HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

**BÁO CÁO BÀI TẬP LỚN**

**Tìm đường đi trên bản đồ phường Nguyễn Trung Trực**

**NHÓM 16**

**Học phần: Nhập môn Trí tuệ nhân tạo - IT3160**

**Mã lớp: 139312**

|  |  |  |
| --- | --- | --- |
| **Giảng viên:** | PGS.Trần Đình Khang |  |
| **Thành viên nhóm:** |  |  |
| **MSSV:** | **Họ và tên** | **Vai trò** |
| 20200392 | Đào Nhật Minh | Nhóm trưởng |
| 20205134 | Hoàng Vân Trường | Thành viên |
| 20205035 | Nguyễn Thu Trang | Thanh viên |
| 20205041 | Trần Anh Tuấn | Thành viên |
| 20205103 | Nguyễn Giang Nam | Thành viên |

Hà Nội, 5/2023

GIỚI THIỆU BÀI TOÁN

Bài toán tìm kiếm đường đi trên bản đồ là một bài toán quan trọng trong lĩnh vực trí tuệ nhân tạo và khoa học máy tính. Bài toán nhằm tìm hiểu cách tìm ra đường đi tối ưu hoặc đường đi ngắn nhất giữa hai điểm trên một bản đồ được mô phỏng trong không gian hai chiều.

Trong đó, nhóm 16 chúng em được giao chủ đề "Tìm đường đi trên bản đồ phường Nguyễn Trung Trực". Bằng cách mô phỏng lại dữ liệu bản đồ từ hình ảnh bản đồ thông qua ngôn ngữ lập trình python, tìm đường đi ngắn nhất dựa trên thuật toán Dijkstra, chúng em đã hoàn thành project trong thời gian cho phép.

Sau đây là các nội dung mà nhóm đã tìm hiểu về lý thuyết thuật toán tìm đường. Cũng như các kết quả mô phỏng và những nhận xét về phương pháp này.

BIỂU DIỄN BÀI TOÁN

Sau khi nghiên cứu yêu cầu của bài toán, chúng em đã mô hình hoá bài toán dựa trên ngôn ngữ lập trình python tạo thành một ứng dụng tìm đường đi như sau:

* Chuyển bản đồ ban đầu thành một đồ thị có hướng, trong đó tập các đỉnh là các ngã rẽ trên bản đồ, tập các cạnh là các tuyến đường được đi. Dựa vào toạ độ thực tế, chúng em xác định được vị trí biên của bản đồ (là một ảnh hình chữ nhật), vị trí của các ngã rẽ thực tế, từ đó chuyển thành các đỉnh của đồ thị dựa vào vị trí tương đối toạ độ của các ngã rẽ với các biên.
* Sau khi chuyển dữ liệu từ bản đồ thành một đồ thị có hướng, chúng em tính toán toạ độ của điểm bắt đầu, kết thúc ở vị trí nào trên bản đồ rồi tiếp tục tính khoảng cách ngắn nhất đến một con đường đã được mô hình sẵn.
* Sau đó, sử dụng thuật toán dijkstra để tìm đường đi ngắn nhất giữa 2 điểm vừa tìm được (khi đó 2 điểm được coi như 2 đỉnh của đồ thị và 2 đỉnh đó với các đầu mút của đường mà điểm đó nằm trên được coi là 4 cạnh mới).

PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI

Chúng em nhận thấy việc tìm đường đi thực tế là lựa chọn một số con đường để di chuyển từ điểm bắt đầu, qua các ngã rẽ tới điểm kết thúc. Từ đó, chúng ta hoàn toàn có thể đưa bài toán trở thành việc **tìm đường đi ngắn nhất giữa 2 điểm trên một đồ thị có hướng**. Trong đó, các ngã rẽ là các đỉnh của đồ thị, các tuyến đường, con phố di chuyển qua là các cạnh của đồ thị có hướng.

Dựa vào kiến thức đã được học tại môn nhập môn Trí tuệ nhân tạo, chúng em nhận thấy việc tìm đường đi trên bản đồ (hoặc trên đồ thị sau khi đã được mô hình hoá trên máy tính) **là một tác tử hoàn chỉnh** với đủ 4 yếu tố PEAS :

