

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN



Dự án Phát Triển Phần Mềm Mã Nguồn Mở

The Caro Games

Với các thành viên:

Lê Gia Hào (3120410153)

Nguyễn Anh Kiệt (3120410269)

Võ Trần Tấn Lộc (3120410299)

Giáo viên hướng dẫn : Từ Lăng Phiêu

Ngày 2 tháng 5 năm 2024

Mục lục

Lời Mở Đầu	4
1 Giới thiệu về Game	5
1.1 Giới thiệu	5
1.2 Mục tiêu dự án	5
1.3 Tính cạnh tranh	5
1.4 Kết luận	5
2 Cơ Sở Lý Thuyết	5
2.1 Ngôn ngữ lập trình Python	5
2.1.1 Dễ học và sử dụng	7
2.1.2 Đa Mục Đích	7
2.1.3 Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ	7
2.1.4 Bảng Ecosystem	7
2.1.5 Hỗ Trợ Nền Tảng Rộng Rãi	7
2.1.6 Bảng Ecosystem	7
2.2 Ưu và Nhược điểm của Python	7
2.2.1 Ưu điểm	8
2.2.2 Nhược điểm	8
2.3 Môi trường làm việc	8
2.3.1 Trình biên dịch và trình thông dịch	8
2.3.2 Công cụ biên tập mã (IDEs)	9
2.3.3 Thư viện và framework	9
2.3.4 Hệ thống quản lý gói:	10
2.3.5 Cộng đồng và tài nguyên học	11
2.3.6 Môi trường ảo (Virtual environments)	11
2.3.7 Công cụ gỡ lỗi (Debugging tools)	11
3 Xây dựng Game Caro	11
3.1 Tổng quan về game	11
3.2 Cấu trúc dự án game cờ Caro	12
3.2.1 Client	12
3.2.2 Server	19
3.2.3 NetWork	21
3.2.4 Game	23
3.3 Giao diện game cờ Caro	26
3.4 Hướng dẫn cài đặt game	30
3.4.1 Trên Window	30
3.4.2 Trên Linux	32
3.4.3 Để chơi game:	32

Danh sách Bảng

1	Bảng phân công	4
---	--------------------------	---

Danh sách Hình Ảnh

1	Minh họa Python	6
2	PyCharm bản miễn phí sử dụng	9
3	Minh họa thư viện Random trong Python	11
4	Màn hình chính	26
5	Màn hình chơi với máy	27
6	Màn hình chơi online khi đang đợi người chơi khác	27
7	Màn hình chơi online ở phía X	28
8	Màn hình chơi online ở phía O	28
9	Màn hình ở mục chat	29
10	Đường dẫn để tải game	30
11	Tạo thư mục chứa game	30
12	Mở Terminal	30
13	Lệnh git init	31
14	Lệnh git clone	31
15	Chọn download file zip	31
16	Tải hoàn thành	32
17	Download trên linux	32
18	Extract To	32
19	Địa chỉ Ipv4 cần thay đổi	33
20	Đổi file server	33
21	Đổi file network	33
22	Chạy file server trước rồi mới chạy file client	34

Lời Cảm Ơn

Chúng em xin gửi lời cảm ơn chân thành nhất đến với giảng viên hướng dẫn - ThS. Từ Lăng Phiêu, giảng viên bộ môn Phát Triển Phần Mềm Mã Nguồn Mở, vì những sự giúp đỡ giảng dạy, góp ý, hướng dẫn chúng em trong suốt khoảng thời gian học tập và thực hiện đồ án này cho môn học.

Chúng em cũng xin gửi lời cảm ơn của mình đến các giảng viên, cán bộ công nhân viên của Trường Đại Học Sài Gòn và nhất là các thầy cô trong khoa Công Nghệ Thông Tin đã giúp đỡ giảng dạy, dạy dỗ chúng em các kiến thức và chia sẻ những kinh nghiệm, trải nghiệm của mình để chúng em có nhiều hiểu biết hơn và các mặt khác nhau cũng ngành và nghề liên quan đến Công Nghệ Thông Tin trong suốt quá trình chúng em tham gia học tập tại trường.

