

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Phát triển game cờ Caro sử dụng Socket

GVHD: Từ Lăng Phiêu

SV: Phạm Triều Điền - 3120410129

Trần Đình Toàn - 3120410121

TP. HỒ CHÍ MINH, THÁNG 5/2024

Mục lục

LỜI MỞ ĐẦU	2
1 GIỚI THIỆU DỰ ÁN	3
1.1 Giới thiệu	3
1.2 Mục tiêu dự án	3
1.3 Tính cạnh tranh	3
1.4 Kết luận	4
2 CƠ SỞ LÝ THUYẾT	5
2.1 Giới thiệu ngôn ngữ Python	5
2.2 Tính năng chính của Python	5
2.2.1 Ưu điểm	5
2.2.2 Nhược điểm	6
2.3 Môi trường làm việc của Python	6
2.4 Thư viện Python sử dụng trong dự án	7
2.5 Framework sử dụng trong dự án	7
3 XÂY DỰNG PHẦN MỀM GAME CỜ CARO SỬ DỤNG FLASK FRAMEWORK	8
3.1 Giới thiệu Framework Flask	8
3.1.1 Ưu điểm	8
3.1.2 Nhược điểm	9
3.2 Cấu trúc thư mục dự án	9
3.3 Thiết kế giao diện	9
3.3.1 Giao diện trang chủ	10
3.3.2 Giao diện khi chọn chơi online	10
3.3.3 Giao diện khi chọn chơi với máy	11
3.4 Hàm xử lý và các sự kiện	11
3.4.1 Sự kiện Click và xử lý logic sau mỗi lần chọn vị trí	11
3.4.2 Hàm kiểm tra chiến thắng	12
3.4.3 Hàm kiểm tra số điểm phòng thủ	13
3.4.4 Hàm kiểm tra số điểm cao nhất	14
3.4.5 Hàm tạo phòng	15
3.4.6 Hàm vào phòng	15
4 TÀI LIỆU THAM KHẢO	15

LỜI MỞ ĐẦU

Cờ Caro, hay còn được biết đến với tên gọi Gomoku, là một trò chơi chiến thuật cổ điển có lịch sử lâu đời. Với luật chơi dễ hiểu, không đòi hỏi nhiều công cụ, Cờ Caro từ lâu đã trở thành một hoạt động giải trí phổ biến, giúp rèn luyện tư duy logic và khả năng phân tích cho người chơi ở mọi lứa tuổi. Trong bối cảnh công nghệ số ngày càng phát triển, việc chuyển thể trò chơi này lên nền tảng kỹ thuật số không chỉ giúp bảo tồn giá trị văn hóa mà còn đáp ứng nhu cầu giải trí hiện đại của cộng đồng.

1 GIỚI THIỆU DỰ ÁN

1.1 Giới thiệu

Nhắc đến các tựa game cổ điển gắn liền với tuổi thơ, làm cho chúng ta tốn khá nhiều giấy đó là game cờ Caro. Để hạn chế tốn giấy và chúng ta có thể chơi với các bạn bè thông qua mạng, thông qua máy tính. Tụi em đã tạo ra 1 game cờ Caro sử dụng Socket để kết nối thiết bị giúp 2 người có thể chơi với nhau, giao diện thân thiện và dễ sử dụng, mang lại giải trí cho người chơi.

1.2 Mục tiêu dự án

Dự án game Cờ Caro của chúng tôi sẽ tập trung vào các mục tiêu chính sau:

- Tái hiện chính xác luật chơi truyền thống: Đảm bảo người chơi có thể trải nghiệm trò chơi một cách chân thực và đầy đủ nhất.
- Giao diện thân thiện và dễ sử dụng: Thiết kế giao diện trực quan, đẹp mắt và dễ dàng thao tác cho mọi người dùng, từ trẻ em đến người lớn tuổi.
- Tính năng đa dạng và linh hoạt: Bổ sung các tính năng mới như chế độ chơi online, bảng xếp hạng, và các mức độ khó khác nhau để tăng tính thách thức và hứng thú cho người chơi.

Chúng tôi tin rằng dự án game Cờ Caro sẽ mang lại nhiều giá trị không chỉ về mặt giải trí mà còn góp phần duy trì và phát huy giá trị văn hóa dân gian trong thời đại công nghệ. Rất mong nhận được sự ủng hộ và góp ý từ quý vị để chúng tôi có thể hoàn thiện sản phẩm, mang đến cho cộng đồng một trải nghiệm chơi game tuyệt vời nhất.

1.3 Tính cạnh tranh

Trong bối cảnh thị trường game hiện đại với sự bùng nổ của các tựa game đa dạng về thể loại và phong cách, các trò chơi cổ điển như Cờ Caro vẫn giữ được một vị trí đặc biệt và có tính cạnh tranh riêng. Dưới đây là một số yếu tố giúp Cờ Caro duy trì tính cạnh tranh của mình:

- Sự Đơn Giản và Tiếp Cận Dễ Dàng:

- Luật Chơi Dễ Hiểu: Cờ Caro có luật chơi đơn giản, dễ hiểu, không đòi hỏi người chơi phải mất nhiều thời gian để học. Điều này thu hút được người chơi ở mọi lứa tuổi, từ trẻ em đến người lớn.

