**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN I**

**Thiết kế trò chơi rắn săn mồi ứng dụng thuật toán tìm đường đi A\***

**Lương Đức Trọng**

Trong.ld215489@sis.hust.edu.vn

**Ngành Khoa học máy tính**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Nguyễn Duy Hiệp  Chữ ký của GVHD |
| **Bộ môn:** | Khoa học máy tính |
| **Trường:** | Công nghệ thông tin và truyền thông |
| **HÀ NỘI, 1/2024**  **LỜI CẢM ƠN**  Em xin chân thành cảm ơn thầy Nguyễn Đức Hiệp vì đã dành thời gian xem xét, hướng dẫn, góp ý và định hướng để em có thể hoàn thiện được Project I.  Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để có điều kiện bổ sung, nâng cao kiến thức của mình, giúp bản thân hoàn thiện hơn trong những dự án sau này.  Em xin kính chúc thầy luôn luôn khỏe mạnh và ngày một thành công hơn trên con đường giảng dạy của mình.  Em xin trân trọng cảm ơn! | |
|  | |

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 1](#_Toc156139880)

[1.1 Đặt vấn đề 1](#_Toc156139881)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc156139882)

[1.3 Định hướng giải pháp 1](#_Toc156139883)

[1.4 Bố cục đồ án 2](#_Toc156139884)

[CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH CÁC YÊU CẦU 3](#_Toc156139885)

[2.1 Khảo sát hiện trạng 3](#_Toc156139886)

[2.2 Tổng quan chức năng 3](#_Toc156139887)

[2.2.1 Biểu đồ use case tổng quát 3](#_Toc156139888)

[2.3 Yêu cầu phi chức năng 4](#_Toc156139889)

[CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG 5](#_Toc156139890)

[3.1 Ngôn ngữ Python 5](#_Toc156139891)

[3.2 Thư viện Pygame 5](#_Toc156139892)

[3.3 Các công nghệ khác 6](#_Toc156139893)

[CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG 7](#_Toc156139894)

[4.1 Thiết kế kiến trúc 7](#_Toc156139895)

[4.2 Thiết kế chi tiết 7](#_Toc156139896)

[4.2.1 Thiết kế giao diện 7](#_Toc156139897)

[4.2.2 Thiết kế thuật toán 11](#_Toc156139898)

[4.3 Xây dựng ứng dụng 12](#_Toc156139899)

[4.3.1 Thư viện và công cụ sử dụng 12](#_Toc156139900)

[4.3.2 Kết quả đạt được 12](#_Toc156139901)

[4.4 Kiểm thử 12](#_Toc156139902)

[4.5 Triển khai 13](#_Toc156139903)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 14](#_Toc156139904)

[5.1 Kết luận 14](#_Toc156139905)

[5.2 Hướng phát triển 14](#_Toc156139906)

[5.3 Lời cảm ơn 14](#_Toc156139907)

**DANH MỤC HÌNH VẼ**

[Hình 2.1 Biểu đồ use case tổng quan chương trình 4](#_Toc156139811)

[Hình 4.1 Giao diện khởi đầu 7](#_Toc156139812)

[Hình 4.2 Giao diện người chơi 8](#_Toc156139813)

[Hình 4.3 Giao diện bot điều khiển 9](#_Toc156139814)

[Hình 4.4 Giao diện kết thúc 9](#_Toc156139815)

**DANH MỤC BẢNG BIỂU**

[Bảng 4.1 Danh sách thư viện và công cụ sử dụng 11](#_Toc156139869)

# GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Trò chơi rắn săn mồi (Snake) lần đầu tiên được giới thiệu vào năm 1976, được coi là một trong những tựa game có tuổi đời lâu nhất lịch sử. Kể từ khi ra mắt, tựa game này luôn chiếm được sự ưa chuộng của nhiều thế hệ người chơi và thực sự trở thành một cơn sốt phổ biến trên toàn cầu khi được gắn liền với các thế hệ điện thoại đời đầu của NOKIA – một ông lớn trong ngành sản xuất thiết bị di động vào cuối thế kỉ 20, đầu thế kỉ 21.

