

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: XÂY DỰNG TRÒ CHƠI RẴN SẴN MÔI

Giáo viên hướng dẫn: TS Bùi Hải Phong

Sinh viên thực hiện:

STT	MÃ SV	HỌ VÀ TÊN	LỚP
1	1571020133	Nguyễn Hữu Khanh	CNTT 15 - 02
2	1571020268	Phạm Lê Tú	CNTT 15 - 02
3	1571020278	Nông Thế Vương	CNTT 15 - 02
4	1571020273	Nguyễn Quốc Việt	CNTT 15 - 02
5	1571020075	Vũ Minh Đức	CNTT 15 - 02

Hà Nội, tháng 3 năm 2023

MỤC LỤC

MỤC LỤC	2
LỜI CẢM ƠN.....	4
CHƯƠNG 1: MỞ ĐẦU.....	5
1.1. Lý do chọn đề tài.....	5
1.2. Mục tiêu và đối tượng nghiên cứu.....	5
1.3. Phương pháp thực hiện	6
1.4. Ý nghĩa khoa học và thực tiễn của đề tài.....	6
CHƯƠNG 2: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO.....	7
2.1. Khái niệm về AI.....	7
2.2. Mục đích của AI.	7
2.3. Những đóng góp của AI trong các lĩnh vực trong cuộc sống.....	7
2.4. Một số ứng dụng AI.....	8
2.5. Phân biệt AI, Machine Learning và Deep Learning	8
2.5.1. AI là gì?.....	9
2.5.2. Machine Learning là gì?	9
2.5.3. Deep Learning là gì?	9
CHƯƠNG 3: PHÂN TÍCH VÀ Ý TƯỞNG HÓA BÀI TOÁN.....	11
3.1. Phân tích yêu cầu của game.....	11
3.2. Lựa chọn thuật toán cho game	11
3.1. Ý tưởng hóa bài toán.....	12
3.1.1. Xác định được đối tượng trong game.	12
3.1.2. Trừu tượng hóa các đối tượng.....	13
3.1.3. Xác định các biến và các sự kiện cần sử dụng trong WinForm.....	15
3.1.4. Xây dựng các tính năng.....	16
CHƯƠNG 4: CÀI ĐẶT CHƯƠNG TRÌNH VÀ GIAO DIỆN TRÒ CHƠI.....	17

4.1. Sử lý sự kiện điều khiển con trỏ lên xuống:	17
4.2. Sử lý sự kiện bắt đầu game	18
4.3. Xử lý sự kiện chụp ảnh màn hình với game.	18
4.4. Xử lý sự kiện hết giờ.....	19
4.5. Xử lý sự kiện khởi động lại game.....	20
4.6. Sử lý sự kiện rắn ăn mồi	21
4.7. Sử lý sự kiện kết thúc game.....	21
4.8. Giao diện trò chơi	22
CHƯƠNG 5: KẾT LUẬN.....	24
DANH MỤC TÀI LIỆU THAM KHẢO	25

LỜI CẢM ƠN

Trước tiên, chúng em xin được gửi lời cảm ơn chân thành và sâu sắc đến thầy TS Bùi Hải Phong đã dẫn dắt luôn quan tâm, hỗ trợ và chỉ bảo chúng em trong suốt quá trình thực hiện báo cáo game rắn sắn môi. Nhờ có sự hướng dẫn tận tâm của thầy, chúng em đã có cơ hội được tiếp cận và nâng cao những kiến thức và kỹ năng thiết yếu về lập trình game.

Chúng em cũng xin gửi lời cảm ơn đến các bạn trong nhóm đã cùng nhau nỗ lực và phối hợp tốt để hoàn thành báo cáo. Chúng em rất vinh dự khi được làm việc với các bạn, những người có tinh thần học hỏi, sáng tạo và trách nhiệm cao.

Tuy nhiên, do kiến thức và kinh nghiệm của chúng em còn nhiều hạn chế, báo cáo game rắn sắn môi không tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý và chỉnh sửa của thầy và các bạn để hoàn thiện hơn.

Xin chân thành cảm ơn!

CHƯƠNG 1: MỞ ĐẦU

1.1. Lý do chọn đề tài.

Con người luôn mơ ước sáng tạo ra một cỗ máy có khả năng tự nghĩ và hành động. Do đó, trí tuệ nhân tạo (AI) là một lĩnh vực mới và hấp dẫn trong kỷ nguyên công nghệ thông tin hiện đại. Các ngành nghề trong xã hội đang áp dụng AI để làm cho máy móc thông minh hơn.

Trong giải trí, trong y tế, hay cả trong giáo dục, rất nhiều robot thông minh được phát triển có thể thay thế con người dạy học, chăm sóc người già, thực hiện các công đoạn trong công nghiệp,... Vì những ứng dụng bổ ích của nó, nhóm em chọn đề tài “Xây dựng trò chơi rắn săn mồi” dựa trên các thuật toán cơ bản hay dùng trong trí tuệ nhân tạo. Game rắn săn mồi là một trò chơi điện tử cổ điển, được phát triển lần đầu tiên vào năm 1976 dưới tên Blockade. Trò chơi này có cách chơi đơn giản nhưng hấp dẫn: người chơi phải điều khiển một con rắn di chuyển trên màn hình và ăn các thức ăn để tăng chiều dài. Tuy nhiên, con rắn không được va chạm vào biên của màn hình hoặc vào thân của chính nó, nếu không sẽ thua cuộc. Trò chơi này đã được tái hiện trên nhiều nền tảng khác nhau, từ máy tính cá nhân đến điện thoại di động.