- Không Yêu Cầu Trang Thiết Bị Đặc Biệt: Người chơi chỉ cần một bàn cờ và quân cờ, hoặc thậm chí chỉ cần một tờ giấy và bút. Điều này giúp trò chơi dễ dàng tiếp cận mọi người, đặc biệt là ở những khu vực không có điều kiện tiếp cận các thiết bị điện tử hiện đại.

- Tính Chiến Thuật Cao:

Phát Triển Tư Duy Logic: Cờ Caro không chỉ là một trò chơi giải trí mà còn là một công cụ hữu ích để rèn luyện tư duy logic, khả năng lập kế hoạch và phân tích tình huống. Điều này đặc biệt hấp dẫn với những người yêu thích các trò chơi chiến thuật.

- Khả Năng Chuyển Đổi Sang Nền Tảng Kỹ Thuật Số:

Trò Chơi Online: Việc chuyển đổi Cờ Caro lên các nền tảng kỹ thuật số giúp người chơi có thể dễ dàng tìm kiếm đối thủ trực tuyến, tạo nên một cộng đồng người chơi đông đảo và năng động.

- Bảo Tồn Giá Trị Văn Hóa:

Truyền Thống và Hiện Đại: Cờ Caro không chỉ là một trò chơi mà còn là một phần của văn hóa dân gian. Việc duy trì và phát triển Cờ Caro trên nền tảng số góp phần bảo tồn và lan tỏa giá trị văn hóa truyền thống trong thời đại số.

1.4 Kết luận

Dự án Phát triển Phần mềm Mã nguồn mở: Phát triển ứng dụng game cờ Caro sử dụng Socket sẽ tạo ra 1 game Caro thú vị, đẹp mắt giúp mọi người giải trí, trò chuyện với nhau và nâng cao tư duy chiến thuật.

2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu ngôn ngữ Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Guido Van Rossum, một lập trình viên máy tính ở Hà Lan, đã tạo ra Python. Ông bắt đầu phát triển ngôn ngữ này vào năm 1989 tại Centrum Wiskunde & Informatica (CWI), ban đầu chỉ là một dự án tiêu khiển để giúp ông giết thời gian trong dịp Giáng sinh. Tên gọi của ngôn ngữ này được lấy cảm hứng từ chương trình truyền hình Monty Python's Flying Circus của đài BBC vì Guido Van Rossum là một “fan cứng” của chương trình này.

Lịch sử các phiên bản Python

- Guido Van Rossum cho ra mắt phiên bản đầu tiên của ngôn ngữ Python (phiên bản 0.9.0) vào năm 1991. Ngôn ngữ này đã bao gồm các tính năng hữu ích như một số kiểu dữ liệu và hàm để xử lý lỗi.
- Python 1.0 đã được ra mắt vào năm 1994 với các hàm mới để dễ dàng xử lý danh sách dữ liệu, chẳng hạn như ánh xạ, lọc và lược bỏ.
- Python 2.0 đã được ra mắt vào ngày 16 tháng 10 năm 2000, với các tính năng hữu ích mới cho lập trình viên, chẳng hạn như hỗ trợ ký tự Unicode và cách xử lý chi tiết một danh sách nhanh chóng hơn.
- Python 3.0 đã được ra mắt vào ngày 3 tháng 12 năm 2008. Phiên bản này bao gồm các tính năng như hàm in và hỗ trợ nhiều hơn cho việc phân chia số và xử lý lỗi.

2.2 Tính năng chính của Python

2.2.1 Ưu điểm

- Cú pháp đơn giản và dễ đọc: Python có cú pháp rõ ràng, dễ học và dễ đọc, làm cho việc viết mã và duy trì mã trở nên dễ dàng hơn.
- Thư viện phong phú: Python có một kho thư viện rộng lớn như NumPy, Pandas, Matplotlib, TensorFlow, và nhiều thư viện khác hỗ trợ các lĩnh vực từ xử lý dữ liệu, khoa học dữ liệu, học máy, đến phát triển web.
- Tính linh hoạt cao: Python có thể được sử dụng cho nhiều loại ứng dụng khác nhau, từ ứng dụng web, phần mềm máy tính, đến các ứng dụng khoa học và tính toán phức tạp.
- Cộng đồng lớn và hỗ trợ tốt: Python có một cộng đồng người dùng lớn mạnh, cung cấp nhiều tài liệu học tập, diễn đàn hỗ trợ

và các dự án mã nguồn mở, giúp người mới dễ dàng tiếp cận và học hỏi. - Đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux mà không cần phải thay đổi mã nguồn. - Hỗ trợ tích hợp: Python có thể tích hợp tốt với các ngôn ngữ và công nghệ khác như C, C++, Java, và .NET, giúp mở rộng khả năng sử dụng trong các dự án phức tạp.

