

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA: CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN

XÂY DỰNG CHƯƠNG TRÌNH TRÒ CHƠI CỜ CARÔ BẰNG NGÔN NGỮ PYTHON

MÔN: LẬP TRÌNH PYTHON

Ngành: **KỸ THUẬT PHẦN MỀM**

Chuyên ngành **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: : **ĐẶNG VĂN LỰC**

Sinh viên thực hiện: **NGUYỄN TRẦN HOÀNG THỊNH**

MSSV: **22140341**

Lớp học phần: **221402**

TP. Hồ Chí Minh, 03, 2025

Khoa: CÔNG NGHỆ THÔNG TIN

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN

TIỂU LUẬN MÔN: Lập trình Python

1. Họ và tên sinh viên: Nguyễn Trần Hoàng Thịnh

2. Tên đề tài: Xây Dựng Chương Trình Trò Chơi Cờ Carô Bằng Ngôn Ngữ Python

3. Nhận xét:

a) Những kết quả đạt được:

.....
.....
.....
.....

b) Những hạn chế:

.....
.....
.....
.....

4. Điểm đánh giá (theo thang điểm 10, làm tròn đến 0.5):

Sinh viên: Nguyễn Trần Hoàng Thịnh MSSV: 22140341

Điểm số:..... Điểm chữ:

TP. HCM, ngày ... tháng 03 năm 2025

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đề tài tiểu luận: “Xây Dựng Chương Trình Trò Chơi Cờ Carô Bằng Ngôn Ngữ Python” do Nguyễn Trần Hoàng Thịnh tìm hiểu và thực hiện.

Kết quả bài làm của đề tài “Xây Dựng Chương Trình Trò Chơi Cờ Carô Bằng Ngôn Ngữ Python” là trung thực và không sao chép từ bất kì bài tập của các cá nhân khác.

TP. HCM, ... tháng 03 năm 2025.

Sinh viên cam đoan

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến thầy Đặng Văn Lực. Trong quá trình học tập và tìm hiểu môn “Lập trình Python”, em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của cô. Thầy đã giúp em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài tiểu luận về đề tài “Xây Dựng Chương Trình Trò Chơi Cờ Carô Bằng Ngôn Ngữ Python”.

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của thầy để bài tiểu luận của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ NỀN TẢNG CÔNG NGHỆ.....	1
1.1. Lý Do Chọn Đề Tài.....	1
1.2. Giới thiệu về ngôn ngữ lập trình Python.....	3
1.3. Cấu trúc của python	5
1.4. Ứng dụng của python.....	6
1.5. Đặc điểm của Python	7
1.6. Công Cụ Và Thư Viện Sử Dụng – Visual Studio	8
1.7. Lý do sử dụng Visual Studio vào dự án.....	10
1.8. Tkinter – Thư viện xây dựng giao diện.....	13
1.9. Ứng dụng Tkinter vào dự án cờ Caro	14
1.10. Giới thiệu trò chơi cờ Caro	15
1.11. Quy luật trò chơi Cờ Caro.....	17
1.12. Cách chơi và mục tiêu.....	18
1.13. Các quy tắc quan trọng trong cách tính thắng thua.....	18
1.14. Chiến thuật và phong cách chơi	19
1.15. Trường hợp đặc biệt và kết thúc ván đấu.....	20
CHƯƠNG 2: PHÂN TÍCH VÀ GIẢI PHÁP CHO TRÒ CHƠI.....	20
2.1. Xác định bài toán và mục tiêu.....	20
2.1.1. Mục tiêu của trò chơi	20
2.1.2. Đối tượng chính	22
2.1.3. Giới thiệu về AI và thuật toán minimax.....	23
CHƯƠNG 3: PHÁT TRIỂN VÀ MÔ TẢ TÍNH NĂNG TRÒ CHƠI.....	32
3.1. Phát triển đề tài	32
3.2. Chạy thử trò chơi, giao diện trò chơi	33
3.3. Những thiếu sót của dự án	57
3.4. Giải pháp cải thiện và nâng cấp trong tương lai	58
CHƯƠNG 4: TỔNG KẾT VÀ ĐỊNH HƯỚNG TƯƠNG LAI	59
4.1. Kết quả đạt được của dự án.....	59
4.2. Hạn chế và giải pháp khắc phục.....	60
TÀI LIỆU THAM KHẢO.....	62

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ NỀN TẢNG CÔNG NGHỆ

1.1. Lý Do Chọn Đề Tài

Trong bối cảnh thời đại công nghệ số đang phát triển vượt bậc, ngành công nghệ thông tin (CNTT) đã khẳng định vai trò không thể thiếu trong mọi khía cạnh của đời sống hiện đại. Từ các lĩnh vực quan trọng như khoa học, kỹ thuật, kinh tế, y tế, giáo dục cho đến quân sự, CNTT không chỉ là công cụ hỗ trợ mà còn là động lực thúc đẩy sự đổi mới và tiến bộ của xã hội. Đặc biệt, trong ngành công nghiệp giải trí, CNTT đã tạo nên một cuộc cách mạng với sự phát triển mạnh mẽ của lĩnh vực trò chơi điện tử. Đây là một ngành công nghiệp trị giá hàng tỷ đô la, thu hút sự quan tâm và tham gia của hàng triệu người trên toàn cầu, từ trẻ em, thanh thiếu niên cho đến người trưởng thành. Các trò chơi điện tử không chỉ mang lại giá trị giải trí mà còn trở thành một phương tiện để phát triển kỹ năng tư duy, sáng tạo và giải quyết vấn đề. Trong những năm gần đây, sự phổ biến của Internet cùng với những bước tiến vượt bậc trong công nghệ đồ họa và phần cứng đã tạo điều kiện cho ngành công nghiệp game bùng nổ. Các trò chơi ngày càng trở nên sống động, chân thực và phức tạp, đáp ứng nhu cầu ngày càng cao của người chơi. Đặc biệt, sự xuất hiện và phát triển của trí tuệ nhân tạo (AI) đã mang đến một bước ngoặt lớn trong quá trình phát triển game. AI không chỉ giúp tối ưu hóa các thuật toán, nâng cao trải nghiệm người chơi mà còn hỗ trợ các nhà phát triển trong việc thiết kế và xây dựng trò chơi một cách hiệu quả hơn. Nhờ đó, việc lập trình game không còn là lĩnh vực dành riêng cho các chuyên gia mà đã trở nên dễ tiếp cận hơn với những người mới bắt đầu, đặc biệt là những sinh viên đam mê CNTT và lập trình.

Xuất phát từ thực tế đó, cùng với niềm đam mê tìm hiểu về lập trình game và mong muốn ứng dụng những kiến thức đã học vào thực tiễn, nhóm chúng em quyết định lựa chọn đề tài “Xây dựng chương trình trò chơi Cờ Caro bằng ngôn ngữ Python” cho bài tiểu luận lần này. Trước đây, nhóm đã tìm hiểu và thực hiện một phiên bản đơn giản hơn của trò chơi này là Tic Tac Toe. Tuy nhiên, sau quá trình nghiên cứu, nhóm nhận thấy rằng Tic Tac Toe có giới hạn về mặt chiến thuật do không gian chơi nhỏ và số nước đi ít, dẫn đến trò chơi dễ bị rơi vào trạng thái hòa khi cả hai người chơi đều có chiến thuật tốt. Do đó,

nhóm quyết định mở rộng và nâng cấp dự án lên trò chơi Cờ Caro - một biến thể phức tạp hơn, đòi hỏi khả năng lập trình thuật toán AI mạnh mẽ hơn để có thể tạo ra một đối thủ máy tính có khả năng chơi thông minh và thách thức người chơi. Cờ Caro, với luật chơi đơn giản nhưng độ phức tạp cao hơn Tic Tac Toe, là một bài toán thú vị trong lập trình AI. Việc phát triển thuật toán AI để giúp máy có thể đưa ra các nước đi hợp lý, có chiến thuật rõ ràng đòi hỏi nhóm phải tìm hiểu và ứng dụng các thuật toán tìm kiếm như Minimax, Alpha-Beta Pruning. Điều này không chỉ giúp nhóm củng cố kiến thức về thuật toán mà còn mang đến cơ hội tiếp cận với những kỹ thuật được sử dụng rộng rãi trong lập trình game thực tế. Hơn nữa, việc mở rộng trò chơi từ Tic Tac Toe sang Cờ Caro cũng giúp nhóm hiểu sâu hơn về cách xử lý bàn cờ lớn, tối ưu hóa hiệu suất tính toán và cải thiện trải nghiệm người chơi.

Bên cạnh đó, việc lựa chọn ngôn ngữ lập trình Python cho dự án này cũng mang ý nghĩa quan trọng. Python là một ngôn ngữ lập trình phổ biến, dễ học, cú pháp rõ ràng và có cộng đồng hỗ trợ rộng lớn, rất phù hợp với những người mới bắt đầu như nhóm chúng em. Đồng thời, Python cung cấp nhiều thư viện mạnh mẽ, giúp đơn giản hóa quá trình phát triển phần mềm, từ đó cho phép nhóm tập trung vào việc xây dựng logic trò chơi thay vì sa lầy vào các vấn đề kỹ thuật phức tạp. Thông qua đề tài này, nhóm chúng em không chỉ củng cố kiến thức về lập trình Python mà còn rèn luyện tư duy thuật toán, kỹ năng phân tích vấn đề và khả năng làm việc nhóm – những yếu tố quan trọng để phát triển phần mềm trong thực tế. Hơn nữa, việc thực hiện đề tài “Xây dựng chương trình trò chơi Cờ Caro” còn là bước đệm để nhóm chúng em làm quen với quy trình phát triển một dự án phần mềm từ ý tưởng ban đầu đến sản phẩm hoàn chỉnh. Đây sẽ là nền tảng vững chắc để chúng em tiếp tục khám phá và chinh phục những dự án lập trình phức tạp hơn trong tương lai, chẳng hạn như các trò chơi có đồ họa cao cấp hoặc tích hợp các công nghệ tiên tiến như AI và thực tế ảo. Với tất cả những lý do trên, nhóm chúng em tin rằng đề tài này không chỉ mang tính thực tiễn mà còn là một cơ hội quý báu để học hỏi, phát triển kỹ năng và nuôi dưỡng đam mê trong lĩnh vực lập trình game nói riêng cũng như công nghệ thông tin nói chung.

1.2. Giới thiệu về ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, đa năng, được tạo ra bởi Guido van Rossum và lần đầu tiên ra mắt vào năm 1991. Python được thiết kế với triết lý "đơn giản và dễ đọc", giúp các lập trình viên dễ dàng tiếp cận, ngay cả với những người mới bắt đầu. Một trong những đặc điểm nổi bật nhất của Python là cú pháp rõ ràng, dễ hiểu, gần gũi với ngôn ngữ tự nhiên, điều này giúp giảm thiểu thời gian viết mã và nâng cao hiệu suất làm việc của lập trình viên.

Python là một ngôn ngữ thông dịch, có nghĩa là mã nguồn Python không cần biên dịch trước khi chạy mà được thực thi trực tiếp bởi trình thông dịch Python. Điều này giúp quá trình phát triển phần mềm trở nên linh hoạt hơn, cho phép lập trình viên thử nghiệm, sửa lỗi và kiểm tra kết quả ngay lập tức mà không cần trải qua các bước biên dịch rườm rà như các ngôn ngữ lập trình biên dịch truyền thống (ví dụ: C, C++ hoặc Java). Chính vì tính linh hoạt này mà Python rất phổ biến trong các lĩnh vực như phát triển web, trí tuệ nhân tạo, khoa học dữ liệu, và tự động hóa.

Python hỗ trợ nhiều mô hình lập trình khác nhau, bao gồm lập trình hướng đối tượng (Object-Oriented Programming - OOP), lập trình thủ tục (Procedural Programming) và lập trình hàm (Functional Programming). Với mô hình lập trình hướng đối tượng, Python cho phép người dùng tạo ra các lớp (class) và đối tượng (object) để mô phỏng các thực thể trong thế giới thực, giúp mã nguồn trở nên có tổ chức, dễ quản lý và tái sử dụng hơn. Ngoài ra, lập trình hàm trong Python cũng được hỗ trợ mạnh mẽ thông qua các hàm ẩn danh (lambda functions), các công cụ như map(), filter() và các thư viện hỗ trợ lập trình hàm như functools.

Python có một thư viện tiêu chuẩn vô cùng phong phú, giúp lập trình viên thực hiện nhiều tác vụ khác nhau mà không cần phải viết lại mã từ đầu. Ví dụ, thư viện os hỗ trợ thao tác với hệ điều hành, thư viện sys giúp làm việc với các thông số hệ thống, thư viện math cung cấp các hàm toán học, và thư viện random giúp tạo số ngẫu nhiên. Ngoài thư viện tiêu chuẩn, Python còn có một hệ sinh thái rộng lớn với hàng ngàn thư viện bên thứ ba được cung cấp thông qua kho lưu trữ PyPI (Python Package Index). Các thư viện nổi tiếng

như NumPy, Pandas, Matplotlib, TensorFlow, PyTorch, Django, Flask giúp Python trở thành lựa chọn hàng đầu trong nhiều lĩnh vực khác nhau.

Một trong những lý do khiến Python trở nên phổ biến là khả năng mở rộng và tích hợp với các ngôn ngữ lập trình khác. Python có thể dễ dàng kết hợp với C, C++, Java, hoặc thậm chí là JavaScript để tận dụng hiệu suất của các ngôn ngữ này. Điều này làm cho Python trở thành một lựa chọn lý tưởng trong các dự án yêu cầu hiệu năng cao nhưng vẫn muốn giữ lại sự linh hoạt và dễ sử dụng của Python.

Python cũng là một công cụ mạnh mẽ trong lĩnh vực trí tuệ nhân tạo và học máy (Machine Learning). Với các thư viện như TensorFlow, Keras, Scikit-learn, PyTorch, Python cho phép các nhà nghiên cứu và kỹ sư dễ dàng triển khai các mô hình AI phức tạp để giải quyết các bài toán như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và phân tích dữ liệu lớn. Nhờ tính đơn giản của Python, các thuật toán AI có thể được triển khai một cách dễ dàng mà không mất quá nhiều thời gian viết mã.

Trong phát triển web, Python cũng đóng một vai trò quan trọng với các framework mạnh mẽ như Django và Flask. Django là một framework full-stack giúp xây dựng các ứng dụng web lớn một cách nhanh chóng và bảo mật, trong khi Flask là một framework nhẹ, linh hoạt, phù hợp với các dự án nhỏ hoặc các API đơn giản. Python cũng được sử dụng rộng rãi trong lĩnh vực tự động hóa, từ viết script để quản lý hệ thống, tự động hóa các công việc lặp đi lặp lại, đến phát triển bot trên các nền tảng mạng xã hội.

Ngoài ra, Python còn có mặt trong nhiều lĩnh vực khác như khoa học dữ liệu, tài chính, an ninh mạng và thậm chí là phát triển trò chơi. Với các công cụ như Pygame, lập trình viên có thể tạo ra các trò chơi 2D đơn giản bằng Python. Trong tài chính, Python được sử dụng để phân tích dữ liệu tài chính, xây dựng mô hình dự báo, và tự động hóa các giao dịch chứng khoán.

1.3. Cấu trúc của python

Cấu trúc của Python được tổ chức theo hướng đơn giản nhưng mạnh mẽ. Một chương trình Python có thể chỉ là một tệp .py đơn giản chứa một vài dòng mã, hoặc có thể là một dự án lớn với nhiều module và gói được tổ chức một cách chặt chẽ. Python hỗ trợ tổ chức mã nguồn theo các module và package, giúp lập trình viên dễ dàng quản lý các dự án lớn.

Một module trong Python là một tệp .py chứa các hàm và lớp có thể tái sử dụng. Các module này có thể được nhập vào trong các chương trình khác nhau bằng cách sử dụng lệnh import.

```
import math
print(math.sqrt(16)) # Kết quả: 4.0
```

Ngoài module, Python còn hỗ trợ tổ chức mã theo package. Một package đơn giản là một thư mục chứa một hoặc nhiều module, cùng với một tệp __init__.py giúp Python nhận diện đó là một package hợp lệ. Việc sử dụng package giúp quản lý mã nguồn hiệu quả hơn, đặc biệt khi làm việc với các dự án lớn.

- Một chương trình Python cơ bản bao gồm các thành phần sau:

- + Biến (Variables): Lưu trữ dữ liệu.
- + Kiểu dữ liệu (Data Types): Python hỗ trợ nhiều kiểu dữ liệu như số nguyên (int), số thực (float), chuỗi (str), danh sách (list), tuple (tuple), từ điển (dict).
- + Toán tử (Operators): Bao gồm toán tử số học (+, -, *, /), toán tử so sánh (==, !=, >, <), toán tử logic (and, or, not).
- + Cấu trúc điều khiển (Control Structures): Gồm câu lệnh điều kiện (if-else), vòng lặp (for, while).
- + Hàm (Functions): Các đoạn mã có thể tái sử dụng, được định nghĩa bằng từ khóa def.

+ Lập trình hướng đối tượng (OOP): Hỗ trợ lớp (class) và đối tượng (object).

Python là một trong những ngôn ngữ lập trình phổ biến nhất thế giới, với một cộng đồng lớn mạnh hỗ trợ và phát triển liên tục. Sự phổ biến của Python không chỉ nhờ vào cú pháp dễ hiểu mà còn nhờ vào sự đóng góp của hàng triệu lập trình viên trên toàn cầu, tạo ra một hệ sinh thái phong phú giúp ngôn ngữ này ngày càng phát triển mạnh mẽ hơn. Với tất cả những đặc điểm trên, Python thực sự là một công cụ mạnh mẽ, đa dụng và đáng học tập cho bất kỳ ai muốn tham gia vào lĩnh vực lập trình.

1.4. Ứng dụng của python

Python là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay và được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ phát triển phần mềm đến trí tuệ nhân tạo. Dưới đây là một số lĩnh vực chính mà Python được sử dụng:

- Phân tích dữ liệu và học máy (Machine learning & Science)

Python là ngôn ngữ hàng đầu trong lĩnh vực khoa học dữ liệu và học máy nhờ các thư viện mạnh mẽ như NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras. Nó giúp các nhà phân tích dữ liệu xử lý, trực quan hóa và xây dựng các mô hình trí tuệ nhân tạo một cách hiệu quả.

- Phát triển web

Python là ngôn ngữ hàng đầu trong lĩnh vực khoa học dữ liệu và học máy nhờ các thư viện mạnh mẽ như NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras. Nó giúp các nhà phân tích dữ liệu xử lý, trực quan hóa và xây dựng các mô hình trí tuệ nhân tạo một cách hiệu quả.

