

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

TRÒ CHƠI ĐUA XE

Sinh viên thực hiện : Nguyễn Tiến Lực
Vũ Duy Mạnh
Nguyễn Trung Sơn
Nguyễn Vọng

Ngành : Công nghệ thông tin

Giảng viên hướng dẫn : ThS. Lê Thị Thùy Trang
Ths. Trần Thái Khánh

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến ThS. **Lê Thị Thùy Trang** và ThS. **Nguyễn Thái Khánh**, những người đã tận tình hướng dẫn, hỗ trợ em trong quá trình thực hiện đề tài "*Viết chương trình trò chơi đua xe sử dụng thuật toán A* để tìm đường tối ưu*" trong khuôn khổ môn **Trí tuệ nhân tạo**.

Trong suốt thời gian nghiên cứu và phát triển đề tài, em đã nhận được sự chỉ dẫn tận tình từ hai thầy cô, không chỉ về mặt kiến thức chuyên môn mà còn về phương pháp tư duy, cách tiếp cận vấn đề và kỹ năng giải quyết các tình huống thực tế. Những bài giảng và tài liệu quý báu của thầy cô đã giúp em hiểu rõ hơn về thuật toán A*, cách ứng dụng nó trong các bài toán tìm đường cũng như cách xây dựng một trò chơi đua xe có tính thử thách và hấp dẫn.

Em đặc biệt biết ơn thầy cô vì sự kiên nhẫn trong việc giải đáp những thắc mắc, gợi ý những hướng đi đúng đắn để em có thể hoàn thiện đồ án một cách tốt nhất. Nhờ sự hướng dẫn tận tình ấy, em không chỉ hoàn thành đề tài mà còn học hỏi được rất nhiều kiến thức bổ ích, giúp em có cái nhìn sâu sắc hơn về lĩnh vực **Trí tuệ nhân tạo** và lập trình game.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến nhà trường, bạn bè và gia đình đã luôn động viên, tạo điều kiện thuận lợi để em có thể tập trung vào nghiên cứu và phát triển dự án này.

Cuối cùng, em xin kính chúc ThS. **Lê Thị Thùy Trang** và ThS. **Nguyễn Thái Khánh** dồi dào sức khỏe, luôn thành công trong sự nghiệp giảng dạy và nghiên cứu.

Em xin chân thành cảm ơn!

Tóm tắt

Nghiên cứu này trình bày quá trình phát triển một trò chơi đua xe, trong đó người chơi điều khiển một chiếc xe đua để di chuyển từ điểm xuất phát đến đích trong thời gian nhanh nhất có thể. Mục tiêu của trò chơi là giúp người chơi tìm ra con đường ngắn nhất để về đích trong khi phải tránh các chướng ngại vật trên đường đua. Để giải quyết bài toán tìm đường tối ưu, nghiên cứu áp dụng thuật toán A^* (A-star), một trong những thuật toán tìm đường hiệu quả nhất hiện nay, giúp xác định lộ trình tối ưu dựa trên chi phí di chuyển ước tính.

Quá trình phát triển trò chơi bao gồm nhiều giai đoạn quan trọng. Đầu tiên, mô hình đường đua được thiết kế với các chướng ngại vật và các ô lưới (grid) đại diện cho không gian di chuyển của xe. Các trạng thái của xe đua, bao gồm vị trí, vận tốc và hướng di chuyển, được cập nhật liên tục dựa trên hành động của người chơi. Thuật toán A^* được sử dụng để tìm kiếm đường đi tối ưu, kết hợp giữa chi phí thực tế từ điểm xuất phát đến vị trí hiện tại và chi phí ước tính từ vị trí đó đến đích.

Trong quá trình triển khai, thuật toán A^* được thử nghiệm trên nhiều dạng bản đồ khác nhau nhằm đánh giá hiệu suất và độ chính xác của thuật toán trong môi trường trò chơi. Kết quả cho thấy thuật toán này có khả năng tìm ra lộ trình tối ưu nhanh chóng, ngay cả khi có nhiều chướng ngại vật phức tạp trên đường đua. Để tăng tính tương tác, trò chơi cũng được tích hợp các yếu tố như điểm số, thời gian hoàn thành và các cấp độ khó khác nhau, giúp người chơi có trải nghiệm thú vị hơn.

Bên cạnh những kết quả đạt được, nghiên cứu cũng chỉ ra một số thách thức và hạn chế của mô hình. Đầu tiên, thuật toán A^* có thể gặp khó khăn khi số lượng chướng ngại vật quá lớn hoặc bản đồ có kích thước quá rộng, dẫn đến thời gian tính toán tăng đáng kể. Ngoài ra, việc cân bằng giữa độ chính xác và tốc độ xử lý là một vấn đề quan trọng cần được tối ưu hóa để đảm bảo trải nghiệm mượt mà cho người chơi.

Tóm lại, nghiên cứu đã thành công trong việc ứng dụng thuật toán A^* để phát triển trò chơi đua xe, giúp người chơi tìm đường nhanh nhất trong môi trường có chướng ngại vật. Kết quả đạt được không chỉ đóng góp vào lĩnh vực **Trí tuệ nhân tạo** mà còn mở ra tiềm năng ứng dụng trong các trò chơi và hệ thống điều hướng thông minh khác.

Mục lục

1	Giới thiệu	1
1.1	Đặt vấn đề	1
1.2	Lý do chọn đề tài	1
1.3	Mục tiêu của đề tài	2
2	Tổng quan	3
2.1	Giới thiệu về thuật toán A*	3
2.2	Ứng dụng A* trong tìm đường đi tối ưu	5
2.3	Cấu trúc bản đồ trò chơi	6
2.4	Xây dựng thuật toán điều khiển xe đua	7
3	Phân tích thiết kế chương trình	9
3.1	Mô tả trò chơi A Star Racing	9
3.2	Phân tích các đối tượng trong trò chơi	10
3.2.1	Phân tích các đối tượng trong trò chơi	10
3.2.2	Đánh giá kết quả	10
3.2.3	Hạn chế và hướng phát triển	11
3.3	Cài đặt thuật toán A*	11
3.3.1	Phân tích các đối tượng trong trò chơi	11
3.3.2	Giao diện trò chơi	11
3.3.3	Phân tích các đối tượng trong trò chơi	11
3.4	Lập kế hoạch thuật toán*	12
3.4.1	Xác định đầu vào và đầu ra	12
3.4.2	Các bước thực hiện	12
3.5	Thiết kế giao diện	12
3.5.1	Màn hình chính	12
3.5.2	Các thành phần đồ họa	13
3.5.3	Các nút điều khiển	13
3.5.4	Hiển thị trạng thái	13
3.5.5	Hướng phát triển giao diện	13
3.6	Đánh giá kết quả	13
3.6.1	Hiệu suất thuật toán	14
3.6.2	Ưu điểm của thuật toán A*	14