2.2.2 Nhược điểm

- Tốc độ chậm: Python là một ngôn ngữ thông dịch, vì vậy nó thường chậm hơn so với các ngôn ngữ biên dịch như C hoặc C++. Điều này có thể là một vấn đề đối với các ứng dụng đòi hỏi hiệu suất cao.
- Sử dụng bộ nhớ lớn: Python thường sử dụng nhiều bộ nhớ hơn so với các ngôn ngữ khác, điều này có thể là một nhược điểm khi làm việc với các ứng dụng cần tiết kiệm tài nguyên hệ thống.
- Không phù hợp cho lập trình di động và game: Mặc dù có một số thư viện và công cụ hỗ trợ, Python không được coi là ngôn ngữ tốt nhất cho việc phát triển ứng dụng di động hoặc trò chơi điện tử.
- Thiếu tính bảo mật: Python không có nhiều tính năng bảo mật tích hợp như một số ngôn ngữ khác, điều này đòi hỏi các nhà phát triển phải đặc biệt chú ý đến bảo mật khi phát triển ứng dụng.

2.3 Môi trường làm việc của Python

Python là một ngôn ngữ lập trình đa dụng và mạnh mẽ, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau nhờ vào cú pháp dễ hiểu và cộng đồng phát triển mạnh mẽ. Dưới đây là các lĩnh vực chính mà Python được ứng dụng:

- Lập trình ứng dụng web: Python là một lựa chọn tuyệt vời để phát triển các ứng dụng web có khả năng mở rộng cao. Với sự hỗ trợ của các framework mạnh mẽ như Django và Flask, lập trình viên có thể xây dựng các trang web và dịch vụ web phức tạp một cách nhanh chóng và hiệu quả. Các trang web nổi tiếng như Mozilla, Reddit, và Instagram đều được phát triển bằng Python. Django, với hệ thống quản lý nội dung mạnh mẽ, giúp phát triển các ứng dụng web có tính năng phong phú mà không cần quá nhiều cấu hình ban đầu. Flask, mặt khác, mang đến sự linh hoạt và nhẹ nhàng, thích hợp cho các dự án nhỏ hơn hoặc những người muốn tùy biến nhiều hơn.
- Khoa học và tính toán: Python nổi bật trong lĩnh vực khoa học và tính toán số liệu nhờ vào nhiều thư viện mạnh mẽ. SciPy và NumPy cung cấp các công cụ cần thiết cho tính toán khoa học và xử lý số liệu hiệu quả. Trong lĩnh vực học máy (machine learning), khai thác dữ liệu (data mining), và học sâu (deep learning), Python gần như không có đối thủ nhờ vào các thư

viện như scikit-learn, TensorFlow, Keras, và PyTorch. Scikit-learn đơn giản hóa việc triển khai các mô hình học máy, trong khi TensorFlow và PyTorch cung cấp khả năng xây dựng và huấn luyện các mạng nơ-ron phức tạp. Những thư viện này không chỉ mạnh mẽ mà còn có cộng đồng người dùng rộng lớn, giúp việc học tập và áp dụng chúng trở nên dễ dàng hơn. - Tạo nguyên mẫu phần mềm: Python là ngôn ngữ lý tưởng để tạo các nguyên mẫu phần mềm (prototypes). Với khả năng phát triển nhanh và dễ dàng, lập trình viên có thể nhanh chóng xây dựng và thử nghiệm các ý tưởng mới. Ví dụ, Pygame là một thư viện phổ biến cho việc viết game đơn giản và nhanh chóng tạo ra các nguyên mẫu game. Sau khi nguyên mẫu đã được hoàn thiện và kiểm thử, bạn có thể chuyển sang các ngôn ngữ lập trình khác như C++ để phát triển phiên bản chính thức của game với hiệu năng cao hơn. Việc này giúp tiết kiệm thời gian và nguồn lực, đồng thời cho phép bạn dễ dàng chỉnh sửa và thử nghiệm các ý tưởng trước khi cam kết vào quá trình phát triển dài hạn.

- Ngôn ngữ tốt để dạy lập trình: Python được xem là một trong những ngôn ngữ tốt nhất để dạy lập trình, đặc biệt cho người mới bắt đầu và trẻ em. Cú pháp của Python rõ ràng, dễ đọc và dễ hiểu, giúp người học tập trung vào các khái niệm lập trình cơ bản mà không bị rối bởi cú pháp phức tạp. Nhiều trường học, đại học, và tổ chức giáo dục sử dụng Python như là ngôn ngữ đầu tiên để giảng dạy lập trình. Các khóa học trực tuyến và sách giáo khoa cũng thường sử dụng Python làm ngôn ngữ chính vì sự đơn giản và hiệu quả của nó. Python không chỉ giúp người học nắm vững các khái niệm lập trình cơ bản mà còn mở ra cơ hội cho họ tiếp cận các lĩnh vực nâng cao như phát triển web, khoa học dữ liệu, và trí tuệ nhân tạo. Nhờ vào tính linh hoạt và khả năng ứng dụng rộng rãi, Python tiếp tục phát triển và trở thành một trong những ngôn ngữ lập trình quan trọng và phổ biến nhất hiện nay.

