BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



BỘ MÔN KHOA HỌC MÁY TÍNH

**Học phần TRÍ TUỆ NHÂN TẠO**

**Đề tài**

**ỨNG DỤNG TRÍ TUỆ NHÂN TẠO**

**TÌM ĐƯỜNG ĐI TRONG MÊ CUNG**

**GVHD:**

Ths.Võ Tri Thức

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | | **MSSV** | | **HỌ VÀ TÊN** | |
| 1 | | B1709549 | | Tăng Nghiệp Minh | |
| 2 | | B1709569 | | Lê Văn Chí Thiện | |
| 3 | | B1709583 | | Lê Quốc Vương | |
| 4 | | B1710368 | | Phan Ngọc Thảo | |

**Học kì: 02, Năm học (2019-2020)**

**NHẬN XÉT CỦA GIẢNG VIÊN**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**LỜI CẢM ƠN**

Để có được bài niên luận này, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến THẦY VÕ TRI THỨC – người đã trực tiếp tận tình hướng dẫn, giúp đỡ em.Trong suốt quá trình thực hiện niên luận, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài niên luận này được hoàn thành một cách tốt nhất.

Em cũng xin gửi lời cám ơn chân thành đến các Thầy Cô Giảng viên Đại học Cần Thơ, đặc biệt là các Thầy Cô ở Khoa CNTT & TT, những người đã truyền đạt những kiếnthức quý báu trong thời gian qua.

Em cũng xin chân thành cảm ơn bạn bè cùng với gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để em có thể hoàn thành bài niên luận một cách tốt nhất.

Tuy có nhiều cố gắng trong quá trình thực hiện niên luận, nhưng không thể tránh khỏi những sai sót. Em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy Cô và các bạn để bài niên luận hoàn thiện hơn.

Cần Thơ, ngày 07 tháng 07 năm 2020

Người viết

Lê Quốc Vương

Phan Ngọc Thảo

Lê Văn Chí Thiện

Tăng Nghiệp Minh

MỤC LỤC

[MỤC LỤC iv](#_Toc44949761)

[DANH MỤC ẢNH v](#_Toc44949762)

[Chương I. TỔNG QUAN 1](#_Toc44949763)

[I. Đặt vấn đề 1](#_Toc44949764)

[II. Lịch sử giải quyết vấn đề 1](#_Toc44949765)

[III. Đối tượng và phạm vi nghiên cứu 2](#_Toc44949766)

[IV. Phương pháp nghiên cứu 2](#_Toc44949767)

[V. Kết quả đạt được 2](#_Toc44949768)

[VI. Bố cục báo cáo 2](#_Toc44949769)

[Chương II. PHẦN MÔ TẢ BÀI TOÁN 3](#_Toc44949770)

[I. Mô tả chi tiết bài toán 3](#_Toc44949771)

[II. Vấn đề liên quan và giải pháp 5](#_Toc44949772)

[Chương III. THIẾT KẾ VÀ CÀI ĐẶT 9](#_Toc44949773)

[I. Thiết kế hệ thống 9](#_Toc44949774)

[II. Thiết kế và cài đặt giải thuật 9](#_Toc44949775)

[Chương IV. KẾT LUẬN 15](#_Toc44949776)

[I. Kết quả đạt được 15](#_Toc44949777)

[II. Hướng phát triển 15](#_Toc44949778)

[TÀI LIỆU THAM KHẢO 16](#_Toc44949779)

DANH MỤC ẢNH

[Hình 1. Game đưa xe dung giải thuật A\* 1](file:////Users/chithien/Documents/TTNT/baocaotrituenhantao.docx#_Toc44950103)

[Hình 2. Tìm đường đi robot 2](#_Toc44950104)

[Hình 3. Tính heuristic 5](#_Toc44950105)

Chương I. TỔNG QUAN

* + 1. Đặt vấn đề

Ngày nay, việc tích hợp Trí tuệ nhân tạo vào các thiết bị điện tử cũng như các ứng dụng thực tế ngày càng nhiều. Nghiên cứu Trí tuệ nhân tạo ngày càng chứng tỏ thế mạnh trong mọi công việc đòi hỏi khả năng và trí tuệ con người.

Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc. Trong đó, áp dụng Trí tuệ nhân tạo vào các trò chơi tìm đường đi được áp dụng rộng rãi.

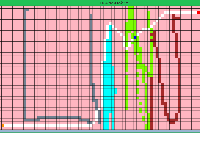
Bản đồ án này được xây dựng dựa trên những kiến thức được học từ môn Trí tuệ nhân tạo để tìm đường đi trong mê cung. Từ đó xây dựng chương trình đưa ra những giải thuật để tìm lời giải cho bài toán.