Nhóm em đã quyết định chọn đề tài “Xây dựng trò chơi rắn săn mồi” vì nhóm em yêu thích trò chơi này và muốn hiểu rõ hơn về cách thiết kế và lập trình game bằng ngôn ngữ C#. Đồng thời, nhóm em cũng muốn áp dụng các kiến thức và kỹ thuật về trí tuệ nhân tạo để làm cho game có tính năng mới và thú vị hơn. Với mong muốn thông qua game này, chúng em sẽ có thể có những cái nhìn rõ ràng về đề tài AI, từ đó mở rộng để nghiên cứu và phát triển thêm.

1.2. Mục tiêu và đối tượng nghiên cứu

Đề tài này nhằm mục đích thiết kế và lập trình game rắn săn mồi bằng ngôn ngữ C#, với sự áp dụng của các kiến thức và kỹ thuật về trí tuệ nhân tạo. Để làm được điều này, nhóm em đã nghiên cứu về lịch sử và thuật toán của game rắn săn mồi, cũng như các công cụ và thư viện hỗ trợ cho việc lập trình game. Nhóm em đã áp dụng các kiến thức về .NET để xây dựng giao diện và logic của game. Nhóm em cũng đã sử dụng các kỹ thuật về trí tuệ nhân tạo để thiết kế con rắn thông minh có khả năng tự động di chuyển và ăn thức ăn trong game.

Mục tiêu của đề tài là thiết kế và lập trình game rắn sẵn môi bằng ngôn ngữ ... (tùy theo yêu cầu của giáo viên), với sự áp dụng của các kiến thức và kỹ thuật về trí tuệ nhân tạo. Đối tượng nghiên cứu là các thuật toán liên quan đến game rắn sẵn môi và trí tuệ nhân tạo.

1.3. Phương pháp thực hiện

Nhóm em đã sử dụng các phương pháp sau để thực hiện đề tài:

Nghiên cứu lý thuyết: Nhóm em đã tổng hợp và phân tích các kiến thức về lịch sử và thuật toán của game rắn sẵn môi, cũng như các công cụ và thư viện hỗ trợ cho việc lập trình game. Nhóm em cũng đã nghiên cứu các kỹ thuật về trí tuệ nhân tạo để thiết kế con rắn thông minh có khả năng tự động di chuyển và ăn thức ăn trong game.

Thiết kế game: Nhóm em đã xây dựng bản thiết kế chi tiết cho game, bao gồm các yếu tố sau: giao diện người dùng, âm thanh, luật chơi, tính điểm, con rắn thông minh.

Lập trình game: Nhóm em đã áp dụng các kiến thức về C# để viết mã nguồn cho game theo bản thiết kế đã có. Nhóm em đã sử dụng công cụ Visual Studio 2022 để biên soạn và kiểm tra mã nguồn.

Kiểm tra và đánh giá game: Nhóm em đã tiến hành kiểm tra hoạt động của game trên nhiều nền tảng khác nhau, từ máy tính cá nhân đến điện thoại di động. Nhóm em cũng đã thu thập ý kiến phản hồi từ người dùng để đánh giá mức độ hấp dẫn và khả thi của game. Nhóm em đã sử dụng các tiêu chí sau để đánh giá game: tính hoạt động, tính hấp dẫn và tính khả thi.

1.4. Ý nghĩa khoa học và thực tiễn của đề tài

Về mặt khoa học: Đề tài này giúp nhóm em nâng cao kiến thức về lập trình game bằng ngôn ngữ C#, cũng như về các kỹ thuật trí tuệ nhân tạo liên quan. Đề tài này cũng góp phần vào việc nghiên cứu và phát triển các ứng dụng của trí tuệ nhân tạo trong lĩnh vực giải trí.

Về mặt thực tiễn: Đề tài này mang lại sản phẩm là game rắn sẵn môi có tính năng mới và thú vị hơn so với các phiên bản trước. Game có thể chạy được trên nhiều nền tảng khác nhau, từ máy tính cá nhân đến điện thoại di động. Game có thể thu hút được sự quan tâm và tham gia của người dùng, góp phần vào việc giải trí và rèn luyện kỹ năng suy luận cho người chơi.

CHƯƠNG 2: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

2.1. Khái niệm về AI

AI - Artificial Intelligence hay còn gọi là Trí tuệ nhân tạo là một ngành khoa học, kỹ thuật chế tạo máy móc thông minh, đặc biệt là các chương trình máy tính thông minh.

AI được thực hiện bằng cách nghiên cứu cách suy nghĩ của con người, cách con người học hỏi, quyết định và làm việc trong khi giải quyết một vấn đề nào đó, và sử dụng những kết quả nghiên cứu này như một nền tảng để phát triển các phần mềm và hệ thống thông minh, từ đó áp dụng vào các mục đích khác nhau trong cuộc sống. Nói một cách dễ hiểu thì AI là việc sử dụng, phân tích các dữ liệu đầu vào nhằm đưa ra sự dự đoán rồi đi đến quyết định cuối cùng.

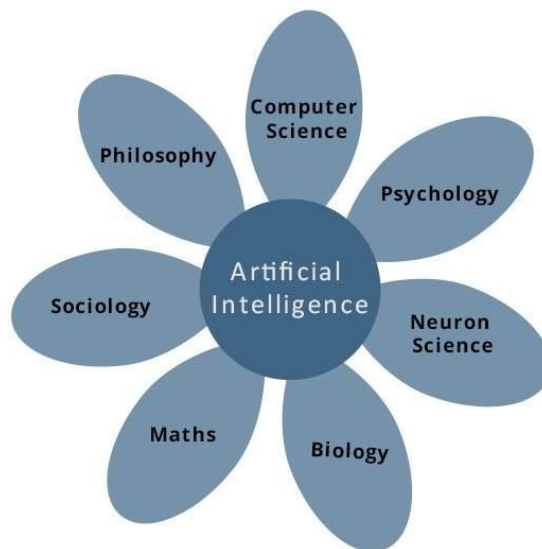
2.2. Mục đích của AI.