Người chơi cần điều khiển con rắn để đi đến thức ăn và đạt được số điểm cao nhất, trò chơi kết thúc khi con rắn tự cắn vào mình hoặc va phải chướng ngại vật.

Với luật chơi khá đơn giản, trò chơi này vẫn trở thành một hiện tượng gây nghiện bởi độ khó của nó khi tốc độ di chuyển của rắn và độ dài con rắn tăng dần theo số điểm đạt được.

Do đó, đã có nhiều người chơi muốn tìm ra cách chơi phù hợp, sao cho tối ưu nhất, chính là tiền đề để bài toán tìm đường đi cho rắn ra đời.

Mặt khác, khi áp dụng các cách thức để giải quyết bài toán trên, cũng là một cơ hội để có thể kiểm thử các phương pháp mới. Vậy nên bài toán trên đã được nghiên cứu và áp dụng các giải thuật khác nhau.

## Mục tiêu và phạm vi đề tài

Trước đây, đã có nhiều thuật toán tìm đường được áp dụng cho trò chơi rắn săn mồi để tìm đường đi tối ưu nhất, ví dụ như: Breadth-First Search, Best-First Search, Deepth-First Search, Hamilton Cycle,…

Các thuật toán trên đều có thể tìm được đường đi cho rắn, tuy nhiên mỗi thuật toán lại có các ưu, nhược điểm khác nhau, chỉ tối ưu trong một vài trường hợp, hoặc gặp các vấn đề về hiệu năng, thời gian thực hiện trong không gian tìm kiếm lớn.

Do đó, đề tài này được xây dựng, nhằm thực hành lập trình trò chơi rắn săn mồi theo các luật cho trước, đồng thời thử nghiệm và áp dụng thuật toán tìm kiếm heuristic vào trò chơi để cải thiện hiệu suất.

## Định hướng giải pháp

Theo mục tiêu đã xác định từ phần trước, trước tiên cần phải xây dựng phần mềm trò chơi rắn săn mồi, sau đó áp dụng thuật toán để tìm đường đi cho rắn phục vụ cho các mục đích khác nhau của trò chơi.

Từ yêu cầu đó đề tài này sử dụng thư viện PyGame – một thư viện hỗ trợ xây dựng các trò chơi của ngôn ngữ Python – để xây dựng nên trò chơi. Sau đó áp dụng thuật toán A\* - một thuật toán heuristic - để tìm đường đi cho rắn. Sở dĩ chọn ngôn ngữ Python là vì đây là ngôn ngữ dễ học, có khả năng phát triển nhanh phần mềm, đồng thời thư viện PyGame được xây dựng sẳn trên ngôn ngữ này cung cấp các tác vụ cụ thể, dễ hiểu thân thiện với người mới, dễ dàng xây dụng nên các phần mềm trò chơi đơn giản. Thuật toán A\* được sử dụng có hiệu suất tốt hơn so với các thuật toán tìm kiếm thông thường nhờ vào việc sử dụng hàm heuristic để đánh giá.

## Bố cục đồ án

Phần còn lại của báo cáo này được tổ chức như sau:

Trong chương 2, em sẽ phân tích, đánh giá các yêu cầu của bài toán, từ đó xác định các chức năng cần thiết.

Chương 3 sẽ nói về các công nghệ được sử dụng cho đề tài này.

Quá trình thiết kế, triển khai, đánh giá hệ thống và xây dựng ứng dụng sẽ được đề cập đến trong chương 4.

# KHẢO SÁT VÀ PHÂN TÍCH CÁC YÊU CẦU

Với chương trước, phạm vi, mục tiêu và giải pháp của đề tài đã được xác định. Trong chương 2, em sẽ đi sâu vào khảo sát gặp phân tích các yêu cầu của bài toán, từ đó xác định các chức năng cần thực hiện và yêu cầu phi chức năng.