- Tự động hóa và phát triển phần mềm

Python có thể giúp tự động hóa các tác vụ lặp đi lặp lại như kiểm tra lỗi, chuyển đổi tệp, thu thập dữ liệu web (web scraping), xử lý văn bản, hoặc quản lý hệ thống. Các thư viện như Selenium, BeautifulSoup, OpenCV giúp Python trở thành một công cụ hữu ích trong tự động hóa.

- Phát triển game

Python có thể được sử dụng để tạo ra các trò chơi từ đơn giản đến phức tạp nhờ thư viện Pygame. Một số trò chơi nổi tiếng đã được phát triển bằng Python như "EVE Online", "CARO" và "Battlefield 2".

- Bảo mật và an ninh mạng

Python cũng được sử dụng rộng rãi trong lĩnh vực an ninh mạng để phát triển các công cụ kiểm tra bảo mật, phát hiện lỗ hổng, phân tích mã độc nhờ các thư viện như Scapy, PyCrypto, Pwntools.

- Ứng dụng trong IoT (Internet of Things):

Với sự phát triển của các thiết bị thông minh, Python trở thành một lựa chọn phổ biến để lập trình trên các thiết bị nhúng như Raspberry Pi, Arduino nhờ các thư viện như MicroPython.

1.5. Đặc điểm của Python

- Các đặc điểm của python

- + Dễ học, thân thiện với người mới

- + Python có cú pháp đơn giản, gần với ngôn ngữ tự nhiên, giúp lập trình viên dễ học và viết mã nhanh chóng.

- Ngôn ngữ thông dịch (Interpreted Language)

Python không cần biên dịch trước khi chạy như các ngôn ngữ C, C++, mà được thực thi trực tiếp bởi trình thông dịch. Điều này giúp lập trình viên dễ dàng thử nghiệm và phát triển ứng dụng nhanh hơn.

- Hỗ trợ lập trình hướng đối tượng (OOP - Object-Oriented Programming)

Python hỗ trợ lập trình hướng đối tượng, cho phép tổ chức mã nguồn theo mô hình lớp (Class) và đối tượng (Object), giúp mã dễ quản lý và mở rộng.

- Mã nguồn mở (Open Source)

Python là một ngôn ngữ mã nguồn mở, miễn phí để sử dụng, sửa đổi và phân phối. Điều này giúp Python có một cộng đồng lớn mạnh và liên tục được cải tiến.

- Đa nền tảng (Cross -platform)

Python có thể chạy trên nhiều hệ điều hành như Windows, macOS, Linux mà không cần thay đổi mã nguồn.

- Thư viện phong phú (Rich Libraries & Frameworks)

Python có một hệ sinh thái thư viện đa dạng hỗ trợ nhiều lĩnh vực khác nhau như NumPy, Pandas (khoa học dữ liệu), TensorFlow, Keras (trí tuệ nhân tạo), Django, Flask (phát triển web), Pygame, Tkinter (phát triển game), v.v.

- Tính mở rộng và nhúng (Extensibility & Embeddability)

Python có thể tích hợp với các ngôn ngữ khác như C, C++, Java để tăng hiệu suất hoặc mở rộng chức năng của ứng dụng.

1.6. Công Cụ Và Thư Viện Sử Dụng – Visual Studio

Đề Trong lĩnh vực phát triển phần mềm, một môi trường phát triển tích hợp (Integrated Development Environment - IDE) đóng vai trò vô cùng quan trọng, giúp lập trình viên có thể viết mã nguồn, kiểm thử, gỡ lỗi và triển khai các ứng dụng một cách hiệu quả. Một trong những IDE nổi bật và được sử dụng rộng rãi nhất hiện nay chính là Visual Studio, một sản phẩm mạnh mẽ do Microsoft phát triển. Đây không chỉ là một công cụ hỗ trợ lập trình mà còn là một hệ sinh thái toàn diện dành cho các nhà phát triển phần mềm trên nhiều nền tảng khác nhau, từ các ứng dụng desktop, web, di động cho đến các hệ thống nhúng và trí tuệ nhân tạo. Với giao diện trực quan, tính năng phong phú và khả năng mở rộng linh hoạt, Visual Studio đã trở thành lựa chọn hàng đầu cho cả lập trình viên chuyên nghiệp lẫn những người mới học lập trình.

Visual Studio lần đầu tiên ra mắt vào năm 1997 với phiên bản Visual Studio 97. Đây là một bước tiến lớn của Microsoft nhằm thống nhất các bộ công cụ lập trình riêng lẻ

như Visual Basic, Visual C++, Visual FoxPro vào một môi trường làm việc chung, giúp lập trình viên dễ dàng quản lý và phát triển dự án. Trải qua hơn hai thập kỷ phát triển, Visual Studio đã liên tục được nâng cấp với nhiều tính năng tiên tiến, cải thiện hiệu suất và mở rộng khả năng hỗ trợ nhiều ngôn ngữ lập trình khác nhau. Các phiên bản đáng chú ý có thể kể đến như Visual Studio 6.0 (1998), Visual Studio .NET (2002) – đánh dấu sự chuyển đổi sang nền tảng .NET, Visual Studio 2010, Visual Studio 2015, và các phiên bản gần đây như Visual Studio 2019 và Visual Studio 2022, với sự tối ưu hóa mạnh mẽ về tốc độ, giao diện và khả năng tích hợp với các công nghệ hiện đại. Một trong những ưu điểm nổi bật của Visual Studio chính là khả năng hỗ trợ đa nền tảng. Ban đầu, Visual Studio chủ yếu được thiết kế dành cho hệ điều hành Windows, nhưng với sự phát triển của công nghệ và nhu cầu mở rộng, Microsoft đã tích hợp khả năng hỗ trợ nhiều nền tảng khác nhau, đặc biệt là với Visual Studio Code – một phiên bản IDE nhẹ hơn, có thể chạy trên Windows, macOS và Linux. Điều này giúp lập trình viên có thể phát triển phần mềm trên nhiều hệ điều hành mà không gặp trở ngại về môi trường làm việc. Visual Studio cung cấp nhiều công cụ mạnh mẽ giúp lập trình viên làm việc hiệu quả hơn. Trình soạn thảo mã nguồn được tích hợp nhiều tính năng hỗ trợ như đánh dấu cú pháp (syntax highlighting), tự động hoàn thành mã (IntelliSense), kiểm tra lỗi cú pháp theo thời gian thực, giúp lập trình viên viết mã nhanh hơn và chính xác hơn. Bên cạnh đó, Visual Studio còn có trình gỡ lỗi mạnh mẽ, giúp phát hiện và sửa lỗi trong quá trình chạy chương trình một cách dễ dàng. Ngoài ra, nó còn hỗ trợ quản lý dự án với hệ thống kiểm soát phiên bản như Git, giúp các nhóm lập trình viên có thể làm việc cùng nhau trên cùng một dự án mà không gặp phải xung đột mã nguồn. Một điểm đặc biệt quan trọng của Visual Studio là khả năng hỗ trợ nhiều ngôn ngữ lập trình khác nhau. Ban đầu, nó chủ yếu được sử dụng cho các ngôn ngữ do Microsoft phát triển như C#, Visual Basic, F#, nhưng theo thời gian, Microsoft đã mở rộng và hỗ trợ nhiều ngôn ngữ khác như C++, Python, JavaScript, TypeScript, PHP, Java, và nhiều ngôn ngữ khác thông qua các tiện ích mở rộng (extensions). Điều này giúp lập trình viên có thể sử dụng Visual Studio cho nhiều loại dự án khác nhau mà không bị giới hạn bởi ngôn ngữ lập trình cụ thể.

Không chỉ là một công cụ dành riêng cho lập trình viên cá nhân, Visual Studio còn được thiết kế để phục vụ nhu cầu phát triển phần mềm chuyên nghiệp với quy mô lớn. Phiên bản Visual Studio Enterprise cung cấp các công cụ tiên tiến để phát triển phần mềm doanh nghiệp, bao gồm kiểm thử tự động, phân tích mã nguồn, tối ưu hóa hiệu suất ứng dụng và tích hợp với các hệ thống DevOps. Microsoft cũng cung cấp phiên bản miễn phí có tên Visual Studio Community, giúp sinh viên, lập trình viên cá nhân và các nhóm nhỏ có thể sử dụng mà không phải trả phí, góp phần mở rộng cộng đồng lập trình viên sử dụng Visual Studio trên toàn thế giới. Nhờ vào sự phát triển không ngừng và khả năng tích hợp với các công nghệ hiện đại như trí tuệ nhân tạo (AI), điện toán đám mây (Cloud Computing), Visual Studio đã trở thành một công cụ không thể thiếu đối với các nhà phát triển phần mềm. Với giao diện thân thiện, tính năng đa dạng và hỗ trợ mạnh mẽ từ cộng đồng, Visual Studio không chỉ giúp lập trình viên nâng cao hiệu suất làm việc mà còn mở ra nhiều cơ hội trong lĩnh vực phát triển phần mềm chuyên nghiệp.

1.7. Lý do sử dụng Visual Studio vào dự án

Trong quá trình phát triển phần mềm, việc lựa chọn một môi trường lập trình phù hợp đóng vai trò vô cùng quan trọng, ảnh hưởng trực tiếp đến hiệu suất làm việc, khả năng mở rộng cũng như tính ổn định của dự án. Đối với dự án xây dựng trò chơi Cờ Caro ứng dụng thuật toán Minimax bằng ngôn ngữ Python, nhóm chúng em đã quyết định sử dụng Visual Studio làm môi trường phát triển chính. Quyết định này được đưa ra dựa trên nhiều yếu tố quan trọng, trong đó bao gồm tính tiện lợi, khả năng hỗ trợ lập trình mạnh mẽ, tích hợp các công cụ hữu ích và phù hợp với nhu cầu của dự án. Trước hết, Visual Studio là một trong những môi trường phát triển tích hợp (IDE) mạnh mẽ và phổ biến nhất hiện nay, được nhiều lập trình viên chuyên nghiệp tin dùng. Nó cung cấp một hệ sinh thái hoàn chỉnh, không chỉ hỗ trợ viết mã mà còn giúp kiểm tra lỗi, gỡ lỗi, quản lý dự án, tích hợp hệ thống kiểm soát phiên bản và tối ưu hóa mã nguồn. Điều này giúp cho nhóm chúng em có thể làm việc một cách hiệu quả hơn, giảm thiểu thời gian xử lý lỗi và tập trung vào việc phát triển thuật toán cho trò chơi. Một trong những lý do quan trọng khiến nhóm quyết định chọn Visual Studio là khả năng hỗ trợ Python mạnh mẽ. Python là một trong những ngôn

ngữ lập trình phổ biến nhất hiện nay, được sử dụng rộng rãi trong nhiều lĩnh vực, từ khoa học dữ liệu, trí tuệ nhân tạo đến phát triển ứng dụng và trò chơi. Visual Studio hỗ trợ tốt Python thông qua tiện ích mở rộng Python Tools for Visual Studio (PTVS), giúp lập trình viên có thể viết, kiểm thử và gỡ lỗi mã nguồn một cách dễ dàng. Trình biên tập mã nguồn của Visual Studio còn tích hợp sẵn các tính năng như IntelliSense, giúp tự động hoàn thành mã, phát hiện lỗi cú pháp theo thời gian thực, hỗ trợ tìm kiếm và thay thế thông minh, giúp cho quá trình lập trình trở nên nhanh chóng và hiệu quả hơn.

Ngoài ra, khả năng gỡ lỗi mạnh mẽ của Visual Studio cũng là một trong những yếu tố quan trọng giúp nhóm chúng em lựa chọn công cụ này. Khi xây dựng trò chơi, đặc biệt là với sự tích hợp của thuật toán Minimax, việc xuất hiện lỗi logic trong quá trình tính toán nước đi của AI là điều không thể tránh khỏi. Visual Studio cung cấp một trình gỡ lỗi trực quan, cho phép theo dõi từng dòng lệnh, kiểm tra giá trị của các biến trong quá trình chạy chương trình, giúp lập trình viên dễ dàng phát hiện và sửa lỗi. Điều này giúp nhóm chúng em tiết kiệm được nhiều thời gian trong quá trình kiểm thử và hoàn thiện thuật toán AI cho trò chơi. Một ưu điểm khác của Visual Studio là khả năng quản lý dự án tốt. Trong quá trình phát triển phần mềm, việc tổ chức mã nguồn một cách khoa học và dễ dàng quản lý là điều rất quan trọng. Visual Studio cung cấp giao diện trực quan để quản lý các tệp tin trong dự án, giúp lập trình viên có thể dễ dàng theo dõi cấu trúc mã nguồn, chỉnh sửa các thành phần và kiểm soát tiến trình phát triển. Hơn nữa, Visual Studio còn hỗ trợ tích hợp với Git, một hệ thống quản lý phiên bản phổ biến, giúp nhóm chúng em có thể theo dõi sự thay đổi của mã nguồn, làm việc nhóm một cách hiệu quả hơn, đồng thời đảm bảo tính nhất quán trong quá trình phát triển. Bên cạnh đó, Visual Studio còn cung cấp một môi trường làm việc thân thiện với người dùng. Giao diện của Visual Studio được thiết kế trực quan, dễ sử dụng và có khả năng tùy chỉnh theo sở thích của từng lập trình viên. Điều này giúp cho nhóm chúng em, dù có sự chênh lệch về kinh nghiệm lập trình, vẫn có thể nhanh chóng làm quen với công cụ và tập trung vào việc phát triển dự án.

Một lý do không thể bỏ qua khi lựa chọn Visual Studio là khả năng mở rộng mạnh mẽ thông qua các tiện ích mở rộng (Extensions). Microsoft đã phát triển một kho tiện ích mở

rộng phong phú, giúp lập trình viên có thể bổ sung các tính năng cần thiết cho từng loại dự án. Đối với nhóm chúng em, các tiện ích hỗ trợ Python, kiểm tra hiệu suất mã nguồn, và gỡ lỗi nâng cao đã giúp ích rất nhiều trong quá trình xây dựng trò chơi Cờ Caro.

Tuy nhiên, khi lựa chọn Visual Studio, nhóm cũng cân nhắc đến các môi trường phát triển khác, trong đó Anaconda là một cái tên đáng chú ý. Anaconda là một phân phối mã nguồn mở dành riêng cho Python và R, được thiết kế đặc biệt để phục vụ các ứng dụng khoa học dữ liệu, trí tuệ nhân tạo (AI) và học máy (Machine Learning). Anaconda tích hợp nhiều thư viện và công cụ quan trọng như NumPy, Pandas, Matplotlib, TensorFlow, Scikit-learn, giúp lập trình viên dễ dàng xây dựng các mô hình phân tích dữ liệu hoặc thuật toán AI. Mặc dù Anaconda có nhiều ưu điểm, đặc biệt là trong lĩnh vực khoa học dữ liệu, nhưng khi xét đến phát triển trò chơi, Visual Studio lại có nhiều lợi thế hơn. Một số lý do chính khiến nhóm không chọn Anaconda bao gồm:

Hạn chế về giao diện phát triển: Anaconda chủ yếu hỗ trợ Jupyter Notebook và Spyder, phù hợp với việc nghiên cứu dữ liệu hơn là phát triển trò chơi có giao diện trực quan.

Quản lý dự án kém hơn Visual Studio: Anaconda không có trình quản lý tệp tin và dự án mạnh mẽ như Visual Studio, gây khó khăn trong việc tổ chức mã nguồn một cách khoa học.

Khả năng gỡ lỗi không mạnh bằng Visual Studio: Trong khi Visual Studio cung cấp trình gỡ lỗi trực quan và tích hợp, Anaconda chỉ hỗ trợ gỡ lỗi ở mức cơ bản, gây khó khăn khi theo dõi và sửa lỗi trong quá trình phát triển trò chơi.

Không hỗ trợ mạnh mẽ cho Git: Visual Studio có tích hợp sẵn Git giúp dễ dàng quản lý phiên bản và làm việc nhóm, trong khi Anaconda yêu cầu cấu hình bổ sung để sử dụng Git hiệu quả.

Với tất cả những lý do trên, nhóm chúng em tin rằng việc lựa chọn Visual Studio làm môi trường phát triển cho dự án Cờ Caro là một quyết định đúng đắn. Công cụ này không chỉ giúp tối ưu hóa quy trình lập trình mà còn hỗ trợ mạnh mẽ trong việc phát triển

và kiểm thử thuật toán AI, giúp nhóm xây dựng một sản phẩm hoàn chỉnh, hoạt động hiệu quả và mang tính thực tiễn cao.

1.8. Tkinter – Thư viện xây dựng giao diện

Đề Trong quá trình phát triển phần mềm, đặc biệt là các ứng dụng có giao diện đồ họa người dùng (GUI – Graphical User Interface), việc lựa chọn một thư viện phù hợp đóng vai trò vô cùng quan trọng. Đối với lập trình Python, có nhiều thư viện hỗ trợ xây dựng giao diện đồ họa như PyQt, Kivy, wxPython, nhưng Tkinter vẫn là một trong những lựa chọn phổ biến nhất nhờ vào sự đơn giản, dễ học và khả năng tích hợp mạnh mẽ với Python. Tkinter (viết tắt của "Tk Interface") là thư viện GUI mặc định của Python, tức là khi cài đặt Python, Tkinter đã được tích hợp sẵn mà không cần cài đặt thêm. Điều này giúp nó trở thành lựa chọn tối ưu cho những ai muốn nhanh chóng xây dựng giao diện mà không phải mất thời gian thiết lập các thư viện bên ngoài. Tkinter được xây dựng dựa trên Tk, một bộ công cụ GUI đa nền tảng phát triển từ những năm 1990. Mặc dù có giao diện khá đơn giản so với các thư viện như PyQt hoặc Kivy, nhưng Tkinter vẫn cung cấp đầy đủ các thành phần cơ bản để xây dựng một ứng dụng GUI hoàn chỉnh, từ nút bấm (Button), ô nhập liệu (Entry), nhãn (Label), hộp thoại (Dialog) cho đến khung vẽ đồ họa (Canvas) để tạo ra các giao diện phức tạp hơn.