Dù đã được sự hỗ trợ giúp đỡ tận tình của thầy, nhưng với những kiến thức nhiều như vô tận mà chúng em vẫn chưa nắm bắt được cũng nhưng với kinh nghiệm còn thiếu và yếu của chúng em sẽ không thể tránh khỏi những thiếu sót. Chúng em mong sẽ nhận được sự chỉ bảo cũng như đóng góp ý kiến của thầy cô để nhóm chúng em có thể bổ sung, sửa chữa cũng như lấy đó làm kinh nghiệm để hoàn thành những đồ án, dự án của hiện tại và cả tương lai sau này.

Lần nữa Chúng em xin gửi lời cảm ơn chân thành nhất của mình ạ.

Lời Mở Đầu

Thế giới hiện đại ngày nay khi con người càng ngày phải chịu nhiều áp lực về cả thể chất và tinh thần, thì nhiều người đã tìm đến games để giải trí, ngoài các hoạt động thể thao như bóng đá, bóng rổ,... Thì người ta còn tìm đến các trò chơi điện tử để giải trí. Games không còn chỉ là một thứ làm ảnh hưởng đến sự phát triển của trẻ nhỏ, tạo nên những kẻ lười làm ham chơi mà games là phương tiện cách thức để giảm áp lực cuộc sống, giúp ta yêu đời hơn. Hiện nay khi thế giới đã phát triển đến mức độ xuất hiện các thiết bị như Vision Pro của Apple, PlayStation VR dùng cho PS4, Oculus Rift S có thể giúp người dùng cảm thấy như chính mình đang đắm chìm trong thế giới ảo. Tuy vậy vẫn có một sự cuốn hút không thể cưỡng lại trước các trò chơi cổ điển, các trò chơi như rambo lùn, cờ tướng, cờ vua đều là những trò chơi đã ra đời từ lâu nhưng chúng lại có sức cuốn hút khiến người ta không thể nào cưỡng lại được.

Họ và Tên	Công việc đã làm	Tỉ lệ phân công
Lê Gia Hào	Làm báo cáo và Phần Game	33%
Võ Trần Tấn Lộc	Làm báo cáo và Phần Network	33%
Nguyễn Anh Kiệt	Làm báo cáo và Phần Server,Client	33%

Bảng 1: Bảng phân công

1 Giới thiệu về Game

1.1 Giới thiệu

Game Caro là một ý tưởng hồi sinh lại dòng game cổ điển thay vì trên giấy chúng ta có thể chơi trên máy tính, chơi cùng bạn bè hoặc chơi với máy. Với chế độ chơi đa dạng cùng giao diện thân thiện, Đây sẽ là một tựa game hứa hẹn mang lại tuổi thơ, sự vui vẻ cho người chơi.

1.2 Mục tiêu dự án

- Tạo Ra Trải Nghiệm Caro Tuyệt Vời: Xây dựng một tựa game Caro trên PC với đồ họa đẹp mắt, tính năng đa dạng và giao diện thân thiện để mang lại trải nghiệm chơi game thú vị và sâu sắc cho người chơi.
- Kết Nối Cộng Đồng: Tạo ra một không gian trực tuyến cho cộng đồng người chơi Caro kết nối, cạnh tranh và chia sẻ niềm đam mê với nhau.
- Phát Triển Kỹ Năng: Cung cấp một nền tảng cho người chơi phát triển kỹ năng chiến thuật, tư duy và thực hiện các chiến lược trong trò chơi Caro.

1.3 Tính cạnh tranh

- Đồ Họa và Âm Thanh: Sử dụng đồ họa chất lượng cao và hiệu ứng âm thanh sống động để tạo ra một trải nghiệm trò chơi hấp dẫn và chân thực.
- Chế Độ Chơi Trực Tuyến: Cung cấp một nền tảng cho người chơi thách đấu và cạnh tranh với nhau trực tuyến.

1.4 Kết luận

"Caro" không chỉ là một tựa game Caro thông thường, mà còn là một không gian kết nối cộng đồng, nơi người chơi có thể thể hiện kỹ năng, thách đấu và chia sẻ niềm đam mê với những người chơi khác. Với mục tiêu tạo ra một trải nghiệm chơi game đẳng cấp và tính cạnh tranh cao, "Caro" hứa hẹn sẽ trở thành điểm đến lý tưởng cho những người yêu thích trò chơi Caro trên nền tảng PC.

2 Cơ Sở Lý Thuyết

2.1 Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào.