## Khảo sát hiện trạng

Trong các chương trình trò chơi con rắn đã được phát triển, người dùng thông thường sẽ điều khiển con rắn di chuyển để ăn thức ăn. Tuy nhiên, trong một số chương trình xuất hiện chướng ngại vật đòi hỏi người dùng cần phải điều khiển khéo léo hơn, đồng thời trò chơi phải luôn đảm bảo rằng luôn tồn tại đường đi từ rắn đến thức ăn.

Ngoài ra, để thử nghiệm hoạt động của thuật toán cũng cần phải mô phỏng quá trình rắn di chuyển theo con đường đã tìm, vì vậy, cũng cần phải xây dựng chương trình sao cho rắn đi theo con đường đã được tìm (khác với mục trên là người chơi điều khiển).

## Tổng quan chức năng

Phần mềm được thiết kế với nhiều chức năng hữu ích để đáp ứng nhu cầu đa dạng của người dùng. Trước hết, giao diện của phần mềm sẽ hiển thị các lựa chọn đa dạng, cho phép người dùng chọn giữa việc tham gia trò chơi hoặc theo dõi thuật toán trong quá trình hoạt động. Ngoài ra, phần mềm cũng cung cấp mục hướng dẫn sử dụng chi tiết, giúp người dùng dễ dàng làm quen và trải nghiệm sử dụng. Cuối cùng, phần mềm sẽ thực hiện mọi yêu cầu theo lựa chọn của người dùng, đảm bảo tính linh hoạt và sự thuận tiện. Điều này bao gồm việc xử lý các yêu cầu thao tác, cập nhật thông tin, và thực hiện các tác vụ khác nhau để đáp ứng mong muốn của người dùng một cách hiệu quả.

Phần mềm sẽ cung cấp các chức năng cụ thể tùy thuộc vào lựa chọn của người dùng. Trong trường hợp người dùng muốn điều khiển trò chơi, họ có thể di chuyển rắn theo ý muốn của mình. Giao diện sẽ hiển thị các thông tin quan trọng như số điểm, độ dài của rắn, và chướng ngại vật. Đặc biệt, độ khó của trò chơi sẽ tăng dần theo thời gian, bao gồm tăng tốc độ di chuyển của rắn, độ dài của rắn và số lượng chướng ngại vật, phụ thuộc vào số điểm mà người chơi đạt được.

Nếu người dùng chọn trò chơi do bot điều khiển, chức năng sẽ tương tự như trò chơi do người điều khiển. Tuy nhiên, trong trường hợp này, bot sẽ tự động di chuyển theo một đường đi được tính toán trước.

### Biểu đồ use case tổng quát

Từ các chức năng đã được xác định, có thể dễ dàng tổng hợp lại các yêu cầu chính của hệ thống. Hình 2.1 dưới đây được gọi là biểu đồ use case tổng quát, mô hình hoá các yêu cầu của hệ thống theo góc nhìn chung nhất với các nhiệm vụ vừa nêu.

A diagram of a diagram

Description automatically generated

Hình . Biểu đồ use case tổng quan chương trình

## Yêu cầu phi chức năng

Phần mềm cần thuận tiện, dễ sử dụng, có đầy đủ các hướng dẫn điều khiển và giao diện hài hoà.

# CÔNG NGHỆ SỬ DỤNG

Chương 2 đã xác định rõ các chức năng, yêu cầu của hệ thống cũng như ca sử dụng chung của toàn bộ hệ thống. Từ đó xác định các công nghệ, công cụ cần thiết để xây dựng chương trình và sẽ được đề cập trong chương này. Với ngôn ngữ Python ở mục 3.1, thư viện Pygame ở mục 3.2 và cuối cùng, các công nghệ khác sẽ được trình bày ở mục 3.3.