2.4 Thư viện Python sử dụng trong dự án

Python-SocketIO: Python-SocketIO là một thư viện Python cho phép bạn xây dựng các ứng dụng sử dụng WebSocket. Nó cung cấp các công cụ để giao tiếp thông qua WebSocket cả trên phía máy chủ và phía client trong Python. Python-SocketIO thường được sử dụng kết hợp với Flask-SocketIO để xây dựng các ứng dụng web thời gian thực.

- Flask-SocketIO: Flask-SocketIO là một extension cho Flask, cho phép bạn thêm tính năng WebSocket vào ứng dụng Flask của mình. WebSocket là một giao thức cho phép giao tiếp hai chiều giữa client và server thông qua một kết nối đơn. Flask-SocketIO làm cho việc xây dựng các ứng dụng web thời gian thực trở nên dễ dàng hơn.

2.5 Framework sử dụng trong dự án

Flask: Đây là một framework web phổ biến được sử dụng để xây dựng các ứng dụng web trong Python. Flask là một framework nhẹ, linh hoạt và dễ dàng học, nó cung cấp các công cụ cần thiết để xây dựng các ứng dụng web từ các ứng dụng nhỏ đến các hệ thống lớn.

Ngoài ra còn sử dụng Jinja2: Jinja2 là một hệ thống template engine trong Python. Nó cho phép bạn tách biệt

logic của ứng dụng web của mình khỏi giao diện người dùng. Bằng cách sử dụng Jinja2, bạn có thể tạo ra các mẫu HTML linh hoạt và dễ bảo trì bằng cách chèn các biến và lệnh điều khiển vào trong chúng.

3 XÂY DỰNG PHẦN MỀM GAME CỜ CARO SỬ DỤNG FLASK FRAMEWORK

3.1 Giới thiệu Framework Flask

Flask là loại framework web phổ biến được viết bằng trình lập ngôn ngữ Python. Công nghệ thường được sử dụng để xây dựng trang web từ những ứng dụng đơn giản đến những hệ thống phức tạp hơn.

Flask Framework sở hữu một số tính năng quan trọng mà nhà phát triển thường sử dụng để xây dựng hiệu ứng web. Dưới đây là một số tính năng chính của Flask:

- Nhẹ và dễ sử dụng: Công nghệ có cấu trúc nhẹ nhàng và mã nguồn dễ đọc, giúp người phát triển dễ dàng tiếp cận và tùy chỉnh theo nhu cầu cụ thể của họ.
- Định tuyến linh hoạt: Flask cung cấp cơ chế hoạt động định tuyến, cho phép người phát triển xác định các mẫu URL và phân bổ chúng cho các hàm xử lý tương ứng. Điều này giúp quản lý và xử lý yêu cầu HTTP một cách hiệu quả.
- Công cụ mẫu: Flask tích hợp Jinja2, đây là một loại trình biên dịch mẫu mạnh mẽ cho phép tạo ra các giao diện người dùng.
- Được mở rộng rộng rãi: Mặc dù mang đặc điểm rút gọn nhưng Flask vẫn có khả năng mở rộng mạnh mẽ thông qua việc sử dụng các tiện ích và thư viện của cộng đồng. Người dùng có thể phân tích các tính năng như xác thực, đăng nhập, điều hướng, cơ sở dữ liệu tương tác và nhiều tính năng khác.
- Máy chủ phát triển tích hợp: Flask cung cấp máy chủ phát triển hợp đồng, giúp người phát triển dễ dàng kiểm tra và phát triển ứng dụng mà không cần cấu hình bổ sung.
- Cộng đồng lớn và tích cực: Số lượng người dùng Flask rất đông và luôn nhận được hỗ trợ mạnh mẽ từ cộng đồng Python, điều này giúp người phát triển tìm kiếm thông tin và tài liệu một cách dễ dàng.