Hình 1: Minh họa Python

Tính linh hoạt này, cùng với sự thân thiện với người mới bắt đầu, đã khiến nó trở thành một trong những ngôn ngữ lập trình được sử dụng nhiều nhất hiện nay. Một cuộc khảo sát được thực hiện bởi công ty phân tích ngành RedMonk cho thấy rằng đây là ngôn ngữ lập trình phổ biến thứ hai đối với các nhà phát triển vào năm 2021.

Một vài sự thật thú vị về Python

- Python được phát triển vào cuối những năm 1980 bởi Guido van Rossum tại Viện Nghiên cứu Quốc gia về Toán học và Khoa học Máy tính ở Hà Lan với tư cách là người kế thừa ngôn ngữ ABC có khả năng xử lý và giao tiếp ngoại lệ.
- Python có nguồn gốc từ các ngôn ngữ lập trình như ABC, Modula 3, small talk, Algol-68.
- Van Rossum đã chọn tên Python cho ngôn ngữ mới từ một chương trình truyền hình, Monty Python's Flying Circus.
- Trang Python là một tệp có phần mở rộng .py chứa có thể là sự kết hợp của Thẻ HTML và tập lệnh Python.
- Vào tháng 12 năm 1989, người sáng tạo đã phát triển trình thông dịch python đầu tiên như một sở thích, và sau đó vào ngày 16 tháng 10 năm 2000, Python 2.0 được phát hành với nhiều tính năng mới.
- Vào ngày 3 tháng 12 năm 2008, Python 3.0 được phát hành với nhiều thử nghiệm hơn và bao gồm các tính năng mới.
- Python là một ngôn ngữ kịch bản mã nguồn mở.
- Python là mã nguồn mở, có nghĩa là bất kỳ ai cũng có thể tải xuống miễn phí từ trang chủ và sử dụng nó để phát triển các chương trình. Mã nguồn của nó có thể được truy cập và sửa đổi theo yêu cầu trong dự án.

- Python là một trong những ngôn ngữ chính thức tại Google.

2.1.1 Dễ học và sử dụng

Python được thiết kế để dễ học và đọc. Cú pháp của nó rất đơn giản và gần gũi với ngôn ngữ tự nhiên, giúp người mới bắt đầu nhanh chóng làm quen và phát triển kỹ năng lập trình.

2.1.2 Đa Mục Đích

Python là một ngôn ngữ đa mục đích, có thể được sử dụng để phát triển nhiều loại ứng dụng, từ web và mobile đến khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa, và nhiều hơn nữa.

2.1.3 Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ

- Python có một cộng đồng lớn và nhiệt tình, với nhiều nguồn tài liệu, sách, và tài nguyên trực tuyến miễn phí giúp học và phát triển.
- Có rất nhiều thư viện và framework mạnh mẽ được xây dựng cho Python, giúp bạn nhanh chóng xây dựng các ứng dụng phức tạp mà không cần phải viết lại từ đầu

2.1.4 Bảng Ecosystem

Python có một ecosystem phong phú của các công cụ và thư viện, bao gồm Django và Flask cho phát triển web, TensorFlow và PyTorch cho học máy và trí tuệ nhân tạo, Pandas và NumPy cho xử lý dữ liệu, và nhiều thứ khác.

2.1.5 Hỗ Trợ Nền Tảng Rộng Rãi

Python chạy trên hầu hết các hệ điều hành, bao gồm Windows, macOS, Linux, cũng như trên nhiều nền tảng di động.

2.1.6 Bảng Ecosystem

Python hỗ trợ lập trình hướng đối tượng, cho phép bạn tổ chức mã của mình thành các lớp và đối tượng, tạo ra mã dễ bảo trì và tái sử dụng.

2.2 Ưu và Nhược điểm của Python

2.2.1 Ưu điểm

- Dễ học và đọc: Python có cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, làm cho nó dễ học và dễ đọc, đặc biệt là đối với người mới bắt đầu trong lập trình.
- Đa mục đích: Python là một ngôn ngữ đa mục đích, được sử dụng cho nhiều mục đích, bao gồm phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa và nhiều hơn nữa.
- Cộng đồng lớn và hỗ trợ mạnh mẽ: Python có một cộng đồng lớn và nhiệt tình, cung cấp nhiều tài liệu, thư viện và framework, giúp bạn giải quyết các vấn đề và xây dựng các ứng dụng một cách nhanh chóng và dễ dàng.
- Bảng sinh thái phong phú: Python có một bảng sinh thái phong phú của các thư viện và framework, bao gồm Django và Flask cho phát triển web, NumPy và Pandas cho xử lý dữ liệu, TensorFlow và PyTorch cho học máy và trí tuệ nhân tạo, và nhiều thứ khác nữa.
- Hỗ trợ đa nền tảng: Python chạy trên hầu hết các hệ điều hành, bao gồm Windows, macOS và Linux, cũng như trên nhiều nền tảng di động.