* Performance measure: Hiệu suất của việc tìm đường đi trên bản đồ là việc ứng dụng thuật toán tìm đường đi ngắn nhất để tìm đường đi ngắn nhất giữa 2 đỉnh. Mục tiêu là tối ưu hoá về thời gian bằng cách tìm quảng đường di chuyển ngắn nhất giữa điểm bắt đầu và điểm xuất phát.
* Environment: Môi trường trong trường hợp này là bản đồ hoặc đồ thị chứa các nút hoặc cạnh. Các nút đại diện cho các vị trí hoặc các ngã rẽ. Các cạnh là đường đi nối các ngã rẽ hoặc nối các vị trí với nhau (thực tế là các phố, đường).
* Actuators: Cơ chế thực thi là việc di chuyển qua các đỉnh và cạnh của đồ thị (ngã rẽ, đường, phố). Nó sẽ đi qua các đỉnh (các ngã rẽ) để tìm đường đi ngắn nhất từ điểm bắt đầu đến điểm kết thúc.
* Sensors: Cảm biến trong trường hợp này là việc thu thập thông tin về các cạnh và đỉnh của đồ thị. Nó giúp xác định khoảng cách từ điểm bắt đầu đến các đỉnh và cạnh liền kề, cung cấp thông tin cho thuật toán tìm đường đi ngắn nhất để xác định đường đi tốt nhất.

Sau khi xác định và hiểu được bản chất bài toán, chúng em quyết định sử dụng thuật toán tìm đường đi ngắn nhất trên đồ thị Dijkstra để giải quyết bài toán tìm đường này.

BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT

Thuật toán Dijkstra được sử dụng để tìm đường đi ngắn nhất từ một điểm bắt đầu đến tất cả các đỉnh còn lại trong một đồ thị có trọng số không âm. Dưới đây là mô tả ngắn gọn của thuật toán Dijkstra:

1. Khởi tạo:
   * Đặt điểm bắt đầu là "s".
   * Đặt khoảng cách từ điểm bắt đầu đến tất cả các đỉnh khác là vô cùng (infinitive).
   * Đặt khoảng cách từ điểm bắt đầu đến chính nó là 0.
2. Lặp cho đến khi tất cả các đỉnh đã được khám phá:
   * Chọn đỉnh u chưa được khám phá có khoảng cách nhỏ nhất.
   * Đánh dấu đỉnh u là đã được khám phá.
3. Duyệt qua tất cả các đỉnh v liền kề với đỉnh u:
   * Nếu khoảng cách từ điểm bắt đầu đến v lớn hơn khoảng cách từ điểm bắt đầu đến u cộng với trọng số của cạnh (u, v), thì cập nhật khoảng cách từ điểm bắt đầu đến v thành khoảng cách mới này.
4. Lặp lại bước 2 và 3 cho đến khi tất cả các đỉnh trong đồ thị đã được khám phá.

Sau khi hoàn thành các bước trên, ta sẽ có khoảng cách ngắn nhất từ điểm bắt đầu đến tất cả các đỉnh trong đồ thị.

Mô hình toán học:

* Đi từ (x­­1­, y­­1) → (x­­2, y­­2)
* Trạng thái ban đầu: (x­­1­, y­­1)
* Trạng thái đích: (x­­2, y­­2)
* Không gian bài toán:

∀ (x,y) ∈ V

A = {(x’,y’)|(x’,y’)∈V,(x,y)(x’,y’)∈E}

* Toán tử chuyển trạng thái:

(x,y) → (x’,y’) nếu d[(x,y)(x’,y’)] min

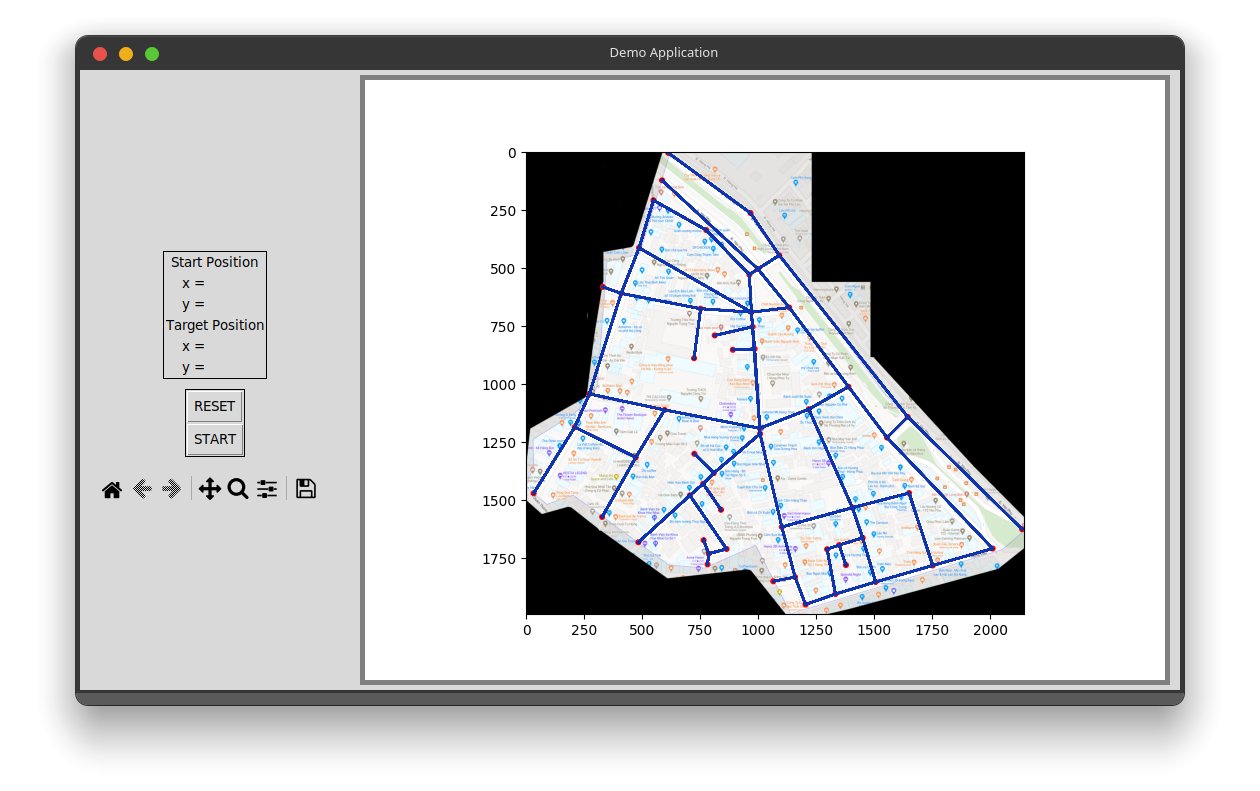
CÀI ĐẶT CHƯƠNG TRÌNH

Chúng em đã cài đặt chương trình. Dưới đây là link github dẫn đến source Code chương trình của chúng em:

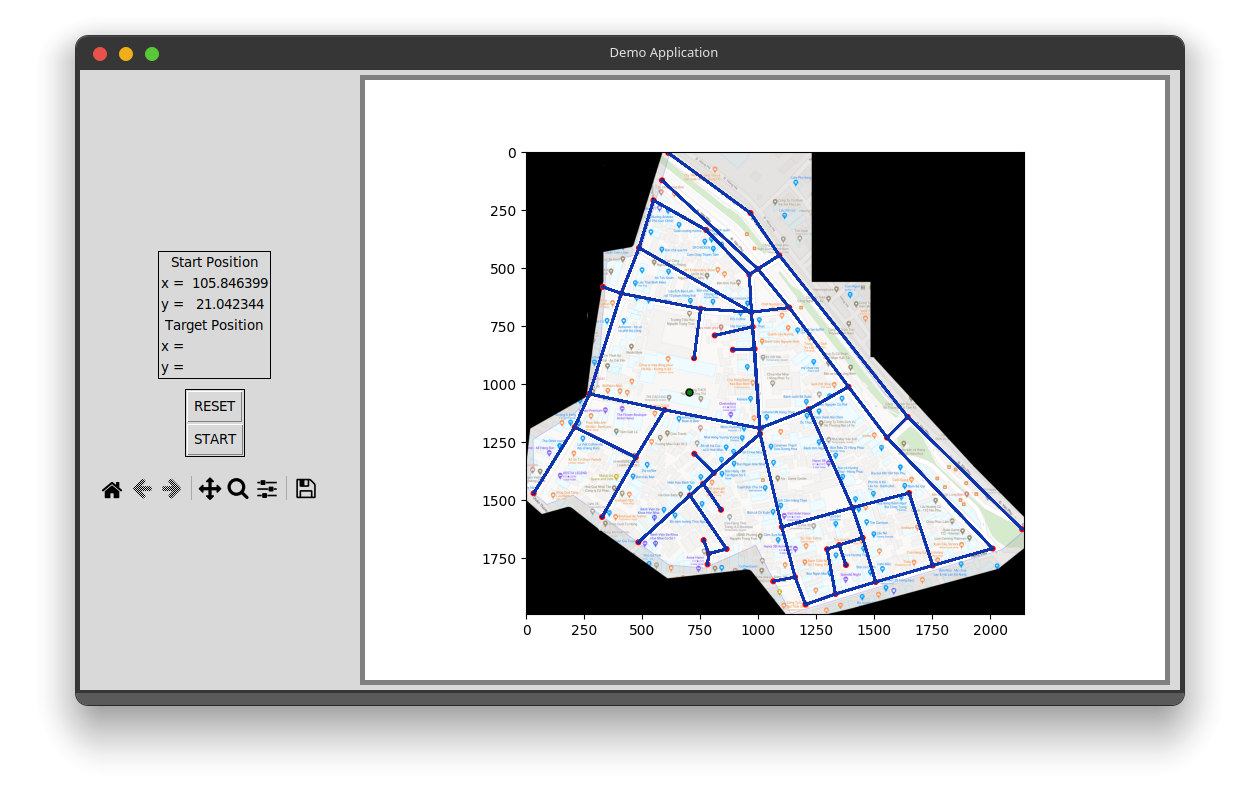
[https://github.com/Minh-19102/HUST\_AI\_Intro.git](https://github.com/Minh-19102/HUST_AI_Intro.git?fbclid=IwAR0oJbwuGKuHOBEjgbhyzIuz8wt7FgN4UIK_r5gmp1lvQqJxXDKgvZagiRc)