Tạo ra các hệ thống chuyên gia - là các ứng dụng máy tính được phát triển để giải quyết các vấn đề phức tạp trong một lĩnh vực cụ thể, ở mức độ thông minh và chuyên môn của con người.

Thực hiện trí thông minh của con người trong máy móc - Tạo ra các hệ thống có thể hiểu, suy nghĩ, học hỏi và hành xử như con người.

2.3. Những đóng góp của AI trong các lĩnh vực trong cuộc sống.

AI là một ngành khoa học và công nghệ dựa trên nhiều ngành khác như Khoa học máy tính, Toán học, Sinh học, Kỹ thuật...



Ảnh 1. Những đóng góp của AI trong các lĩnh vực trong cuộc sống

2.4. Một số ứng dụng AI

Ví dụ ứng dụng trong lĩnh vực y tế:

Quản trị: Các hệ thống AI trợ giúp các công việc hành chính hàng ngày, để giảm thiểu lỗi của con người và tối đa hóa hiệu quả.

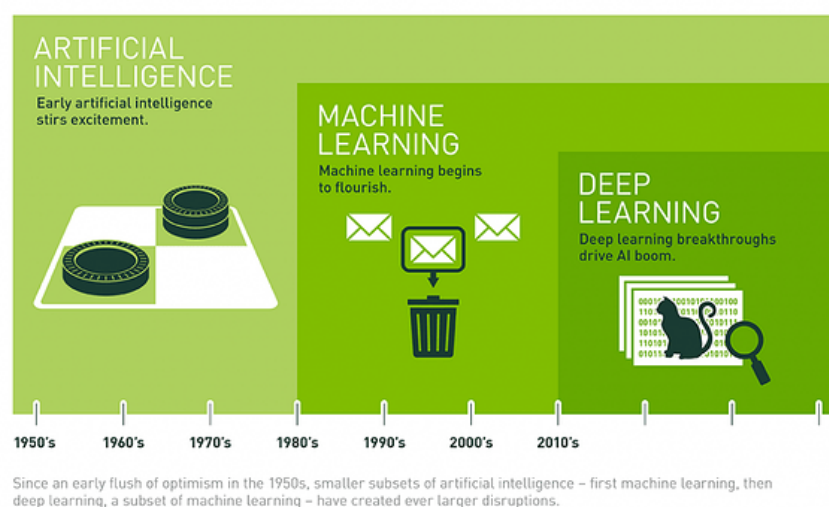
Điều trị từ xa: Đối với các tình huống không khẩn cấp, bệnh nhân có thể liên hệ với hệ thống AI của bệnh viện để phân tích các triệu chứng của họ, nhập các dấu hiệu quan trọng của họ và đánh giá xem có cần phải chăm sóc y tế hay không. Điều này làm giảm khối lượng công việc của các chuyên gia y tế bằng cách chỉ đưa các trường hợp quan trọng đến họ.

Hỗ trợ chuẩn đoán: Thông qua thị giác máy tính và mạng lưới thần kinh tích chập, AI hiện có khả năng đọc quét hình ảnh cộng hưởng từ để kiểm tra khối u và sự phát triển ác tính khác của nó, với tốc độ nhanh hơn so với các bác sĩ x-quang và sai số thấp hơn đáng kể.

Phẫu thuật có sự trợ giúp của robot: Robot phẫu thuật có sai số rất nhỏ và có thể thực hiện phẫu thuật suốt ngày đêm mà không bị kiệt sức.

Giám sát các chỉ số quan trọng. Ngoài ra còn rất nhiều những ứng dụng trong các lĩnh vực khác trong đời sống như nhận diện khuôn mặt, nhận diện giọng nói, ô tô tự lái...

2.5. Phân biệt AI, Machine Learning và Deep Learning



Ảnh 2. Phân biệt AI, Machine Learning và Deep Learning

2.5.1. AI là gì?

Trí tuệ nhân tạo là trí tuệ máy móc được tạo ra bởi con người. Trí tuệ này có thể tư duy, suy nghĩ, học hỏi,... như con người. Xử lý dữ liệu ở mức độ rộng hơn, quy mô hơn, hệ thống, khoa học và nhanh hơn so với con người.

AI có ba mức độ khác nhau:

- **Narrow AI:** Trí tuệ nhân tạo được cho là hẹp khi máy có thể thực hiện một nhiệm vụ cụ thể tốt hơn so với con người. Nghiên cứu hiện tại về AI hiện đang ở cấp độ này.
- **General AI:** Trí tuệ nhân tạo đạt đến trạng thái chung khi nó có thể thực hiện bất kỳ nhiệm vụ sử dụng trí tuệ nào có cùng độ chính xác như con người.
- **Strong AI:** AI rất mạnh khi nó có thể đánh bại con người trong nhiều nhiệm vụ cụ thể.

2.5.2. Machine Learning là gì?

Machine Learning còn được gọi là học máy. Bạn có thể viết ứng dụng có AI mà không sử dụng học máy, nhưng bạn phải viết cả triệu triệu dòng code để xây dựng các trường hợp xảy ra.

Học máy là cách để có được AI, máy tự học mà không cần phải code nhiều như không có học máy. Nói cách khác, nếu AI là mục tiêu thì học máy là phương tiện để đạt được mục tiêu đó.

Máy sẽ được “học” bằng cách train nó với một lượng dữ liệu khổng lồ với một thuật toán, thuật toán có khả năng điều chỉnh và xây dựng model. Tuy nhiên, nếu như trong training dữ liệu có ngôn ngữ khác trong thực tế (tiếng Việt thay vì tiếng Anh...) thì rất có thể máy sẽ dự báo không chính xác nữa.