2.2.2 Nhược điểm

- Tốc độ chậm: Python thường chậm hơn so với các ngôn ngữ lập trình khác như C++ hoặc Java, đặc biệt là trong các ứng dụng yêu cầu xử lý đồ họa nặng hoặc tính toán lớn.
- Quản lý bộ nhớ kém: Python không hiệu quả trong việc quản lý bộ nhớ so với một số ngôn ngữ khác, do đó có thể gặp vấn đề với các ứng dụng yêu cầu xử lý lượng dữ liệu lớn hoặc yêu cầu hiệu suất cao.
- Giới hạn về tính toán đa luồng: Python không hỗ trợ tính toán đa luồng tốt như một số ngôn ngữ lập trình khác, điều này có thể làm giảm hiệu suất trong các ứng dụng đòi hỏi tính toán đa luồng.

2.3 Môi trường làm việc

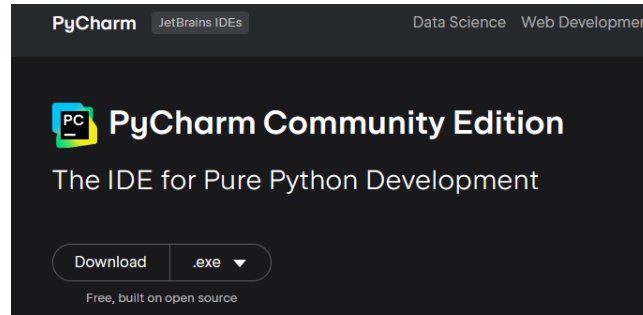
2.3.1 Trình biên dịch và trình thông dịch

Python có hai cách thực thi chương trình: thông qua trình thông dịch (interpreter) và trình biên dịch (compiler). Trình thông dịch cho phép bạn chạy mã Python dưới dạng nguyên bản, trong khi trình biên dịch biên dịch mã Python thành mã máy trước khi thực thi.

2.3.2 Công cụ biên tập mã (IDEs)

Có nhiều công cụ biên tập mã dành cho Python như PyCharm, Visual Studio Code, Sublime Text, và IDLE. Những công cụ này cung cấp các tính năng hỗ trợ lập trình như gợi ý mã, kiểm tra cú pháp, và gỡ lỗi mã. Ở trong dự án này chúng em sử dụng PyCharm vì tính dễ dụng, đầy đủ các tính năng hỗ trợ cho Python.

<https://www.jetbrains.com/pycharm/download/?section=windows#section=windows>



Hình 2: PyCharm bản miễn phí sử dụng

2.3.3 Thư viện và framework

Python có một bộ thư viện phong phú và mạnh mẽ, bao gồm NumPy, Pandas, Matplotlib, và scikit-learn cho xử lý dữ liệu và học máy; Django và Flask cho phát triển web; TensorFlow và PyTorch cho trí tuệ nhân tạo; và nhiều thư viện khác nữa. **Pygame** Pygame là một thư viện Python phổ biến được sử dụng để phát triển trò chơi và ứng dụng đa phương tiện tương tác. Với Pygame, ta có thể phát triển các trò chơi 2D đơn giản hoặc phức tạp, từ trò chơi arcade đơn giản đến các ứng dụng giáo dục và giải trí phức tạp.

- Đa nền tảng: Pygame hoạt động trên nhiều nền tảng khác nhau bao gồm Windows, macOS và Linux, cho phép bạn phát triển trò chơi và ứng dụng trên nhiều hệ thống khác nhau mà không cần phải thay đổi mã nguồn.
- Đồ họa và âm thanh: Pygame cung cấp các chức năng để vẽ đồ họa, xử lý hình ảnh, và phát lại âm thanh. Bạn có thể tạo ra các đối tượng hình học, văn bản, hình ảnh và video, cũng như điều khiển âm thanh và nhạc trong trò chơi của mình.
- Xử lý sự kiện: Pygame cho phép bạn xử lý sự kiện người dùng như nhấn phím, di chuyển chuột và nhấn nút, giúp bạn tạo ra các tương tác người dùng tương tác trong trò chơi của mình.
- Hiệu suất: Pygame được viết bằng ngôn ngữ C và Python, cho phép bạn tận dụng các tính năng hiệu suất của ngôn ngữ C trong khi vẫn có thể sử dụng Python để viết mã ngắn gọn và dễ đọc.

