Maze

A1045141 賴愉婧

一. 程式解說

主程式：Maze.cpp
副程式：Search.cpp, DFS.cpp, BFS.cpp, Greedy.cpp, AStar.cpp
header 檔：Search.h

前置作業

1. 輸入迷宮的檔案名稱
2. 讀檔儲存迷宮，並記錄原始迷宮的狀態
3. 定義 struct Point 儲存點的位置、狀態
4. 設起點 S，目標為找到終點 G

二. 解題原理

DFS：

- 變數 vector<Point> passed 儲存已經走過的點，設起點 S 為 passed 的第一個點

1. 從第一個節點開始找
2. 設定目前節點狀態為”被發現”
3. 搜尋節點周圍可走的方向(與節點相連的點)
4. 4-1. 若遇到第一個相連可走的點為”未發現”
   a. 設 next Point(下一節點)的 previous Vertex(前一節點在 passed 中的位置)為目前節點
   b. 將 next Point 加入(push) passed
   c. 判斷 next Point 是否為 G，若不是則繼續找下一個點
   d. 重複步驟 2，從最後一個節點開始找，直到找到 G(結束)

   4-2. 若第一個點已走過(“被發現”或”結束”)
   a. 尋找第二個相連的點，若相連的點都已走過，代表為死路
   b. 設定目前節點狀態為”結束”
5. 標示路徑，在 passed 找到 G，依照 G 的 previous Vertex 找尋到達 G 的路徑

BFS：

- 變數 vector<Point> passed 儲存已經走過的點，設起點 S 為 passed 的第一個點

1. 從第一個節點開始找
2. 設定目前節點狀態為”被發現”
3. 搜尋節點周圍可走的方向(與節點相連的點)
4. 若遇到相連的點為”未發現”
    a. 設 next Point(下一節點)的 previous Vertex(前一節點在 passed 中的位置)為目前節點
    b. 將 next Point 加入 passed
    c. 判斷 next Point 是否為 G
    d. 重複步驟 4，直到找完所有相連的點
5. 設定目前節點狀態為”結束”
6. 尋找下一個節點(節點加 1)，重複步驟 2，直到找到 G(結束)
7. 標示路徑，在 passed 找到 G，依照 G 的 previous Vertex 找尋到達 G 的路徑

Greedy：

- 變數 vector<Point> passed 儲存已經走過的點，設起點 S 為 passed 的第一個點
- 變數 vector<Point> path 儲存可能要走的點，起始等於 passed
- h = heuristic function，預估抵達 G 的距離

1. 從目前預估離 G 最近的點(path 最後的點)current Point 開始找
2. 從 path 移除(pop)預估距離最近的點(path 最後的點)
3. 設定目前狀態為”被發現”
4. 搜尋節點周圍可走的方向(與節點相連的點)
5. 若遇到相連的點為”未發現”
    a. 搜尋 current Point 在 passed 的位置，設為 next Point(下一節點)的 previous Vertex(前一節點在 passed 中的位置)
    b. 將 next Point 加入 passed
    c. 判斷 next Point 是否為 G，不是的話加入 path
    d. 重複步驟 5，直到找完所有相連的點
6. 排序 path 的 h 從大到小，預估距離最近的點放最後
7. 設定目前節點狀態為”結束”，尋找下一個節點
8. 重複步驟 1，直到找到 G(結束)

A*：

- 變數 vector<Point> passed 儲存已經走過的點，設起點 S 為 passed 的第一個點，S 的 g 為 0
- 變數 vector<Point> path 儲存可能要走的點，起始等於 passed
- h = heuristic function，預估抵達 G 的距離
- g = 已走距離
- f = evaluation function = h + g，預估+實際抵達 G 的總距離