2.5.3. Deep Learning là gì?

Deep Learning được bắt nguồn từ thuật toán Neural network của AI, là một ngành nhỏ của Machine Learning.

Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.

Trí tuệ nhân tạo có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, Machine learning nằm ở tầng tiếp theo và Deep learning nằm ở tầng trên cùng.

CHƯƠNG 3: PHÂN TÍCH VÀ Ý TƯỞNG HÓA BÀI TOÁN

3.1. Phân tích yêu cầu của game

Game rắn săn mồi là một trò chơi giải trí kinh điển với nhiệm vụ của người chơi là điều khiển con rắn di chuyển trong bản đồ, ăn các loại trái cây, bánh ngọt, kẹo ngọt để trở nên to lớn hơn và thôn tính các con rắn khác.

Yêu cầu chức năng: là những gì game có thể làm được cho người dùng, ví dụ như:

- Cho phép người dùng bắt đầu và kết thúc game
- Cho phép người dùng điều khiển hướng di chuyển của con rắn
- Cho phép người dùng ăn các loại thức ăn để tăng kích thước và điểm số
- Cho phép người dùng xem bảng xếp hạng và lưu lại kết quả

Yêu cầu phi chức năng: là những gì liên quan đến hiệu suất, độ tin cậy, khả năng sử dụng và an ninh của game, ví dụ như:

- Game có giao diện đẹp mắt và âm thanh sống động
- Game có tốc độ xử lý nhanh và không bị giật lag
- Game có hướng dẫn chi tiết cho người dùng mới
- Game không chứa virus hay mã độc

3.2. Lựa chọn thuật toán cho game

Game rắn săn mồi là một trò chơi giải trí kinh điển với nhiệm vụ của người chơi là điều khiển con rắn di chuyển trong bản đồ, ăn các loại thức ăn để tăng kích thước và điểm số và thôn tính các con rắn khác. Để lựa chọn thuật toán cho game rắn săn mồi, chúng ta cần xem xét các yếu tố sau:

Thuật toán phải đảm bảo được tính logic và hợp lý của game, ví dụ như:

- Con rắn không được di chuyển ngược lại hướng hiện tại
- Con rắn phải tự động di chuyển theo hướng đã quy định
- Con rắn phải dừng lại khi va vào biên hoặc vào thân của mình
- Con rắn phải tăng kích thước và điểm số khi ăn được thức ăn

Thuật toán phải đảm bảo được hiệu suất và khả năng mở rộng của game, ví dụ như:

- Thuật toán không gây ra hiện tượng giật lag hay treo máy
- Thuật toán có thể áp dụng cho nhiều cấp độ khó khác nhau
- Thuật toán có thể bổ sung thêm các tính năng mới cho game

Một trong những thuật toán phổ biến để lập trình game rắn sắn môi là sử dụng danh sách liên kết (linked list) để lưu trữ vị trí của các ô vuông tạo nên con rắn. Danh sách liên kết là một cấu trúc dữ liệu gồm các nút (node) liên kết với nhau bằng các con trỏ (pointer). Mỗi nút sẽ lưu giữ thông tin về vị trí x, y và hướng di chuyển của ô vuông đó. Đầu danh sách sẽ là đầu của con rắn, cuối danh sách sẽ là đuôi của con rắn.

Để di chuyển con rắn, ta chỉ cần xoá nút cuối danh sách (đuôi) và thêm một nút mới vào đầu danh sách (đầu) với vị trí và hướng di chuyển mới. Để kiểm tra va chạm, ta chỉ cần so sánh vị trí của nút đầu danh sách (đầu) với biên hoặc các nút còn lại trong danh sách (thân). Để tăng kích thước con rắn khi ăn được thức ăn, ta chỉ cần không xoá nút cuối danh sách (đuôi) khi di chuyển.

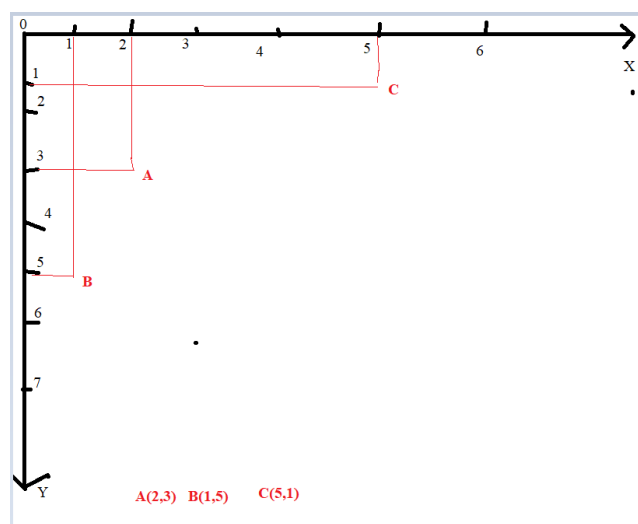
Khai báo class Circle để biểu diễn ô vuông.

Khai báo class Settings để biểu diễn con rắn và đường đi của rắn.

3.1. Ý tưởng hóa bài toán

3.1.1. Xác định được đối tượng trong game.

- **Hệ tọa độ màn hình:** Để có thể triển khai code, trước tiên chúng ta cần phân tích trò chơi này, xem nó có những đối tượng nào. Mỗi đối tượng có những hành vi gì,... Bạn cần nắm được hệ tọa độ của màn hình đồ họa, bởi chúng ta làm việc với nó rất nhiều. Bạn muốn vẽ cái gì lên màn hình thì đều cần chỉ định tọa độ mà bạn cần vẽ.