- Cộng đồng và tài liệu: Pygame có một cộng đồng lớn và nhiệt tình, cung cấp nhiều tài liệu, hướng dẫn và ví dụ mã nguồn cho người mới bắt đầu và những người muốn phát triển trò chơi và ứng dụng đa phương tiện sử dụng Pygame.
- Đồ họa 2D: Pygame cung cấp các công cụ để vẽ đồ họa 2D, bao gồm hình ảnh, văn bản và hình học cơ bản như hình tròn và hình vuông.
- Âm thanh: Bạn có thể sử dụng Pygame để phát và điều khiển âm thanh trong trò chơi của mình
- Collision detection: Pygame hỗ trợ phát hiện va chạm giữa các đối tượng trong trò chơi, giúp bạn xây dựng các tính năng như va chạm và phản ứng.
- Các công cụ phát triển: Pygame đi kèm với các công cụ hỗ trợ như sprite và group để quản lý các đối tượng trong trò chơi.

Random Thư viện random trong Python cung cấp các chức năng để tạo ra các số ngẫu nhiên và thực hiện các hoạt động liên quan đến ngẫu nhiên.

- random(): Chức năng này trả về một số ngẫu nhiên từ phân phối đều trong khoảng từ 0 đến 1.
- randint(a, b): Trả về một số nguyên ngẫu nhiên nằm trong khoảng từ a đến b (bao gồm cả a và b).
- choice(seq): Chọn một phần tử ngẫu nhiên từ một chuỗi hoặc danh sách seq.
- shuffle(seq): Trộn ngẫu nhiên các phần tử trong một chuỗi hoặc danh sách seq.
- sample(population, k): Trả về một danh sách con ngẫu nhiên với k phần tử từ một tập hợp (population) không trùng lặp.
- uniform(a, b): Trả về một số ngẫu nhiên có phân phối đều trong khoảng từ a đến b, bao gồm cả a và b.
- gauss(mu, sigma): Trả về một số ngẫu nhiên từ phân phối Gaussian (chuẩn) với giá trị trung bình mu và độ lệch chuẩn sigma.

2.3.4 Hệ thống quản lý gói:

Pip là một công cụ quản lý gói cho Python, cho phép bạn cài đặt và quản lý các gói và dependencies cho dự án của mình.

```

import random

# Tạo một số ngẫu nhiên từ 0 đến 1
print(random.random())

# Tạo một số nguyên ngẫu nhiên từ 1 đến 100
print(random.randint(1, 100))

# Chọn một phần tử ngẫu nhiên từ danh sách
my_list = [1, 2, 3, 4, 5]
print(random.choice(my_list))

# Trộn ngẫu nhiên các phần tử trong danh sách
random.shuffle(my_list)
print(my_list)

# Lấy một mẫu ngẫu nhiên không trùng lặp từ danh sách
print(random.sample(my_list, 3))

```

Hình 3: Minh họa thư viện Random trong Python

2.3.5 Cộng đồng và tài nguyên học

Python có một cộng đồng lớn và nhiệt tình, với nhiều diễn đàn, mailing list, và các nguồn tài liệu học như sách, video hướng dẫn, và bài giảng trực tuyến giúp bạn học và giải quyết vấn đề trong quá trình phát triển.

2.3.6 Môi trường ảo (Virtual environments)

Môi trường ảo cho phép bạn tạo ra một môi trường lập trình riêng biệt, với các phiên bản Python và các thư viện cụ thể cho mỗi dự án, giúp tránh xung đột và xung đột giữa các dependencies của dự án.

2.3.7 Công cụ gỡ lỗi (Debugging tools)

Python cung cấp các công cụ gỡ lỗi như pdb và PyCharm Debugger giúp bạn xác định và sửa lỗi trong mã của mình.