3.6.3	Nhược điểm của thuật toán	14
3.6.4	Đánh giá giao diện trò chơi	14
3.6.5	Hướng phát triển trong tương lai	15
3.6.6	Kết luận	15
3.7	Hạn chế và hướng phát triển	15
3.7.1	Hạn chế của trò chơi	15
3.7.2	Hướng phát triển trong tương lai	16
3.7.3	Tổng kết	16
3.8	Cách hoạt động của code	17
3.8.1	Tổng quan về chương trình	17
3.8.2	Khởi tạo chương trình	17
3.8.3	Tạo bản đồ trò chơi	18
3.8.4	Thuật toán A*	18
3.8.5	Xử lý giao diện và hiển thị trò chơi	18
3.8.6	Kiểm tra điều kiện thắng cuộc	19
3.8.7	Thoát chương trình	19
3.8.8	Tóm tắt cách hoạt động của code	19
3.8.9	kết quả	19
Kết Luận		21
	Kết Luận	21
Tài Liệu Tham Khảo		23
	Tài Liệu Tham Khảo	23

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Đua xe là một trong những môn thể thao tốc độ thu hút sự quan tâm của nhiều người trên toàn thế giới. Không chỉ là một bộ môn giải trí, đua xe còn là nền tảng để phát triển công nghệ ô tô hiện đại, từ thiết kế khí động học, động cơ hiệu suất cao đến các hệ thống an toàn tiên tiến. Đua xe không chỉ là cuộc đua về tốc độ mà còn là cuộc đua về công nghệ, nơi các kỹ sư và nhà nghiên cứu liên tục tìm kiếm những giải pháp mới để cải thiện hiệu suất và độ an toàn của xe.

Mô hình xe đua đóng vai trò quan trọng trong việc nghiên cứu và thử nghiệm các nguyên lý vận hành thực tế của xe đua. Việc chế tạo mô hình giúp tái hiện quá trình thiết kế, thử nghiệm và cải tiến của xe đua thực tế, tạo điều kiện cho người yêu thích công nghệ và tốc độ có thể nghiên cứu chuyên sâu hơn về lĩnh vực này. Mô hình xe đua không chỉ là công cụ để học tập và nghiên cứu mà còn là nền tảng để phát triển các công nghệ mới, từ đó ứng dụng vào thực tế sản xuất xe đua và ô tô thương mại.

1.2 Lý do chọn đề tài

- **Đua xe là ngành công nghiệp phát triển mạnh mẽ:** Đua xe không chỉ là một môn thể thao mà còn là một ngành công nghiệp phát triển mạnh mẽ, nơi các công nghệ tiên tiến liên tục được cải tiến và ứng dụng vào thực tế. Các giải đua xe như Formula 1, NASCAR, và Le Mans là những ví dụ điển hình về sự kết hợp giữa thể thao và công nghệ cao.
- **Nghiên cứu các yếu tố kỹ thuật quan trọng:** Mô hình xe đua giúp nghiên cứu các yếu tố quan trọng như khí động học, động cơ, hệ thống truyền động, và vật liệu trong xe. Những yếu tố này không chỉ ảnh hưởng đến tốc độ mà còn đến độ ổn định và an toàn của xe.
- **Tính ứng dụng cao:** Việc chế tạo mô hình xe đua mang tính ứng dụng cao, có thể sử dụng trong giáo dục, nghiên cứu kỹ thuật cũng như làm nền tảng để phát triển các sản phẩm thực tế. Mô hình xe đua có thể được sử dụng trong các phòng thí nghiệm,

trường đại học, và các trung tâm nghiên cứu để giảng dạy và thử nghiệm các công nghệ mới.

- **Cơ hội cho người đam mê tốc độ:** Giúp người đam mê tốc độ có cơ hội tìm hiểu, thực hành và phát triển ý tưởng sáng tạo của riêng mình. Mô hình xe đua là công cụ tuyệt vời để những người yêu thích công nghệ và tốc độ có thể thử nghiệm và cải tiến các thiết kế của mình.
- **Phát triển kỹ năng kỹ thuật:** Việc nghiên cứu và chế tạo mô hình xe đua giúp phát triển các kỹ năng kỹ thuật như thiết kế cơ khí, lập trình, và phân tích dữ liệu. Đây là những kỹ năng quan trọng trong ngành công nghiệp ô tô và các lĩnh vực kỹ thuật khác.

1.3 Mục tiêu của đề tài

Nghiên cứu cấu tạo và nguyên lý hoạt động của mô hình xe đua: Tìm hiểu sâu về cấu trúc cơ bản của xe đua, bao gồm khung xe, hệ thống truyền động, động cơ, và các hệ thống điều khiển. Nghiên cứu nguyên lý hoạt động của các hệ thống này để hiểu rõ cách chúng tương tác với nhau để tạo ra hiệu suất tối ưu. Phân tích ảnh hưởng của thiết kế khí động học đến tốc độ và hiệu suất xe: Khí động học là yếu tố quan trọng quyết định đến tốc độ và độ ổn định của xe. Nghiên cứu các yếu tố khí động học như lực cản, lực nâng, và độ ổn định để tối ưu hóa thiết kế của mô hình xe đua. Thử nghiệm các vật liệu và công nghệ mới trong chế tạo mô hình xe đua: Vật liệu và công nghệ mới có thể cải thiện đáng kể hiệu suất của xe. Nghiên cứu và thử nghiệm các vật liệu nhẹ, bền, và các công nghệ tiên tiến như in 3D, composite, và vật liệu nano để áp dụng vào mô hình xe đua. Ứng dụng mô hình xe đua trong nghiên cứu, giáo dục và phát triển công nghệ ô tô: Mô hình xe đua có thể được sử dụng như một công cụ giảng dạy trong các trường đại học và trung tâm nghiên cứu. Nó cũng có thể là nền tảng để phát triển các công nghệ mới trong ngành công nghiệp ô tô.

