is labelled by '* '
. . . . . . . . . . . . . . . . . . *
. . . . . . . . . . . . . . . . . . *
. . . . . . . . . . . . . . . . . . *
. . . . . . . . . . . . . . . . . . *
. . . . . . . . . . . ----------. . *
. . . . . A . . . . . . * * * | . . *
. . . . . . . . . . . . . * . * * * *
. . . . . . * * * * * * * . . . . .
. . . . . . * | . . . . . . . . . .
. . . . . . * | . . . . . . . . . .
. . . . . . * | . . . . . . . . . .
. . . . . . * | . . . . . . . . . .
. . . . . * * | . . . . D . . . . .
. . . . . * . . . . . . . . . . . .
. . . . . * . . . . . . . . . . . .
. . . . . * . . . . . . . . . . . .
. . . . . * . . . . . . . . . . . .
* * * * * * . . . . . . . . . . . .
The shortest path is :
(0,0) - (1,0) - (2,0) - (3,0) - (4,0) - (5,0) - (5,1) - (5,2) - (5,3) - (5,4) - (5,5) - (5,5) - (6,5) - (6,6) - (6,7) - (6,8) - (6,9) - (6,10) -
(7,10) - (8,10) - (9,10) - (10,10) - (11,10) - (12,10) - (12,11) - (12,12) - (13,12) - (14,12) - (14,11) - (15,11) - (16,11) - (17,11) - (17,12) -
(17,13) - (17,14) - (17,15) - (17,16) - (17,17) -

The length of the shortest path is 36

The number of cells on the frontier that is from start cell to nearest landmark: 66
The number of cells on the frontier that is from goal cell to nearest landmark: 80
The total number of nodes is 146 which is places on the frontier
```

==========Output==========

Finally, I put Part A and Part B of the assignment in the main function. In Part A, we should output the shortest path with the lowest length from the start cell to the goal cell.

From the output in Part B, we have four landmarks(A, B, C and D) in the grid map and we need to find a more efficient path maybe it is not necessarily optimal. I set the cell from the start cell (0,0) to the goal cell (17,17) and the shortest way needs 34 steps but the total number of nodes on the frontier is 312.

The more efficient way from (0,0) to (17,17) needs 36 steps but the total number of nodes on the frontier is 146.So we choose the second one as our output.For this path, the length of it may not be lower, but the time efficiency is better. This means that we need to find the path with lower expansion which is more efficient.