3 Xây dựng Game Caro

3.1 Tổng quan về game

Game Caro, còn được biết đến với tên gọi Gomoku hoặc Three in a Row, là một trò chơi bàn cờ hai người chơi, thường được chơi trên một bảng ô vuông. Trò chơi này yêu cầu người chơi đặt các quân cờ của mình trên bảng theo lượt, với mục tiêu là tạo thành một dãy liên tiếp gồm n quân cờ theo hàng ngang, dọc hoặc chéo trên bảng. Trò chơi Caro là một trò chơi giải trí tuyệt vời, kết hợp giữa sự đơn giản và sâu sắc, thu hút người chơi mọi lứa tuổi.

- Bảng chơi của Caro thường có kích thước lớn và được chia thành lưới ô vuông. Có hai loại quân cờ, thường là 'X' và 'O', tương ứng với hai người chơi.
- Người chơi đặt quân cờ của mình lên bảng theo lượt. Mục tiêu của mỗi người chơi là tạo thành một chuỗi liên tiếp gồm n quân cờ trên bảng (thường là năm), theo hàng ngang, dọc hoặc chéo. Người chơi nào tạo thành chuỗi n quân cờ đầu tiên sẽ chiến thắng trò chơi.
- Trò chơi Caro yêu cầu sự kỹ năng, chiến thuật và dự đoán của người chơi. Các người chơi thường cố gắng tạo ra các chuỗi liên tiếp trên bảng của họ trong khi ngăn chặn đối thủ làm điều tương tự.
- Trò chơi Caro có thể được chơi trên nhiều nền tảng khác nhau, bao gồm bảng giấy, bảng điện tử và trò chơi trực tuyến. Có các phiên bản trò chơi Caro được phát triển với đồ họa hấp dẫn và tính năng mạng, cho phép người chơi kết nối và thi đấu với nhau từ xa.

3.2 Cấu trúc dự án game cờ Caro

Dự án Game cờ Caro một ứng dụng đơn giản cho trò chơi cờ Caro (tic-tac-toe) với các tính năng chơi trực tuyến và gửi tin nhắn trong game. Thông qua socket để kết nối giữa clients và server, qua đó có thể đồng bộ bàn cờ và tin nhắn trong thời gian thực giữa các người chơi.

Cấu trúc dự án chia làm các thành phần có các chức năng riêng:

- Client xử lý hiển thị giao diện người dùng
- Server xử lý logic liên quan đến server
- Network bao gồm các phương thức kết nối giao tiếp giữa client và server
- Game xử lý logic game

3.2.1 Client

Game xử lý logic game

Khởi tạo màn hình và các biến cần thiết: Code bắt đầu bằng việc khai báo và khởi tạo một số biến cần thiết như màn hình, font chữ, kích thước bảng cờ, và một số màu sắc cơ bản.

```

pygame.font.init()

ChatPanel = 400
width = 600
height = 600
win = pygame.display.set_mode((width + ChatPanel, height))
pygame.init()
pygame.display.set_caption("Caro Game")
WIDTH = 600
HEIGHT = 600
LINE_WIDTH = 15
BOARD_ROWS = 3
BOARD_COLS = 3
SQUARE_SIZE = WIDTH // BOARD_COLS
CIRCLE_RADIUS = SQUARE_SIZE // 3
CROSS_WIDTH = 25
SPACE = SQUARE_SIZE // 4
font = pygame.font.SysFont(None, 24)
GRAY = (200, 200, 200)

players = ["X", "O"]
ChatMessage = []
input_text = ""
StartMessage = []
game = Game(0)

# Màu sắc
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
BLUE = (0, 0, 255)
GREEN = (0, 255, 0)

# Khởi tạo input box và send button
input_box = pygame.Rect(WIDTH + 10, HEIGHT - 50, 280, 40)
send_button = pygame.Rect(WIDTH + 300, HEIGHT - 50, 90, 40)