Chương 2

Tổng quan

2.1 Giới thiệu về thuật toán A*

Thuật toán A* là một thuật toán tìm kiếm trên đồ thị được sử dụng rộng rãi trong lĩnh vực trí tuệ nhân tạo, hệ thống điều hướng và lập kế hoạch di chuyển. Nó được Peter Hart, Nils Nilsson và Bertram Raphael giới thiệu vào năm 1968 như một phần của nghiên cứu về trí tuệ nhân tạo. Thuật toán A* kết hợp giữa thuật toán Dijkstra (tìm đường đi tối ưu) và thuật toán tìm kiếm tham lam (Greedy Search), giúp cân bằng giữa tốc độ tìm kiếm và độ chính xác.

Nguyên lý hoạt động

A* hoạt động bằng cách đánh giá mỗi nút theo công thức:

$$f(n) = g(n) + h(n)$$

Trong đó:

- $f(n)$ là tổng chi phí ước tính từ điểm xuất phát đến đích qua nút n .
- $g(n)$ là chi phí thực tế từ điểm xuất phát đến nút n .
- $h(n)$ là hàm heuristic, ước lượng chi phí từ n đến đích.

Các hàm heuristic phổ biến

Khoảng cách Manhattan:

$$h(n) = |x_1 - x_2| + |y_1 - y_2|$$

Sử dụng khi chỉ có thể di chuyển theo trục ngang và dọc.

Khoảng cách Euclidean:

$$h(n) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Dùng khi di chuyển theo cả đường chéo.

Khoảng cách Chebyshev:

$$h(n) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

Phù hợp khi có thể di chuyển theo tám hướng.

Quy trình thực hiện thuật toán A*

- **Khởi tạo danh sách mở (Open Set) và danh sách đóng (Closed Set).**
- **Lặp lại các bước sau cho đến khi tìm thấy đường đi hoặc danh sách mở rỗng:**
 - * Chọn nút có giá trị $f(n)$ nhỏ nhất.
 - * Nếu nút đó là đích, dừng thuật toán và truy vết đường đi.
 - * Chuyển nút đó sang danh sách đóng.
 - * Mở rộng các nút lân cận, tính toán giá trị $f(n)$ và thêm vào danh sách mở nếu chưa có.
- **Nếu danh sách mở rỗng, kết luận không tìm được đường đi.**

Ví dụ minh họa

Giả sử có một bản đồ dạng lưới với chướng ngại vật:

S	0	0	1	0	0	D
0	1	0	1	0	1	0
0	0	0	1	0	0	0
1	1	0	0	0	1	0

- **S:** Điểm xuất phát.
- **D:** Điểm đích.
- **1:** Chướng ngại vật.
- **0:** Ô trống có thể di chuyển.

Thuật toán A* sẽ tìm đường đi tối ưu từ S đến D bằng cách mở rộng các nút có giá trị $f(n)$ nhỏ nhất và tránh chướng ngại vật.

Ưu điểm của thuật toán A*

- Luôn tìm được đường đi ngắn nhất nếu hàm heuristic phù hợp.
- Hiệu quả trong không gian tìm kiếm lớn.
- Có thể áp dụng cho nhiều bài toán khác nhau.

Nhược điểm

- Cần nhiều bộ nhớ vì phải lưu danh sách mở và danh sách đóng.
- Nếu hàm heuristic không chính xác, có thể làm chậm quá trình tìm kiếm hoặc đưa ra lộ trình không tối ưu.

2.2 Ứng dụng A* trong tìm đường đi tối ưu

Thuật toán A* có nhiều ứng dụng quan trọng trong thực tế, đặc biệt trong các lĩnh vực yêu cầu tìm kiếm đường đi tối ưu một cách nhanh chóng và hiệu quả. Dưới đây là một số ứng dụng chính:

Trò chơi điện tử

- Điều hướng nhân vật NPC (Non-Player Character) tránh chướng ngại vật.
- AI đối thủ trong game chiến thuật tìm đường đi tối ưu.
- Trò chơi đua xe dùng A* để tính toán tuyến đường nhanh nhất.

Robot di động và phương tiện tự hành

- Robot hút bụi dùng A* để lập kế hoạch di chuyển.
- Xe tự lái tìm tuyến đường tốt nhất trong môi trường thực tế.
- Drone/UAV sử dụng A* để tránh vật cản trong không gian.

Hệ thống GPS và điều hướng giao thông

- Google Maps, Apple Maps tính toán tuyến đường tối ưu.
- Ứng dụng giao thông thông minh giảm tắc nghẽn.
- Logistics tối ưu hóa tuyến đường vận chuyển.

Trí tuệ nhân tạo và khoa học dữ liệu

- AI tìm kiếm dữ liệu trong không gian trạng thái lớn.
- Chatbot tối ưu hóa phản hồi theo chuỗi hội thoại.
- NLP cải thiện dịch máy và phân tích cú pháp.

Lập lịch và tối ưu hóa

- Lập lịch CPU giúp tối ưu hóa xử lý tiến trình.
- Quản lý kho hàng tìm đường đi nhanh nhất.
- Tối ưu dây chuyền sản xuất để tăng năng suất.

An ninh mạng và kiểm thử phần mềm

- Phát hiện lỗ hổng bảo mật bằng tìm kiếm tấn công tối ưu.
- Kiểm thử phần mềm bằng cách xác định đường dẫn kiểm thử quan trọng.

Nhờ khả năng tìm đường hiệu quả, thuật toán A* là một trong những phương pháp tìm kiếm mạnh mẽ nhất, được ứng dụng rộng rãi trong nhiều lĩnh vực.

2.3 Cấu trúc bản đồ trò chơi

Bản đồ trong trò chơi đua xe được biểu diễn dưới dạng ma trận 2D, trong đó mỗi ô trên bản đồ đại diện cho một loại địa hình hoặc chướng ngại vật khác nhau. Việc thiết kế bản đồ hợp lý giúp tăng độ thử thách và tính chiến thuật trong trò chơi.