## Ngôn ngữ Python

**Python** là một ngôn ngữ lập trình thông dịch độc lập, linh hoạt, mã nguồn mở và vô cùng đa năng. Python ra đời bởi Guido van Rossum (một nhà khoa học máy tính và nhà phát triển phần mềm người Hà Lan) và được phát hành lần đầu tiên vào năm 1991. Được phát triển trên nền tảng C++, Python nhanh chóng trở thành ngôn ngữ được ưa chuộng và phổ biến hàng đầu trên thế giới bởi cú pháp ngắn gọn, sát với mô phỏng ngữ nghĩa Tiếng Anh, cộng đồng lớn, hỗ trợ cực kỳ mạnh mẽ các thư viện toán học, đa nền tảng và đa dạng phong cách lập trình. Thật vậy, có thể xây dựng chương trình Python theo lập trình tuần tự, lập trình hướng cấu trúc, hướng function (thủ tục) và cả hướng đối tượng. Chúng ta hay nói vui với nhau về lý do lựa chọn Python hàng đầu là "code ít nhất, được nhiều nhất".

Trong nghiên cứu, Python cũng thường được lựa chọn là ngôn ngữ để kiểm thử vì tính nhanh, gọn, dễ dàng triển khai và đánh giá dù có nhược điểm về mặt tốc độ và bộ nhớ. Các thuật toán cần bộ số liệu lớn, nhiều bước tính toán cũng ưu tiên được thử nghiệm bằng Python nhờ tính linh hoạt, ít ràng buộc và sự hỗ trợ từ cộng đồng mạnh. Trong đề tài này, toàn bộ mã nguồn của chương trình sẽ được viết bằng Python nhờ vào các đặc điểm ưu việt đã kể trên.

## Thư viện Pygame

**Pygame** là một thư viện lập trình trò chơi của ngôn ngữ Python. Pygame được viết bởi Pete Shinners thay thế cho chương trình PySDL sau khi quá trình phát triển dự án này bị đình trệ. Pygame có thể chạy trên nhiều nền tảng và hệ điều hành khác nhau, giúp các nhà phát triển có thể sử dụng công cụ và chức năng mở rộng để tạo ra các trò chơi nhập vai ấn tượng. Pygame sử dụng Simple DirectMedia Layer (SDL), một thư viện phát triển đa nền tảng cho phép các nhà phát triển có thể truy cập vào phần cứng máy tính như đồ họa, âm thanh và thiết bị đầu vào. Pygame cung cấp nhiều chức năng mở rộng hỗ trợ nhà phát triển tập trung phát triển trò chơi API trực quan và dễ hiểu, hỗ trợ người mới sử dụng hay cả những nhà phát triển có kinh nghiệm đều có thể truy cập được. Tính đơn giản, linh hoạt, dễ sử dụng của Pygame đang ngày càng phổ biến với các nhà phát triển.

Với những đặc điểm đó, Pygame đặc biệt rất phù hợp với yêu cầu xây dựng một trò chơi rắn săn mồi với đầy đủ các tính năng cơ bản. Dễ dàng sửa chữa, bảo trì, vận hành và thuận tiện cho các lập trình viên mới trong quá trình nghiên cứu, phát triển.

## Các công nghệ khác

Đề tài này được phát triển trên **Visual Studio Code (VS Code**). VS Code là một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ trên 3 nền tảng là Windows, Linux và macOS được phát triển bởi Microsoft. VS Code hỗ trợ cho JavaScript, Node.js và TypeScript, cũng như cung cấp một hệ sinh thái mở rộng vô cùng phong phú cho nhiều ngôn ngữ lập trình khác. Nó có thể chạy trên nhiều nền tảng và được tích hợp với Git, có chức năng nổi bật cú pháp, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. VS Code cũng có hệ thống phím tắt sinh động, trực quan giúp thao tác trở nên nhanh chóng hơn, từ đó năng suất và hiệu quả lập trình cũng được tăng lên.

Sử dụng VS Code, chúng ta hoàn toàn có thể chạy nhiều ngôn ngữ khác nhau, sử dụng các extension để hỗ trợ và dễ dàng soạn thảo, kiểm tra, gỡ lỗi. Python là một ngôn ngữ phổ biến, do vậy ngôn ngữ này cũng được VS Code hỗ trợ rất tốt và là môi trường thích hợp để viết mã nguồn cho đề tài.