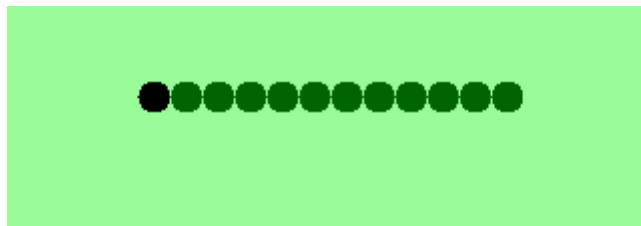
Hệ tọa độ màn hình trong game rắn săn mồi có những đặc điểm sau:

- Hệ tọa độ là hệ Oxy với gốc O nằm ở góc trên bên trái của màn hình.
- Trục Ox có chiều hướng sang phải và trục Oy có chiều hướng xuống dưới.
- Đơn vị đo của hệ tọa độ là pixel, tức là khoảng cách giữa hai điểm liên kề trên màn hình bằng 1 pixel.
- Tọa độ của một điểm được biểu diễn bằng cặp số (x,y) với x là khoảng cách từ điểm đó đến trục Oy và y là khoảng cách từ điểm đó đến trục Ox.

Ví dụ: Tọa độ của góc dưới bên phải của màn hình là $(width - 1, height - 1)$ với width và height là chiều rộng và chiều cao của màn hình tính bằng pixel.

- **Phân tích đối tượng rắn trong game**

Con rắn săn mồi của chúng ta sẽ là một chuỗi các hình tròn nhỏ nối lại với nhau (số hình tròn nhỏ chính là độ dài của con rắn). Khởi tạo trò chơi, chúng ta có thể đặt độ dài ban đầu của nó là 3. Trong quá trình trò chơi diễn ra, ta phải lưu vết được tọa độ của từng hình tròn đó. Đối tượng con rắn có thể mô phỏng là chuỗi các hình tròn nhỏ.



Tại mỗi bước dịch chuyển của rắn, mỗi đốt thân của rắn sẽ dịch chuyển đi 1 đơn vị độ dài bằng nhau. Trong đó, đốt thân đầu tiên (đầu của rắn) sẽ tiến lên theo hướng dịch chuyển, các đốt thân phía sau di chuyển đến vị trí cũ của đốt thân phía trước nó. Ví dụ:

Giả sử con rắn có 3 đốt và tọa độ của nó hiện tại là: $x1(3,0)$ – đầu, $x2(2,0)$ và $x3(1,0)$ và đang đi theo hướng trục Ox. Bây giờ rắn đổi hướng di chuyển sang bên trái. Khi đó tọa độ mới của từng đốt là: $x1(3,-1)$, $x2$ sẽ là tọa độ của $x1$ cũ $(3, 0)$ và $x3$ chuyển sang vị trí của $x2$ cũ là $(2, 0)$.

3.1.2. Trừu tượng hóa các đối tượng

Như vậy, ta sẽ xây dựng một đối tượng Circle. Đối tượng này giúp ta lưu được tọa độ của một điểm trên trục tọa độ 2 chiều Oxy. Ta có cấu trúc Circle như sau:

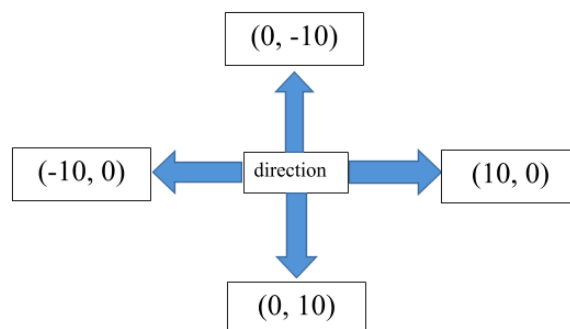
```

internal class Circle
{
    public int X { get; set; }
    public int Y { get; set; }
    public Circle()
    {
        X = 0;
        Y = 0;
    }
}

```

Khi đó:

- Một con rắn sẽ là 1 mảng các đối tượng Circle tương ứng với các đốt của rắn. Tại mỗi đốt, cặp tọa độ (x,y) sẽ lưu vị trí đốt hiện tại và cặp tọa độ (x0,y0) sẽ lưu vị trí trước đó của đốt hiện tại để các đốt sau đó của con rắn có thể sử dụng.
- Mỗi đối tượng thức ăn sẽ là 1 đối tượng Circle. Ta chỉ cần sử dụng cặp biến (x, y) để lưu 1 đối tượng thức ăn. Tại một thời điểm trên màn hình chỉ có 1 thức ăn, và thức ăn đó xuất hiện ở 1 vị trí ngẫu nhiên bất kỳ.
- Để rắn di chuyển được trên màn hình thì ta cần thêm một biến lưu hướng đi của nó. Ta sẽ tận dụng luôn đối tượng Circle để xác định hướng theo tọa độ (x,y). Ví dụ nếu rắn đang di chuyển theo hướng trái sang phải, như vậy đối tượng direction của ta sẽ là (10, 0). Tức là ở mỗi bước đi, tọa độ x sẽ tăng thêm 10 đơn vị và tọa độ y không đổi. Khi ta thay đổi hướng đi, ta chỉ cần thay đổi giá trị (x, y) của đối tượng direction như hình dưới đây:



Mô phỏng hướng đi của rắn, tại mỗi bước đi ta sẽ cho Circle đầu của rắn cộng thêm với Circle hướng đi của nó. Các bạn lưu ý rằng trục Oy trong Console của chúng ta hướng xuống dưới nhé (Tức là y tăng khi xuống dưới và giảm khi đi lên trên).

Rắn sẽ ăn được thức ăn khi tọa độ đầu snake[0] trùng với tọa độ food. Khi đó ta cần:

1. Tăng chiều dài của rắn
2. Khởi tạo ngẫu nhiên tọa độ thức ăn mới – Lưu ý không khởi tạo trùng vào các đốt thân của rắn.
3. Tăng điểm số.