```

Vẽ bảng cờ và các ký tự 'X' và 'O': Có các hàm `draw_board()` và `draw_figures()` được sử dụng để vẽ bảng cờ và các ký tự 'X' và 'O' lên màn hình.

```
# Vẽ bảng cờ
def draw_board():
    for row in range(1, BOARD_ROWS + 1):
        pygame.draw.line(win, BLACK, (0, row * SQUARE_SIZE), (WIDTH, row *
SQUARE_SIZE), LINE_WIDTH)
    for col in range(1, BOARD_COLS + 1):
        pygame.draw.line(win, BLACK, (col * SQUARE_SIZE, 0), (col *
SQUARE_SIZE, HEIGHT), LINE_WIDTH)

def draw_figures(board):
    for row in range(BOARD_ROWS):
        for col in range(BOARD_COLS):
            if board[row][col] == 'X':
                pygame.draw.line(win, RED, (col * SQUARE_SIZE + SPACE, row *
SQUARE_SIZE + SQUARE_SIZE - SPACE), (col * SQUARE_SIZE + SQUARE_SIZE - SPACE,
row * SQUARE_SIZE + SPACE), CROSS_WIDTH)
                pygame.draw.line(win, RED, (col * SQUARE_SIZE + SPACE, row *
SQUARE_SIZE + SPACE), (col * SQUARE_SIZE + SQUARE_SIZE - SPACE, row *
SQUARE_SIZE + SQUARE_SIZE - SPACE), CROSS_WIDTH)
            elif board[row][col] == 'O':
                pygame.draw.circle(win, BLUE, (int(col * SQUARE_SIZE +
SQUARE_SIZE // 2), int(row * SQUARE_SIZE + SQUARE_SIZE // 2)), CIRCLE_RADIUS,
LINE_WIDTH)
```

Vẽ giao diện người dùng: Có hàm `redraw_Window()` để vẽ giao diện người dùng, bao gồm bảng cờ và hộp chat.

```
def redrawWindow(win, game, p):
    win.fill((128,128,128))

    if not(game.connected()):
        display_message(" Waiting for Player...")
    else:
        draw_board()
        draw_figures(game.board)
        draw_chat()
    pygame.display.update()
```

Vẽ giao diện khung chat nằm bên phải màn hình: Hàm `draw_chat()`.


```
# Hàm vẽ giao diện chat
def draw_chat():
    pygame.draw.rect(win, GRAY, input_box, 0)
    pygame.draw.rect(win, BLACK, input_box, 2)
    pygame.draw.rect(win, BLACK, send_button, 2)

    #Player turn
    global game
    playerTurn_surface = font.render("PLAYER TURN:
"+players[game.playerTurn], True, GREEN)
    win.blit(playerTurn_surface, (WIDTH + 10, 10))

    # Vẽ tin nhắn
    if len(StartMessage) > 0:
        for i, message in enumerate(StartMessage):
            text_surface = font.render(message, True, BLACK)
            win.blit(text_surface, (WIDTH + 10, 30 + i * 24))
    else:
        for i, message in enumerate(ChatMessage):
            text_surface = font.render(message, True, BLACK)
            win.blit(text_surface, (WIDTH + 10, 30 + i * 24))

    # Vẽ input text
    input_surface = font.render(input_text, True, BLACK)
    win.blit(input_surface, (input_box.x + 5, input_box.y + 10))

    # Vẽ nút Send
    send_text = font.render("Send", True, BLACK)
    win.blit(send_text, (send_button.x + 10, send_button.y + 10))
```

Hiển thị tin nhắn: Có hàm `display_message()` để hiển thị tin nhắn trên màn hình.

```
# Hiển thị tin nhắn
def display_message(message):
    pygame.draw.rect(win, BLACK, (0, HEIGHT // 2 - 50, WIDTH, 100))
    font = pygame.font.Font(None, 36)
    text = font.render(message, True, WHITE)
    win.blit(text, (WIDTH // 2 - text.get_width() // 2, HEIGHT // 2 -
text.get_height() // 2))
    pygame.display.update()
    pygame.time.delay(2000)
```

Chơi trực tuyến và gửi tin nhắn: Hàm `PlayOnline()` cho phép người chơi kết nối và chơi trực tuyến với một người chơi khác thông qua mạng. Người chơi có thể gửi tin nhắn trong quá trình chơi.

- Đầu tiên kết nối đến server để xác định xem người chơi hiện tại là X hay O
- Sau khi kết nối đến server thì cập nhật bàn cờ liên tục bằng cách lấy thông tin bàn cờ từ server
- Chỉ khi đến lượt thì người chơi mới có quyền chọn nước đi và thay đổi dữ liệu của bàn cờ


```

# chơi online
def PlayOnline():
    win = pygame.display.set_mode((width + ChatPanel, height))
    run = True
    clock = pygame.time.Clock()
    n = Network()
    player = "X"
    player = n.getP()
    global ChatMessage
    global input_text
    global StartMessage
    global game
    StartMessage = ["You are player " + player, "X play first"]

    while run:
        clock.tick(60)
        try:
            game = n.send("get")
            ChatMessage = game.messages

            #kiểm tra khi có người chơi chat
            if len(ChatMessage) > 0:
                StartMessage = []

        except:
            run = False
            print("Couldn't get game")
            break

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
                pygame.quit()

            if event.type == pygame.MOUSEBUTTONDOWN:
                mouseX = event.pos[0] // SQUARE_SIZE
                mouseY = event.pos[1] // SQUARE_SIZE
                if(mouseX < 3 and mouseY < 3):
                    if game.board[mouseY][mouseX] == '' and game.connected():
                        if(game.playerTurn == 0 and player ==
players[game.playerTurn]):