Ngoài ra đề tài cũng sử dụng module auto-py-to-exe – một module dùng để đóng gói file python dựa trên thư viên **PyInstaller** - để xây dựng file thực thi trong môi trường Windows.

# THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Chương 3 đã mô tả về các công nghệ sẽ được sử dụng trong đề tài này. Trong chương 4, chúng ta sẽ đi sâu vào phần thiết kế, triển khai và đánh giá hệ thống. Kiến trúc chung của chương trình sẽ được đề cập ở mục 4.1. Mục 4.2 sẽ trình bày về các chi tiết trong chương trình. Quá trình xây dựng sẽ được nói đến ở mục 4.3 và các mục 4.4, 4.5 lần lượt là kiểm thử và triển khai hệ thống.

## Thiết kế kiến trúc

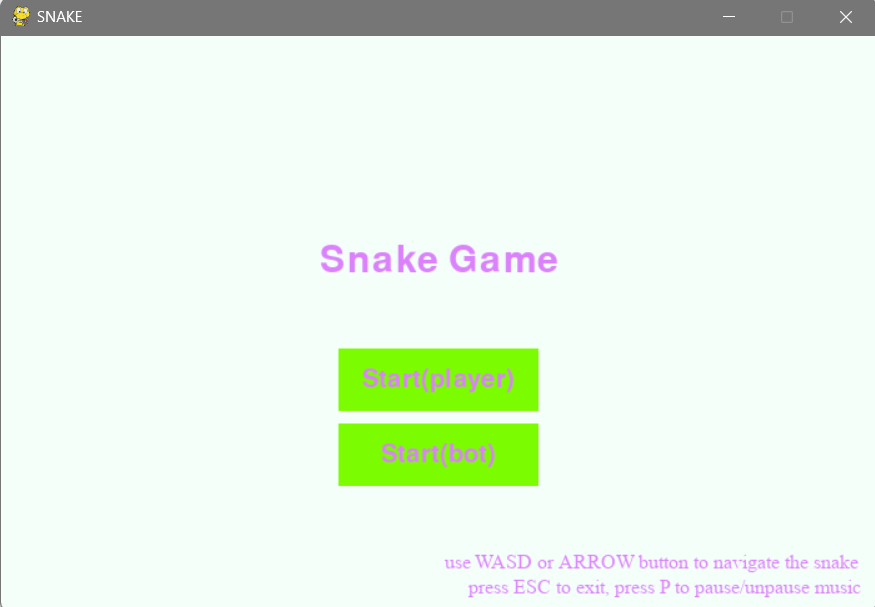
Do chương trình trò chơi rắn săn mồi tương đối đơn giản, cùng với đó là ngôn ngữ Python có cú pháp ngắn gọn, linh hoạt, dễ hiểu nên em đã quyết định triển khai đề tài này nằm trong duy nhất một tập tin.

Các hàm được sử dụng sẽ đều nằm trong thân của tập tin, chỉ có duy nhất một hàm chính để bắt đầu chương trình sẽ được gọi, và các hàm bổ trợ khác sẽ được gọi trong thân của hàm chính. Hai lựa chọn của người chơi cũng được tách thành hai hàm riêng biệt để chạy các tác vụ khác nhau.

## Thiết kế chi tiết

### Thiết kế giao diện

Khi chương trình bắt đầu, màn hình bắt đầu trò chơi sẽ xuất hiện, bao gồm các lựa chọn chạy chương trình, hướng dẫn điều khiển cho người chơi, được mô tả như hình 4.1.



Hình . Giao diện khởi đầu

Tiếp đến, khi người chơi lựa chọn vào một trong hai nút, giao diện sẽ chuyển thành giao diện tương ứng. Nếu người chơi chọn vào nút phía trên (tức là người chơi điều khiển) thì giao diện mới sẽ xuất hiện như hình 4.2.