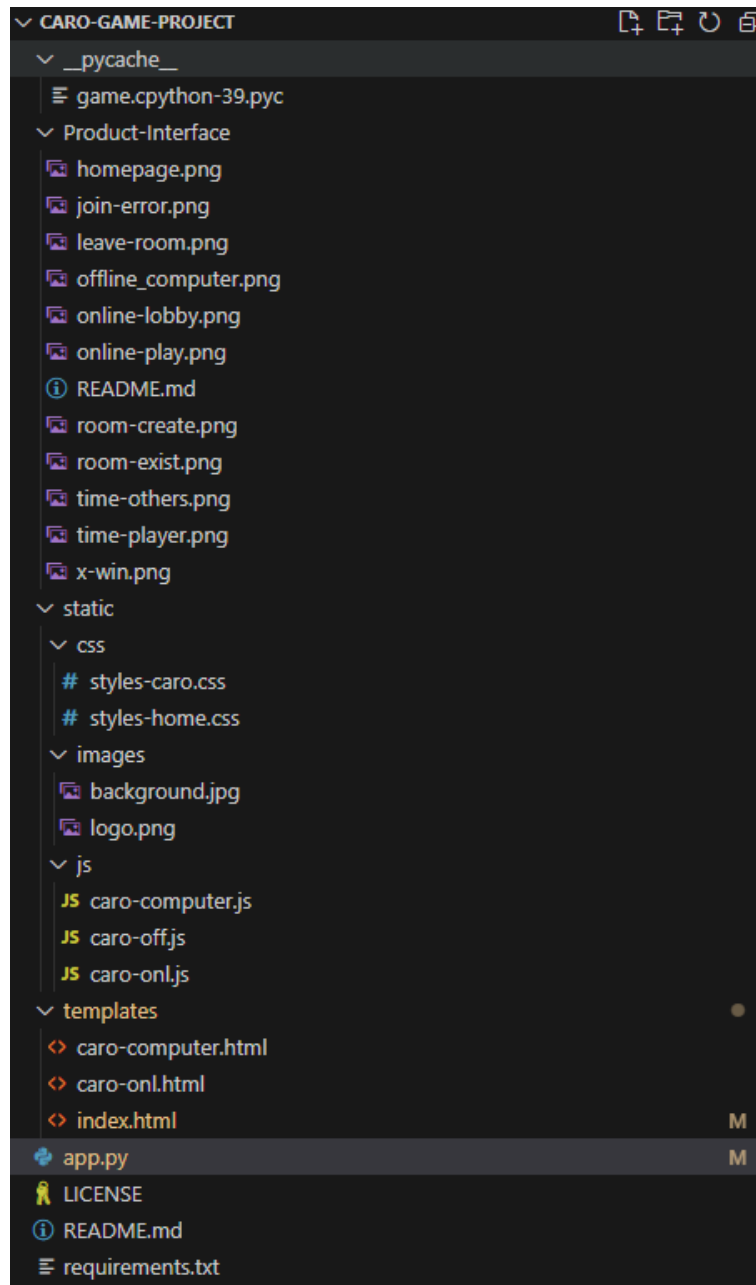
3.1.1 Ưu điểm

- Linh hoạt: Flask được thiết kế để linh hoạt và dễ dàng tùy chỉnh theo nhu cầu cụ thể của từng dự án.
- Dễ học và sử dụng: Flask có mã cấu trúc đơn giản, dễ hiểu và tài liệu hướng dẫn phong phú, giúp người dùng dễ dàng tiếp cận và sử dụng.
- Đa chức năng: Flask cho phép tích hợp nhiều công nghệ và thư viện bổ sung khác nhau, từ các công cụ front-end cho cơ sở dữ liệu và mạng dịch vụ.
- Mạnh mẽ cho ứng dụng nhỏ đến trung bình: Flask thích hợp để xây dựng các trang web ứng dụng với quy mô từ nhỏ đến trung bình với hiệu suất tốt.

3.1.2 Nhược điểm

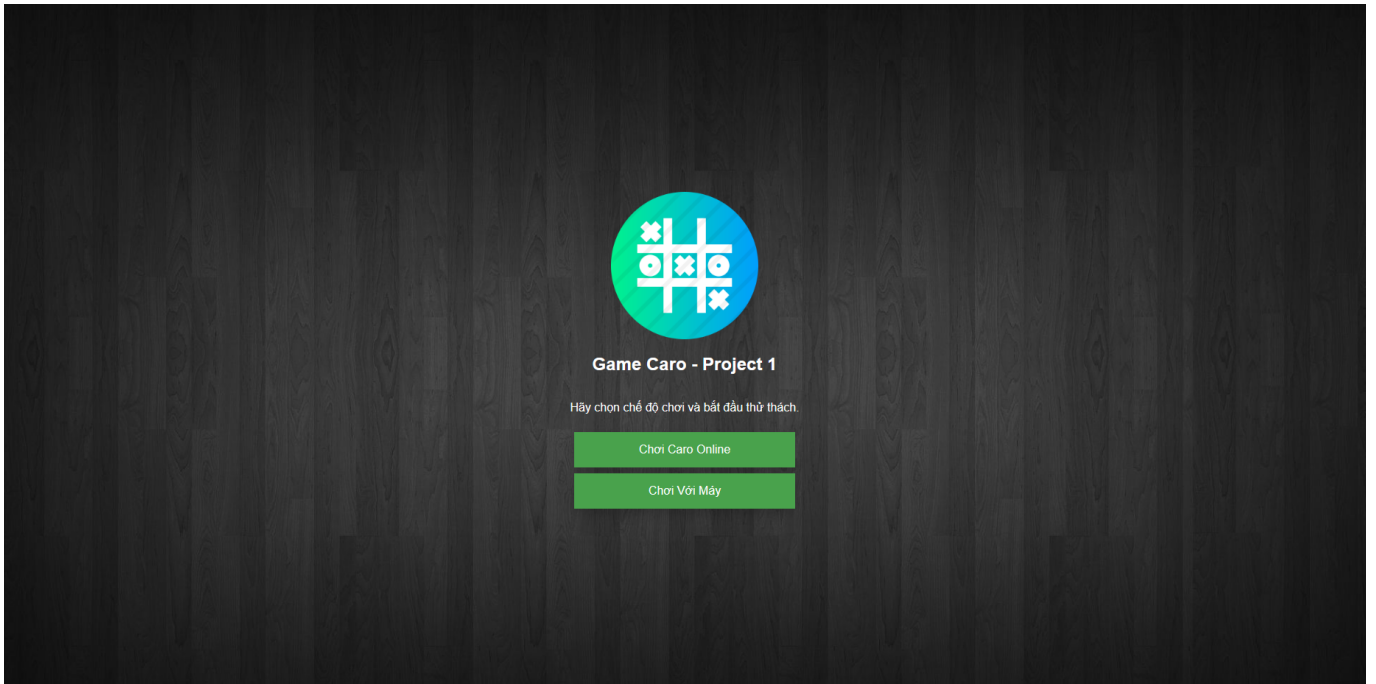
- Khả năng mở rộng: Flask không cung cấp sẵn các tính năng lớn hơn như các khung, điều này có thể làm tăng mức độ phức tạp khi cần tích hợp nhiều tính năng mở rộng.
- Không phù hợp cho ứng dụng lớn: Bình không phù hợp cho các ứng dụng lớn với quy mô phức tạp và yêu cầu hiệu suất cao.

3.2 Cấu trúc thư mục dự án



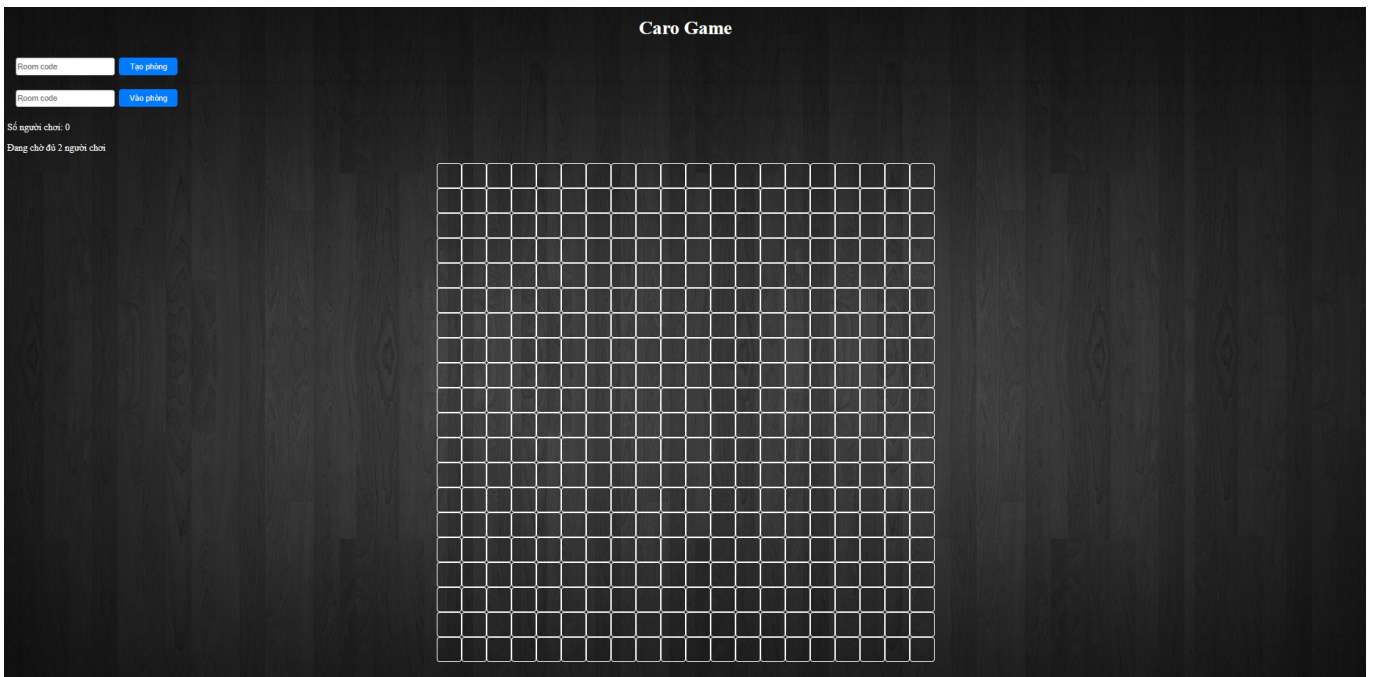
3.3 Thiết kế giao diện

3.3.1 Giao diện trang chủ

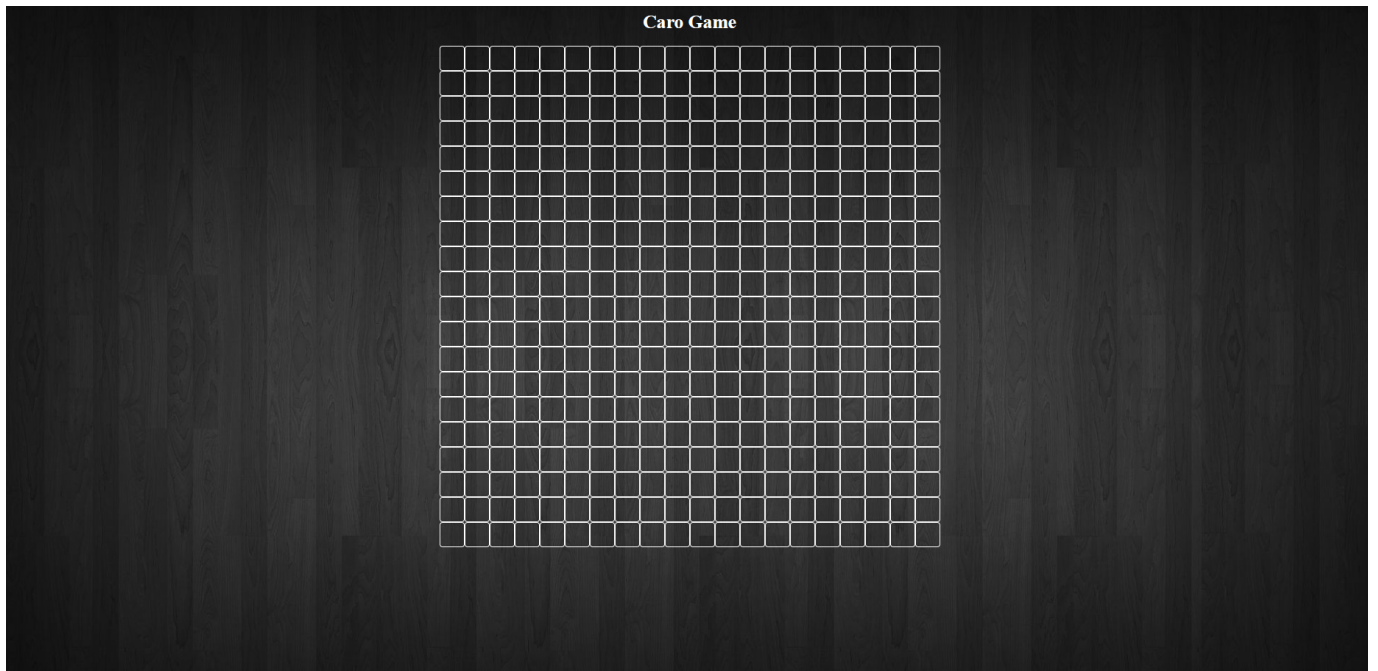


- Hình ảnh trên là giao diện của trang chủ dùng HTML/CSS

3.3.2 Giao diện khi chọn chơi online



3.3.3 Giao diện khi chọn chơi với máy



3.4 Hàm xử lý và các sự kiện

3.4.1 Sự kiện Click và xử lý logic sau mỗi lần chọn vị trí

```
function handleClick(e) {
  // Kiểm tra xem ô đã được đánh chưa
  if (e.target.textContent === '') {
    e.target.textContent = currentPlayer;
    e.target.classList.add(currentPlayer.toLowerCase());
    // Highlight cell sau khi đánh
    e.target.classList.add('highlight');
    let index = board.indexOf(e.target);
    if (checkWin(index, currentPlayer)) {
      setTimeout(function() {
        alert('You win!');
        resetGame();
      }, 100); // Thêm trễ 100ms
    }

    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
    socket.emit('move_computer', { index: index, player: currentPlayer, target: e.target });
    // Máy tính đánh ngay
    function getComputerMove(): any {
      const computerMove = getComputerMove();
      // Kiểm tra xem ô đã được đánh chưa
      if (board[computerMove[0] * 20 + computerMove[1]].textContent === '') {
        board[computerMove[0] * 20 + computerMove[1]].textContent = currentPlayer;
        board[computerMove[0] * 20 + computerMove[1]].classList.add(currentPlayer.toLowerCase());
        board[computerMove[0] * 20 + computerMove[1]].classList.add('highlight'); // Highlight the cell
        if (checkWin(computerMove[0] * 20 + computerMove[1], currentPlayer)) {
          setTimeout(function() {
            alert('Computer wins!');
            resetGame();
          }, 100); // Thêm trễ 100ms
        }
      }
      currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
    }
  }
}
```

3.4.2 Hàm kiểm tra chiến thắng

```
function checkWin(index, player) {
  let row = Math.floor(index / 20); // Lấy số hàng
  let col = index % 20; // Lấy số cột
  let directions = [
    [-1, -1],
    [-1, 0],
    [-1, 1],
    [0, 1]
  ];
  for (let [dx, dy] of directions) {
    let count = 1;
    for (let i = 1; i < 5; i++) {
      let x = row + dx * i;
      let y = col + dy * i;
      if (x < 0 || x >= 20 || y < 0 || y >= 20 || board[x * 20 + y].textContent !== player) {
        break;
      }
      count++;
    }
    for (let i = 1; i < 5; i++) {
      let x = row - dx * i;
      let y = col - dy * i;
      if (x < 0 || x >= 20 || y < 0 || y >= 20 || board[x * 20 + y].textContent !== player) {
        break;
      }
      count++;
    }
    if (count >= 5) {
      return true;
    }
  }
  return false;
}
```

3.4.3 Hàm kiểm tra số điểm phòng thủ

```
// Hàm kiểm tra số điểm phòng thủ
function evaluateDefensePosition(row, col, player) {
  const directions = [
    [0, 1], // Ngang
    [1, 0], // Dọc
    [1, 1], // Chéo phải
    [1, -1] // Chéo trái
  ];
  let maxScore = 0; // Đặt giá trị thấp để ưu tiên phòng thủ
  for (const [dx, dy] of directions) {
    let count = 1;
    for (let i = 1; i < 5; i++) {
      let x = row + dx * i;
      let y = col + dy * i;
      if (x < 0 || x >= 20 || y < 0 || y >= 20 || board[x * 20 + y].textContent !== player) {
        break;
      }
      count++;
    }
    for (let i = 1; i < 5; i++) {
      let x = row - dx * i;
      let y = col - dy * i;

      if (x < 0 || x >= 20 || y < 0 || y >= 20 || board[x * 20 + y].textContent !== player) {
        break;
      }
      count++;
    }
    maxScore = Math.max(maxScore, count);
  }
  return maxScore;
}
```

3.4.4 Hàm kiểm tra số điểm cao nhất

```
// Hàm lấy danh sách các điểm có điểm số cao nhất
function getBestPoints() {
    let maxAttackScore = -Infinity;
    let maxDefenseScore = -Infinity;
    let bestAttackPoints = [];
    let bestDefensePoints = [];

    for (let i = 0; i < 20; i++) {
        for (let j = 0; j < 20; j++) {
            if (board[i * 20 + j].textContent === "") {
                const attackScore = evaluatePosition(i, j, 'O');
                const defenseScore = evaluateDefensePosition(i, j, 'X');

                if (attackScore > maxAttackScore) {
                    maxAttackScore = attackScore;
                    bestAttackPoints = [
                        [i, j]
                    ];
                } else if (attackScore === maxAttackScore) {
                    bestAttackPoints.push([i, j]);
                }

                if (defenseScore > maxDefenseScore) {
                    maxDefenseScore = defenseScore;
                    bestDefensePoints = [
                        [i, j]
                    ];
                } else if (defenseScore === maxDefenseScore) {
                    bestDefensePoints.push([i, j]);
                }
            }
        }
    }
}
```

3.4.5 Hàm tạo phòng

```
document.getElementById('create-room-form').addEventListener('submit', function(e) {
    e.preventDefault();
    let roomCode = document.getElementById('create-room-code').value;
    fetch('/create', {
        method: 'POST',
        body: new URLSearchParams({
            'room_code': roomCode,
        })
    }).then(response => {
        if (!response.ok) {
            alert('Tạo phòng không thành công! (Phòng đã tồn tại)');
        } else {
            alert('Tạo phòng thành công --> Hãy tham gia phòng này!');
        }
    });
});
```

3.4.6 Hàm vào phòng

```
document.getElementById('join-room-form').addEventListener('submit', function(e) {
    e.preventDefault();
    let roomCode = document.getElementById('join-room-code').value;
    socket.emit('join', { room_code: roomCode }, function(error) {
        if (error) {
            alert("Phòng không tồn tại hoặc đã đủ số người chơi!");
        } else {
            // Ẩn cả khối chứa form tạo phòng và form tham gia phòng
            document.getElementById('create-room-form').parentElement.style.display = 'none';
            document.getElementById('join-room-form').parentElement.style.display = 'none';
            // Hiển thị thông tin về phòng hiện tại và nút để rời phòng
            let currentRoomElement = document.getElementById('current-room');
            currentRoomElement.innerText = 'Phòng hiện tại: ' + roomCode;
            currentRoomElement.style.fontWeight = 'bold';
            document.getElementById('current-room-container').style.display = 'block';
        }
    });
});
```

4 TÀI LIỆU THAM KHẢO

<https://fptshop.com.vn/tin-tuc/danh-gia/flask-la-gi-175382> <https://aws.amazon.com/vi/what-is/python/>