

**BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

----□□□□□----



BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI: ỨNG DỤNG THUẬT TOÁN A* ĐỂ TÌM ĐƯỜNG

Môn học: Nhập môn trí tuệ nhân tạo
Giảng viên hướng dẫn: TS Nguyễn Thị Tuyết Hải
Lớp: D22CQCN02-N

Sinh viên thực hiện:	1. Văn Duy Khánh	N22DCCN139
	2. Vũ Đình Khoa	N22DCCN142
	3. Phạm Văn Phú	N22DCCN160
	4. Phạm Hùng Thiên	N22DCCN180

TPHCM, tháng 05/2025

Mục lục

Mục lục	2
Chương 1: Trình bày cơ sở lý thuyết và trình bày về thuật toán	4
1.1. Giới thiệu thuật toán	4
1.2. Trình bày thuật toán A*	5
1.3. Ưu và nhược điểm của thuật toán A*	6
Chương 2: Trình bày về thiết kế và cấu trúc và thư viện	7
Chương 3: Phân công công việc, link github và demo của chương trình.....	8

Danh mục hình ảnh

Hình 3.1: Bảng phân công nhiệm vụ

Hình 3.2: Lưu đồ giải thuật chương trình

Chương 1: Trình bày cơ sở lý thuyết và trình bày về thuật toán

1.1. Giới thiệu thuật toán

Thuật toán A* là một trong những thuật toán tìm kiếm đường đi ngắn nhất phổ biến nhất và hiệu quả nhất trong lĩnh vực trí tuệ nhân tạo và khoa học máy tính. Thuật toán này kết hợp các ưu điểm của tìm kiếm ưu tiên theo độ sâu (Depth-First Search) và tìm kiếm theo chiều rộng (Breadth-First Search) để tìm ra đường đi tối ưu từ điểm bắt đầu đến điểm đích trong một không gian tìm kiếm.

- **Cơ sở lý thuyết:** Thuật toán A* sử dụng hai hàm chi phí để đánh giá các bước đi trong không gian tìm kiếm:

+ $g(n)$: Chi phí từ điểm bắt đầu đến đỉnh n .

+ $h(n)$: Ước lượng chi phí từ đỉnh n đến đích. Hàm này thường được gọi là hàm heuristic.

- Tổng chi phí của một đỉnh n được tính bằng hàm $f(n)$: $f(n) = g(n) + h(n)$

Trong đó:

- $g(n)$ là chi phí thực tế để đi từ điểm bắt đầu đến đỉnh n .

- $h(n)$ là chi phí ước lượng từ đỉnh n đến đích, dựa trên một hàm heuristic.

Để đảm bảo thuật toán A* tìm được đường đi ngắn nhất, hàm heuristic phải chấp nhận được (admissible), tức là nó không bao giờ được đánh giá cao hơn chi phí thực tế.

1.2. Trình bày thuật toán A*

Thuật toán A* có thể được trình bày theo các bước sau:

- Khởi tạo: Đặt đỉnh bắt đầu vào danh sách mở (open list), là một hàng đợi ưu tiên dựa trên hàm $f(n)$. Đặt chi phí ban đầu của đỉnh này là 0 và ước lượng chi phí đến đích bằng hàm heuristic.

- Duyệt đỉnh: Trong khi danh sách mở không rỗng:

- + Lấy đỉnh n có giá trị $f(n)$ thấp nhất ra khỏi danh sách mở.

- + Nếu n là đích, thuật toán kết thúc và đường đi tối ưu đã được tìm thấy.

- + Di chuyển đỉnh n vào danh sách đóng (closed list) để đánh dấu rằng nó đã được duyệt.

- + Mở rộng đỉnh: Đối với mỗi đỉnh kề m của n :

- Tính giá trị $g(m)$ bằng $g(n)$ cộng với chi phí từ n đến m .

- + Nếu m nằm trong danh sách đóng và giá trị $g(m)$ mới thấp hơn giá trị $g(m)$ hiện tại, bỏ qua đỉnh m .

- + Nếu m không nằm trong danh sách mở hoặc giá trị $g(m)$ mới thấp hơn giá trị $g(m)$ hiện tại: + Cập nhật $g(m)$ và tính lại $f(m)$.

- + Đặt n là cha của m .

- + Nếu m chưa có trong danh sách mở, thêm m vào danh sách mở.

- Thuật toán kết thúc: Khi đỉnh đích được lấy ra từ danh sách mở, đường đi tối ưu đã được xác định bằng cách truy vết từ đỉnh đích về điểm bắt đầu thông qua các đỉnh cha.

1.3. Ưu và nhược điểm của thuật toán A*

- **Ưu điểm:** Một thuật giải linh động, tổng quát, trong đó hàm chứa cả tìm kiếm chiều sâu, tìm kiếm chiều rộng và những nguyên lý Heuristic khác. Nhanh chóng tìm đến lời giải với sự định hướng của hàm Heuristic. Chính vì thế mà người ta thường nói A* chính là thuật giải tiêu biểu cho Heuristic.