Hình . Giao diện người chơi

Trong giao diện trên, chúng ta có thể thấy con rắn là một chuỗi các ô hình vuông với vị trí thân màu xanh lá cây, vị trí đầu màu xanh nước biển đậm. Các chướng ngại vật mang màu đen và thức ăn – mục tiêu di chuyển của con rắn – mang màu đỏ. Ngoài ra, phía trên bên trái màn hình biểu thị số điểm có được. Mỗi khi chướng ngại vật thay đổi, trò chơi sẽ tạm dừng để người chơi có thể định hình và yêu cầu nhấn nút để tiếp tục.

Giao diện rắn (Hình 4.3) được điều khiển bởi bot cũng mang các đặc điểm tượng tự người chơi, duy nhất khác ở chỗ rắn sẽ chạy liên tục không có khoảng dừng chờ phím vì bot đã nhận được thông tin môi trường từ các biến chứ không cần quan sát như con người.

A screenshot of a computer

Description automatically generated

Hình . Giao diện bot điều khiển

Cuối cùng, ghi trò chơi kết thúc, sẽ xuất hiện giao diện hiển thị số điểm như hình 4.4.

A screenshot of a video game

Description automatically generated

Hình . Giao diện kết thúc

### Thiết kế thuật toán

Thuật toán A\* là một giải thuật tìm kiếm trong đồ thị, được sử dụng để tìm đường đi từ một nút khởi đầu tới một nút đích cho trước. Thuật toán này sử dụng một “đánh giá heuristic” để xếp loại từng nút theo ước lượng về tuyến đường tốt nhất đi qua nút đó. Thuật toán này duyệt các nút theo thứ tự của đánh giá heuristic này. Do đó, thuật toán A\* là một ví dụ của tìm kiếm theo lựa chọn tốt nhất (best-first search). Thuật toán A\* được mô tả lần đầu vào năm 1968 bởi Peter Hart, Nils Nilsson, và Bertram Raphael. Thuật toán này có độ phức tạp thuật toán tương đối thấp và được sử dụng rộng rãi trong các ứng dụng thực tế như trò chơi điện tử, robot, và xe tự hành.

Dựa vào những ưu điểm đó, trong đề tài này, em đã lựa chọn thuật toán A\* là thuật toán để tìm đường đi cho rắn. Với mục đích tối ưu hiệu suất của chương trình, hàm đánh giá của thuật toán đã được lựa chọn như sau:

Trong đó:

Hàm heuristic được tính bằng khoảng cách Euclidean:

Với *p,q* lần lượt là vị trí của đầu rắn và thức ăn

Trong quá trình tìm kiếm của thuật toán A\*, cũng cần phải bỏ qua việc xét đến các chướng ngại vật, thành chương trình cũng như thân của rắn.

Mã giả của thuật toán:

**function** A\*(điểm\_xuất\_phát,đích)

**var** đóng:= *tập rỗng*

**var** q:= tạo\_hàng\_đợi(tạo\_đường\_đi(điểm\_xuất\_phát))

**while** q *không phải tập rỗng*

**var** p:= lấy\_phần\_tử\_đầu\_tiên(q)

**var** x:= *nút cuối cùng của p*

**if** x **in** đóng

**continue**

**if** x = đích

**return** p

*bổ sung x vào tập đóng*

**foreach** y **in** các\_đường\_đi\_tiếp\_theo(p)

đưa\_vào\_hàng\_đợi(q, y)

**return** failure

Thuật toán A\* là một thuật toán hoàn chỉnh (complete), do đó, nếu tồn tại đường đi từ rắn đến thức ăn, thuật toán sẽ luôn tìm được.