* + 1. Lịch sử giải quyết vấn đề

Trong thực tế đã có nhiều trò chơi tìm đường áp dụng Trí tuệ nhân tạo vào.

Game đua xe dựa vào giải thuật A\*.

Hình 1. Game đưa xe dung giải thuật A\*



Hình . Tìm đường đi robot

* + 1. Đối tượng và phạm vi nghiên cứu
* Đối tượng nghiên cứu: Đường đi trong mê cung
* Phạm vi nghiên cứu: Xây dựng giải thuật tìm đường đi trong mê cung, cho ra tốc độ, so sánh trực quan giữa hai thuật toán BFS (Best first search), A Star. Xây dụng thuật toán sinh ma trận ngẫu nhiên, đa dạng dữ liệu đầu vaò
  + 1. Phương pháp nghiên cứu

Từ các ma trận thực hiện mô hình hoá bài toán. Kết hợp kiến thực đã học trong môn Trí tuệ nhân tạo áp dụng thuật toán đưa ra quyết định đường đi trong mê cung.

* + 1. Kết quả đạt được
* Tìm đường đi trong mê cung với giải thuật A\*, BFS, số bước di chuyển, số nút đã duyệt, thời gian tính toán
* Đọc dữ liệu ma trận từ file
* Tạo ma trận ngẫu nhiên
* Giao diện đồ hoạ mô hình hoá cách tìm đường đi của thuật toán
  + 1. Bố cục báo cáo

Tài liệu này cung cấp thông tin về mô hình hoá bài toán, xây dụng giải thuật, cài đặt giải thuật, các thành phần giao diện. Cấu trúc tài liệu này gồm có 3 chương.

* **Chương 1. Tổng quan**, giới thiệu tổng quát về đề tài
* **Chương 2. Mô tả bài toán**, chi tiết bài toán và những vấn đề liên quan
* **Chương 3. Thiết kế và cài đặt**, xây dựng giải thuật tìm kiếm không gian trạng thái, sinh mê cung, giao diện đồ hoạ
* **Chương 4. Kết luận**, trình bày kết quả đạt được cũng như hạn chế và hướng phát triển

Chương II. PHẦN MÔ TẢ BÀI TOÁN

* + 1. Mô tả chi tiết bài toán

Chúng ta bắt đầu từ bước chuẩn bị dữ liệu, chuyển mô hình ma trận sang dạng ma trận m x n. Xây dựng mô hình với mong muốn cuối là tìm được đường đi trong mê trong từ một điểm đầu bất kì tới điểm mục tiêu. Lựa chọn thuật toán phù hợp để áp dụng vào bài toán. Dữ liệu có được từ các trang trên internet, với nhiều ma trận khác nhau và kết quả để có thể áp dụng để kiểm tra tính đúng đắn giải thuật sau khi thực hiện. Mỗi dữ liệu sẽ chứa số dòng, số cột, vị trí ô bắt đầu tìm, vị trí ô mục tiêu, ma trận gồm hai số 0 và 1. Một là tường, không thể di chuyển sang được, 0 là ô trống nơi đây có thể di chuyển. Như vậy, chúng ta đã có được dữ liệu và thông tin cần thiết về bài toán, tiến hành giải quyết trong các bước tiếp theo:

**- Đọc dữ liệu từ file:**

+ Dòng đầu tiên: giá trị thứ nhất tương ứng với số lượng hàng (m), giá trị thứ hai tương ứng với số lượng cột (n), giá trị thứ ba tương ứng toạ độ x của điểm bắt đầu, giá trị thứ tư toạ độ y của điểm bắt đầu, giá trị thứ 5 toạ độ x của điểm mục tiêu, giá trị thứ 6 toạ độ x của điểm. Các tiếp theo sẽ còn lại m dòng mỗi dòng có n phần tử.

+ Đưa dữ liệu biểu diễn dưới dạng mạng hai chiều m hàng, n cột.

**- Xây dựng giải thuật tìm đường đi**

+ Heuristic: hàm giúp đưa ra quyết định ô nào sẽ được chọn tiếp theo

+ Các hành động:

Up: di chuyển sang ô phía trên

Down: di chuyển sang ô phía dưới

Left: di chuyển sang ô bên trái

Right: di chuyển sang ô bên phải

+ Giải thuật BFS: có hàm f = h, dựa vào f để chọn bước đi tiếp theo.