Tkinter có nhiều ưu điểm khiến nó trở thành lựa chọn hàng đầu cho các lập trình viên khi phát triển ứng dụng có giao diện đồ họa. Một trong những ưu điểm quan trọng nhất là dễ học, dễ sử dụng. Nhờ có cú pháp đơn giản, dễ hiểu, lập trình viên – đặc biệt là người mới học Python – có thể nhanh chóng làm quen với Tkinter và tạo ra các giao diện trực quan một cách dễ dàng. Điều này giúp họ tập trung vào phát triển chức năng chính của ứng dụng thay vì phải mất quá nhiều thời gian cho việc thiết kế giao diện. Bên cạnh đó, Tkinter có lợi thế lớn khi tích hợp sẵn trong Python, không giống như các thư viện GUI khác yêu cầu cài đặt bổ sung. Điều này giúp tiết kiệm thời gian thiết lập môi trường lập trình và đảm bảo tính tương thích cao giữa các phiên bản Python khác nhau. Một ưu điểm quan trọng khác của Tkinter là khả năng hoạt động trên nhiều hệ điều hành. Thư viện này có thể chạy trên Windows, macOS và Linux mà không cần thay đổi mã nguồn, đảm bảo

tính linh hoạt cao cho các ứng dụng cần triển khai trên nhiều nền tảng. Hơn nữa, Tkinter cung cấp đầy đủ các widget cần thiết cho việc xây dựng giao diện như nút bấm, ô nhập liệu, danh sách lựa chọn, thanh cuộn, cửa sổ hộp thoại và nhiều thành phần khác giúp lập trình viên dễ dàng thiết kế giao diện người dùng một cách trực quan.

Mặc dù có nhiều ưu điểm, nhưng Tkinter cũng tồn tại một số hạn chế nhất định. Một trong số đó là giao diện có phần đơn giản và không hiện đại bằng các thư viện GUI khác như PyQt hoặc Kivy. Các thành phần giao diện trong Tkinter có phong cách thiết kế khá cũ kỹ, khó tùy chỉnh nếu so với các framework GUI cao cấp hơn. Bên cạnh đó, hiệu suất của Tkinter cũng không thực sự tối ưu khi xây dựng các ứng dụng đồ họa phức tạp hoặc yêu cầu cập nhật giao diện theo thời gian thực.

1.9. Ứng dụng Tkinter vào dự án cờ Caro

Tkinter là một lựa chọn phù hợp để xây dựng giao diện đồ họa cho trò chơi cờ caro trong dự án này. Nhờ vào khả năng cung cấp các thành phần giao diện cơ bản, nhóm có thể dễ dàng thiết kế bàn cờ, hiển thị nước đi của người chơi và AI, cũng như cập nhật trạng thái trò chơi một cách trực quan. Việc sử dụng Canvas của Tkinter giúp nhóm vẽ bàn cờ và các quân cờ một cách linh hoạt, đồng thời cho phép cập nhật giao diện mỗi khi người chơi thực hiện một nước đi mới. Bên cạnh đó, Tkinter còn giúp nhóm dễ dàng tạo các thành phần giao diện bổ sung như nút bắt đầu ván mới, nút thoát, hiển thị điểm số hoặc các thông báo trạng thái của trò chơi. Ngoài ra, nhờ có hệ thống sự kiện trong Tkinter, nhóm có thể dễ dàng xử lý các thao tác chuột, giúp người chơi có thể nhấp vào ô cờ để thực hiện nước đi. Điều này giúp tăng trải nghiệm người dùng, khiến trò chơi trở nên sinh động hơn. Một lợi thế quan trọng khác của việc sử dụng Tkinter trong dự án này là khả năng tích hợp với thuật toán AI Minimax. Nhóm có thể lập trình AI để tự động phân tích thế cờ và thực hiện nước đi hợp lý nhất, sau đó hiển thị trực tiếp trên giao diện Tkinter một cách mượt mà. Sự kết hợp giữa Tkinter và thuật toán Minimax giúp trò chơi không chỉ có giao diện trực quan mà còn mang lại trải nghiệm thử thách cho người chơi khi đối đầu với AI. Tuy nhiên, do Tkinter không được tối ưu cho các ứng dụng đồ họa phức tạp, nhóm cần đảm bảo rằng trò chơi vẫn duy trì được hiệu suất ổn định, tránh tình trạng lag khi AI tính toán nước đi hoặc

khi cập nhật giao diện liên tục. Việc tối ưu code để đảm bảo chương trình chạy mượt mà là một yếu tố quan trọng khi sử dụng Tkinter cho dự án này.

Tóm lại, với sự đơn giản, dễ sử dụng và khả năng tích hợp linh hoạt với Python, Tkinter là một lựa chọn hợp lý để xây dựng giao diện cho trò chơi cờ caro. Nó giúp nhóm nhanh chóng thiết kế giao diện, xử lý sự kiện chuột, hiển thị nước đi của AI và tạo ra một trò chơi hoàn chỉnh mà không phải tốn quá nhiều công sức cho việc lập trình giao diện phức tạp. Mặc dù có một số hạn chế nhất định, nhưng với quy mô của dự án này, Tkinter vẫn là một giải pháp hiệu quả, giúp nhóm có thể tập trung vào phần thuật toán và logic trò chơi mà không phải mất quá nhiều thời gian cho việc thiết kế giao diện.

1.10. Giới thiệu trò chơi cờ Caro

Cờ Caro là một trong những trò chơi trí tuệ lâu đời và phổ biến, đặc biệt được ưa chuộng ở các nước châu Á như Việt Nam, Nhật Bản, Trung Quốc và Hàn Quốc. Trò chơi này không chỉ mang tính giải trí mà còn giúp rèn luyện tư duy chiến thuật, khả năng quan sát và phân tích tình huống. Đặc biệt, cờ Caro là một trò chơi đơn giản về mặt luật lệ nhưng lại vô cùng phức tạp khi đi sâu vào chiến thuật và cách chơi thực tế. Điều này làm cho cờ Caro trở thành một trò chơi phù hợp với mọi lứa tuổi, từ trẻ em cho đến người lớn, từ người mới bắt đầu cho đến những người chơi chuyên nghiệp có khả năng tính toán nước đi một cách bài bản.

Trò chơi cờ Caro có nguồn gốc từ trò chơi cờ vây của Trung Quốc và cờ gomoku của Nhật Bản. Tuy nhiên, trong quá trình phát triển, cờ Caro đã có những biến thể và quy tắc riêng biệt, phù hợp với từng vùng miền và đối tượng người chơi. Ở Việt Nam, cờ Caro trở thành một trò chơi phổ biến trong học đường, thường được học sinh chơi trên giấy ô ly trong những giờ ra chơi hoặc trên bàn cờ khi có thời gian rảnh rỗi. Với sự phát triển của công nghệ, cờ Caro ngày nay không chỉ được chơi trên bàn cờ vật lý mà còn xuất hiện trên nhiều nền tảng trực tuyến, từ ứng dụng di động, trò chơi máy tính cho đến các trang web chuyên về cờ Caro. Cờ Caro có ưu điểm là không yêu cầu các thiết bị chuyên dụng đắt tiền. Chỉ cần một tờ giấy ô vuông và một cây bút hoặc một bàn cờ đơn giản với các quân cờ X và O là người chơi đã có thể tham gia ngay lập tức. Chính vì sự tiện lợi này mà trò chơi cờ

Caro đã trở thành một phần không thể thiếu trong đời sống hàng ngày của nhiều người, đặc biệt là các bạn học sinh, sinh viên. Ngoài ra, cờ Caro còn có thể được chơi trên nhiều loại bàn cờ khác nhau, từ bàn cờ giấy, bàn cờ gỗ, bàn cờ điện tử cho đến các ứng dụng trên điện thoại thông minh.

Một trong những điểm hấp dẫn của cờ Caro là tính đối kháng cao. Khi hai người chơi đối đầu với nhau, mỗi nước đi đều có thể quyết định đến cục diện của cả ván đấu. Người chơi không chỉ cần tìm cách xây dựng chiến lược tấn công mà còn phải có kế hoạch phòng thủ hợp lý để ngăn cản đối phương giành chiến thắng. Điều này đòi hỏi người chơi phải có sự tập trung cao độ, khả năng phán đoán nhanh chóng và tư duy logic sắc bén. Một nước đi sai lầm có thể dẫn đến thất bại ngay lập tức, trong khi một nước đi chính xác có thể mang lại chiến thắng một cách bất ngờ. Không giống như nhiều trò chơi trí tuệ khác có yếu tố may mắn, cờ Caro hoàn toàn dựa vào khả năng tính toán và chiến lược của người chơi. Mỗi ván cờ là một cuộc đấu trí thực sự, nơi mà người chơi phải liên tục suy nghĩ, phân tích các nước đi và dự đoán những chiến thuật mà đối thủ có thể sử dụng. Khi chơi cờ Caro ở mức độ cao, người chơi thường không chỉ tập trung vào một vài nước đi trước mắt mà còn phải tính toán chiến lược dài hơi, lên kế hoạch cho cả một chuỗi nước đi nhằm đạt được lợi thế trên bàn cờ.

Ngoài yếu tố cá nhân, cờ Caro còn là một trò chơi giúp rèn luyện kỹ năng giao tiếp và tinh thần thể thao. Khi chơi cờ Caro với bạn bè hoặc gia đình, người chơi có thể học cách chấp nhận thắng thua, tôn trọng đối thủ và rút kinh nghiệm từ những ván cờ đã chơi. Điều này không chỉ giúp nâng cao kỹ năng chơi cờ mà còn có tác động tích cực đến tư duy và thái độ trong cuộc sống hàng ngày. Ngày nay, với sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI), cờ Caro đã được đưa vào các chương trình máy tính, nơi mà người chơi có thể đối đầu với AI thông minh, giúp nâng cao kỹ năng chơi cờ của mình. Các thuật toán như Minimax, Alpha-Beta Pruning, Monte Carlo Tree Search đã được ứng dụng vào trò chơi cờ Caro, tạo ra những đối thủ máy tính có khả năng phân tích và phản ứng rất nhanh trước các nước đi của người chơi. Điều này giúp những người yêu thích cờ Caro có thể rèn luyện kỹ năng của mình mọi lúc mọi nơi mà không cần phải tìm kiếm đối thủ thực tế.

Nhìn chung, cờ Caro là một trò chơi có tính trí tuệ cao nhưng lại vô cùng gần gũi và dễ tiếp cận. Từ một trò chơi dân gian đơn giản, cờ Caro đã phát triển thành một bộ môn trí tuệ có sức hút lớn, không chỉ trong đời sống thường ngày mà còn trong cả lĩnh vực nghiên cứu trí tuệ nhân tạo. Sự kết hợp giữa truyền thống và công nghệ đã giúp cờ Caro duy trì sức hấp dẫn của nó, trở thành một trò chơi không bao giờ lỗi thời và luôn có chỗ đứng vững chắc trong lòng người hâm mộ trên toàn thế giới.

1.11. Quy luật trò chơi Cờ Caro

Cờ Caro là một trò chơi chiến thuật hai người, trong đó người chơi lần lượt đặt quân cờ của mình lên một bàn cờ dạng lưới ô vuông. Mặc dù có vẻ ngoài đơn giản, luật chơi cờ Caro có thể trở nên khá phức tạp khi đi sâu vào các quy tắc chi tiết và chiến thuật chơi. Việc nắm vững luật chơi không chỉ giúp người chơi hiểu rõ cách vận hành của trò chơi mà còn giúp họ xây dựng chiến lược hợp lý để giành chiến thắng. Dưới đây là những quy tắc chi tiết của cờ Caro, được trình bày một cách cụ thể và tỉ mỉ để đảm bảo mọi khía cạnh của trò chơi đều được làm rõ.

- Bàn cờ và quân cờ:

- + Cờ Caro được chơi trên một bàn cờ có dạng lưới ô vuông, mỗi ô tương ứng với một vị trí đặt quân cờ.
- + Kích thước bàn cờ có thể thay đổi tùy theo thỏa thuận giữa hai người chơi.
- + Các kích thước phổ biến nhất bao gồm 15x15, 19x19, hoặc trên một tờ giấy kẻ ô ly (không giới hạn số lượng ô).
- + Hai người chơi sẽ sử dụng hai loại quân cờ khác nhau, thông thường được ký hiệu là “X” và “O”. Trong phiên bản cờ thực tế, quân cờ có thể là hai loại màu khác nhau, như đen và trắng.
- + Mỗi lượt đi, người chơi chỉ được phép đặt một quân cờ của mình vào một ô trống trên bàn cờ.

1.12. Cách chơi và mục tiêu

- Hai người chơi sẽ lần lượt thực hiện nước đi của mình. Người đi trước có thể được quyết định bằng cách tung đồng xu, thỏa thuận hoặc theo quy tắc truyền thống là người chơi quân X luôn đi trước.

- Mục tiêu của mỗi người chơi là sắp xếp các quân cờ của mình thành một chuỗi liên tiếp ít nhất 5 quân theo một trong ba hướng sau:

+ Hàng ngang (tức là nằm trên cùng một hàng).

+ Hàng dọc (tức là nằm trên cùng một cột).

+ Đường chéo (có thể là chéo từ trái sang phải hoặc từ phải sang trái).

Người chơi nào đạt được điều kiện này trước sẽ là người chiến thắng. Nếu toàn bộ bàn cờ đã được lấp đầy nhưng không có người chơi nào tạo được một chuỗi 5 quân liên tiếp, ván cờ sẽ được xem là hòa.

1.13. Các quy tắc quan trọng trong cách tính thắng thua

Trong cờ Caro, ngoài quy tắc cơ bản là sắp xếp được 5 quân liên tiếp, một số biến thể của trò chơi còn áp dụng các luật đặc biệt để tăng tính cân bằng và hấp dẫn:

Luật cấm chặn hai đầu (Áp dụng trong một số phiên bản cờ Caro hiện đại)

Nếu một chuỗi quân đã đạt đủ 5 quân liên tiếp nhưng bị đối thủ chặn cả hai đầu (không có ô trống ở hai đầu của dãy quân đó), thì nước đi đó không được tính là chiến thắng. Luật này nhằm hạn chế việc giành chiến thắng một cách dễ dàng và đòi hỏi người chơi phải có chiến thuật thông minh hơn.

- Luật thắng tuyệt đối:

Nếu một người chơi tạo được một chuỗi 6 quân liên tiếp hoặc dài hơn, thì họ vẫn được tính là thắng cuộc ngay lập tức, bất kể có bị chặn hay không. Quy tắc này áp dụng trong phiên bản cờ Caro truyền thống, đặc biệt là ở Việt Nam.

Luật cấm nước đi "4-4" và "3-3" (Áp dụng trong một số giải đấu chuyên nghiệp):

Người chơi không được phép thực hiện nước đi tạo ra hai chuỗi 4 quân liên tiếp (4-4) hoặc 3 quân liên tiếp (3-3) ở hai vị trí khác nhau cùng một lúc. Điều này nhằm ngăn chặn chiến thuật "dồn ép" đối thủ quá mức và tạo thêm chiều sâu chiến thuật cho trò chơi. Nếu một nước đi vi phạm quy tắc này, người chơi sẽ phải chọn một nước đi khác.

- Luật chặn một đầu:

Trong một số trường hợp, một người chơi có thể giành chiến thắng với một chuỗi 5 quân nếu chỉ bị chặn một đầu (tức là một đầu vẫn còn ô trống). Luật này thường được sử dụng trong các trận đấu không chính thức hoặc khi chơi giải trí.

1.14. Chiến thuật và phong cách chơi

Mặc dù luật chơi cờ Caro đơn giản, nhưng việc áp dụng các chiến thuật hợp lý là điều tối quan trọng để có thể giành chiến thắng. Một số chiến thuật phổ biến bao gồm:

Tấn công chủ động:

Người chơi nên chủ động xây dựng chuỗi quân cờ ngay từ đầu để tạo áp lực lên đối thủ. Việc chiếm các vị trí trung tâm của bàn cờ sẽ giúp mở rộng cơ hội tấn công theo nhiều hướng khác nhau.

- Phòng thủ chiến thuật:

Nếu không thể chủ động tấn công, người chơi cần phải tìm cách phá vỡ chuỗi quân của đối thủ bằng cách chặn trước khi họ kịp xếp đủ 5 quân liên tiếp. Một chiến thuật hiệu quả là đặt quân cờ sao cho đối phương buộc phải thay đổi chiến lược, từ đó mất đi lợi thế ban đầu.

- Bẫy đối thủ bằng nước đi kép:

Một chiến thuật quan trọng trong cờ Caro là tạo ra những nước đi kép, nơi mà bất kỳ lựa chọn nào của đối phương cũng sẽ dẫn đến một tình huống có lợi cho mình.

Ví dụ, nếu người chơi tạo ra hai chuỗi "4 quân liên tiếp" ở hai vị trí khác nhau, đối phương sẽ không thể chặn cả hai cùng một lúc, dẫn đến chiến thắng tất yếu.

- Duy trì sự linh hoạt trong chiến thuật:

Một người chơi giỏi sẽ không chỉ tập trung vào một chiến thuật cố định mà sẽ điều chỉnh linh hoạt dựa trên cách chơi của đối thủ. Đôi khi, việc phòng thủ đúng lúc có thể giúp tạo ra cơ hội phản công mạnh mẽ hơn.

1.15. Trường hợp đặc biệt và kết thúc ván đấu

- Một ván đấu có thể kết thúc theo ba cách chính:

- + Thắng cuộc: Khi một người chơi tạo ra được 5 quân liên tiếp theo đúng quy tắc.
- + Hòa cờ: Nếu toàn bộ bàn cờ đã kín chỗ mà không có ai chiến thắng.
- + Bị xử thua: Nếu một người chơi vi phạm luật chơi (chẳng hạn như đi vào ô đã có quân cờ hoặc phạm luật trong các giải đấu có quy tắc khắt khe hơn).

Một số phiên bản cờ Caro trực tuyến có thể áp dụng giới hạn thời gian cho mỗi lượt đi, giúp tránh tình trạng kéo dài quá lâu và tăng cường tính thử thách. Nhìn chung, luật chơi cờ Caro có thể được tóm gọn một cách đơn giản, nhưng khi đi sâu vào các biến thể và chiến thuật, trò chơi này trở nên cực kỳ thú vị và đầy thử thách. Việc nắm vững quy tắc sẽ giúp người chơi không chỉ tham gia một cách dễ dàng mà còn có thể áp dụng những chiến thuật thông minh để giành chiến thắng. Chính sự kết hợp giữa luật chơi đơn giản và chiều sâu chiến thuật đã giúp cờ Caro trở thành một trò chơi kinh điển, thu hút nhiều thế hệ người chơi trên toàn thế giới.