Cấu trúc dữ liệu bản đồ

Bản đồ được lưu dưới dạng ma trận 2D, mỗi ô trong ma trận có một giá trị đại diện cho loại địa hình tương ứng:

- **0**: Ô trống - Xe có thể di chuyển vào.
- **1**: Chướng ngại vật - Xe không thể đi qua.
- **S**: Vạch xuất phát - Điểm bắt đầu của xe.
- **D**: Vạch đích - Mục tiêu cần đến.
- **2**: Đường trơn - Xe có thể di chuyển nhanh hơn nhưng khó kiểm soát.
- **3**: Bùn lầy - Xe di chuyển chậm lại.
- **4**: Đường tắt - Có thể giúp xe đi nhanh hơn nhưng có nguy cơ gặp chướng ngại vật.

Ví dụ về bản đồ trò chơi

Ví dụ về một bản đồ đua xe đơn giản:

```
S  0  0  1  0  0  D
0  1  0  1  0  1  0
0  0  0  1  0  0  0
1  1  0  0  0  1  0
```

Trong đó:

- Xe bắt đầu tại vị trí **S** và phải tìm đường đến **D**.
- Chướng ngại vật (**1**) cản trở lộ trình di chuyển.
- Người chơi cần chọn lộ trình tối ưu để đến đích nhanh nhất.

Mô tả các địa hình nâng cao

Để tăng độ phức tạp và thử thách, có thể sử dụng các loại địa hình khác nhau:

- **Đường trơn (2)**: Xe có thể di chuyển nhanh hơn nhưng dễ mất kiểm soát.
- **Bùn lầy (3)**: Xe bị giảm tốc độ đáng kể.
- **Đường tắt (4)**: Có thể giúp xe rút ngắn quãng đường nhưng có nguy cơ gặp chướng ngại vật bất ngờ.

Ảnh hưởng của bản đồ đến thuật toán A*

- Nếu bản đồ có quá nhiều chướng ngại vật, thuật toán A* sẽ mất nhiều thời gian hơn để tìm ra lộ trình.

- Nếu bản đồ có nhiều đường tắt, người chơi cần cân nhắc giữa rủi ro và tốc độ để quyết định lộ trình tối ưu.
- Các loại địa hình có trọng số khác nhau sẽ ảnh hưởng đến thuật toán tìm đường, đòi hỏi phải điều chỉnh hàm đánh giá heuristic để có kết quả tốt nhất.

Bản đồ đóng vai trò quan trọng trong việc định hình cách thức chơi và chiến thuật của người chơi. Một bản đồ được thiết kế hợp lý sẽ mang lại trải nghiệm thú vị và đầy thử thách cho người chơi.

2.4 Xây dựng thuật toán điều khiển xe đua

Trong trò chơi đua xe, việc điều khiển xe đua và tìm đường đi tối ưu là yếu tố quan trọng quyết định trải nghiệm của người chơi. Để thực hiện điều này, thuật toán A* được sử dụng nhằm tìm kiếm lộ trình ngắn nhất từ điểm xuất phát đến đích, đồng thời tránh các chướng ngại vật trên đường đua.

Các bước triển khai thuật toán điều khiển xe đua

- 1. Khởi tạo môi trường**
 - Tải bản đồ từ dữ liệu ma trận 2D.
 - Xác định vị trí bắt đầu (**S**) và vị trí đích (**D**).
- 2. Khởi tạo thuật toán A***
 - Tạo danh sách mở (**Open Set**) chứa các nút cần kiểm tra.
 - Tạo danh sách đóng (**Closed Set**) để lưu các nút đã được kiểm tra.
- 3. Thực hiện thuật toán**
 - Lặp lại các bước sau cho đến khi tìm thấy đường đi hoặc danh sách mở rỗng:
 - (a) Chọn nút có giá trị $f(n) = g(n) + h(n)$ nhỏ nhất.
 - (b) Nếu nút hiện tại là điểm đích, truy vết lại đường đi và kết thúc thuật toán.
 - (c) Thêm nút hiện tại vào danh sách đóng.
 - (d) Mở rộng các nút lân cận, cập nhật giá trị $g(n)$ và $h(n)$, rồi thêm vào danh sách mở nếu chưa có.
- 4. Xuất kết quả**
 - Nếu tìm thấy đường đi, hiển thị lộ trình tối ưu cho xe đua.
 - Nếu không có đường đi, thông báo rằng xe không thể về đích.

Điều chỉnh tốc độ xe theo địa hình

Mỗi loại địa hình có thể ảnh hưởng đến tốc độ của xe đua:

- **Đường bình thường (0):** Xe di chuyển với tốc độ tiêu chuẩn.
- **Đường trơn (2):** Xe có thể tăng tốc nhưng dễ bị mất kiểm soát.
- **Bùn lầy (3):** Giảm tốc độ xe, tăng thời gian hoàn thành đường đua.
- **Đường tắt (4):** Giúp xe di chuyển nhanh hơn nhưng có nguy cơ gặp chướng ngại vật.

Tối ưu hóa thuật toán điều khiển xe

- **Cải tiến hàm heuristic:** Điều chỉnh heuristic để phản ánh đúng ảnh hưởng của địa hình đến quãng đường di chuyển.

Chương 3

Phân tích thiết kế chương trình

Trò chơi **A Star Racing** là một trò chơi đua xe đơn giản, trong đó người chơi điều khiển một chiếc xe di chuyển trên bản đồ lưới để tìm đường đi tối ưu từ điểm xuất phát đến đích. Trò chơi sử dụng thuật toán A* để tìm kiếm đường đi ngắn nhất và tránh các chướng ngại vật trên đường.

Nguyên tắc hoạt động

- Trò chơi được xây dựng trên nền tảng Python và sử dụng thư viện Pygame để hiển thị đồ họa.
- Bản đồ trò chơi được chia thành các ô lưới, trong đó:
 - * **Ô màu trắng:** Đường có thể đi được.
 - * **Ô màu đen:** Chướng ngại vật.
 - * **Ô màu đỏ:** Điểm xuất phát của xe.
 - * **Ô màu xanh lá cây:** Điểm đích.
 - * **Ô màu vàng:** Đường đi tối ưu được tìm bởi thuật toán A*.
- Thuật toán A* sẽ được áp dụng để tính toán đường đi ngắn nhất từ vị trí xuất phát đến đích.

Thiết kế bản đồ trò chơi

- Bản đồ có kích thước 15×15 ô lưới.
- Mỗi ô có giá trị 0 (đường đi được) hoặc 1 (chướng ngại vật).
- Người chơi không thể di chuyển vào các ô có chướng ngại vật.
- Bản đồ có thể được tùy chỉnh với các bố cục khác nhau để tạo ra thử thách mới.