+ Giải thuật A\*: có hàm f = g + h. g là số bước đã đi.

- Xây dựng giải thuật sinh mê cung

+ Sinh mê cung từ một số lượng hàng và cột bất kì

+ Sử dụng giải thuật quay lui

**- Tạo ứng dụng:**

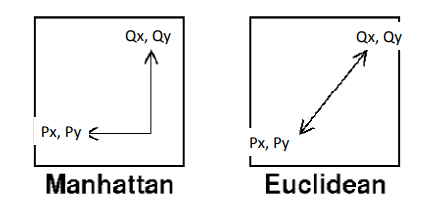
+ Đưa dữ liệu sau khi thực lên bài toán

+ Cho phép lựa chọn giải thuật thực hiện.

1. Vấn đề liên quan và giải pháp
   * + 1. Vấn đề 1: Đọc file

Sử dụng XMLHttpRequest với phương thức get để đọc file. Tách các phần tử trong file thành các phần tử dưới dạng mảng. Lấy 2 giá trị đầu tiên tương ứng số hàng và số cột, giá trị thứ 3 và 4 dùng khởi tạo điểm bắt đầu. Giá trị 5, 6 dùng khởi tạo nút mục tiêu. Duyệt các giá trị còn lại đưa vào ma trận m x n.

2. Vấn đề 2: Hàm tính heuristic

1. start (Px, Py) điểm bắt đầu
2. goal(Qx, Qy) điểm mục tiêu.
3. h = | Qy – Py | + | Qx - Px | (Phương pháp Mahanttan),
4. h =(| Qy – Py |2 + | Qx- Px |2 )^(1/2) (Phương pháp Euclidean)
5. 

Hình . Tính heuristic

1. Vấn đề 3: Các hành động trong bài toán

* Up: Di chuyển ô trống lên trên, hành động thành công khi và chỉ khi sau khi di chuyển lên trên mà ô đó không nằm ngoài biên (toạ độ x >= 0). Và ô đó không phải là tường.
* Down: Di chuyển ô trống xuống dưới, hành động thành công khi và chỉ khi sau khi di chuyển lên trên mà ô đó không nằm ngoài biên (toạ độ x < số dòng). Và ô đó không phải là tường.
* Left: Di chuyển ô trống sang trái, hành động công khi và chỉ khi sau khi di chuyển lên trên mà ô đó không nằm ngoài bien (toạ độ y >= 0). Và ô đó không phải là tường.
* Right: Di chuyển ô trống sang phải, hành động công khi và chỉ khi sau khi di chuyển lên trên mà ô đó không nằm ngoài biên (toạ độ y < số cột). Và ô đó không phải là tường.

1. Vấn đề 4: Giải thuật tìm kiếm
2. Best first search (BFS)

Giải thuật BFS gồm một openList chứa các ô đã duyệt đến nhưng ô chờ duyệt, closeList chứa những ô đã duyệt qua. Mỗi ô duyệt sẽ thực hiện các hành động di chuyển lên, di chuyển xuống, di chuyển sang trái, di chuyển sang phải để tìm ra các lận cận, sau đó tính giá trị f (f = h) dựa vào heuristic. Bước chọn hướng đi tiếp theo sẽ dựa vào ô có f nhỏ nhất. Sau khi lấy ô đó ra kiểm tra xem nếu là mục tiêu thì dừng. Nếu không là mục tiêu thì tìm các lận cận dựa vào hành động up, down, left, right. Điểm lận cận nếu không tồn tại trong openList và closeList thì đưa nó vào openList, nếu lận cận đã tồn tại trong openList thì so sánh giá trị f cũ và mới, nếu f mới nhỏ hơn thì tiến hành cập nhật lại giá trị. Ngược lại nó nằm trong closeList chúng ta so sánh giá trị f cũ và mới, nếu f mới nhỏ hơn thì xoá bỏ nút ra khỏi closeList và đưa nút mới vào openList.

1. A star (A\*)

Tương tự với BFS, nhưng có thêm giá trị g là khoảng cách từ bước cha tới ô hiện tại, hàm f = g + h, quá trình cập nhật lại nút vào openList dựa vào giá trị g thay vì f.

1. Vấn đề 5. Sinh mê cung ngẫu nhiên

Sinh mê cung ngẫu nhiên áp dụng phương pháp quay lui (backtracking) để sinh ma trận:

- Khởi tạo ma trận m x n có giá trị bằng 1

- Tạo danh sách rỗng - **list**, chọn ngẫu nhiên ô đầu tiên với toạ độ (x, y) ngẫu nhiên. Đưa ô đó vào **list**.