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Dưới đây là bảng mô tả các thư viện và công cụ đã được sử dụng trong đề tài này:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Mục đích** | **Công cụ** | **Phiên bản** | **Địa chỉ URL** |
| IDE lập trình | Visual Studio Code | 1.85.1 | https://code.visualstudio.com/ |
| Ngôn ngữ lập trình | Python | 3.12.1 | https://www.python.org/ |
| Thư viện lập trình game | Pygame | 2.5.2 | https://www.pygame.org |
| Thư viện đóng gói | PyInstaller | 6.3.0 | https://www.pyinstaller.org/ |

Bảng . Danh sách thư viện và công cụ sử dụng

Các công cụ được sử dụng đã góp phần không nhỏ giúp đề tài này hoàn thiện. Chương trình được soạn thảo vốn dĩ là một chương trình đơn giản, các thư viện và công cụ trên tuy ít nhưng đã đáp ứng đủ yêu cầu đầu ra của chương trình. Từ đó đã đạt được các kết quả được trình bày ở phần sau.

### Kết quả đạt được

Sau quá trình phát triển, chương trình được xây dựng đã bám sát với các thiết kế đã nêu, thực hiện đầy đủ các chức năng đã được xác định. Khối lượng mã nguồn thu được là khoảng 580 dòng code, chiếm dung lượng xấp xỉ 23 KB.

Phần mềm được sinh ra tuy đơn giản nhưng có khả năng bảo trì, cập nhật tính năng linh hoạt với cấu trúc đã thiết kế.

File thực thi duy nhất (.EXE) được build có dung lượng xấp xỉ 32 MB, bao gồm cả các thư viện và tài nguyên đính kèm.

## Kiểm thử

Sau khi chương trình đã hoàn thành, quá trình kiểm thử cũng không mất quá nhiều thời gian do chương trình chạy khá đơn giản. Tuy nhiên vẫn suất hiện một số vấn đề hiệu suất khi chương trình chạy lâu dài, tức là khi điểm càng cao, con rắn càng dài, thuật toán sẽ càng mất thời gian để tìm đường đi đến thức ăn. Ngoài ra, sẽ có một số tình huống rắn bị bịt hết các đường do thuật toán chưa đảm bảo rắn sẽ luôn sống. Do đó, cần bổ sung các thuật toán khác để khắc phục trong tương lai.

## Triển khai

Sau cùng, sản phẩm đã được đóng gói hoàn thiện thành một file thực thi duy nhất, bao gồm cả các tài nguyên đi kèm, giúp dễ dàng triển khai, di chuyển.

Toàn bộ mã nguồn của sản phẩm đã được công khai tại github cá nhân: <https://github.com/trongldwork>

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Thuật toán A\* được sử dụng trong trò chơi đã hoạt động hiệu quả, vượt trội hơn các thuật toán tìm kiếm bình thường khác như BFS, DFS,… Song, thuật toán vẫn còn đơn giản, chưa giải quyết triệt để vấn đề đường đi kế tiếp sau khi rắn đã đi theo đường đi thuật toán.

Đề tài “thiết kế trò chơi rắn săn mồi ứng dụng thuật toán tìm đường đi A\*” là một đề tài thú vị, có nhiều hướng phát triển. Sau quá trình nghiên cứu, tìm hiểu và xây dựng, em đã học được thêm nhiều kiến thức mới, trau dồi kiến thức lập trình của bản thân và vận dụng được các kiến thức đã học trên lớp vào các dự án của bản thân.

Tuy nhiên, do hiểu biết còn hạn chế, đề tài tuy đã hoàn thành nhưng vẫn còn sơ sài, đơn giản. Em sẽ cố gắng để cải tiến và ngày một hoàn thiện hơn.

## Hướng phát triển

Trong tương lai, em sẽ bổ sung các ràng buộc, các luật lệ, thêm các tính năng để trò chơi trở nên sinh động hơn và không bị nhàm chán. Đồng thời, sẽ xem xét sử dụng các công cụ lập trình game khác để trò chơi trở nên sinh động hơn. Thuật toán A\* cũng cần phải kết hợp cùng các giải thuật khác như forward checking, các thuật toán tìm kiếm với tri thức bổ sung,… để nâng cao hiệu suất chương trình. Tốc độ tính toán của chương trình trong không gian trò chơi lớn cũng là vấn đề cần quan tâm, có thể giải quyết bằng cách migrate trò chơi sang các ngôn ngữ lập trình khác như C#, C++, Rust,…