Cài đặt thuật toán A*

- Thuật toán A* sử dụng hàm chi phí để tìm đường đi tối ưu:
 - * Chi phí thực tế từ điểm bắt đầu đến ô hiện tại.
 - * Ước tính chi phí từ ô hiện tại đến điểm đích (sử dụng heuristic Manhattan).
- Các ô lân cận sẽ được kiểm tra để tìm ra hướng đi tốt nhất.

- Khi tìm thấy đường đi, xe sẽ di chuyển theo đường đã được đánh dấu.

Giao diện trò chơi

- Trò chơi hiển thị một cửa sổ 600×600 pixel.
- Các yếu tố đồ họa bao gồm:
 - * Xe đua.
 - * Chướng ngại vật.
 - * Điểm xuất phát và đích.
 - * Đường đi ngắn nhất.
- Xe sẽ tự động di chuyển theo đường đi tìm được.

Đánh giá kết quả

- Chương trình có thể tìm đường đi nhanh và hiệu quả trên bản đồ kích thước nhỏ.
- Với bản đồ lớn hơn, hiệu suất thuật toán có thể giảm do số lượng ô cần kiểm tra tăng lên.
- Kết quả được hiển thị trực quan để người chơi quan sát quá trình tìm đường.

3.2 Phân tích các đối tượng trong trò chơi

3.2.1 Phân tích các đối tượng trong trò chơi

- **Người chơi:** Là người quan sát quá trình di chuyển của xe theo thuật toán A^* . Trong các phiên bản nâng cao, người chơi có thể điều chỉnh các yếu tố như vị trí xuất phát, đích đến hoặc bản đồ.
- **Xe đua:** Là đối tượng chính của trò chơi, có khả năng di chuyển trên bản đồ theo đường đi được thuật toán tìm ra.
- **Bản đồ:** Được chia thành các ô vuông, mỗi ô có thể là đường đi hoặc chướng ngại vật, ảnh hưởng đến việc tính toán đường đi của xe.
- **Chướng ngại vật:** Là các ô không thể đi qua, tạo ra thử thách cho thuật toán tìm đường.
- **Thuật toán A^* :** Điều khiển quá trình tìm kiếm đường đi tối ưu từ điểm xuất phát đến đích.

3.2.2 Đánh giá kết quả

- Chương trình có thể tìm đường đi nhanh và hiệu quả trên bản đồ kích thước nhỏ.
- Với bản đồ lớn hơn, hiệu suất thuật toán có thể giảm do số lượng ô cần kiểm tra tăng lên.
- Kết quả được hiển thị trực quan để người chơi quan sát quá trình tìm đường.

3.2.3 Hạn chế và hướng phát triển

- Chương trình chưa hỗ trợ bản đồ động (có chướng ngại vật thay đổi theo thời gian).
- Hiện tại chỉ có một cách di chuyển, có thể bổ sung nhiều chiến lược tìm đường hơn.
- Trong tương lai có thể cải tiến thuật toán để tăng tốc độ xử lý.
- Bổ sung hiệu ứng đồ họa và âm thanh để trò chơi sinh động hơn.

3.3 Cài đặt thuật toán A*

3.3.1 Phân tích các đối tượng trong trò chơi

- **Bước 1:** Khởi tạo danh sách mở (open list) và danh sách đóng (closed list).
- **Bước 2:** Đưa điểm bắt đầu vào danh sách mở.
- **Bước 3:** Duyệt qua các ô lân cận, tính toán chi phí và thêm vào danh sách mở nếu ô đó chưa được duyệt.
- **Bước 4:** Chọn ô có tổng chi phí thấp nhất ($f = g + h$) để tiếp tục duyệt.
- **Bước 5:** Lặp lại quá trình cho đến khi tìm thấy điểm đích hoặc danh sách mở rỗng.
- **Bước 6:** Truy vết lại đường đi từ điểm đích về điểm xuất phát.

3.3.2 Giao diện trò chơi

- Trò chơi hiển thị một cửa sổ 600x600 pixel.
- Các yếu tố đồ họa bao gồm:
 - * Xe đua.
 - * Chướng ngại vật.
 - * Điểm xuất phát và đích.
 - * Đường đi ngắn nhất.
- Xe sẽ tự động di chuyển theo đường đi tìm được.

3.3.3 Phân tích các đối tượng trong trò chơi

- **Người chơi:** Là người quan sát quá trình di chuyển của xe theo thuật toán A*. Trong các phiên bản nâng cao, người chơi có thể điều chỉnh các yếu tố như vị trí xuất phát, đích đến hoặc bản đồ.
- **Xe đua:** Là đối tượng chính của trò chơi, có khả năng di chuyển trên bản đồ theo đường đi được thuật toán tìm ra.
- **Bản đồ:** Được chia thành các ô vuông, mỗi ô có thể là đường đi hoặc chướng ngại vật, ảnh hưởng đến việc tính toán đường đi của xe.

- **Chướng ngại vật:** Là các ô không thể đi qua, tạo ra thử thách cho thuật toán tìm đường.
- **Thuật toán A*:** Điều khiển quá trình tìm kiếm đường đi tối ưu từ điểm xuất phát đến đích.

3.4 Lập kế hoạch thuật toán

3.4.1 Xác định đầu vào và đầu ra

- **Đầu vào:** Bản đồ lưới, vị trí xuất phát, vị trí đích.
- **Đầu ra:** Đường đi tối ưu từ điểm xuất phát đến đích.

3.4.2 Các bước thực hiện

1. Khởi tạo danh sách mở (các ô cần kiểm tra) và danh sách đóng (các ô đã kiểm tra).
2. Thêm ô xuất phát vào danh sách mở và thiết lập chi phí ban đầu.
3. Lặp lại các bước sau cho đến khi danh sách mở rỗng hoặc tìm thấy điểm đích:
 - Chọn ô có chi phí thấp nhất trong danh sách mở.
 - Loại bỏ ô đó khỏi danh sách mở và thêm vào danh sách đóng.
 - Xem xét các ô lân cận:
 - * Nếu ô lân cận là điểm đích, kết thúc thuật toán.
 - * Nếu ô lân cận không phải chướng ngại vật và chưa có trong danh sách đóng, tính toán chi phí mới và thêm vào danh sách mở nếu cần.
4. Truy vết lại đường đi từ điểm đích về điểm xuất phát.
5. Xuất kết quả đường đi.