- Lặp cho đến khi **list** rỗng, mỗi bước lặp sẽ thuật hiện xáo trộn thứ tự ưu tiên của các hành động up, down, left, right (tăng tính ngẫu nhiên của mê cung). - - Lặp qua từng hành đồng nếu ô đó tìm ra được lân cận thì đưa lận cận đó vào **list** và dừng các hành động còn lại với ô hiện tại, cập nhật vị trí ô hiện trong ma trận là 0, và ô mới tìm thấy là 0. Sau khi không còn ô nào trong danh sách. Thực hiện trả về ma trận.

1. Vấn đề 6. Tạo giao diện cho bài toán

**Giao diện gồm:**

- Start game: cho phép bắt đầu thực hiện bài toán và giải tìm đường đi (mặc định sẽ là BFS với ma trận ngẫu nhiên 70x70)

- Run: chạy thuật toán, hoặc chạy lại thuật toán vừa chọn.

- Generate maze: Sinh ma trận ngẫu nhiên

- Run again: chạy lại thuật toán với điểm bắt đầu và mục tiêu ngẫu nhiên.

- Select:

+ Bfs: Chọn giải thuật BFS để giải bài toán

+ A star: Chọn giải thuật A Star để giải bài toán

- Choose maze: chọn các mê cung có sẵn (mê cung đọc từ file)

- Các ô input:

Rows: chọn số hàng để sinh mê cung

Colums: chọn số cột để sinh mê cung

- Các ô output:

Open list: Số lượng các nút (node) đã tìm nhưng chưa duyệt

Close list: Số lượng các nút (node) đã được duyệt qua

Path length: Số ô để đến được ô mục tiêu (ngoại trừ ô bắt đầu và mục tiêu)

Solve time: Thời gian thực hiện lời giải, tính bằng mili giây

- Khung ma trận: kích thước ngẫu nhiên tuỳ vào bài toán.

+ Ô màu xám  là ô trống có thể di chuyển được

+ Ô màu đen  là tường, không thể di chuyển

+ Ô màu đỏ  là điểm bắt đầu

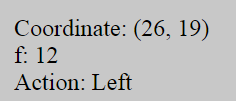
+ Ô màu xanh lá  điểm đích

+ Ô màu cam đậm  các ô đang ở trong closeList

+ Ô màu cam nhạt  các ô đang ở trong openList

+ Ô màu xanh dương  đường đến mục tiêu

+ Khi chọn vào các ô đường đi sẽ hiện thị ra các số liệu: Coordinate toạ độ x ,y, giá trị f, hành động đã thực hiện để được ô hiện tại.



Chương III. THIẾT KẾ VÀ CÀI ĐẶT

* + 1. Thiết kế hệ thống

Hệ thống được thiết kế và cài đặt bằng JavaScript nên việc thiết kế chương trình và kiểm thử không quá phức tạp. Thiết lập chương trình có thể nhận input từ file, hay tự tạo input, sau đó thể hiện kết quả tìm kiếm lên web,