- Nhược điểm:

+ Bộ nhớ: A* có thể tiêu tốn nhiều bộ nhớ vì phải lưu trữ nhiều đỉnh trong danh sách mở và danh sách đóng.

+ Hiệu suất: Nếu hàm heuristic không được chọn đúng cách, thuật toán có thể trở nên không hiệu quả.

Như vậy, thuật toán A* là một công cụ mạnh mẽ cho việc tìm kiếm đường đi trong không gian tìm kiếm, đặc biệt hữu ích trong các ứng dụng như lập kế hoạch robot, trò chơi điện tử, và các hệ thống GIS.

Chương 2: Trình bày về thiết kế và cấu trúc và thư viện

Danh sách thư viện áp dụng:

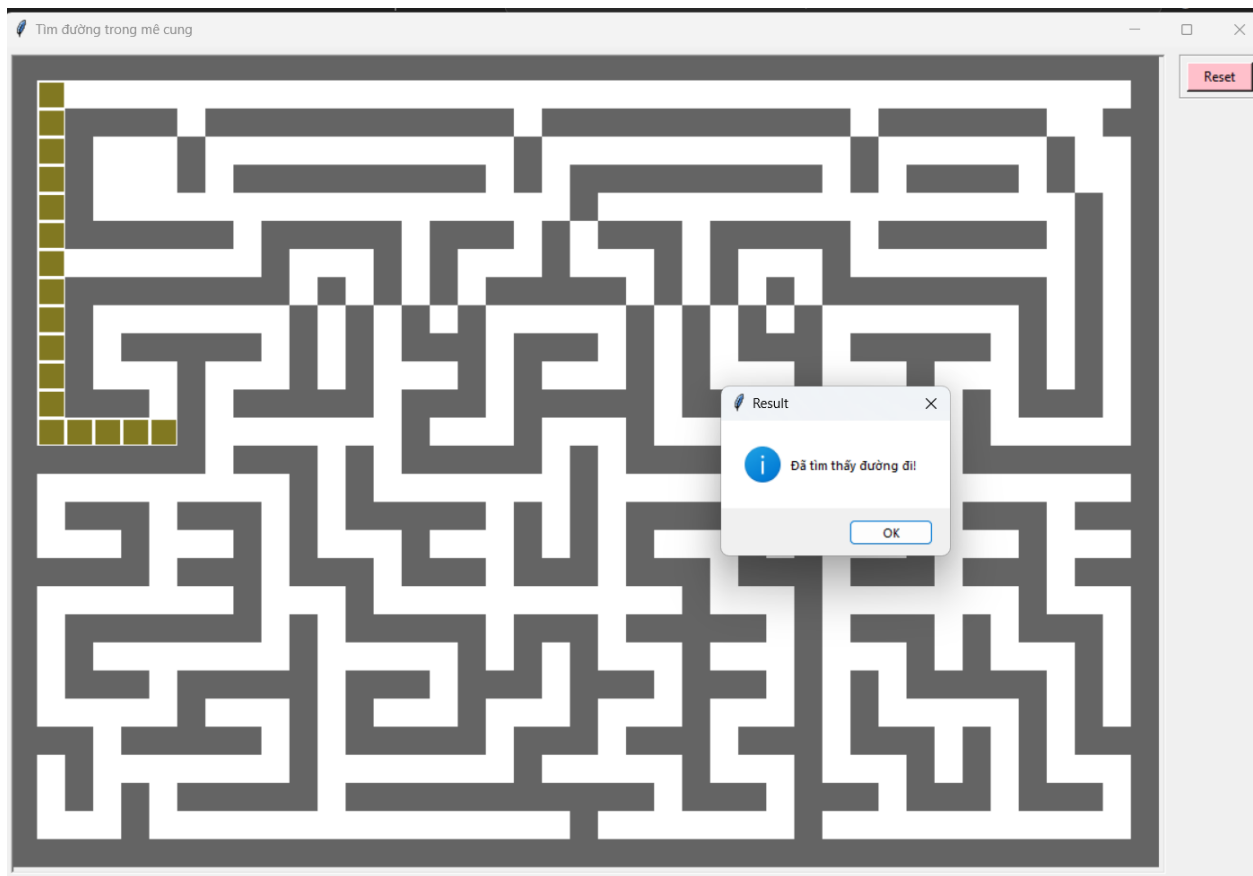
- **math**: Để tính toán các giá trị toán học, đặc biệt là trong hàm heuristic (sử dụng hàm sqrt).
- **simpleai.search**: Cung cấp các thuật toán tìm kiếm, cụ thể là thuật toán A* (astar), và lớp cơ sở. - SearchProblem để định nghĩa bài toán tìm kiếm.
- **numpy**: Để tạo và xử lý các mảng số học, được sử dụng trong việc tạo hình ảnh mê cung.
- **cv2 (OpenCV)**: Để xử lý và chuyển đổi hình ảnh.
- **tkinter**: Để tạo giao diện người dùng (GUI) và xử lý các sự kiện. - PIL (Pillow): Để chuyển đổi hình ảnh từ định dạng OpenCV sang định dạng có thể hiển thị trên tkinter.