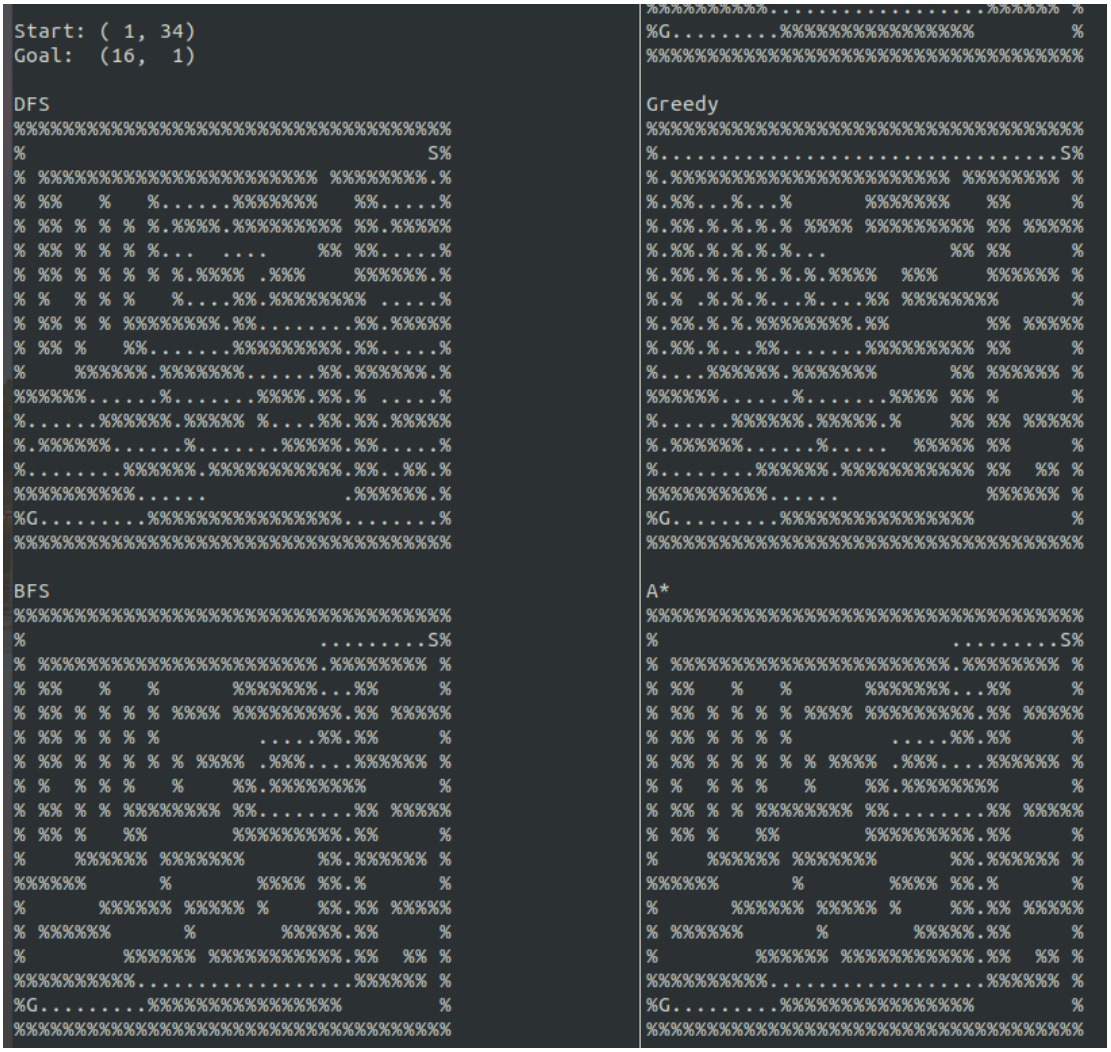
1. 從目前離 G 最近的點(path 最後的點)current Point 開始找
2. 從 path 移除(pop)總距離最近的點(path 最後的點)
3. 設定目前狀態為”被發現”
4. 搜尋節點周圍可走的方向(與節點相連的點)
5. 若遇到相連的點為”未發現”
    a. next Point(下一節點)的 g 等於 current Point 的 g 加 1
    b. 搜尋 current Point 在 passed 的位置，設為 next Point 的 previous Vertex(前一節點在 passed 中的位置)
    c. 將 next Point 加入 passed
    d. 判斷 next Point 是否為 G，不是的話加入 path
    e. 重複步驟 5，直到找完所有相連的點
6. 排序 path 的 f 從大到小，總距離最近的點放最後
7. 設定目前節點狀態為”結束”，尋找下一個節點
8. 重複步驟 1，直到找到 G(結束)

三. 印出結果

Small Maze

```
Start: ( 3, 11)
Goal:  ( 8,  1)

DFS
%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%          % %        %
%     %%%%%%% % %%%%%%% %
%%%%%%.....S   %        %
%     %.%%%%%%% %% %%%%%%
% %%%%.%.....      %    %
%       ...%%%.%%%    % %
%%%%%%%%%%%... %%%%%%% %
%G.........%%          %
%%%%%%%%%%%%%%%%%%%%%%%%%%

BFS
%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%          % %        %
%     %%%%%%% % %%%%%%% %
%%%%%%%       S..%        %
%     % %%%%%%%.%% %%%%%%
% %%%% %      ..    %    %
%          %%%.%%%    % %
%%%%%%%%%%%... %%%%%%% %
%G.........%%          %
%%%%%%%%%%%%%%%%%%%%%%%%%%

Greedy
%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%          % %        %
%     %%%%%%% % %%%%%%% %
%%%%%%.....S   %        %
%     %.%%%%%%% %% %%%%%%
% %%%%.%.....      %    %
%       ...%%%.%%%    % %
%%%%%%%%%%%... %%%%%%% %
%G.........%%          %
%%%%%%%%%%%%%%%%%%%%%%%%%%

A*
%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%          % %        %
%     %%%%%%% % %%%%%%% %
%%%%%%%       S..%        %
%     % %%%%%%%.%% %%%%%%
% %%%% %      ..    %    %
%          %%%.%%%    % %
%%%%%%%%%%%... %%%%%%% %
%G.........%%          %
%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Midium Maze



Big Maze