CHƯƠNG 2: PHÂN TÍCH VÀ GIẢI PHÁP CHO TRÒ CHƠI

2.1. Xác định bài toán và mục tiêu

2.1.1. Mục tiêu của trò chơi

Trò chơi Cờ Caro được xây dựng với mục tiêu tái hiện một trò chơi chiến thuật cổ điển, có lịch sử lâu đời và phổ biến rộng rãi trong nhiều nền văn hóa. Khác với Tic Tac Toe - trò chơi sử dụng lưới 3x3 với luật thắng đơn giản, Cờ Caro được chơi trên một bàn cờ lớn hơn (thường là 15x15) và yêu cầu người chơi phải tạo được một chuỗi liên tiếp 5 quân theo hàng ngang, hàng dọc hoặc đường chéo để giành chiến thắng. Chính sự mở rộng

này làm tăng đáng kể độ phức tạp của trò chơi, đòi hỏi người chơi không chỉ có tư duy chiến thuật mà còn phải có khả năng phân tích sâu và lên kế hoạch dài hạn cho từng nước đi.

Mục tiêu chính của dự án này không chỉ dừng lại ở việc tạo ra một trò chơi giải trí đơn thuần mà còn hướng đến nhiều khía cạnh khác:

Phát triển kỹ năng lập trình Python: Thông qua việc xây dựng trò chơi, người thực hiện có cơ hội rèn luyện và nâng cao kỹ năng lập trình Python, từ cách tổ chức mã nguồn, làm việc với các thư viện có sẵn đến việc xây dựng các thuật toán phức tạp.

Ứng dụng thuật toán AI trong trò chơi: Một trong những điểm quan trọng nhất của dự án là triển khai trí tuệ nhân tạo (AI) để tạo ra đối thủ máy tính có khả năng chơi một cách thông minh. AI trong trò chơi này sử dụng thuật toán Minimax kết hợp với heuristic để đánh giá và đưa ra nước đi tối ưu. Điều này giúp người thực hiện hiểu rõ hơn về cách hoạt động của các thuật toán tìm kiếm và đánh giá trạng thái trong trí tuệ nhân tạo.

Xây dựng giao diện đồ họa (GUI) thân thiện: Để tăng tính tương tác và trải nghiệm người dùng, trò chơi sử dụng thư viện Tkinter để xây dựng giao diện trực quan. Việc này giúp người thực hiện làm quen với lập trình giao diện đồ họa, xử lý sự kiện chuột và cập nhật màn hình động theo thời gian thực.

Rèn luyện tư duy thuật toán và kỹ năng giải quyết vấn đề: Việc thiết kế và lập trình trò chơi Cờ Caro đòi hỏi khả năng tư duy thuật toán, từ xây dựng logic kiểm tra thắng thua đến tối ưu hóa thuật toán AI để tăng tốc độ xử lý và nâng cao hiệu suất.

Áp dụng kiến thức vào thực tế: Thay vì chỉ học lý thuyết lập trình, việc triển khai một dự án thực tế như trò chơi Cờ Caro giúp người thực hiện có cơ hội áp dụng những kiến thức đã học vào một sản phẩm hoàn chỉnh, từ đó tăng cường khả năng làm việc thực tế và sáng tạo.

Mở rộng khả năng phát triển trong tương lai: Dự án này không chỉ dừng lại ở phiên bản cơ bản mà còn có tiềm năng nâng cấp và mở rộng, chẳng hạn như cải thiện giao diện,

bổ sung nhiều cấp độ AI, tích hợp khả năng chơi trực tuyến hoặc sử dụng đồ họa hiện đại hơn bằng các thư viện như Pygame hay PyQt.i.

2.1.2. Đối tượng chính

- Bàn cờ (Board)

Là khu vực chính của trò chơi, nơi người chơi thực hiện các nước đi. Trong chương trình, bàn cờ thường được biểu diễn dưới dạng một ma trận hai chiều (mảng 2D) để theo dõi trạng thái của từng ô (trống, X, O). Bàn cờ có thể có nhiều kích thước khác nhau, nhưng phổ biến nhất là 15x15 hoặc lớn hơn. Giao diện bàn cờ có thể được hiển thị bằng thư viện Tkinter, giúp người chơi dễ dàng tương tác với trò chơi.

- Có hai chế độ chơi:

- + Người chơi thật (PVP): Do người dùng điều khiển, đặt X hoặc O lên bàn cờ.

- + Trí tuệ nhân tạo (AI): Được lập trình để đưa ra nước đi dựa trên thuật toán Minimax kết hợp Alpha-Beta Pruning. Người chơi thực hiện nước đi theo lượt, tuân theo luật chơi để tìm cách chiến thắng hoặc ngăn chặn đối thủ.

- Bộ xử lý trò chơi (Game Controller)

Chịu trách nhiệm quản lý trạng thái trò chơi, bao gồm:

- + Xác định ai đang chơi.

- + Kiểm tra điều kiện thắng/thua/hòa.

- + Kiểm soát lượt đi giữa người chơi và AI.

- + Giao diện người dùng (User Interface – UI)

- Giao diện giúp người chơi tương tác với trò chơi một cách trực quan, có thể được thiết kế bằng thư viện Tkinter với các thành phần:

- + Menu chính (chơi với người/AI, thiết lập bàn cờ).

- + Bàn cờ hiển thị trạng thái hiện tại.

- + Các nút chức năng như "Bắt đầu mới", "Gợi ý nước đi".

- + Bộ đánh giá nước đi (Evaluator)

Thành phần quan trọng của AI, giúp xác định nước đi nào là tốt nhất. Sử dụng các thuật toán như Minimax để phân tích giá trị của từng ô trên bàn cờ. Đưa ra điểm số cho các vị trí tiềm năng, ưu tiên những nước có khả năng dẫn đến chiến thắng nhanh nhất.

- Quản lý dữ liệu trò chơi (Game Model)

Chứa thông tin về bàn cờ, trạng thái trò chơi và lịch sử nước đi. Hỗ trợ tính năng lưu và tải lại trò chơi. Có thể dùng các cấu trúc dữ liệu như danh sách (list) hoặc từ điển (dictionary) để lưu trạng thái bàn cờ.

2.1.3. Giới thiệu về AI và thuật toán minimax

- Giới thiệu về trí tuệ nhân tạo AI

Trong thời đại công nghệ phát triển mạnh mẽ, Trí tuệ nhân tạo (Artificial Intelligence - AI) đã và đang trở thành một trong những lĩnh vực quan trọng nhất, có ảnh hưởng sâu rộng đến hầu hết các khía cạnh của cuộc sống. AI không chỉ là một công nghệ mang tính đột phá trong ngành công nghiệp phần mềm mà còn đang dần trở thành một phần không thể thiếu trong nhiều lĩnh vực như y tế, giáo dục, tài chính, sản xuất, và đặc biệt là ngành công nghiệp trò chơi điện tử. Về bản chất, AI là khả năng của máy tính hoặc hệ thống có thể tự động học hỏi từ dữ liệu, phân tích, nhận diện các mẫu và đưa ra quyết định thông minh mà không cần sự can thiệp trực tiếp của con người. AI có thể mô phỏng tư duy con người, tự học hỏi và cải thiện hiệu suất làm việc theo thời gian, giúp giải quyết nhiều bài toán phức tạp mà trước đây chỉ con người mới có thể thực hiện.

- Dựa trên khả năng và mức độ thông minh, AI có thể được chia thành các loại sau:

- + AI yếu (Weak AI): Là loại AI được thiết kế để thực hiện một nhiệm vụ cụ thể, chẳng hạn như trợ lý ảo (Siri, Google Assistant), hệ thống gợi ý nội dung trên YouTube hoặc thuật toán AI trong game.

+ AI mạnh (Strong AI): Là loại AI có khả năng tư duy và hành động như con người, có thể tự học và đưa ra các quyết định phức tạp hơn.

+ Siêu AI (Super AI): Đây là một khái niệm giả thuyết về AI có trí thông minh vượt xa con người. Hiện tại, loại AI này vẫn chưa tồn tại trong thực tế.

- Ứng dụng của AI trong đời sống:

+ Trí tuệ nhân tạo không chỉ là một khái niệm lý thuyết mà đã được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống, góp phần thay đổi cách con người làm việc và sinh hoạt hằng ngày:

+ Trong giáo dục: AI hỗ trợ xây dựng các hệ thống học tập thông minh, giúp cá nhân hóa quá trình học tập cho từng học sinh, đưa ra các bài giảng phù hợp với năng lực và nhu cầu học tập của mỗi người.

+ Trong y tế: AI được sử dụng để phân tích hình ảnh y khoa, hỗ trợ chẩn đoán bệnh sớm và giúp bác sĩ đưa ra phác đồ điều trị hiệu quả.

+ Trong tài chính: AI giúp phân tích dữ liệu tài chính khổng lồ, dự đoán xu hướng thị trường và hỗ trợ các quyết định đầu tư.

+ Trong sản xuất công nghiệp: AI được ứng dụng trong dây chuyền sản xuất tự động, giúp tối ưu hóa hiệu suất và giảm thiểu chi phí lao động.

+ Trong công nghiệp giải trí và trò chơi điện tử: AI giúp tạo ra các nhân vật NPC (Non-Player Character) thông minh hơn, tạo ra trải nghiệm chơi game thực tế hơn và phát triển các thuật toán giúp trò chơi trở nên thử thách hơn.

- Ứng dụng của AI trong dự án này

Trong phạm vi của bài tiểu luận này, AI được ứng dụng vào việc phát triển một đối thủ máy tính thông minh cho trò chơi Cờ Caro, giúp người chơi có thể thử thách bản thân trước một hệ thống có khả năng phân tích và đưa ra nước đi hợp lý. Việc ứng dụng AI trong trò chơi này không chỉ giúp nâng cao trải nghiệm chơi game mà còn là cơ hội để tìm

hiểu cách thức hoạt động của các thuật toán trí tuệ nhân tạo trong thực tế. Trí tuệ nhân tạo trong trò chơi Cờ Caro được xây dựng dựa trên thuật toán Minimax kết hợp với cắt tỉa Alpha-Beta để giúp AI tìm ra nước đi tối ưu nhất. Cụ thể:

AI sẽ phân tích toàn bộ trạng thái của bàn cờ, đánh giá từng nước đi có thể xảy ra và chọn nước đi mang lại lợi thế cao nhất.

Thuật toán Minimax giúp AI tính toán tất cả các kịch bản có thể xảy ra sau mỗi nước đi và đưa ra quyết định dựa trên việc tối ưu hóa kết quả cho chính nó trong khi cố gắng giảm cơ hội thắng của đối thủ.

Cắt tỉa Alpha-Beta giúp giảm số lượng trạng thái cần kiểm tra, từ đó tăng tốc quá trình tính toán và giúp AI chơi nhanh hơn.

- Giới thiệu về thuật toán minimax

Trong lĩnh vực trí tuệ nhân tạo, có rất nhiều phương pháp khác nhau để giúp máy tính có thể đưa ra quyết định một cách thông minh. Một trong những phương pháp cổ điển và quan trọng nhất trong việc phát triển AI cho các trò chơi chiến lược là thuật toán Minimax. Đây là một thuật toán thuộc nhóm AI sơ khai (Weak AI), bởi vì nó không có khả năng học hỏi hay thích nghi như các hệ thống AI hiện đại (ví dụ: học máy - Machine Learning hoặc học sâu - Deep Learning), mà chỉ đơn thuần thực hiện tính toán dựa trên việc xem xét tất cả các nước đi có thể xảy ra và lựa chọn nước đi tốt nhất dựa trên logic cố định. Mặc dù Minimax không phải là AI mạnh có khả năng học tập và thích nghi, nhưng nó vẫn giúp máy tính có thể đưa ra quyết định tối ưu bằng cách phân tích tất cả các kịch bản có thể xảy ra. Nhờ vậy, Minimax được ứng dụng rộng rãi trong nhiều trò chơi đối kháng hai người như Cờ Caro, Cờ vua, Cờ tướng, Tic-Tac-Toe, và nhiều loại trò chơi khác có tính chất chiến lược.

- Nguyên lý hoạt động của thuật toán minimax

Thuật toán Minimax là một thuật toán đệ quy mạnh mẽ được sử dụng rộng rãi trong trí tuệ nhân tạo (AI) để ra quyết định tối ưu trong các trò chơi chiến lược hai người chơi.

Đây là một trong những phương pháp nền tảng của lý thuyết trò chơi, giúp máy tính có thể lựa chọn nước đi tốt nhất bằng cách giả định rằng đối thủ cũng sẽ chơi một cách hoàn hảo.

- Về cơ hoạt động:

Thuật toán Minimax hoạt động dựa trên Game Tree (cây trạng thái trò chơi), trong đó mỗi nút trong cây đại diện cho một trạng thái của bàn cờ tại một thời điểm cụ thể.

Ở mỗi lượt đi, thuật toán sẽ tạo ra tất cả các trạng thái có thể có trong tương lai, sau đó đánh giá và lựa chọn trạng thái tốt nhất dựa trên giả định rằng đối thủ cũng đang chơi một cách tối ưu.

Minimax là một thuật toán đệ quy, trong đó thuật toán sẽ lặp lại quy trình này cho đến khi đạt đến trạng thái kết thúc của trò chơi (tức là có người thắng, hòa, hoặc không còn nước đi hợp lệ).

Sau khi khám phá toàn bộ cây trạng thái trò chơi, thuật toán sẽ quay ngược lại từ các trạng thái cuối cùng và chọn nước đi tối ưu nhất cho người chơi.

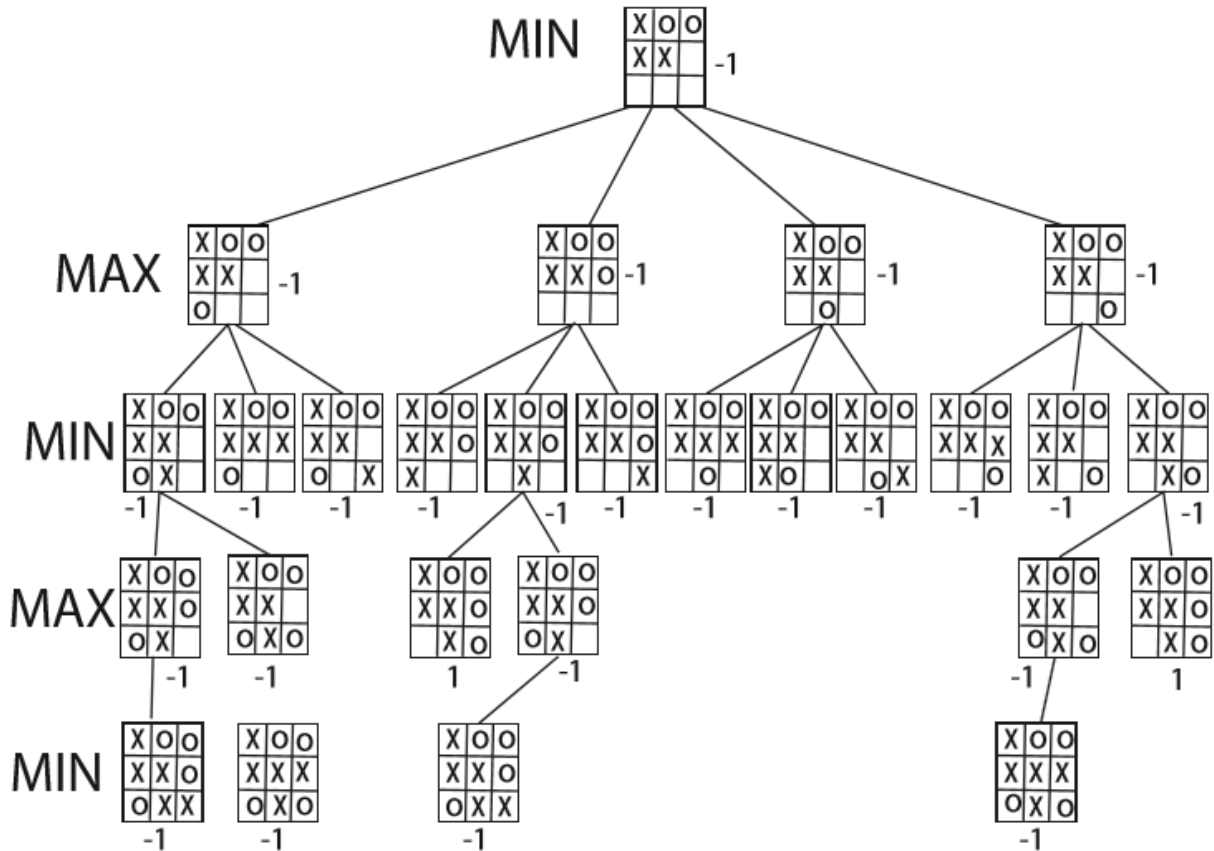
- Cấu trúc cơ bản của thuật toán Minimax trong thuật toán này, hai người chơi trong trò chơi được định nghĩa như sau:

- + Người chơi MAX (thường là AI) luôn muốn tối đa hóa lợi thế của mình.
- + Người chơi MIN (đối thủ) luôn muốn tối thiểu hóa lợi thế của AI, tức là tìm cách cản trở chiến lược của AI.

Cả hai người chơi hoạt động đối lập nhau, trong đó:

- + MAX sẽ chọn nước đi có điểm số cao nhất để gia tăng lợi thế.
- + MIN sẽ chọn nước đi có điểm số thấp nhất để hạn chế lợi thế của MAX.

Minimax đảm bảo rằng AI không chỉ tìm nước đi tốt nhất cho mình, mà còn phải xem xét các phản ứng có thể có của đối thủ, từ đó đưa ra quyết định tốt nhất trong mọi tình huống.



Mô tả cây xử lý minimax trong trò chơi

- Thuật toán Minimax hoạt động như thế nào?

+ Mỗi nút trong cây biểu diễn một trạng thái cụ thể của bàn cờ.

+ Mỗi nhánh tương ứng với một nước đi khả thi từ trạng thái hiện tại.

- Tạo hai loại nút trong cây trạng thái:

+ Nút MAX: Đại diện cho lượt chơi của AI, nơi AI sẽ cố gắng chọn nước đi có điểm số cao nhất.

+ Nút MIN: Đại diện cho lượt chơi của đối thủ, nơi AI giả định rằng đối thủ sẽ chọn nước đi có điểm số thấp nhất để giảm lợi thế của AI.

Duyệt cây trạng thái bằng tìm kiếm theo chiều sâu (Depth-First Search - DFS): Thuật toán Minimax sẽ đệ quy kiểm tra tất cả các nước đi có thể xảy ra, mở rộng cây trạng thái trò chơi cho đến khi gặp trạng thái kết thúc (thắng, thua hoặc hòa).

- Khi đạt đến trạng thái kết thúc, thuật toán sẽ gán điểm số dựa trên kết quả:

+ AI thắng → Điểm số +10

+ AI thua → Điểm số -10

+ Hòa → Điểm số 0

- Khi thuật toán duyệt đến các trạng thái kết thúc, nó sẽ quay ngược lại dần dần, tại mỗi lượt nó sẽ chọn nước đi tối ưu theo nguyên tắc:

+ MAX chọn điểm số cao nhất (nước đi có lợi nhất cho AI).