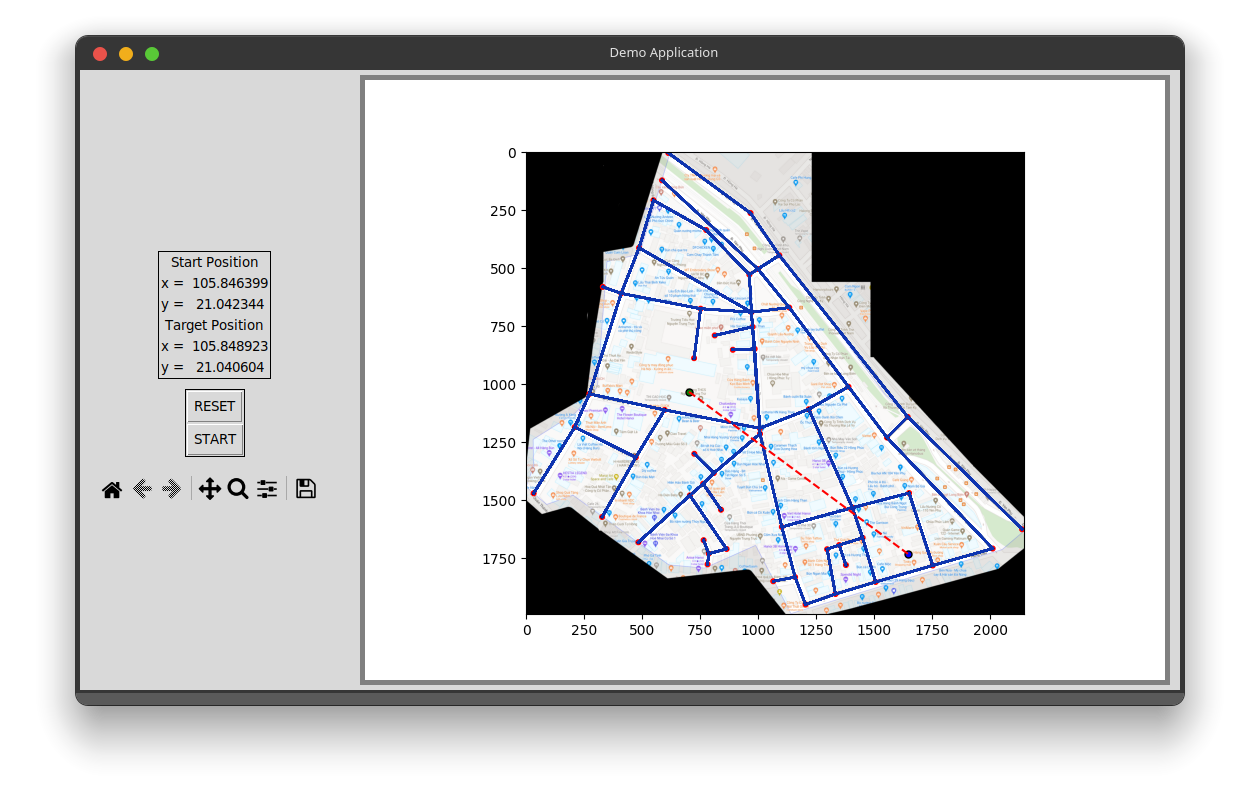
KẾT QUẢ CHẠY

* Chương trình sẽ hiển thị lên màn hình bản đồ và các tuyến đường di chuyển.



* Người đùng chọn vị trí trên bản đồ bằng cách di chuột và bấm chuột để chọn vị trí bắt đầu và kết thúc:





* Sau đó chương trình sẽ bắt đầu tìm đường đi bằng thuật toán dijkstra: