****

**BÁO CÁO ĐỒ ÁN 1:**

***A\* Robot PathFinding***

*Sinh viên thực hiện:*

*Lê Hồng Quang-18127190*

*Nguyễn Lê Đức Hoàng-18127219*

***1.MỨC ĐỘ HOÀN THÀNH***

|  |  |
| --- | --- |
| TASK 1 | 100% |
| TASK 2 | 100% |

***2.PHÂN CÔNG CÔNG VIỆC***

|  |  |  |
| --- | --- | --- |
| **SINH VIÊN** | **CÔNG VIỆC** | **ĐÁNH GIÁ** |
| 18127190 | Code thuật toán,Viết báo cáo,Hỗ trợ giao diện | 100% |
| 18127274 | Code giao diện,Hỗ trợ viết báo cáo | 100% |

***3.TEST CASE***

|  |  |  |
| --- | --- | --- |
| Trường hợp | Input | Ouput |
| 1 | 7 7  0 0  6 6  0 0 0 0 0 0 1  0 0 1 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 | 9  (0, 0) (1, 1) (0, 2) (1, 3) (2, 4) (3, 5) (4, 6) (5, 6) (6, 6)  S - x - - - o  - x o x - o o  o o - - x - o  - o o - - x -  - o o - - o x  - o - - - o x  - - - - - - G |
| 2 | 7 7  0 0  6 6  0 0 0 0 0 0 0  0 0 0 0 0 1 0  0 1 1 0 0 1 0  0 1 1 0 0 1 0  0 1 1 0 0 1 0  0 1 0 1 1 1 0  0 0 0 0 0 0 0 | -1 |
| 3 | 5 5  4 1  0 2  0 0 0 0 0  0 1 1 0 0  0 0 0 0 1  0 0 1 0 0  0 0 1 0 0 | 5  (4, 1) (3, 1) (2, 2) (1, 3) (0, 2)  - - G - -  - o o x -  - - x - o  - x o - -  - S o - - |
| 4 | 8 8  0 0  6 6  0 0 0 0 0 0 1 0  0 0 0 0 0 1 1 1  1 1 0 0 0 0 1 0  0 1 1 0 0 0 0 1  0 1 1 0 0 1 0 1  0 1 0 0 0 1 0 0  0 0 0 0 0 0 0 0  0 1 1 0 0 0 0 0 | 8  (0, 0) (1, 1) (2, 2) (3, 3) (4, 4) (5, 4) (6, 5) (6, 6)  S - - - - - o -  - x - - - o o o  o o x - - - o -  - o o x - - - o  - o o - x o - o  - o - - x o - -  - - - - - x G -  - o o - - - - - |

***4.SƠ ĐỒ LUỒNG CHƯƠNG TRÌNH***

|  |  |
| --- | --- |
| Hàm | Chức năng |
| readfile(filename,mode)->tuple | Hàm mở file để đọc dữ liệu gồm tên file (filename), chế độ đọc (mode). Hàm trả về tuple gồm: (ma trận điểm, điểm đầu, điểm đích) |
| writefile(filename,mode,data,matrix,start,goal) | Hàm ghi dữ liệu vào file gồm tên file, chế độ,dữ liệu ghi vào, ma trận , điểm đầu, điểm cuối |
| main | Hàm main |
| duongdingannhat(node\_ht) | Trả về đường đi lùi từ node đích về tới node bắt đầu |
| create\_child\_node(matran,node\_ht) | Mở rộng các node xung quanh node hiện tại với các điều kiện thỏa yêu cầu bài toán |
| thuattoan(matran,start,end) | Thuật toán A\* áp dụng euclid distance với ma trận , điểm đầu và điểm cuối |

***5.MIÊU TẢ CẤU TRÚC DỮ LIỆU***

|  |  |
| --- | --- |
| **CLASS NODE** | |
| THUỘC TÍNH | +parent: Node cha  +vitri: vị trí node là một tuple (vitri[0],vitri[1])  +g:Khoảng cách từ node hiện tại đến node bắt đầu  +h:Chi phí heuristic(Euclid distance)->Khoảng cách từ node hiện tại đến node đích  +f:g+h |
| METHODS | +\_\_eq\_\_(self,other): so sánh vị trí của self với other trả về True hoặc False |

***6.MIÊU TẢ THUẬT TOÁN***

A\* algorithm with Euclid distance:

+Tạo node bắt đầu , node kết thúc

+Khởi tạo open\_list và closed\_list với open\_list là danh sách lưu các node đang xét và closed\_list lưu các node đã xét

+Thêm node bắt đầu vào open\_list

+Thiết lặp điều kiện dừng để vòng lặp không chạy vô hạn

+Lặp cho đến khi tìm thấy điểm kết thúc

+Trong vòng lặp:

-Tìm kiếm node có chi phí f thấp nhất trong open\_list

-Gán node hiện tại với node đã tìm được

-Loại bỏ node hiện tại khỏi open\_list sau đó thêm vào closed\_list

-Nếu node hiện tại bằng node kết thúc-> trả về đường đi có chi phí thấp nhất

-Khởi tạo danh sách node con(hàng xóm) của node hiện tại với các phần tử trong danh sách thỏa các điều kiện của bài toán

-Với mỗi phần tử trong danh sách hàng xóm , nếu nó có trong closed\_list->tiếp tục vòng lặp, sau đó khởi tạo các chi phí f,g,h

-Nếu các phân tử trong danh sách hàng xóm bằng với phần tử trong open\_list mà chi phí của nó lớn hơn chi phí của node kia trong open\_list->tiếp tục vòng lặp

-Thêm phần tử child vào open\_list

***7.THAM KHẢO***

<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>->Thuật toán

<https://www.youtube.com/?gl=VN> ->Giao diện

***8.CÁCH CHẠY CHƯƠNG TRÌNH***

a) Thuật toán A\*

-Vào cmd-> gõ đường dẫn thư mục chứa 3 file: file thuật toán, file input, file output

-Gõ cú pháp: python .\<Ten file thuật toán> <Tên file input> <Tên file output> -> enter

-Nếu muốn thay đổi test case thì sửa file input sau đó lưu lại rồi mới chạy dòng lệnh trên

b) Giao diện:

-Tạo node bắt đầu: First-Left-Click Mouse

-Tạo node kết thúc:Second-Left-Click Mouse

-Tạo chướng ngại vật: Left-Click Mouse sau khi đã tạo node bắt đầu và node kết thúc

-Xóa chướng ngại vật: Right-Click

-Chạy thuật toán: nhấn Space

-Xóa ma trận : Ctrl+C