CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập – Tự do – Hạnh Phúc

------0------

BÁO CÁO ĐỒ ÁN 1

Môn: Cơ Sở Trí Tuệ Nhân Tạo

Tên đề tài: TÌM KIẾM HEURISTIC VỚI A*

Thực hiện: 1612286 Nguyễn Hồng Khang

1612778 Nguyễn Anh Tuấn

1. **Phân công:**

	Công việc
Hồng Khang	Viết hàm isValid, isEmptySquare, isDestination, calculateHeuristic, Output
Anh Tuấn	Viết hàm aStarSearch và báo cáo

2. **Mức độ hoàn thành đồ án:**

   Hoàn thành 100% yêu cầu

3. **Các bộ test:**

   a. Bộ test 1:

   7
   9 0
   6 6
   0 0 0 0 0 0 1
   0 0 0 0 0 1 1
   1 1 0 0 0 0 1
   0 1 1 0 0 0 0
   0 1 1 0 0 1 0
   0 1 0 0 0 1 0
   0 0 0 0 0 0 0

   b. Bộ test 2:

   9
   8 2
   1 7
   0 0 0 0 0 0 1 0 1
   0 0 0 0 0 1 1 0 0
   1 1 0 0 0 0 1 1 0
   0 1 1 0 0 0 0 0 1
   0 1 1 0 0 1 0 0 1
   0 1 0 0 0 1 0 0 0

0 0 0 0 0 0 0 0 1
1 0 0 1 0 1 0 1 0
0 1 0 0 0 1 0 0 0

c. Bộ test 3:

9
5 0
1 7
0 0 0 0 0 0 1 0 1
0 0 0 0 0 1 1 0 0
1 1 0 0 0 0 1 1 0
0 1 1 0 0 0 0 1 1
0 1 1 0 0 1 0 0 1
0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1
1 0 0 1 0 1 0 1 0
0 1 0 0 0 1 0 0 0

4. Sơ đồ biểu diễn hệ thống phần mềm:

a. Hàm bool isValid(int row, int col):
   Trả về true nếu ô đó hợp lệ và ngược lại trả về false.
b. Hàm bool isEmptySquare(vector<vector<int>> grid, int row, int col):
   Kiểm trả ô đó có chướng ngại vật không, trả về true nếu ô đó trống và ngược lại trả về false
c. Hàm bool isDestination(int row, int col, Pair dest):
   Kiểm tra ô đó có phải là Goal hay không, trả về true nếu phải và ngược lại trả về false.
d. Hàm double calculateHeuristic(int row, int col, Pair dest):
   Hàm tính heuristic Euclidean tại ô đó.
e. Hàm void Output(vector<vector<cell>> cellDetails, vector<vector<int>>grid, Pair dest):
   Hàm xuất ra kết quả nếu tìm được đường đến đích.

f. Hàm int aStarSearch(vector<vector<int>> grid, Pair src, Pair dest):
Hàm trả về 0 nếu tìm được đường đến Goal và trả về -1 nếu không đến được Goal.

5. Mô tả:

a. Cấu trúc dữ liệu:
- Bản đồ nhập vào ta sẽ lưu dưới dạng vector các vector (giống mảng 2 chiều)
- Dùng pair<int,int> để lưu tọa độ các điểm
- Dùng vector các vector để lưu closeList (Danh sách các điểm mà ta sẽ không mở nữa)
- Dùng set để lưu openList (danh sách các điểm ta đã thăm qua và đưa vào để chờ được mở), cấu trúc set sắp xếp phần tử nhỏ nhất lên đầu, ở đây phần tử được xem là nhỏ khi có f nhỏ (f=g+h).

b. Thuật toán:
Đầu tiên ta kiểm tra Start và Goal có hợp lệ không, nếu không thì trả về -1 nếu có thì đi tiếp. Kiểm tra Start có trùng với Goal hay không, nếu có thì không cần xét các điểm lân cận mà xuất kết quả ra luôn, nếu không thì ta thêm Start vào openList. Bắt đầu vào vòng lặp cho đến khi openList rỗng, xóa ô đang xét khỏi openList và đặt closeList tại vị trí đó bằng true. Xét các ô lân cận của ô đang xét, nếu ô đó là đích thì xuất kết quả, nếu không thì cập nhật giá trị tại ô này khi ô này chưa từng ở trong openList hoặc giá trị hiện tại tốt hơn giá trị đã lưu trong openList. Cứ chạy đến khi hết openList, nếu vẫn chưa tìm được Goal thì trả về -1, còn nếu tìm được Goal thì gọi hàm Output và trả về 0. Hàm Output dùng cấu trúc Stack để truy vết đường đi từ Goal quay về Start và xuất ra cấu trúc đúng theo đề bài.

6. Tài liệu tham khảo:

https://en.wikipedia.org/wiki/A*_search_algorithm
https://www.hackerearth.com/fr/practice/notes/a-search-algorithm/