3.5 Thiết kế giao diện

3.5.1 Màn hình chính

- Màn hình chính của trò chơi hiển thị bản đồ lưới với kích thước 15×15 ô vuông.
- Mỗi ô vuông có thể là đường đi, chướng ngại vật, điểm xuất phát hoặc điểm đích.
- Giao diện được thiết kế đơn giản, tập trung vào việc thể hiện trực quan đường đi của xe theo thuật toán A*.
- Màu sắc được sử dụng để giúp người chơi dễ dàng nhận biết các yếu tố trên bản đồ:
 - * **Màu trắng:** Đường đi có thể di chuyển.
 - * **Màu đen:** Chướng ngại vật.
 - * **Màu đỏ:** Điểm xuất phát của xe.
 - * **Màu xanh lá cây:** Điểm đích.

- * **Màu vàng:** Đường đi tối ưu được thuật toán tìm ra.

3.5.2 Các thành phần đồ họa

- **Xe đua:**
 - * Được hiển thị dưới dạng hình vuông hoặc biểu tượng xe nhỏ, có thể di chuyển theo đường đi tìm được.
 - * Màu sắc nổi bật để dễ dàng theo dõi trong quá trình di chuyển.
- **Chướng ngại vật:**
 - * Là các ô vuông màu đen trên bản đồ, xe không thể đi qua.
 - * Bố trí ngẫu nhiên hoặc theo thiết kế sẵn để tạo thử thách cho thuật toán.
- **Điểm xuất phát và đích:**
 - * Xuất phát (màu đỏ) và đích (màu xanh lá cây) được đặt cố định hoặc có thể thay đổi tùy vào thiết lập của người chơi.
- **Đường đi tối ưu:**
 - * Sau khi thuật toán A* tính toán xong, đường đi tối ưu được hiển thị bằng màu vàng để dễ dàng quan sát.

3.5.3 Các nút điều khiển

- **Bắt đầu (Start):** Kích hoạt thuật toán A* để xe bắt đầu di chuyển theo đường đi được tìm thấy.
- **Làm mới (Reset):** Đặt lại bản đồ và vị trí xe về trạng thái ban đầu.
- **Thoát (Exit):** Thoát khỏi trò chơi.

3.5.4 Hiển thị trạng thái

- Hiển thị **số bước di chuyển** của xe từ điểm xuất phát đến đích.
- Cập nhật **thời gian hoàn thành** sau khi xe đến đích.
- Hiển thị **thông báo lỗi** nếu không tìm được đường đi.
- Nếu người chơi có thể thiết lập bản đồ tùy chỉnh, giao diện cũng sẽ hiển thị các công cụ hỗ trợ chỉnh sửa bản đồ.

3.5.5 Hướng phát triển giao diện

- **Cải thiện đồ họa:** Thêm hiệu ứng hình ảnh để làm cho trò chơi hấp dẫn hơn.
- **Tăng độ tùy chỉnh:** Cho phép người chơi thay đổi vị trí xuất phát, đích và bố trí chướng ngại vật.
- **Thêm chế độ chơi:** Cung cấp các chế độ chơi khác nhau như chế độ thử thách với thời gian giới hạn.
- **Bổ sung âm thanh:** Hiệu ứng âm thanh khi xe di chuyển hoặc khi đạt đến đích để tạo trải nghiệm sinh động hơn.

3.6 Đánh giá kết quả

3.6.1 Hiệu suất thuật toán

- Thuật toán A* hoạt động tốt trên bản đồ có kích thước nhỏ đến trung bình.
- Thời gian tìm đường nhanh, đặc biệt khi số lượng chướng ngại vật không quá lớn.
- Khi bản đồ có nhiều chướng ngại vật, số ô cần kiểm tra tăng lên, dẫn đến thời gian xử lý lâu hơn.

3.6.2 Ưu điểm của thuật toán A*

1. Tìm đường đi tối ưu:

- Thuật toán A* luôn tìm được đường đi ngắn nhất nếu có.
- Kết hợp giữa chi phí thực tế và ước lượng khoảng cách giúp tìm đường hiệu quả hơn so với BFS hoặc Dijkstra.

2. Tốc độ tốt:

- Trong các trường hợp thông thường, thuật toán có thể tìm đường nhanh nhờ sử dụng hàng đợi ưu tiên.
- Có thể điều chỉnh heuristic để cân bằng giữa tốc độ và độ chính xác.

3. Khả năng mở rộng:

- Có thể áp dụng cho các bản đồ lớn hơn nếu tối ưu cấu trúc dữ liệu.
- Có thể kết hợp với các thuật toán khác để tăng hiệu quả.

3.6.3 Nhược điểm của thuật toán

1. Tốn bộ nhớ:

- Thuật toán lưu nhiều trạng thái trong danh sách mở và danh sách đóng, làm tăng dung lượng bộ nhớ sử dụng.

2. Hiệu suất giảm trên bản đồ lớn:

- Khi số lượng ô vuông tăng lên, số bước tính toán cũng tăng, làm giảm hiệu suất.
- Nếu chướng ngại vật dày đặc, thuật toán có thể mất nhiều thời gian để tìm đường đi hợp lý.

3. Chưa tối ưu với địa hình động:

- Nếu bản đồ thay đổi liên tục (ví dụ như chướng ngại vật di chuyển), thuật toán cần được chạy lại từ đầu, làm giảm hiệu suất.

3.6.4 Đánh giá giao diện trò chơi

– Dễ nhìn và trực quan:

- * Các màu sắc được sử dụng rõ ràng giúp người chơi dễ phân biệt các thành phần trên bản đồ.

- * Hiển thị đường đi tối ưu giúp người chơi theo dõi quá trình tìm đường của xe.
- **Dễ thao tác:**
 - * Các nút điều khiển đơn giản, dễ sử dụng.
 - * Có thể thêm các tùy chọn để người chơi chỉnh sửa bản đồ hoặc thay đổi điểm xuất phát và đích.
- **Hạn chế:**
 - * Giao diện còn đơn giản, chưa có hiệu ứng động hoặc âm thanh.
 - * Chưa có chế độ chơi thử thách hoặc tùy chỉnh bản đồ nâng cao.