+ MIN chọn điểm số thấp nhất (nước đi có lợi nhất cho đối thủ).

Quá trình này tiếp tục cho đến khi thuật toán quay trở lại trạng thái ban đầu của bàn cờ, nơi AI sẽ chọn nước đi tối ưu nhất để thực hiện.

- Giả sử AI có ba lựa chọn nước đi:

Nước đi	Kịch bản xảy ra sau nước đi này	Điểm số ước tính
A	AI có thể thắng sau 2 lượt	+10
B	Đối thủ có thể thắng ngay sau 1 lượt	-10
C	Trò chơi có thể dẫn đến hòa	0

Ví dụ về minimax trong caro

+ Nếu AI chơi ở vị trí A, AI có cơ hội thắng cao nhất (+10 điểm).

+ Nếu AI chơi ở vị trí B, đối thủ có thể thắng ngay (-10 điểm), nên đây là nước đi tệ nhất.

+ Nếu AI chơi ở vị trí C, trận đấu có thể hòa (0 điểm), đây là một lựa chọn trung lập.

+ Như vậy, thuật toán Minimax sẽ chọn nước đi A vì đây là nước đi có lợi nhất cho AI.

```
def minimax(self, depth, alpha, beta, is_maximizing, ai_label, board_state):
    score = self.evaluator.evaluate_board(ai_label, board_state)
    if depth >= self.max_depth or abs(score) >= 100000 or self.game.is_tied():
        return score - depth if is_maximizing else score + depth

    moves = self.game.get_nearby_moves()
    start_time = time.time()

    move_scores = []
    for row, col in moves:
        if time.time() - start_time > self.time_limit:
            break
        temp_board = copy.deepcopy(board_state)
        temp_board[row][col] = ai_label if is_maximizing else ("O" if ai_label == "X" else "X")
        score = self.evaluator.evaluate_board(ai_label, temp_board)
        center_distance = math.sqrt((row - self.board_size//2)**2 + (col - self.board_size//2)**2)
        score += int(100 / (center_distance + 1))
        move_scores.append((row, col, score))

    move_scores.sort(key=lambda x: x[2], reverse=is_maximizing)
    sorted_moves = [(row, col) for row, col, _ in move_scores]

    if is_maximizing:
        best_score = float('-inf')
        for row, col in sorted_moves:
            if time.time() - start_time > self.time_limit:
                break
            temp_board = copy.deepcopy(board_state)
            temp_board[row][col] = ai_label
            score = self.minimax(depth + 1, alpha, beta, False, ai_label, temp_board)
            best_score = max(best_score, score)
            alpha = max(alpha, best_score)
            if beta <= alpha:
                break
        return best_score
```

Một phần đoạn code Minimax trong đề tài

- Ứng dụng của Minimax, trong công việc, học tập và vào dự án:

Thuật toán Minimax có nhiều ứng dụng quan trọng trong cả học tập lẫn công việc, đặc biệt trong các lĩnh vực liên quan đến trí tuệ nhân tạo, lập trình game và ra quyết định. Trong học tập, Minimax là một công cụ tuyệt vời giúp sinh viên hiểu rõ về các khái niệm cơ bản của AI, từ việc đánh giá trạng thái trò chơi, lập trình đệ quy đến tối ưu hóa thuật toán. Nhờ vậy, sinh viên có thể rèn luyện tư duy logic, kỹ năng phân tích vấn đề cũng như cách áp dụng thuật toán vào thực tế. Không chỉ giới hạn trong lĩnh vực trò chơi, Minimax còn được sử dụng trong nghiên cứu khoa học dữ liệu và trí tuệ nhân tạo để giải quyết các bài toán tối ưu hóa và ra quyết định dựa trên dữ liệu.

Trong công việc, thuật toán này được áp dụng trong nhiều lĩnh vực khác nhau như phát triển trò chơi điện tử, tài chính và kinh doanh. Trong lập trình game, Minimax được sử dụng để xây dựng AI có khả năng chơi cờ vua, cờ vây, tic-tac-toe hay cờ Caro một cách thông minh. Các nhà phát triển game có thể tận dụng thuật toán này để tạo ra những đối thủ máy có chiến thuật rõ ràng, giúp nâng cao trải nghiệm người chơi. Ngoài ra, trong tài

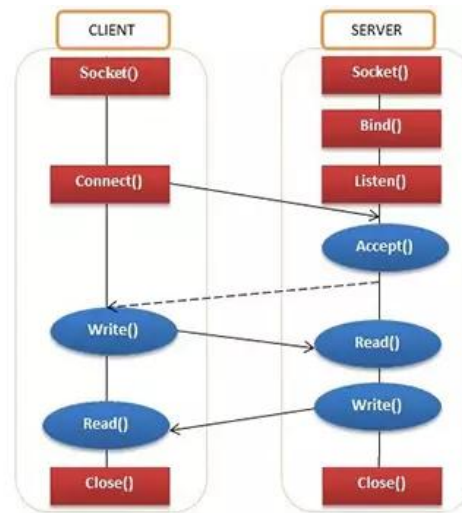
chính và kinh doanh, Minimax có thể được sử dụng để mô phỏng các tình huống cạnh tranh giữa các doanh nghiệp, giúp tìm ra phương án tối ưu nhằm đạt được lợi thế trong thị trường.

Ứng dụng của Minimax trong dự án cờ Caro mang lại nhiều lợi ích quan trọng. Trước hết, nó giúp xây dựng một đối thủ AI thông minh, có khả năng phân tích và đưa ra nước đi tối ưu thay vì chọn ngẫu nhiên. Điều này giúp tăng tính thử thách và hấp dẫn cho trò chơi, khiến người chơi có cảm giác như đang đối đầu với một đối thủ thực sự có chiến thuật. Thứ hai, việc áp dụng Minimax giúp nhóm phát triển hiểu sâu hơn về cách AI hoạt động trong các trò chơi chiến thuật, đồng thời thực hành các khái niệm quan trọng trong lập trình AI như tìm kiếm cây trạng thái, đánh giá nước đi và tối ưu hóa thuật toán. Cuối cùng, dự án này không chỉ giúp củng cố kiến thức về thuật toán Minimax mà còn là bước đệm để nhóm có thể tiếp cận với các kỹ thuật AI nâng cao hơn trong tương lai.

2.1.4. Giới thiệu thư viện socket

Socket là một giao diện lập trình ứng dụng mạng (API) được sử dụng để truyền và nhận dữ liệu trên Internet. Nó đóng vai trò như một "cánh cửa" đặc biệt, cho phép các ứng dụng trên máy tính có thể giao tiếp với nhau qua mạng. Khi hai chương trình muốn trao đổi dữ liệu, chúng cần thiết lập một liên kết giao tiếp hai chiều (two-way communication). Điểm đầu và điểm cuối của liên kết này chính là socket, giúp kết nối hai tiến trình (process) trên hai thiết bị khác nhau hoặc trên cùng một thiết bị. Hãy tưởng tượng socket như hai đầu dây điện thoại giúp hai người có thể trò chuyện với nhau, hoặc như một hệ thống bưu điện giúp gửi thư đến đúng người nhận. Một chức năng quan trọng khác của socket là giúp tăng TCP (Transmission Control Protocol) định danh ứng dụng mà dữ liệu sẽ được gửi tới. Điều này được thực hiện thông qua sự ràng buộc với một cổng (port) – một số nguyên cụ thể giúp xác định ứng dụng đích. Khi một ứng dụng được liên kết với một cổng nhất định, nó có thể thiết lập kết nối giữa client (máy khách) và server (máy chủ), đảm bảo dữ liệu được truyền tải chính xác. Trong dự án cờ caro online, socket chính là "cầu nối" vô hình giúp hai người chơi kết nối qua mạng. Thay vì ngồi đối diện, mỗi người chơi thao tác trên máy tính, và mọi nước đi được truyền tải tức thì. Hãy tưởng tượng socket như một cặp "ống nghe" đặc biệt: khi một người chơi thực hiện nước đi, dữ liệu sẽ được "gửi" qua socket đến

máy tính đối thủ, và ngược lại. Để socket hoạt động chính xác, nó cần xác định địa chỉ IP của máy tính đối thủ (giống như địa chỉ nhà trong hệ thống bưu điện) và cổng (port) của ứng dụng cờ caro (giống như số nhà). Nhờ đó, dữ liệu truyền đi đúng đích, giúp trò chơi vận hành mượt mà, mang lại trải nghiệm tốt nhất cho người chơi. Nhờ có socket, không chỉ trò chơi cờ caro online mà nhiều ứng dụng mạng khác như truyền dữ liệu, ứng dụng chat, dịch vụ web đều có thể hoạt động một cách hiệu quả và ổn định.



Mô hình socket

- Mô tả mô hình:

- + Trước tiên chúng ta sẽ tạo ra một máy chủ bằng cách mở một socket - `Socket()`
- + Sau đó chúng ta sẽ liên kết nó với một host hoặc một máy và một port - `Bind()`
- + Tiếp theo server sẽ bắt đầu lắng nghe trên port đó - `Listen()`
- + Yêu cầu kết nối từ client được gửi tới server - `Connect()`
- + Server sẽ accept yêu cầu từ client và sau đó kết nối được thiết lập - `Accept()`
- + Bây giờ cả hai đều có thể gửi và nhận tin tại thời điểm đó - `Read()` / `Write()`
- + Và cuối cùng khi hoàn thành chúng có thể đóng kết nối - `Close()`

CHƯƠNG 3: PHÁT TRIỂN VÀ MÔ TẢ TÍNH NĂNG TRÒ CHƠI

3.1. Phát triển đề tài

Dự án phát triển trò chơi cờ caro tập trung vào việc ứng dụng thuật toán Minimax để xây dựng một AI thông minh, hỗ trợ hai tính năng chính: cho phép người chơi đấu cờ với máy và gợi ý nước đi tối ưu trong quá trình chơi. Thuật toán Minimax được chọn vì khả năng đưa ra quyết định tối ưu trong các trò chơi đối kháng, giúp AI (MAX) tối đa hóa cơ hội chiến thắng, trong khi giả định người chơi (MIN) cũng chơi một cách tối ưu để giảm thiểu lợi ích của AI.

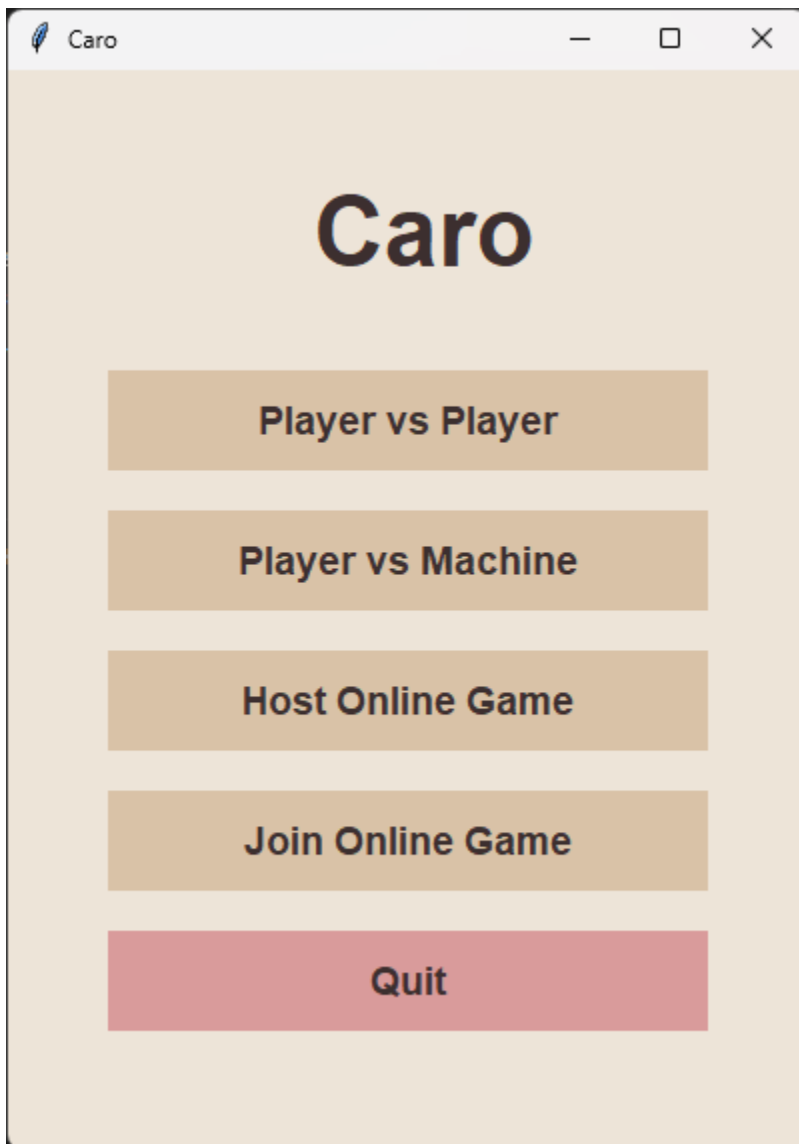
Để tăng hiệu suất đề tài còn ứng dụng thuật toán kết hợp với cắt tỉa Alpha-Beta, giúp giảm số lượng trạng thái cần đánh giá mà vẫn đảm bảo kết quả chính xác.

Về giao diện, dự án sử dụng thư viện Tkinter của Python để tạo một giao diện đồ họa thân thiện, cho phép người chơi dễ dàng tương tác với bàn cờ, hiển thị gợi ý nước đi, và theo dõi trạng thái trò chơi.

Ngoài ra, để hỗ trợ tính năng chơi cùng bạn bè, dự án tích hợp socket để kết nối nhiều người chơi trong cùng mạng, cho phép hai người chơi đấu với nhau trên các thiết bị khác nhau thông qua giao thức mạng.

Sự kết hợp giữa thuật toán Minimax, giao diện Tkinter, và kết nối socket không chỉ mang lại trải nghiệm chơi cờ thú vị mà còn giúp người chơi cải thiện kỹ năng thông qua gợi ý từ AI, đồng thời tạo điều kiện để kết nối và thi đấu với bạn bè một cách dễ dàng.

3.2. Chạy thử trò chơi, giao diện trò chơi



Giao diện chính của trò chơi

- Giao diện trong hình là menu khởi động (Start Menu) của trò chơi Caro, một trò chơi chiến thuật dạng Tic-Tac-Toe mở rộng trên bàn cờ 15x15, nơi người chơi cần xếp 5 ký

hiệu liên tiếp (X hoặc O) để thắng. Menu này là điểm nhập chính, cung cấp các tùy chọn chế độ chơi:

- + Player vs Player (PvP): Hai người chơi đấu với nhau trên cùng một máy.
- + Player vs Machine (PvAI): Người chơi đấu với AI, sử dụng thuật toán thông minh để chọn nước đi.
- + Host Online Game: Người chơi tạo một phòng chơi trực tuyến và chờ người khác tham gia.
- + Join Online Game: Người chơi tham gia vào một phòng chơi trực tuyến đã được tạo.
- + Quit: Thoát trò chơi.

- Giao diện được thiết kế để:

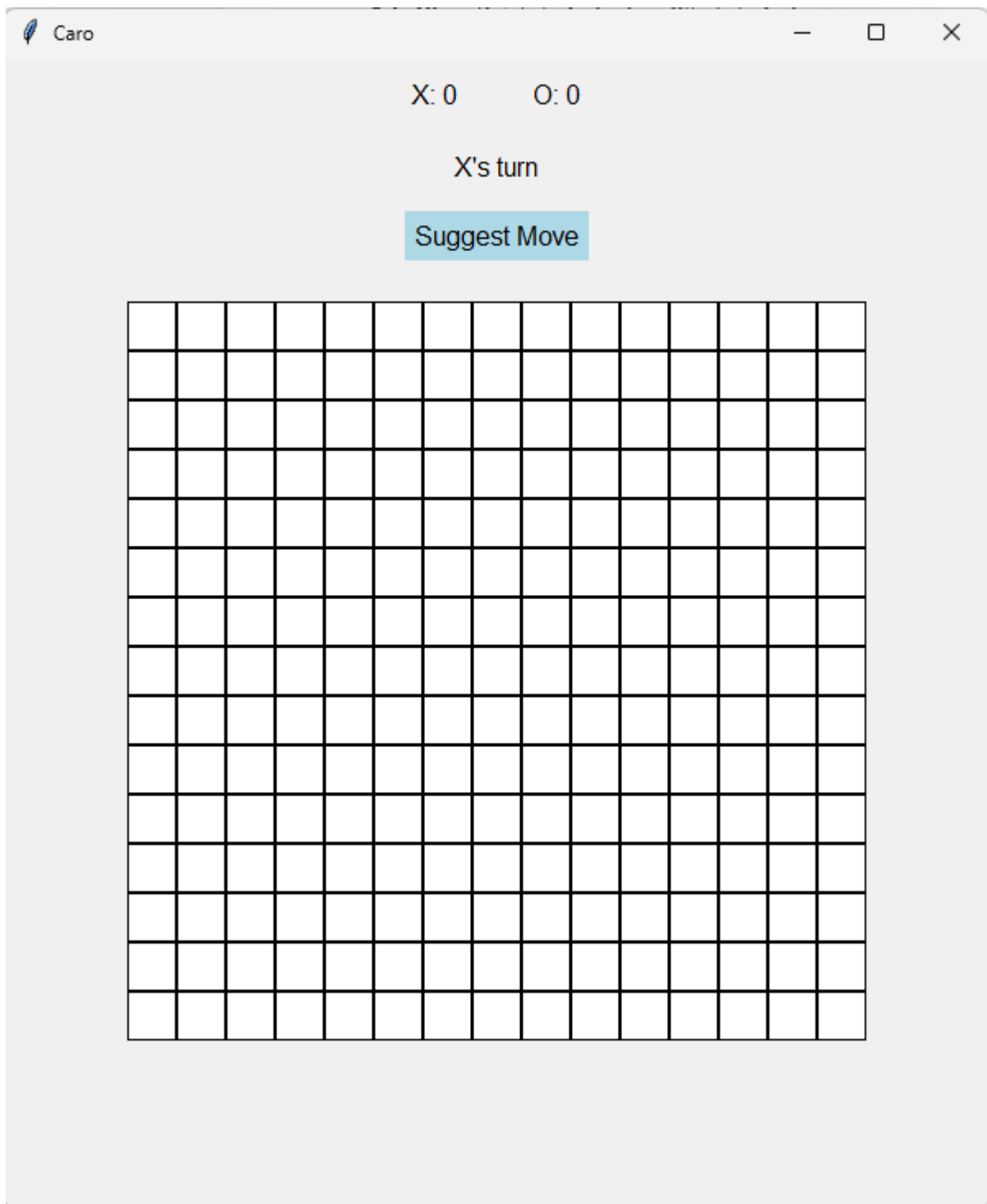
- + Tạo trải nghiệm người dùng trực quan, dễ sử dụng.
- + Hỗ trợ nhiều chế độ chơi, bao gồm cả chơi trực tuyến.
- + Đảm bảo giao diện đơn giản nhưng đầy đủ chức năng.