Nếu đầu Rắn chạm chạm thân thì game kết thúc và chỉ chết khi chạm vào thân.

3.1.3. Xác định các biến và các sự kiện cần sử dụng trong WinForm

Các biến được sử dụng:

```
int maxWidth;  
int maxHeight;  
int score;  
int highScore;  
bool goLeft, goRight, goDown, goUp;
```

Các sự kiện cần sử dụng:

```
private void KeyIsDown(object sender, KeyEventArgs e)  
private void KeyIsUp(object sender, KeyEventArgs e)  
private void StartGame(object sender, EventArgs e)  
private void TakeSnapShot(object sender, EventArgs e)  
private void GameTimerEvent(object sender, EventArgs e)  
private void UpdatePictureBoxGraphics(object sender, PaintEventArgs e)  
private void RestartGame()
```

```
private void EatFood()  
  
private void GameOver()
```

3.1.4. Xây dựng các tính năng

Sử dụng các thư viện hoặc framework phù hợp với ngôn ngữ lập trình bạn chọn để vẽ và di chuyển các đối tượng trên màn hình.

Xử lý các sự kiện từ bàn phím hoặc chuột để điều khiển con rắn và thay đổi hướng di chuyển của nó.

Xử lý va chạm giữa con rắn và các đối tượng khác, như thức ăn, biên của màn hình hoặc chính bản thân con rắn.

Cập nhật điểm số khi con rắn ăn được thức ăn và hiển thị nó trên màn hình.

Phát âm thanh khi con rắn ăn được thức ăn hoặc bị thua cuộc.

Tạo ra các mức độ khó khác nhau cho game bằng cách điều chỉnh kích thước của màn hình, số lượng và vị trí của thức ăn, tốc độ di chuyển của con rắn...

Tùy biến giao diện và thiết kế của game bằng cách sử dụng các màu sắc, hình ảnh, font chữ khác nhau.

CHƯƠNG 4: CÀI ĐẶT CHƯƠNG TRÌNH VÀ GIAO DIỆN TRÒ CHƠI

4.1. Sử lý sự kiện điều khiển con trỏ lên xuống:

Con trỏ xuống:

```
private void KeyIsDown(object sender, EventArgs e)
{
    if (e.KeyCode == Keys.Left && Settings.directions != "right")
    {
        goLeft = true;
    }
    if (e.KeyCode == Keys.Right && Settings.directions != "left")
    {
        goRight = true;
    }
    if (e.KeyCode == Keys.Up && Settings.directions != "down")
    {
        goUp = true;
    }
    if (e.KeyCode == Keys.Down && Settings.directions != "up")
    {
        goDown = true;
    }
}
```

Con trỏ lên:

```
private void KeyIsUp(object sender, EventArgs e)
{
    if (e.KeyCode == Keys.Left)
    {
        goLeft = false;
    }
    if (e.KeyCode == Keys.Right)
    {
```

```

        goRight = false;
    }
    if (e.KeyCode == Keys.Up)
    {
        goUp = false;
    }
    if (e.KeyCode == Keys.Down)
    {
        goDown = false;
    }
}

```

4.2. Xử lý sự kiện bắt đầu game

```

private void StartGame(object sender, EventArgs e)
{
    RestartGame();
}

```

4.3. Xử lý sự kiện chụp ảnh màn hình với game.

```

private void TakeSnapshot(object sender, EventArgs e)
{
    Label caption = new Label();
    caption.Text = "I scored: " + score + " and my Highscore is " +
highScore + " on the SnakesOfPrey By Group 5";
    caption.Font = new Font("Ariel", 12, FontStyle.Bold);
    caption.ForeColor = Color.Purple;
    caption.AutoSize = false;
    caption.Width = picCanvas.Width;
    caption.Height = 30;
    caption.TextAlign = ContentAlignment.MiddleCenter;
    picCanvas.Controls.Add(caption);
}

```

```

SaveFileDialog dialog = new SaveFileDialog();
dialog.FileName = "SnakesOfPrey by G5";
dialog.DefaultExt = ".jpg";
dialog.Filter = "JPG Image File | *.jpg";
dialog.ValidateNames = true;

if (dialog.ShowDialog() == DialogResult.OK)
{
    int width = Convert.ToInt32(picCanvas.Width);
    int height = Convert.ToInt32(picCanvas.Height);
    Bitmap bmp = new Bitmap(width, height);
    picCanvas.DrawToBitmap(bmp, new Rectangle(0, 0, width,
height));
    bmp.Save(dialog.FileName, ImageFormat.Jpeg);
    picCanvas.Controls.Remove(caption);
}
}

```

4.4. Xử lý sự kiện hết giờ.

<pre> Private void GameTimerEvent(object sender, EventArgs e) { if (goLeft) { Settings.directions = "left"; } if (goRight) { Settings.directions = "right"; } if (goDown) { Settings.directions = "down"; } </pre>	<pre> if (Snake[i].X < 0) { Snake[i].X = maxWidth; } if (Snake[i].X > maxWidth) { Snake[i].X = 0; } if (Snake[i].Y < 0) { Snake[i].Y = maxHeight; } if (Snake[i].Y > maxHeight) { Snake[i].Y = 0; } </pre>
--	--

<pre> if (goUp) { Settings.directions = "up"; } for (int i = Snake.Count - 1; i >= 0; i--) { if (i == 0) { switch (Settings.directions) { case "left": Snake[i].X--; break; case "right": Snake[i].X++; break; case "down": Snake[i].Y++; break; case "up": Snake[i].Y--; break; } } } </pre>	<pre> } if (Snake[i].X == food.X && Snake[i].Y == food.Y) { EatFood(); } for (int j = 1; j < Snake.Count; j++) { if (Snake[i].X == Snake[j].X && Snake[i].Y == Snake[j].Y) { GameOver(); } } else { Snake[i].X = Snake[i - 1].X; Snake[i].Y = Snake[i - 1].Y; } } picCanvas.Invalidate(); </pre>
--	---