Chương 3: Phân công công việc, link github và demo của chương trình

Mã sinh viên	Tên sinh viên	Công việc
N22DCCN139	Văn Duy Khánh	Lên ý tưởng, tìm hiểu thư viện và thuật toán, lập trình chương trình
N22DCCN142	Vũ Đình Khoa	Lên ý tưởng, tìm hiểu thư viện và thuật toán, lập trình chương trình
N22DCCN160	Phạm Văn Phú	Lên ý tưởng, tìm hiểu thư viện và thuật toán, lập trình chương trình
N22DCCN180	Phạm Hùng Thiên	Lên ý tưởng, tìm hiểu thư viện và thuật toán, lập trình chương trình

Hình 3.1: Bảng phân công nhiệm vụ

Demo sản phẩm:

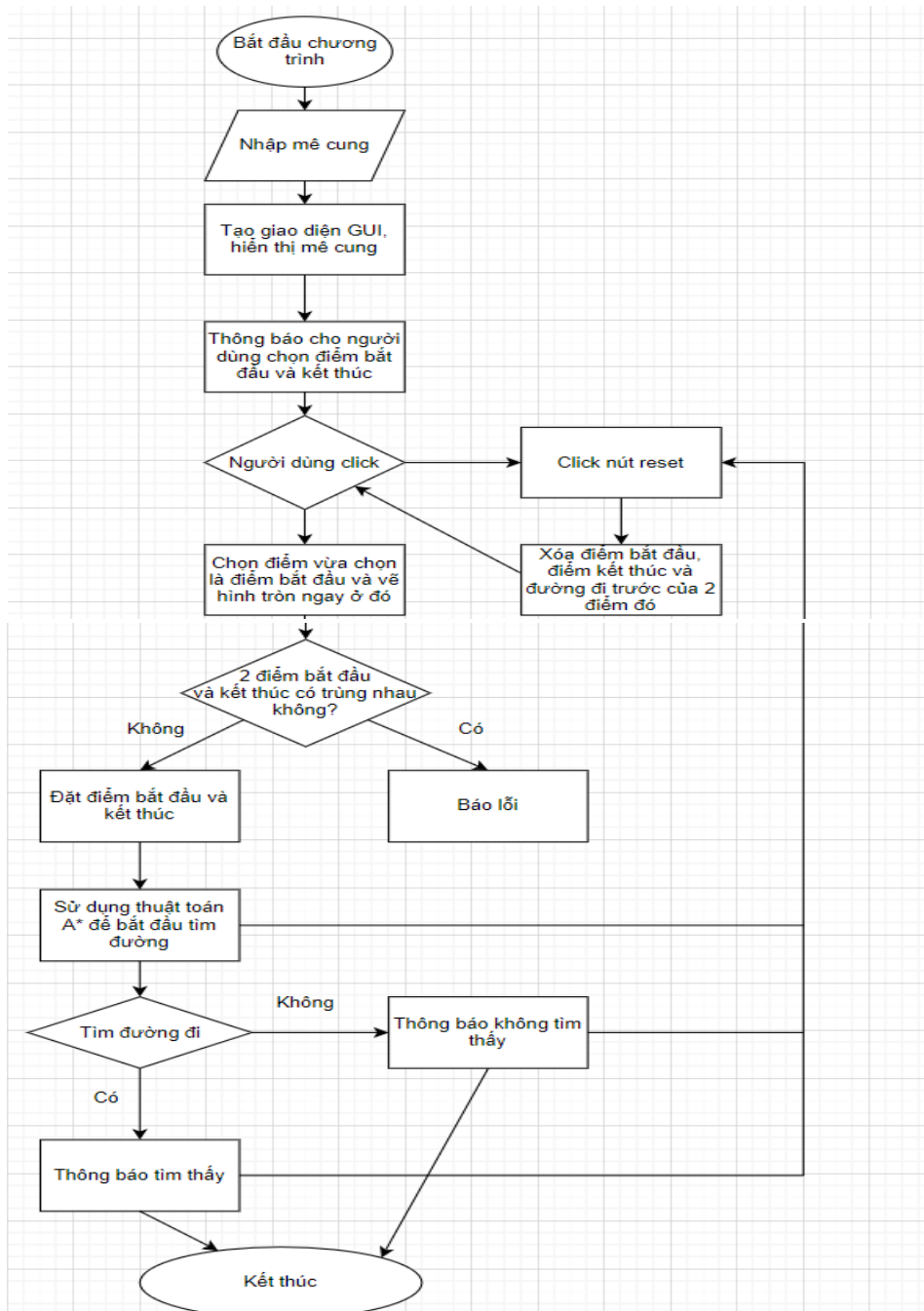


Link github: <https://github.com/VanDuyKhanh2004/AI-Find-way-by-A->

Link lưu đồ:

https://drive.google.com/file/d/13H2uFpp9C2cYPZrK2Dq9UGE0P27eo_q9/view?usp=sharing

Lưu đồ giải thuật:



Hình 3.2: Lưu đồ giải thuật chương trình