- Mục đích:

- + Menu này là bước đầu tiên trong luồng hoạt động của trò chơi:

- Người chơi chọn chế độ từ menu.

- + Hệ thống khởi tạo trò chơi tương ứng (PvP, PvAI, chơi trực tuyến, hoặc thoát).
- + Chuyển sang giao diện bàn cờ hoặc thoát ứng dụng.



Giao diện khi bắt đầu trận đấu

- Giao diện trong hình là bàn cờ Caro của trò chơi, giao diện này được hiển thị sau khi người chơi chọn chế độ chơi từ menu chính (ví dụ: Player vs Player, Player vs Machine, hoặc chơi trực tuyến).

- Các thành phần chính bao gồm:

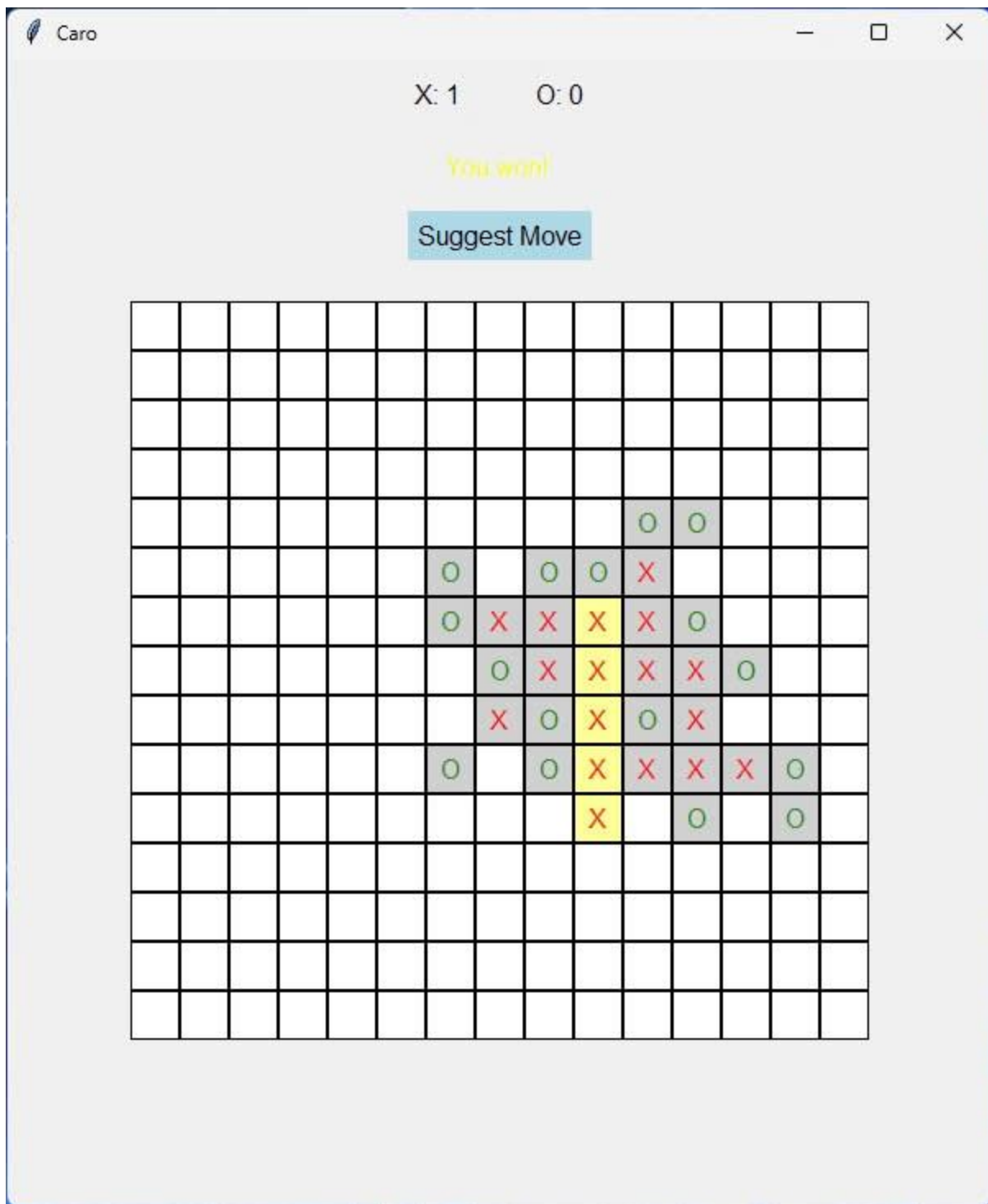
- + Điểm số: Hiển thị điểm của X và O (hiện tại là X: 0, O: 0).
- + Thông báo lượt chơi: Hiển thị lượt hiện tại ("X's turn").
- + Nút "Suggest Move": Cho phép người chơi yêu cầu gợi ý nước đi.
- + Bàn cờ 15x15: Lưới ô vuông để người chơi đặt ký hiệu X hoặc O.

- Giao diện này được thiết kế để:

- + Cung cấp không gian chơi trực quan, dễ thao tác.
- + Hỗ trợ người chơi với tính năng gợi ý nước đi (đặc biệt hữu ích trong chế độ Player vs Machine).
- + Hiển thị trạng thái trò chơi (lượt chơi, điểm số) một cách rõ ràng.

- Mục đích:

- + Giao diện bàn cờ là nơi diễn ra toàn bộ hoạt động chơi chính của trò chơi:
- + Người chơi (hoặc AI) lần lượt đặt ký hiệu X hoặc O vào các ô trên bàn cờ.
- + Hệ thống kiểm tra điều kiện thắng (5 ký hiệu liên tiếp) hoặc hòa (hết ô trống).
- + Cung cấp gợi ý nước đi nếu người chơi yêu cầu, đặc biệt trong chế độ chơi với máy



Giao diện khi chiến thắng

- Giao diện khi chiến thắng hiển thị:

- + Điểm số: Hiển thị điểm của X và O (X: 1, O: 0, cho thấy X đã thắng 1 ván).
- + Thông báo kết quả: "You won!" cho biết người chơi X đã thắng.
- + Nút "Suggest Move": Cho phép người chơi yêu cầu gợi ý nước đi (hiện không khả dụng vì ván đã kết thúc).
- + Bàn cờ 15x15: Lưới ô vuông chứa các ký hiệu X (đỏ) và O (xanh), với 5 ô X liên tiếp được làm nổi bật bằng màu vàng.

- Giao diện này được thiết kế để:

- + Cung cấp không gian chơi trực quan, dễ thao tác.
- + Thông báo rõ ràng kết quả ván đấu (thắng, thua, hoặc hòa).
- + Làm nổi bật các ô dẫn đến chiến thắng để người chơi dễ nhận biết.

Mục đích: Giao diện này thể hiện trạng thái kết thúc của một ván đấu

- + Người chơi X đã xếp được 5 ký hiệu liên tiếp, giành chiến thắng.
- + Hệ thống cập nhật điểm số và hiển thị thông báo chiến thắng.
- + Người chơi có thể bắt đầu ván mới hoặc quay lại menu chính (tùy vào chế độ chơi).

- Phân tích trạng thái bàn cờ ký hiệu

- + X (màu đỏ): Người chơi X (có thể là người chơi hoặc AI, tùy chế độ).
- + O (màu xanh): Người chơi O (có thể là người chơi thứ hai hoặc AI).

- Ô thắng

- + 5 ô X liên tiếp được làm nổi bật bằng màu vàng, nằm ở hàng 8, từ cột 7 đến cột 11 (tọa độ: (8,7), (8,8), (8,9), (8,10), (8,11)).

+ Đây là một hàng ngang, đáp ứng điều kiện thắng của trò chơi Caro (5 ký hiệu liên tiếp).

Trạng thái:

+ Ván đấu đã kết thúc, X thắng, điểm số được cập nhật (X: 1, O: 0).

+ Các ô xung quanh khu vực trung tâm có nhiều X và O, cho thấy một ván đấu tập trung vào chiến thuật tấn công và phòng thủ.

- Giải thích thuật toán

Giao diện này liên quan đến một số thuật toán quan trọng, đặc biệt trong việc kiểm tra chiến thắng và xử lý trạng thái trò chơi. Thuật toán kiểm tra thắng (trong `CaroGame._check_winner`) mục đích xác định xem có người chơi nào thắng sau mỗi nước đi hay không.

- Cách hoạt động

+ Khi một nước đi được thực hiện (ví dụ: X đặt ở ô (8,11)), hệ thống kiểm tra từ ô đó theo 4 hướng: ngang, dọc, chéo chính, chéo phụ.

+ Đếm số ký hiệu liên tiếp cùng loại (X hoặc O)

+ Hướng dương: Từ ô hiện tại, tăng tọa độ theo hướng.

+ Hướng âm: Từ ô hiện tại, giảm tọa độ theo hướng.

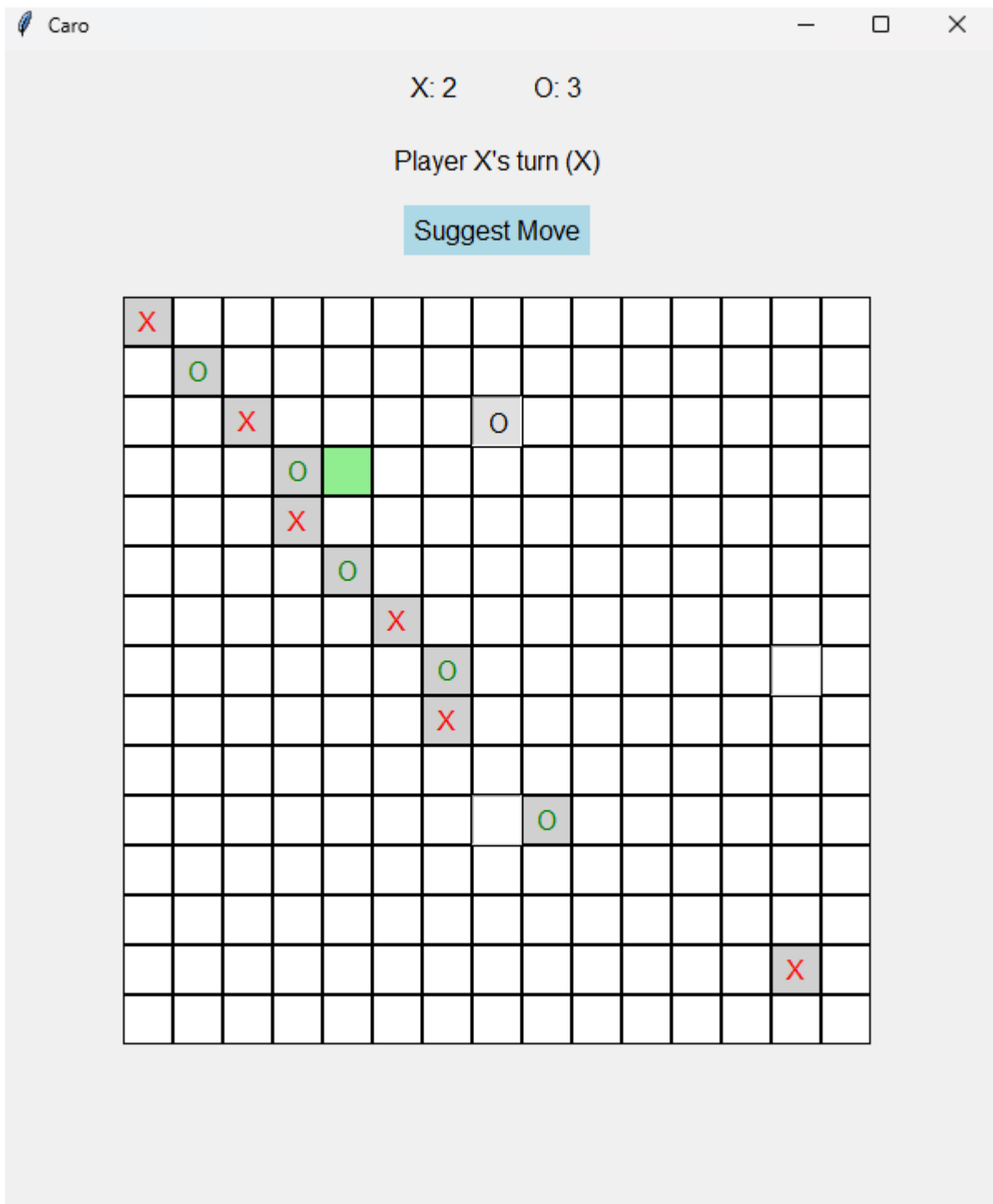
+ Trong hình, kiểm tra hướng ngang:

+ Từ ô (8,11), đếm ngược về bên trái: (8,10), (8,9), (8,8), (8,7) đều là X \rightarrow 5 ô liên tiếp.

+ Đếm tiếp về bên phải: không có ô nào nữa (đã đủ 5).

+ Nếu tìm thấy 5 ký hiệu liên tiếp, người chơi đó thắng. Hệ thống lưu danh sách 5 ô thắng (để làm nổi bật) và đặt trạng thái `has_winner` thành `True`.

- + Nếu không có người thắng và bàn cờ đầy, trò chơi hòa.
 - + Độ phức tạp: $O(1)$ cho mỗi nước đi, vì chỉ kiểm tra một số lượng cố định ô (tối đa 5 ô mỗi hướng).
- Thuật toán cập nhật trạng thái trò chơi xử lý trạng thái trò chơi sau mỗi nước đi, bao gồm cập nhật điểm số và thông báo kết quả. Cách hoạt động:
- + Sau mỗi nước đi, kiểm tra xem có người thắng không (gọi `Carogame.has_Winner`)
 - + Nếu có, làm nổi bật các ô thắng (trong hình là 5 ô X màu vàng).
 - + Cập nhật điểm số: Tăng điểm của người thắng (X: 1, O: 0).
 - + Hiển thị thông báo "You won!" (hoặc "X won!" tùy chế độ).
 - + Nếu không có người thắng, kiểm tra xem trò chơi có hòa không (gọi `CaroGame.is_tired`)
 - + Nếu hòa, hiển thị thông báo "Tied game!".
 - + Nếu trò chơi vẫn tiếp tục, chuyển lượt cho người chơi tiếp theo.



Tính năng gợi ý đường đi

- Giới thiệu lý thuyết và ý nghĩa của giao diện

+ Điểm số: Hiện thị điểm của X và O (X: 2, O: 3, cho thấy X đã thắng 2 ván, O thắng 3 ván).

+ Thông báo lượt chơi: "Player X's turn (X)" cho biết hiện là lượt của người chơi X.

+ Nút "Suggest Move": Cho phép người chơi yêu cầu gợi ý nước đi.

+ Bàn cờ 15x15: Lưới ô vuông chứa các ký hiệu X (đỏ) và O (xanh), với một ô được làm nổi bật màu xanh lá (ô gợi ý).

- Giao diện này được thiết kế để:

+ Cung cấp không gian chơi trực quan, dễ thao tác.

+ Hỗ trợ người chơi với tính năng gợi ý nước đi (đặc biệt hữu ích trong chế độ Player vs Machine).

+ Hiện thị trạng thái trò chơi (lượt chơi, điểm số) một cách rõ ràng.

- Mục đích giao diện này thể hiện trạng thái của một ván đấu đang diễn ra:

+ Người chơi X và O đã thực hiện một số nước đi, tập trung ở góc trên bên trái bàn cờ.

+ Ô (2,3) được làm nổi bật màu xanh lá, cho thấy đây là nước đi được gợi ý sau khi người chơi nhấn "Suggest Move".

+ Người chơi X có thể tiếp tục nước đi để tiến tới chiến thắng hoặc chặn đối thủ.

- Phân tích trạng thái bàn cờ ký hiệu:

+ X (màu đỏ): Người chơi X (có thể là người chơi hoặc AI, tùy chế độ).

+ O (màu xanh): Người chơi O (có thể là người chơi thứ hai hoặc AI).

- Ô gợi ý:

+ Ô (2,3) được làm nổi bật màu xanh lá, cho thấy đây là nước đi được gợi ý cho người chơi X.

- Trạng thái:

+ Ván đấu đang diễn ra, chưa có người thắng (không có 5 ký hiệu liên tiếp).

+ Các nước đi tập trung ở góc trên bên trái, với một số mẫu 2-3 ký hiệu liên tiếp (ví dụ: 3 ô X chéo từ (0,0) đến (2,2)).

+ Điểm số (X: 2, O: 3) cho thấy đây là ván đấu thứ 6 (tổng cộng 5 ván trước đó).

- Giải thích thuật toán

Giao diện này liên quan đến một số thuật toán quan trọng, đặc biệt trong việc gợi ý nước đi (ô (2,3) được làm nổi bật) và xử lý trạng thái trò chơi.

- Thuật toán gợi ý nước đi (trong SuggestAI)

Khi người chơi nhấn nút "Suggest Move", hệ thống sử dụng lớp SuggestAI để tìm nước đi tối ưu, sử dụng thuật toán Minimax với Alpha-Beta Pruning, được tối ưu với bảng chuyển vị và các chiến lược ưu tiên. Tìm nước đi tốt nhất cho người chơi X, ưu tiên thắng, chặn đối thủ, hoặc tạo cơ hội chiến thuật.

- Kiểm tra nước thắng ngay lập tức:

+ Kiểm tra xem có nước đi nào giúp X thắng ngay (5 ký hiệu liên tiếp) không.

+ Trong trường hợp này, chưa có nước thắng ngay, nên bỏ qua bước này.