4.5. Xử lý sự kiện khởi động lại game

```

private void RestartGame()
{
    maxWidth = picCanvas.Width / Settings.Width - 1;
    maxHeight = picCanvas.Height / Settings.Height - 1;
    Snake.Clear();
    startButton.Enabled = false;
    snapButton.Enabled = false;
}

```

```

score = 0;
txtScore.Text = "Score: " + score;
Circle head = new Circle { X = 10, Y = 5 };
Snake.Add(head); // adding the head part of the snake to the list
for (int i = 0; i < 10; i++)
{
    Circle body = new Circle();
    Snake.Add(body);
}
food = new Circle { X = rand.Next(2, maxWidth), Y =
rand.Next(2, maxHeight) };
gameTimer.Start();
}

```

4.6. Sử lý sự kiện rắn ăn mồi

```

private void EatFood()
{
    score += 1;

    txtScore.Text = "Score: " + score;

    Circle body = new Circle
    {
        X = Snake[Snake.Count - 1].X,
        Y = Snake[Snake.Count - 1].Y
    };

    Snake.Add(body);

    food = new Circle { X = rand.Next(2, maxWidth), Y =
rand.Next(2, maxHeight) };
}

```

4.7. Sử lý sự kiện kết thúc game

```

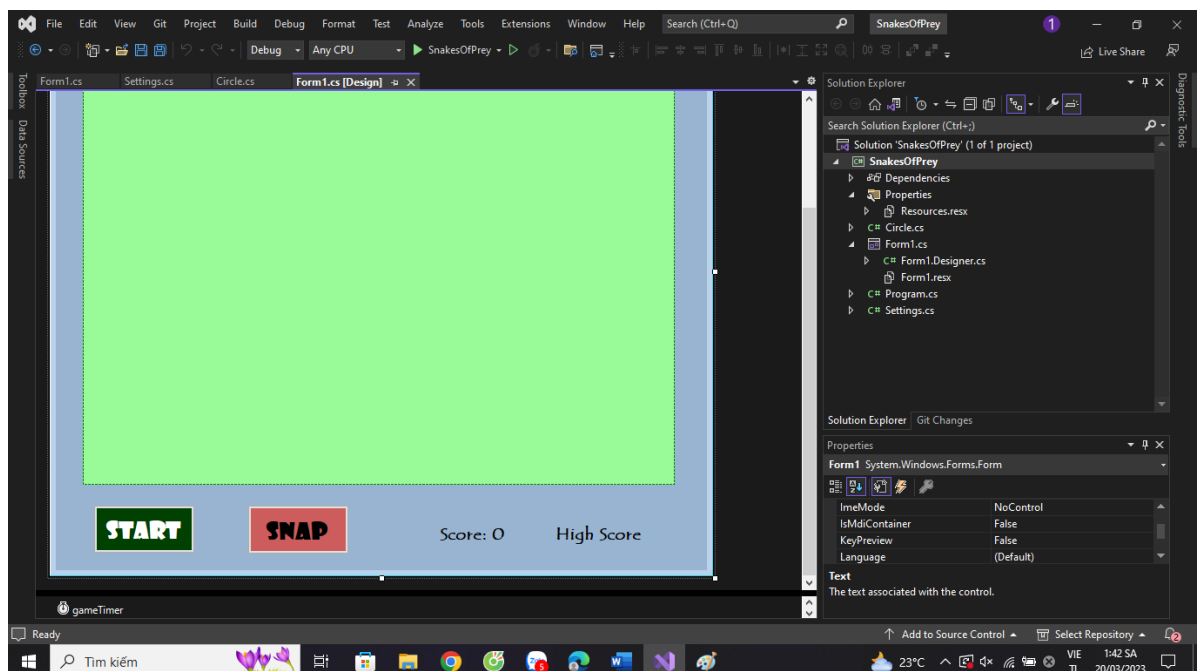
private void GameOver()
{
    gameTimer.Stop();
    startButton.Enabled = true;
    snapButton.Enabled = true;

    if (score > highScore)
    {
        highScore = score;

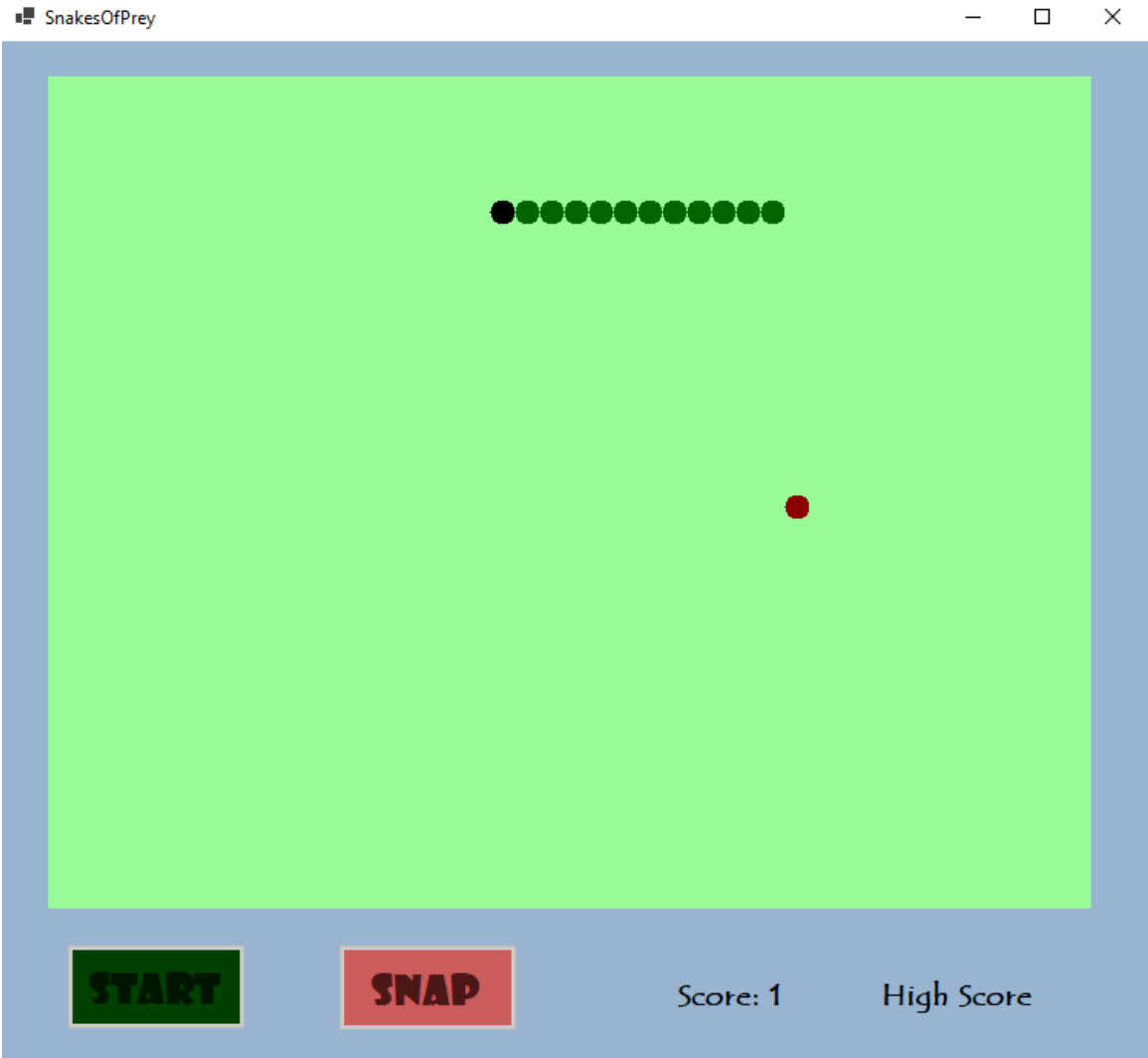
        txtHighScore.Text = "High Score: " + Environment.NewLine +
highScore;
        txtHighScore.ForeColor = Color.Maroon;
        txtHighScore.TextAlign = ContentAlignment.MiddleCenter;
    }
}