3.6.5 Hướng phát triển trong tương lai

1. **Tối ưu thuật toán:**
 - Sử dụng kỹ thuật giảm bộ nhớ để tăng hiệu suất xử lý.
 - Áp dụng thuật toán A* cải tiến như Jump Point Search để tìm đường nhanh hơn trên bản đồ lớn.
2. **Cải thiện giao diện đồ họa:**
 - Thêm hiệu ứng động khi xe di chuyển.
 - Cải thiện thiết kế bản đồ với các chủ đề đa dạng.
3. **Thêm các chế độ chơi:**
 - Chế độ nhiều người chơi với bản đồ lớn hơn.
 - Chế độ thử thách với giới hạn thời gian hoặc bản đồ thay đổi liên tục.
4. **Nâng cao trí tuệ nhân tạo:**
 - Kết hợp nhiều thuật toán tìm đường khác để AI có thể thích nghi với các dạng bản đồ khác nhau.
 - Thêm khả năng học hỏi từ các lần chơi trước để tối ưu hóa đường đi.

3.6.6 Kết luận

- Trò chơi hoạt động tốt với thuật toán A*, giúp xe tìm đường nhanh và hiệu quả.
- Giao diện hiện tại đơn giản nhưng trực quan, phù hợp để minh họa thuật toán.
- Cần phát triển thêm các tính năng mở rộng để nâng cao trải nghiệm người chơi.
- Việc tối ưu thuật toán và bổ sung các yếu tố động sẽ giúp trò chơi trở nên thú vị và thử thách hơn.

3.7 Hạn chế và hướng phát triển

3.7.1 Hạn chế của trò chơi

1. **Giới hạn kích thước bản đồ:**
 - Hiện tại, trò chơi chỉ hỗ trợ bản đồ kích thước 15×15 , gây hạn chế khi mở rộng trò chơi.

- Khi tăng kích thước bản đồ, hiệu suất thuật toán có thể giảm do số lượng ô cần kiểm tra tăng.
- 2. **Thiếu yếu tố động:**
 - Trò chơi chưa có cơ chế thay đổi bản đồ theo thời gian thực.
 - Chướng ngại vật tĩnh làm giảm độ thử thách khi chơi nhiều lần.
- 3. **Chưa có đa dạng chế độ chơi:**
 - Người chơi chỉ có thể xem xe di chuyển theo đường đi tối ưu.
 - Chưa có chế độ chơi tự điều khiển hoặc nhiều màn chơi thử thách.
- 4. **Giao diện đồ họa còn đơn giản:**
 - Thiếu hiệu ứng hình ảnh khi xe di chuyển.
 - Chưa có âm thanh hoặc hiệu ứng đặc biệt khi xe về đích.
- 5. **Chưa có tính năng học máy:**
 - Xe chưa có khả năng học hỏi từ các lần chơi trước để tối ưu hóa đường đi.
 - Không có hệ thống đánh giá hiệu suất của người chơi.

3.7.2 Hướng phát triển trong tương lai

1. **Mở rộng kích thước bản đồ:**
 - Hỗ trợ nhiều kích thước bản đồ khác nhau (20×20 , 30×30 , ...).
 - Tối ưu thuật toán để xử lý nhanh hơn trên bản đồ lớn.
2. **Bổ sung yếu tố động:**
 - Thêm chướng ngại vật di chuyển hoặc thay đổi theo thời gian.
 - Thêm tính năng cập nhật bản đồ theo từng màn chơi.
3. **Nâng cao trí tuệ nhân tạo:**
 - Kết hợp thuật toán D* hoặc thuật toán học tăng cường để xe có thể tự cải thiện khả năng tìm đường.
 - Cho phép xe tránh các chướng ngại vật di chuyển theo thời gian thực.
4. **Thêm chế độ chơi mới:**
 - Chế độ người chơi tự điều khiển xe bằng bàn phím.
 - Chế độ đua xe với nhiều đối thủ AI, yêu cầu tìm đường nhanh hơn để về đích trước.
5. **Cải thiện giao diện đồ họa:**
 - Thêm hiệu ứng động khi xe di chuyển hoặc va chạm với chướng ngại vật.
 - Bổ sung âm thanh để tăng trải nghiệm người chơi.
6. **Xây dựng hệ thống đánh giá người chơi:**
 - Thêm bảng xếp hạng thời gian hoàn thành mỗi màn chơi.
 - Lưu lại lịch sử di chuyển của xe để phân tích và cải thiện chiến thuật.

3.7.3 Tổng kết

- Trò chơi đã đạt được mục tiêu minh họa thuật toán A* một cách trực quan và dễ hiểu.

- Tuy nhiên, vẫn còn nhiều hạn chế cần cải thiện để tăng tính thử thách và hấp dẫn hơn.
- Trong tương lai, việc nâng cấp thuật toán, thêm yếu tố động và cải thiện giao diện sẽ giúp trò chơi trở nên hấp dẫn hơn đối với người chơi.

3.8 Cách hoạt động của code

3.8.1 Tổng quan về chương trình

Chương trình trò chơi **A Star Racing** được viết bằng Python, sử dụng thư viện Pygame để hiển thị đồ họa và thuật toán A* để tìm đường đi tối ưu. Dưới đây là chi tiết cách hoạt động của từng phần trong chương trình.

3.8.2 Khởi tạo chương trình

- **Nhập thư viện cần thiết:**

- pygame: Để tạo giao diện đồ họa và xử lý sự kiện bàn phím.
- heapq: Hỗ trợ hàng đợi ưu tiên trong thuật toán A*.
- time: Để đo thời gian hoàn thành đường đi.

- **Cấu hình kích thước cửa sổ và bản đồ:**

- Cửa sổ có kích thước 600×600 pixel.
- Bản đồ có kích thước 15×15 ô lưới.
- Mỗi ô lưới có kích thước `CELL_SIZE = WIDTH // COLS`.

- **Xác định màu sắc:**

- Màu trắng: Ô đường đi.
- Màu đen: Chướng ngại vật.
- Màu đỏ: Điểm xuất phát.
- Màu xanh lá cây: Điểm đích.
- Màu vàng: Đường đi tối ưu.