- Chặn nước thắng của đối thủ:

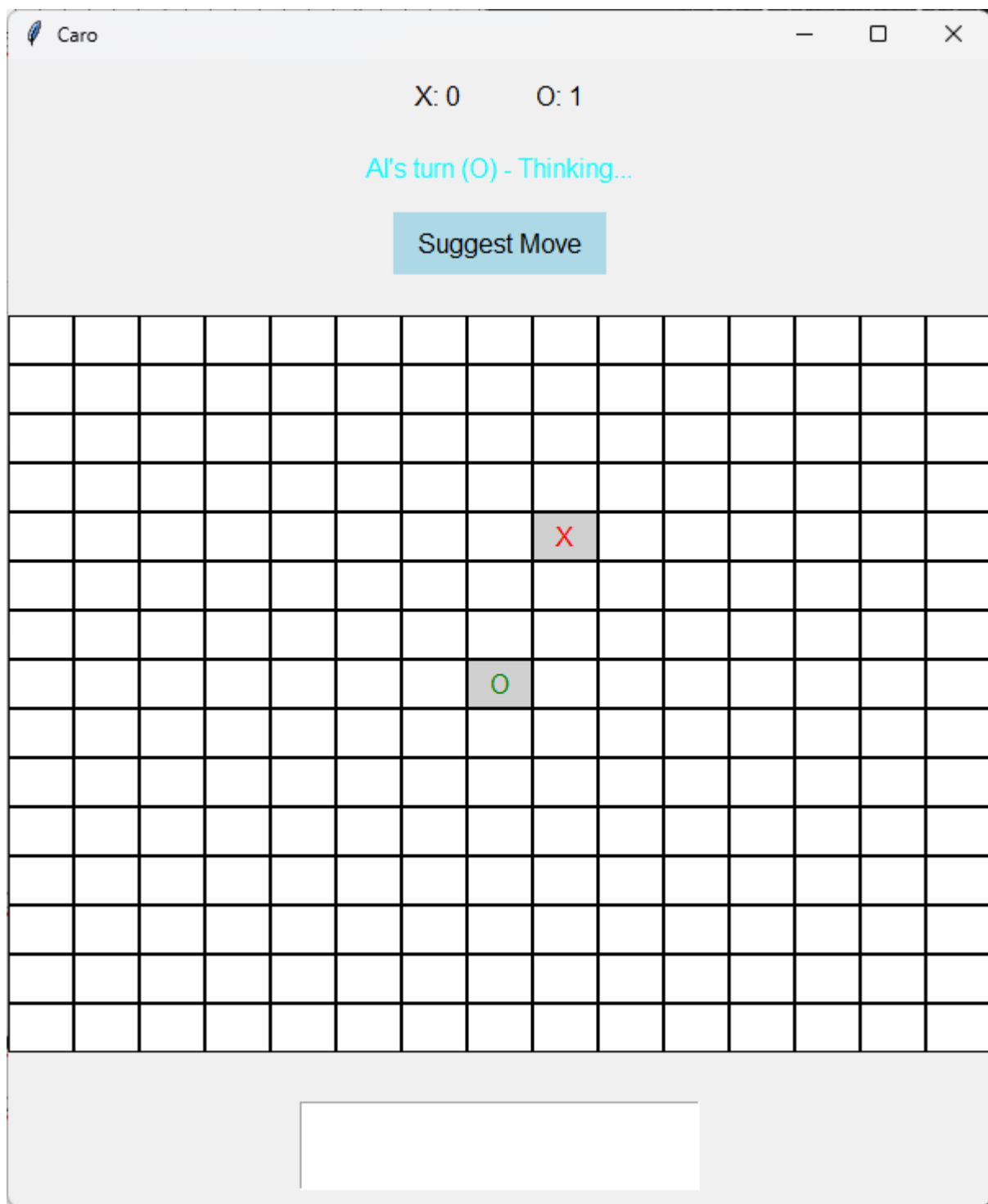
+ Kiểm tra xem O có thể thắng ở lượt tiếp theo không (4 ký hiệu liên tiếp với 1 đầu mở).

- + Trong hình, O có 3 ô liên tiếp chéo (tọa độ (1,1), (2,2), (3,3)), nhưng chưa đủ 4, nên chưa cần chặn ngay.
- Đi ở trung tâm (nếu bàn cờ trống):
 - + Bàn cờ đã có nước đi, nên bỏ qua bước này.
- Minimax với Alpha-Beta Pruning:
 - + Minimax: Thuật toán tìm kiếm đệ quy, mô phỏng các nước đi có thể xảy ra:
 - + Lượt của X (maximizing): Chọn nước đi có điểm cao nhất.
 - + Lượt của O (minimizing): Chọn nước đi có điểm thấp nhất.
- Alpha-Beta Pruning: Cắt bỏ các nhánh không cần thiết trong cây tìm kiếm:
 - + Alpha: Điểm số tốt nhất mà X (maximizing) có thể đạt được.
 - + Beta: Điểm số thấp nhất mà O (minimizing) có thể ép buộc.
 - + Nếu một nhánh có điểm số vượt quá beta (khi maximizing) hoặc nhỏ hơn alpha (khi minimizing), bỏ qua nhánh đó.
 - + Bảng chuyển vị: Lưu trữ kết quả của các trạng thái bàn cờ đã tính toán để tránh lặp lại
- Tối ưu:
 - + Chỉ xem xét các nước đi gần các ô đã chơi (get_nearby_moves), giảm không gian tìm kiếm.
 - + Lọc top 7 nước đi tiềm năng dựa trên đánh giá sơ bộ để giảm thời gian tính toán.
 - + Giới hạn thời gian (3 giây) để đảm bảo phản hồi nhanh.
- Đánh giá bàn cờ sử dụng BoardEvaluator để tính điểm cho trạng thái bàn cờ:
 - + Ưu tiên cao cho 5 ký hiệu liên tiếp (thắng ngay).

- + Điểm cao cho 4 hoặc 3 ký hiệu liên tiếp với đầu mở (cơ hội thắng). Ví dụ: 3 ô X chéo từ (0,0) đến (2,2) có 2 đầu mở (ô (3,3) và ô (-1,-1)), nên được điểm cao.
- + Điểm âm nếu đối thủ có 4 hoặc 3 ký hiệu liên tiếp (cần chặn). Ví dụ: 3 ô O chéo từ (1,1) đến (3,3) có điểm âm.
- + Ưu tiên tạo bẫy (ví dụ: hai chuỗi 3 ký hiệu liên tiếp).
- + Ô (2,3) được chọn vì nó có thể mở rộng chuỗi 3 ô X chéo thành 4 ô (tọa độ (0,0), (1,1), (2,2), (2,3)), tạo cơ hội thắng.

- Độ phức tạp:

- + $O(b^d)$ trong trường hợp xấu nhất, với b là số nhánh (số nước đi có thể) và d là độ sâu tìm kiếm ($\text{max_depth} = 10$).
- + Alpha-Beta Pruning và bảng chuyển vị giảm đáng kể số nhánh cần tính toán, thường xuống $O(b^{(d/2)})$ trong trường hợp tốt.



Tính năng đánh với AI

- Giải thích thuật toán:

Lớp CaroAI sử dụng thuật toán Minimax với Alpha-Beta Pruning làm cốt lõi, kết hợp với các chiến lược ưu tiên để tăng hiệu quả. Dưới đây là chi tiết các thuật toán và cách chúng hoạt động.

- Mục đích: Tìm nước đi tối ưu bằng cách mô phỏng các nước đi có thể xảy ra, tối đa hóa điểm số của AI và tối thiểu hóa điểm số của đối thủ cách hoạt động:

- Điều kiện dừng:

- + Nếu độ sâu đạt `max_depth` (6), hoặc bàn cờ hòa (`self.game.is_tied()`), hoặc điểm số tuyệt đối ≥ 100000 (thắng/thua), trả về điểm số.

- + Điểm số được điều chỉnh theo độ sâu: trừ đi độ sâu nếu là maximizing (AI), cộng độ sâu nếu là minimizing (đối thủ), để ưu tiên chiến thắng nhanh.

- Lấy danh sách nước đi:

- + Sử dụng `self.game.get_nearby_moves()` để lấy các nước đi gần các ô đã chơi, giảm không gian tìm kiếm.

- Đánh giá sơ bộ và sắp xếp:

- + Với mỗi nước đi, tạo một bản sao của bàn cờ (`copy.deepcopy`) và đặt ký hiệu (AI hoặc đối thủ).

- + Tính điểm bằng `self.evaluator.evaluate_board`.

- + Cộng điểm dựa trên khoảng cách đến trung tâm: $100 / (\text{center_distance} + 1)$, ưu tiên các nước đi gần trung tâm.

- + Sắp xếp các nước đi theo điểm số (giảm dần nếu maximizing, tăng dần nếu minimizing).

- Minimax với Alpha-Beta Pruning:

- + Maximizing (AI): Chọn nước đi có điểm cao nhất, cập nhật alpha.

- + Minimizing (đối thủ): Chọn nước đi có điểm thấp nhất, cập nhật beta.
- + Alpha-Beta Pruning: Nếu $\beta \leq \alpha$, cắt bỏ nhánh vì không thể cải thiện kết quả.
- Giới hạn thời gian:
 - + Nếu thời gian vượt quá `self.time_limit` (1.5 giây), dừng tính toán và trả về kết quả hiện tại.
- Độ phức tạp:
 - + $O(b^d)$ trong trường hợp xấu nhất, với b là số nhánh (số nước đi có thể) và d là độ sâu (`max_depth = 6`).
 - + Alpha-Beta Pruning giảm đáng kể số nhánh, thường xuống $O(b^{(d/2)})$ trong trường hợp tốt.
- Thuật toán chọn nước đi cho AI (trong `get_ai_move`) tìm nước đi tốt nhất cho AI, ưu tiên thắng, chặn đối thủ, và tối ưu chiến thuật cách hoạt động:
- Kiểm tra nước thắng ngay lập tức:
 - + Gọi `self.evaluator.check_winning_move` để tìm nước đi giúp AI thắng ngay (5 ký hiệu liên tiếp).
 - + Nếu có, trả về nước đi đó.
- Chặn nước thắng của đối thủ:
 - + Gọi `self.evaluator.check_opponent_winning_move` để tìm nước đi chặn đối thủ thắng (4 ký hiệu liên tiếp với đầu mở, hoặc 3 ký hiệu với 2 đầu mở).
 - + Nếu có, trả về nước đi đó.

- Đi ở trung tâm (nếu bàn cờ trống):

+ Nếu bàn cờ chưa có nước đi nào, chọn ô trung tâm (7,7 trên bàn 15x15) vì đây là vị trí chiến lược.

- Đánh giá sơ bộ các nước đi:

+ Lấy danh sách nước đi hợp lệ (get_nearby_moves hoặc toàn bộ ô trống nếu không có nước đi gần).

+ Với mỗi nước đi, đặt ký hiệu AI, tính điểm bằng BoardEvaluator, và cộng điểm dựa trên khoảng cách đến trung tâm.

+ Sắp xếp các nước đi theo điểm số (giảm dần).

- Minimax:

+ Với mỗi nước đi trong danh sách đã sắp xếp, gọi minimax để tính điểm chính xác hơn (xem xét phản ứng của đối thủ).

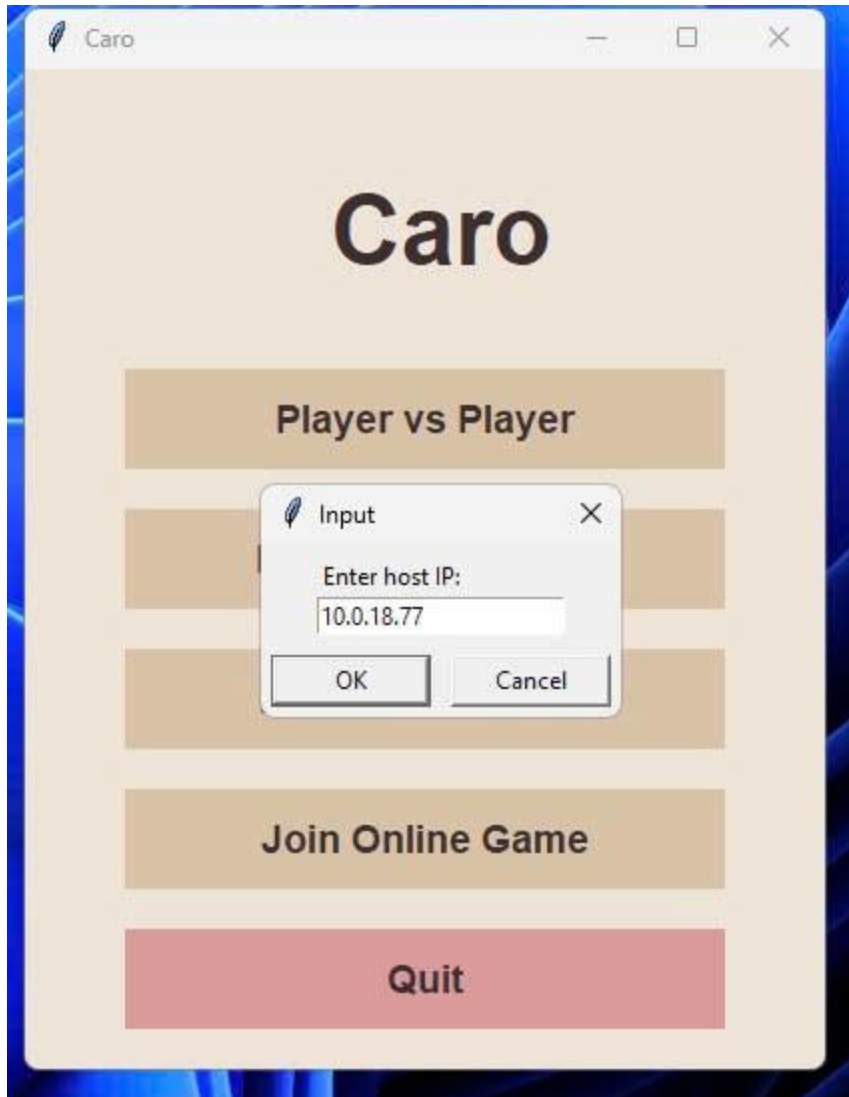
+ Chọn nước đi có điểm cao nhất.

- Giới hạn thời gian:

+ Nếu vượt quá 1.5 giây, chọn nước đi tốt nhất hiện tại (dựa trên đánh giá sơ bộ).

- Độ phức tạp:

+ Tương tự như thuật toán Minimax, nhưng được tối ưu bằng cách kiểm tra thắng/chặn trước và giới hạn không gian tìm kiếm.



Giao diện nhập ip để chơi online

- Giải thích giao diện và ý nghĩa:

- + Menu chính: Giao diện hiển thị các tùy chọn:
- + "Player vs Player": Chơi hai người trên cùng một máy.
- + "Join Online Game": Tham gia một ván đấu trực tuyến bằng cách nhập IP của host.
- + "Quit": Thoát trò chơi.

- Hộp thoại "Input host IP":

- + Hiện thị khi người chơi nhấn "Join Online Game".
- + Yêu cầu người chơi nhập địa chỉ IP của máy host (người tạo phòng chơi trực tuyến).
- + Có hai nút: "OK" (xác nhận IP và kết nối) và "Cancel" (hủy bỏ).

- Ý nghĩa

+ Chế độ chơi trực tuyến: Tính năng "Join Online Game" cho phép người chơi tham gia một ván đấu Caro với người chơi khác qua mạng (LAN hoặc Internet).

+ Địa chỉ IP: Địa chỉ IP (10.0.18.77 trong hình) là địa chỉ của máy host, được sử dụng để thiết lập kết nối giữa client (người chơi tham gia) và host (người tạo phòng).

+ Mục đích: Hộp thoại này là bước đầu tiên để thiết lập kết nối mạng, cho phép hai người chơi ở hai máy khác nhau đấu với nhau.

- Làm sao để lấy địa chỉ IP như trong hình?

Địa chỉ IP 10.0.18.77 trong hình là địa chỉ của máy host (người tạo phòng chơi trực tuyến). Để lấy được địa chỉ IP này, bạn cần thực hiện các bước sau:

- Bước 1: Xác định máy host

- + Máy host là máy của người chơi đã chọn "Host Online Game" từ menu chính.
- + Khi người chơi chọn "Host Online Game", trò chơi sẽ khởi tạo một máy chủ (server) trên máy của họ và hiện thị địa chỉ IP của máy đó.

- Bước 2: Lấy địa chỉ IP của máy host

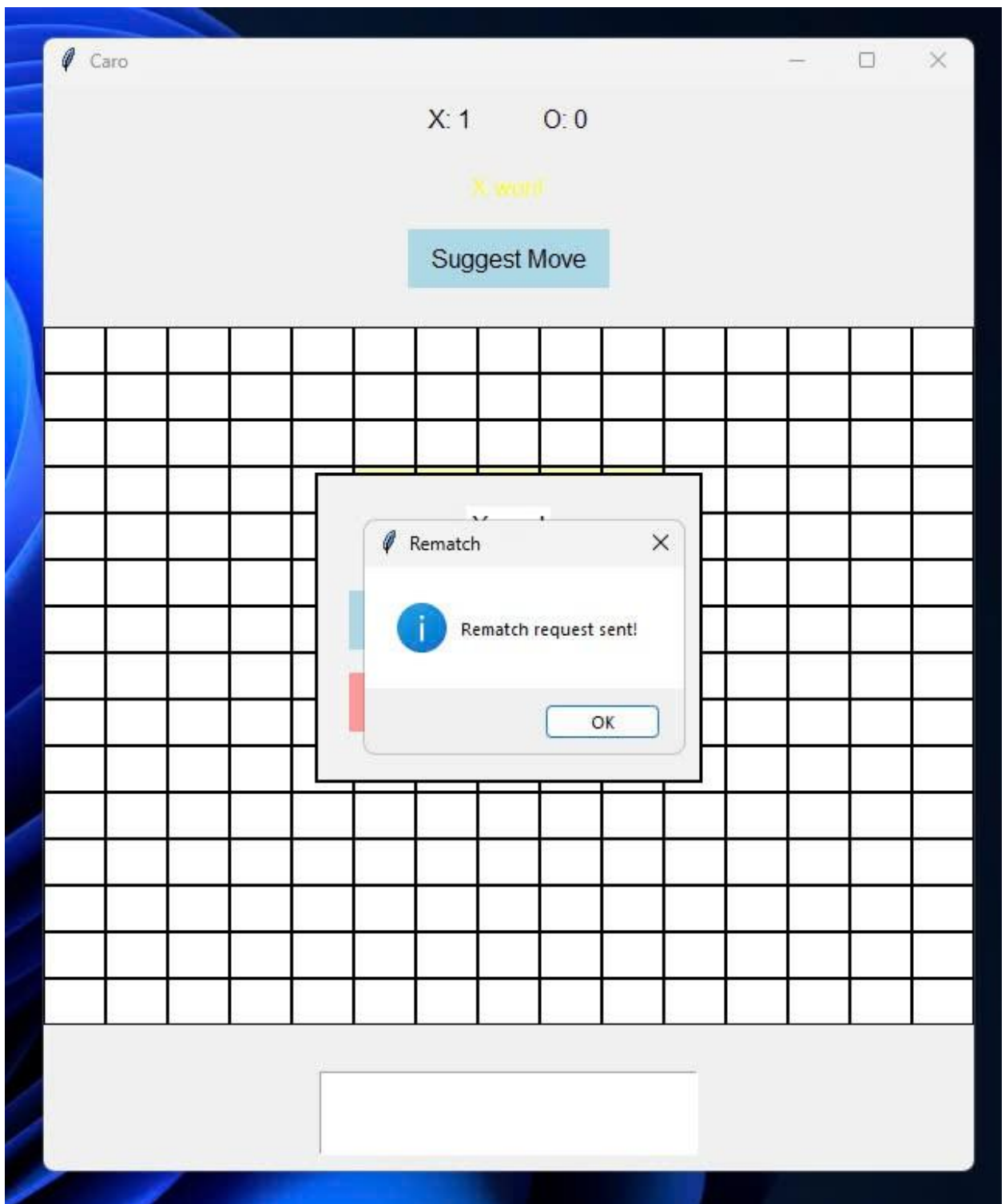
- + Trong trò chơi Caro này, khi người chơi chọn "Host Online Game", trò chơi sẽ sử dụng NetworkManager (trong model/network.py) để tạo một máy chủ.
- + Theo thiết kế, trò chơi có thể hiển thị địa chỉ IP của máy host trên giao diện (ví dụ: một thông báo "Your IP: 10.0.18.77, waiting for opponent...").