* + 1. Thiết kế và cài đặt giải thuật
       1. Cài đặt trạng thái

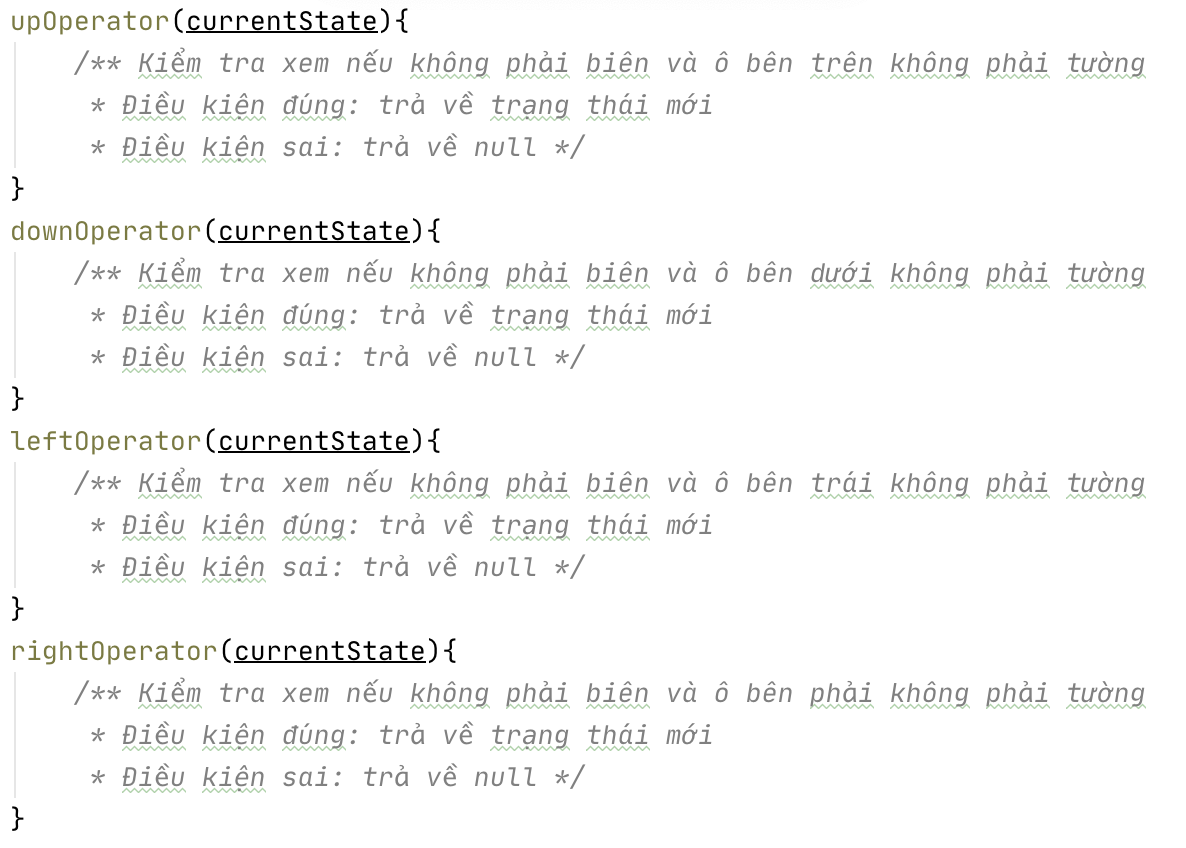
Mỗi trạng thái sẽ bao gồm vị trí hàng hiện tại, vị trí cột hiện tại

A picture containing bird

Description automatically generated

* + - 1. Cài đặt nút (node)A screenshot of a social media post

         Description automatically generated
      2. Cài đặt các hành động

****

* + - 1. Biểu diễn đồ thị

A screenshot of a cell phone

Description automatically generated

* + - 1. Giải thuật Bfs

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

* + - 1. Giải thuật A\*

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

* + - 1. Giải thuật sinh mê cung

A screenshot of a cell phone

Description automatically generated

* + - 1. Vẽ giao diện

1. Hàm vẽ một nútA screenshot of a cell phone

   Description automatically generated
2. Hàm lấy danh sách node đường đi

A picture containing knife

Description automatically generated

1. Hàm vẽ các node đường đi

A screenshot of a cell phone

Description automatically generated

Chương IV. KẾT LUẬN

* + 1. Kết quả đạt được
       1. Ưu điểm:

- Trong quá trình thực hiện biết thêm cách vẽ giao diện với canvas, xây dựng giải thuật backtracking, mô hình trực quan bài toán mê cung, dễ dàng so sánh hiệu quả của từng giải thuật

- Chương trình cài đặt với kiến thức giải thuật đã được học trong môn Trí tuệ nhân tạo

- Sinh ra ma trận ngẫu nhiên, đa dạng trường hợp khác nhau để kiểm thử

* + - 1. Khuyết điểm:
* Giao diện chưa hoàn chỉnh
* Giải thuật tìm kiếm còn ít
  + 1. Hướng phát triển
* Xây dựng giải thuật tìm kiếm với Dijktra, Backtracking,…
* Cải thiện đồ hoạ
* Chủ động lựa chọn ô bắt đầu, ô mục tiêu
* Lựa chọn các ô muốn là vị trí tường

TÀI LIỆU THAM KHẢO

1. Tài liệu Trí tuệ nhân tạo, khoa Công nghệ thông tin, bộ môn Khoa học máy tính, 06/2019
2. [Pathfinding in Strategy Games and Maze Solving Using A\* Search Algorithm](https://www.scirp.org/html/2-1730422_70460.htm), 2020
3. [Maze generation algorithm, wikipedia](https://en.wikipedia.org/wiki/Maze_generation_algorithm), 2020
4. [Pathfinding and mazes](http://web.cse.ohio-state.edu/~wang.3602/courses/cse3541-2017-fall/13-Pathfinding.pdf), 2020