```

4.8. Giao diện trò chơi



Giao diện trò chơi khi đã debug:



CHƯƠNG 5: KẾT LUẬN

Trong báo cáo này, nhóm em đã trình bày về quá trình thiết kế và lập trình game rắn sẵn môi bằng ngôn ngữ C#. Nhóm em đã nghiên cứu về lịch sử và thuật toán của game rắn sẵn môi, cũng như các công cụ và thư viện hỗ trợ cho việc lập trình game. Nhóm em đã áp dụng các kiến thức về .NET để xây dựng giao diện và logic của game. Nhóm em cũng đã kiểm tra và đánh giá kết quả của game qua các tiêu chí như tính hoạt động, tính hấp dẫn và tính khả thi.

Kết quả cho thấy game rắn sẵn môi của nhóm em hoạt động ổn định và mượt mà trên nhiều nền tảng khác nhau. Game có giao diện đơn giản và cũng có tính khả thi cao khi có thể được triển khai và phân phối dễ dàng.

Tuy nhiên, game rắn sẵn môi của nhóm em vẫn còn một số hạn chế và điểm cần cải thiện. Một số hạn chế là: game chỉ có một chế độ chơi duy nhất; game không có tính năng lưu điểm cao hay xếp hạng; game không có tính năng tùy biến con rắn hay màn hình; game không có tính năng multiplayer hay online. Một số điểm cần cải thiện là: tăng độ khó của game theo thời gian; thêm các loại thức ăn khác nhau cho con rắn; thêm các hiệu ứng khi con rắn ăn hoặc va chạm.

Nhìn chung, nhóm em đã hoàn thành được mục tiêu của đề tài là thiết kế và lập trình game rắn sẵn môi bằng ngôn ngữ C#. Nhóm em mong muốn trong tương lai sẽ có thể phát triển thêm các tính năng mới cho game để làm cho nó hay hơn và thu hút được nhiều người chơi hơn.

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thị Thu Hà, Lập trình Windows Forms với C#, NXB Thống Kê, 2016.
- [2]. Nguyễn Văn Hùng, Lập trình C# căn bản và nâng cao, NXB Đại Học Quốc Gia Hà Nội, 2017
- [3]. Chương Trình Game Rắn Săn Mồi. Truy cập ngày 19/3/2023 từ <https://text.123docz.net/document/1435849-chuong-trinh-game-ran-san-moi.htm>
- [4]. Nguyễn Thị Hồng Nhung, Lập trình C# nâng cao, NXB Đại Học Quốc Gia TP.HCM, 2018.
- [5]. Nguyễn Văn Tú, Lập trình game với C# và Unity 3D, NXB Thông Tin và Truyền Thông, 2019.
- [6]. Nguyễn Đức Hùng, Lập trình Windows Forms với Visual Studio .NET 2019 và C#, NXB Khoa Học Tự Nhiên và Công Nghệ, 2020.
- [7]. Bùi Minh Tuấn, Lập trình game cơ bản với C#, NXB Giáo Dục Việt Nam, 2020.