- Cung cấp IP cho người chơi (client)
 - + Người chơi host cần gửi địa chỉ IP (ví dụ: 10.0.18.77) cho người chơi client (người muốn tham gia).
 - + Có thể gửi qua tin nhắn, email, hoặc bất kỳ phương thức nào.
- Nhập IP vào hộp thoại
 - + Người chơi client mở trò chơi, chọn "Join Online Game".
 - + Nhập địa chỉ IP (10.0.18.77) vào hộp thoại "Input host IP".
 - + Nhấn "OK" để kết nối.
- Cách hoạt động của tính năng "Join Online Game"
- Host tạo phòng:
 - + Người chơi host chọn "Host Online Game" từ menu.
 - + Trò chơi gọi `NetworkManager.host_game` (trong `model/network.py`) để tạo một máy chủ socket.
 - + Máy chủ lắng nghe trên một cổng (mặc định là 12345 trong mã nguồn).
 - + Địa chỉ IP của máy host (ví dụ: 10.0.18.77) được sử dụng để client kết nối.
- Client tham gia:
 - + Người chơi client chọn "Join Online Game".
 - + Hộp thoại "Input host IP" hiện lên, yêu cầu nhập IP của host.
 - + Khi nhấn "OK", trò chơi gọi `NetworkManager.join_game` với IP đã nhập (10.0.18.77) và cổng 12345.
 - + Nếu kết nối thành công, client và host bắt đầu trao đổi dữ liệu (nước đi, tin nhắn, v.v.) qua `NetworkService`.

- Chơi trực tuyến:

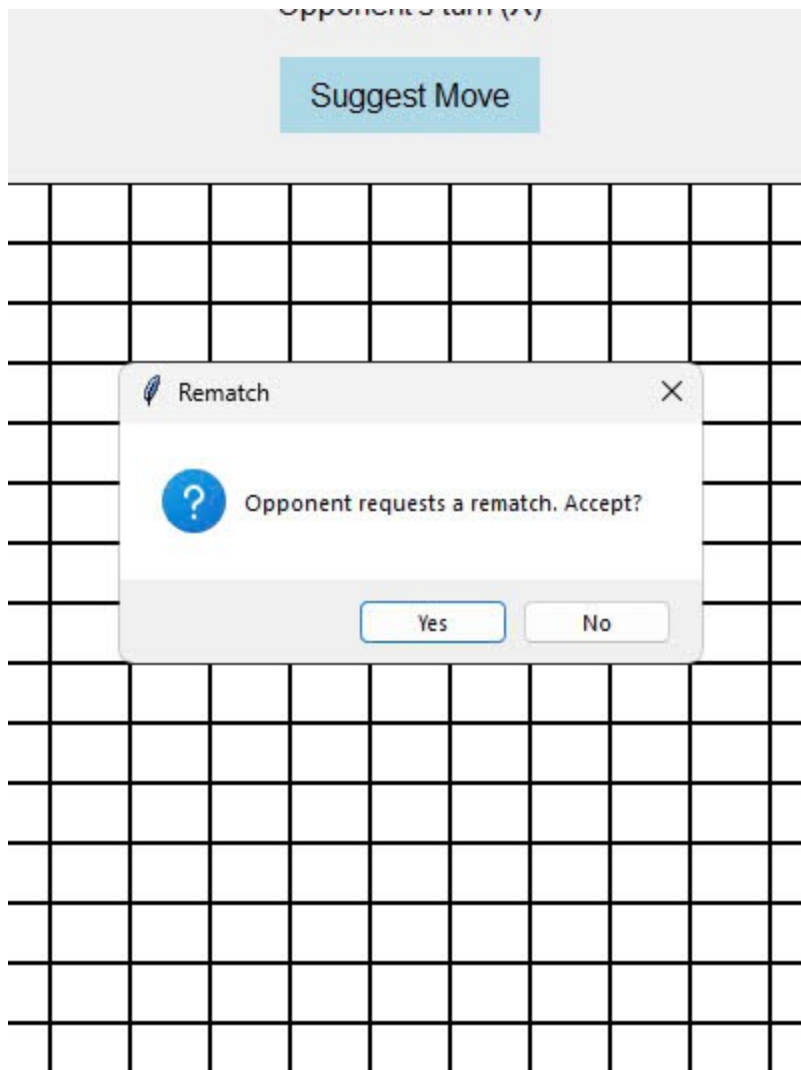
- + Host và client lần lượt gửi nước đi qua mạng.

- + OnlineGameManager (trong model/online_game_manager.py) đồng bộ hóa trạng thái trò chơi giữa hai máy.



Chức năng chơi lại trong chế độ online của host

- Chức năng chơi lại (Rematch) trong chế độ online (Host):
 - + Mô tả: Sau khi ván đấu kết thúc, host gửi yêu cầu chơi lại (rematch) tới client.
 - + Hiện thị: Hộp thoại "Rematch" với thông báo "Rematch request sent!" và nút "OK".
- Chức năng:
 - + Gửi tin nhắn rematch qua mạng tới client.
 - + Chờ client chấp nhận hoặc từ chối.
 - + Nếu chấp nhận, reset bàn cờ và bắt đầu ván mới.
 - + Nếu từ chối, quay lại menu chính.



Chức năng chấp nhận chơi lại từ đối phương

- Chức năng chơi lại :

+ Mô tả: Sau khi ván đấu kết thúc, 2 bên nếu có 1 bên yêu cầu chơi lại sẽ hiện thông báo.

+ Hiện thị: Hộp thoại "Rematch" với thông báo "Opponent requests a rematch. Accept?" và hai nút "Yes"/"No".

- Chức năng

+ Nhận tin nhắn rematch từ host qua mạng.

+ Nếu nhấn "Yes": Gửi xác nhận, reset bàn cờ, bắt đầu ván mới.

+ Nếu nhấn "No": Gửi từ chối, quay lại menu chính.

3.3. Những thiếu sót của dự án

- Chế độ về người dùng giao diện (UI/UX)

+ Không có menu thoát trận trong lúc chơi: Khi người chơi đang ở trong trò chơi, không có nút hoặc menu để thoát trận hoặc quay lại màn hình chính. Điều này gây bất lợi, đặc biệt nếu người chơi muốn dừng giữa chừng hoặc bắt đầu lại.

+ Giao diện chưa thân thiện và hấp dẫn: Hiện tại, giao diện trò chơi khá đơn giản, chưa có ứng dụng đồ họa hoặc hoạt hình để làm trò chơi trở nên sinh động hơn. Việc bổ sung thêm dấu hiệu đánh dấu nước đi, thương mại chuyển động khi đánh cờ hoặc thay đổi màu sắc khi thắng/thua có thể giúp nâng cao trải nghiệm của người dùng.

+ Chưa có âm thanh và ứng dụng: Trò chơi không có âm thanh ứng dụng khi thiết lập quân cờ, chiến thắng, chiến thắng hoặc thông báo nước đi của AI. Điều này làm giảm tính hấp dẫn và thiếu đi sự tương tác trực tiếp giữa người chơi và trò chơi.

- Chế độ trải nghiệm game

+ Không có cấp độ điều chỉnh tùy chỉnh của AI (Chế độ nâng, giảm độ khó): AI luôn sử dụng thuật toán Minimax cùng với tìm kiếm độ sâu tương tự, không có tùy chọn thay đổi độ khó để phù hợp với người chơi mới hoặc những người chơi có kinh nghiệm. Nếu AI quá mạnh, người chơi mới sẽ nhanh chóng chán nản, trong khi nếu AI quá yếu, người chơi có kinh nghiệm sẽ tìm thấy chán.

+ Thiếu tính năng hoàn tác nước đi (Undo/Redo): Trong một số trường hợp, người chơi có thể muốn quay lại nước đi trước đó để sửa lỗi, nhưng hiện tại tính năng này chưa được hỗ trợ trong trò chơi. Công việc bổ sung Undo/Redo giúp tăng tính hoạt động và cải thiện trải nghiệm chơi game.

+ Không có đồng hồ bấm giờ (Hẹn giờ): Trò chơi không có bộ đếm thời gian giới hạn cho mỗi lượt đi, khiến chiến đấu có thể kéo dài dù người chơi hoặc AI suy nghĩ

quá lâu. Thêm một bộ đếm thời gian trợ giúp trò chơi có tốc độ nhanh hơn và tạo thêm nhiều thử thách cho người chơi.

+ Chưa có hệ thống lưu trữ kết quả và số điểm: Trò chơi không có bảng xếp hạng hoặc lưu trữ kết quả chiến đấu, khiến người chơi không thể theo dõi thành tích hoặc so sánh kết quả với những lần chơi trước.

3.4. Giải pháp cải thiện và nâng cấp trong tương lai

- Cải thiện giao diện người dùng (UI/UX) để nâng cao trải nghiệm như:

+ Thêm menu thoát trận khi đang chơi: Cung cấp tùy chọn thoát hoặc quay về menu chính mà không cần đóng ứng dụng.

+ Thêm các nút thoát trận khi đang chơi và các nút reset trận đấu khi đang chơi nút hòa nút đầu hàng.

- Cải thiện đồ họa và hiển thị trạng thái trò chơi như:

+ Thay đổi cách hiển thị bàn cờ bằng các hình ảnh trực quan hơn.

+ Tạo hiệu ứng khi đánh cờ (ví dụ: đổi màu ô vừa được đánh).

+ Thêm hiệu ứng âm thanh: Cải thiện trải nghiệm bằng cách thêm âm thanh khi đánh cờ, khi có người chiến thắng hoặc khi đến lượt chơi.

+ Bổ sung đồng hồ bấm giờ (Timer): Giới hạn thời gian suy nghĩ để tăng độ kịch tính và tránh tình trạng người chơi kéo dài trận đấu quá lâu.

- Mở rộng chế độ chơi

+ Thêm chế độ chơi khác ngoài caro truyền thống, có thêm các biến thể của cờ caro như có thể cho phép 3-4 người chơi cùng lúc thay vì 1-1, các cờ khác như gomoku, caro ultimate

+ Tích hợp bảng xếp hạng và điểm số: Lưu trữ kết quả các trận đấu để người chơi có thể theo dõi thành tích của mình.

- + Thêm chế độ hướng dẫn chơi (Tutorial Mode): Hướng dẫn người mới cách đánh cờ và cung cấp gợi ý chiến thuật để cải thiện khả năng chơi.
- + Chế độ phân tích ván đấu: Sau khi trận đấu kết thúc, cung cấp bảng phân tích các nước đi và đề xuất cách chơi tốt hơn.
- + Chế độ giải đố (Puzzle Mode): Tạo ra các thế cờ đặc biệt để người chơi luyện tập chiến thuật.
- + Lưu và tải lại ván đấu: Hỗ trợ lưu trạng thái trận đấu để tiếp tục chơi sau này.
- + Hỗ trợ nhiều cấp độ AI: Thêm các mức độ khó khác nhau để người chơi có thể thử thách bản thân theo từng trình độ.

CHƯƠNG 4: TỔNG KẾT VÀ ĐỊNH HƯỚNG TƯƠNG LAI

Dự án Cờ Caro sử dụng AI Minimax là một nỗ lực toàn diện nhằm xây dựng một trò chơi chiến thuật truyền thống kết hợp với công nghệ trí tuệ nhân tạo. Qua quá trình nghiên cứu, thiết kế, lập trình và kiểm thử, nhóm đã đạt được nhiều thành tựu đáng kể, đồng thời nhận ra một số hạn chế cần cải thiện để hướng đến sản phẩm hoàn thiện hơn. Dưới đây là phân tổng kết chi tiết về kết quả đạt được cũng như các định hướng phát triển trong tương lai, nhằm mang lại trải nghiệm tốt nhất cho người chơi và mở rộng tiềm năng thương mại của dự án.

4.1. Kết quả đạt được của dự án

Xây dựng thành công trò chơi Cờ Caro với giao diện thân thiện và trực quan, giao diện đồ họa được hoàn thiện. Nhờ việc sử dụng thư viện Tkinter của Python, nhóm đã thiết kế một giao diện trực quan, với bàn cờ được vẽ rõ ràng và các nút chức năng được bố trí hợp lý. Người chơi có thể dễ dàng nhận diện các ô cờ, theo dõi các nước đi và trạng thái trận đấu. Giao diện không quá phức tạp nhưng vẫn đảm bảo tính thẩm mỹ và sự trực quan cần thiết cho một trò chơi chiến thuật.

Ứng dụng áp dụng thành công cơ bản thuật toán Minimax cơ bản, nhóm đã áp dụng thành công thuật toán Minimax để tạo ra một đối thủ AI thông minh. AI không chỉ đưa ra

nước đi một cách ngẫu nhiên mà còn có chiến lược rõ ràng, phân tích trạng thái bàn cờ và đưa ra các quyết định tối ưu. Thuật toán này đã được tích hợp cùng với kỹ thuật cắt tỉa Alpha-Beta nhằm rút gọn số lượng trạng thái cần duyệt, từ đó giúp AI phản hồi nhanh hơn mà vẫn đảm bảo chất lượng của nước đi. Tích lũy kinh nghiệm và nâng cao kỹ năng lập trình, tư duy thuật toán, kỹ năng lập trình Python và thiết kế GUI với thư viện Tkinter.

Dự án đã giúp nhóm củng cố kiến thức về Python, đặc biệt trong việc phát triển giao diện người dùng với Tkinter. Qua đó, các thành viên có cơ hội làm quen với các khái niệm như lập trình hướng đối tượng, xử lý sự kiện và quản lý trạng thái của trò chơi. Hiểu biết về một nhánh nhỏ về trí tuệ nhân tạo trong game, việc triển khai thuật toán Minimax đã mở ra cho nhóm một cái nhìn tổng quát về cách thức AI đưa ra quyết định trong các trò chơi chiến thuật. Điều này không chỉ là bài học lý thuyết mà còn là kinh nghiệm thực tiễn quý báu, giúp nhóm có thể ứng dụng vào các dự án AI phức tạp hơn trong tương lai.

Phát triển kỹ năng làm việc nhóm và quản lý dự án, quá trình thực hiện dự án yêu cầu sự phối hợp chặt chẽ giữa các thành viên, từ phân tích yêu cầu, thiết kế kiến trúc, lập trình cho đến kiểm thử. Nhờ đó, nhóm đã rèn luyện được kỹ năng giao tiếp, phân công nhiệm vụ, và quản lý tiến độ làm việc một cách hiệu quả. Qua các bài kiểm thử, trò chơi đã hoạt động ổn định với khả năng xử lý các nước đi của cả người chơi và AI một cách trơn tru. Điều này chứng tỏ rằng các giải pháp về thuật toán và tối ưu hóa đã được triển khai đúng đắn.

4.2. Hạn chế và giải pháp khắc phục

Mặc dù có các ưu điểm trên nhưng vẫn còn nhiều thiếu sót khi làm và chạy thử chương trình trò chơi caro

- Giao diện người dùng chưa hoàn thiện thiếu các thứ như:

- + Thiếu menu thoát trận trong lúc chơi, điều này gây bất tiện cho người dùng khi muốn dừng trận đấu giữa chừng.
- + Giao diện cơ bản có thể được cải thiện thêm để trở nên hấp dẫn hơn với các hiệu ứng động và đồ họa hiện đại.

- Chức năng lưu và tải lại ván đấu chưa được tích hợp:

Người chơi chưa có khả năng lưu lại trạng thái ván đấu để có thể tiếp tục chơi sau khi tạm dừng, điều này là một hạn chế lớn trong trải nghiệm người dùng.

- AI vẫn có thể chưa tối ưu hoàn toàn:

Mặc dù đã áp dụng thuật toán Minimax, AI đôi khi vẫn gặp khó khăn trong việc nhận diện các thế cờ phức tạp, đặc biệt khi đối mặt với những chiến thuật tinh vi của người chơi.

- Hiệu suất của thuật toán Minimax khi bàn cờ mở rộng:

Khi kích thước bàn cờ tăng lên, số lượng trạng thái cần đánh giá tăng theo cấp số nhân, làm cho thời gian tính toán tăng đáng kể. Để khắc phục những hạn chế này, nhóm đã đưa ra một số giải pháp cải tiến cụ thể như bổ sung các nút chức năng cần thiết, tối ưu hóa thuật toán AI, tích hợp tính năng lưu tải ván đấu, và mở rộng chế độ chơi trực tuyến. Các giải pháp này sẽ được triển khai trong các update sắp tới cho dự án.

TÀI LIỆU THAM KHẢO

1. Sebastian Lague, “Algorithms Explained – minimax and alpha-beta pruning
2. Pygame_menu, “Pygame menu documents”
3. VTI Academy. (n.d.). *Python là gì? Những điều cần biết về lập trình Python*. Truy cập ngày 10 tháng 3 năm 2025
4. Lê Hồng Ngọc Quân. *Visual Studio Code là gì? Tính năng của Visual Studio Code*. Truy cập ngày 21 tháng 3 năm 2025
5. Bro. (2017). *Thuật toán Minimax (AI trong Game)*. Viblo. Truy cập ngày 10 tháng 3 năm 2025