3.8.3 Tạo bản đồ trò chơi

Bản đồ được biểu diễn dưới dạng danh sách hai chiều (MAP), mỗi phần tử có giá trị:

- 0: Ô có thể đi được.
- 1: Chướng ngại vật.

Điểm xuất phát (START) và đích (GOAL) được xác định trước.

3.8.4 Thuật toán A*

- **Hàm heuristic:**

- Sử dụng khoảng cách Manhattan để ước lượng chi phí từ ô hiện tại đến đích:

$$h(n) = |x_{goal} - x_{current}| + |y_{goal} - y_{current}|$$

- **Hàm tìm đường A*:**

1. Khởi tạo danh sách mở (open_list) và danh sách đóng (came_from).
2. Đưa điểm xuất phát vào danh sách mở.
3. Lặp lại quá trình tìm kiếm:
 - Lấy ô có chi phí thấp nhất ($f(n) = g(n) + h(n)$).
 - Nếu ô đó là đích, truy vết lại đường đi và trả về kết quả.
 - Kiểm tra các ô lân cận:
 - * Nếu ô đó có thể đi được, tính toán chi phí mới.
 - * Nếu đường mới tốt hơn đường cũ, cập nhật danh sách mở.
4. Nếu danh sách mở rỗng và chưa tìm thấy đích, kết thúc thuật toán.

3.8.5 Xử lý giao diện và hiển thị trò chơi

- **Vẽ bản đồ:**

- Duyệt qua từng ô trong danh sách MAP để vẽ các ô vuông tương ứng với màu sắc.
- Vẽ đường đi tìm được bằng thuật toán A* bằng màu vàng.
- Vẽ xe đua tại vị trí hiện tại.

- **Xử lý sự kiện bàn phím:**

- Dùng `pygame.KEYDOWN` để phát hiện khi người chơi nhấn phím.
- Nếu nhấn các phím mũi tên (UP, DOWN, LEFT, RIGHT), xe di chuyển theo hướng đó nếu hợp lệ.
- Nếu nhấn phím H, thuật toán A* sẽ chạy và hiển thị đường đi tối ưu.

3.8.6 Kiểm tra điều kiện thắng cuộc

- Nếu vị trí xe trùng với vị trí đích (GOAL), hiển thị thông báo chiến thắng và thời gian hoàn thành.
- Kết thúc vòng lặp chính (`running = False`).

3.8.7 Thoát chương trình

- Khi người chơi đóng cửa sổ hoặc hoàn thành đường đi, `pygame.quit()` được gọi để thoát chương trình.

3.8.8 Tóm tắt cách hoạt động của code

1. Khởi tạo Pygame, bản đồ và thuật toán A*.
2. Hiển thị giao diện bản đồ, vẽ các thành phần.
3. Kiểm tra sự kiện bàn phím để di chuyển xe hoặc chạy thuật toán A*.
4. Tìm đường đi ngắn nhất bằng thuật toán A* và hiển thị lên màn hình.
5. Kiểm tra xem xe đã đến đích hay chưa.
6. Thoát chương trình khi hoàn thành.

3.8.9 kết quả

newpage



Hình 3.1: Mô tả ảnh

Kết Luận

Ưu điểm

Nghiên cứu đã thành công trong việc ứng dụng thuật toán A^* vào trò chơi đua xe để tìm đường đi tối ưu, giúp người chơi nhanh chóng xác định lộ trình hợp lý nhất trong môi trường có chướng ngại vật. Một trong những ưu điểm nổi bật của thuật toán A^* là khả năng tìm kiếm hiệu quả với độ chính xác cao, nhờ vào sự kết hợp giữa chi phí thực tế và chi phí ước tính. Ngoài ra, thuật toán này đảm bảo tìm ra con đường tối ưu nhất nếu có, giúp trò chơi vận hành mượt mà và mang đến trải nghiệm hấp dẫn cho người chơi. Hơn nữa, trò chơi được thiết kế với cơ chế tương tác linh hoạt, giúp nâng cao khả năng phản xạ và tư duy chiến lược của người chơi khi đối mặt với các thử thách trên đường đua.

Nhược điểm

Bên cạnh những ưu điểm, nghiên cứu cũng chỉ ra một số hạn chế cần được khắc phục. Đầu tiên, thuật toán A^* có thể gặp khó khăn về hiệu suất khi bản đồ trở nên quá phức tạp hoặc kích thước quá lớn, dẫn đến thời gian tính toán tăng đáng kể. Điều này có thể ảnh hưởng đến trải nghiệm của người chơi, đặc biệt trong các tình huống yêu cầu phản ứng nhanh. Ngoài ra, trò chơi hiện chưa có sự đa dạng về chiến thuật điều khiển xe, vì thuật toán chủ yếu tập trung vào việc tìm đường mà chưa xét đến các yếu tố động lực học của phương tiện. Bên cạnh đó, việc tối ưu hóa thuật toán để giảm mức tiêu thụ tài nguyên cũng là một thách thức cần được nghiên cứu thêm, đặc biệt khi triển khai trên các nền tảng có cấu hình hạn chế.

Hướng phát triển

Trong tương lai, nghiên cứu có thể mở rộng theo nhiều hướng nhằm cải thiện hiệu suất và tính ứng dụng của trò chơi. Một trong những hướng phát triển quan trọng là tối ưu hóa thuật toán A^* bằng cách kết hợp với các thuật toán tìm kiếm khác như **Dijkstra** hoặc **Jump Point Search (JPS)** để tăng tốc độ xử lý. Ngoài ra, có thể áp dụng các kỹ thuật **Machine Learning** để dự đoán đường đi tối ưu dựa trên dữ liệu từ các ván chơi trước, giúp trò chơi trở nên thông minh và thích ứng hơn với phong cách chơi của từng người. Bên cạnh đó, trò chơi có thể được nâng cấp với nhiều yếu tố mới như chế độ nhiều người chơi, địa hình đa dạng và các cấp độ khó

nâng cao để mang đến trải nghiệm phong phú hơn cho người dùng. Những cải tiến này không chỉ giúp trò chơi trở nên hấp dẫn hơn mà còn góp phần vào việc nghiên cứu và phát triển các thuật toán tìm đường tối ưu trong lĩnh vực **Trí tuệ nhân tạo**.

Tài Liệu Tham Khảo

[1] Bài giảng của ThS. Nguyễn Thái Khánh và ThS. Lê Thị Thùy Trang, 2025.