```

```

        game.board[mouseY][mouseX] = player
        game.changeTurn()
        game = n.send(game)
    if(game.playerTurn == 1 and player ==
players[game.playerTurn]):
        game.board[mouseY][mouseX] = player
        game.changeTurn()
        game = n.send(game)
    if send_button.collidepoint(event.pos):
        # Gửi tin nhắn từ input
        ChatMessage.append(player + " send: " + input_text)
        # Reset input text sau khi gửi
        input_text = ""
        game.messages = ChatMessage
        game = n.send(game)

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_BACKSPACE:
            input_text = input_text[:-1] # Xóa ký tự cuối cùng
        elif event.key == pygame.K_RETURN:
            # Gửi tin nhắn từ input
            ChatMessage.append(player + " send: " + input_text)
            # Reset input text sau khi gửi
            input_text = ""
            game.messages = ChatMessage
            game = n.send(game)
        else:
            input_text += event.unicode

    #check player win
    if game.check_win():
        redrawWindow(win, game, player)
        display_message(players[game.winner] + " wins!")
        game = n.send("reset")
    elif game.checkTie():
        redrawWindow(win, game, player)
        display_message("Tie Game!")
        game = n.send("reset")
    redrawWindow(win, game, player)

```

Chơi với máy (bot): Hàm `PlayWithBot()` cho phép người chơi chơi với một trí tuệ nhân tạo (bot) được xây dựng sẵn. Bot sẽ tự động chọn nước đi của mình sau khi người chơi đã đi.

```

# chơi với máy
def PlayWithBot():
    run = True
    win.fill(WHITE)
    draw_board()
    # Khởi tạo bảng cờ
    global game
    game.resetGame()
    player = 'X'
    draw_board()
    while run:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
                pygame.quit()

            if event.type == pygame.MOUSEBUTTONDOWN:
                mouseX = event.pos[0] // SQUARE_SIZE
                mouseY = event.pos[1] // SQUARE_SIZE

                if game.board[mouseY][mouseX] == '':
                    game.board[mouseY][mouseX] = player
                    draw_figures(game.board)
                    pygame.display.update()
                    if game.check_win():
                        display_message(players[game.winner] + " wins!")
                        run = False
                    else:
                        bot_play()
                        draw_figures(game.board)
                        pygame.display.update()
                        if game.check_win():
                            display_message(players[game.winner] + " wins!")
                            run = False
                        elif game.checkTie():
                            display_message("Tie Game!")
                            run = False
                    pygame.display.update()

```

Giao diện menu: Có hàm `menu_win()` để hiển thị giao diện menu cho người chơi chọn chế độ chơi (trực tuyến hoặc với bot).

```

def menu_win():
    win = pygame.display.set_mode((width, height))
    run = True
    clock = pygame.time.Clock()

    while run:
        clock.tick(60)
        win.fill((128, 128, 128))
        for btn in btns:
            btn.draw(win)
        pygame.display.update()

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                run = False
            if event.type == pygame.MOUSEBUTTONDOWN:
                pos = pygame.mouse.get_pos()
                for btn in btns:
                    if btn.click(pos):
                        run = False
                        if btn.text == "Play Online":
                            PlayOnline()
                        if btn.text == "Play with bot":
                            PlayWithBot()

```

3.2.2 Server

Thực hiện chức năng của server trong trò chơi cờ Caro. Dưới đây là mô tả chi tiết về chức năng của từng phần:

- Thiết lập kết nối và lắng nghe: Server được thiết lập để lắng nghe các kết nối từ client trên một địa chỉ IP và cổng cụ thể.
- Xử lý các kết nối từ client: Khi một client kết nối, server sẽ chấp nhận kết nối đó và tạo một luồng mới để xử lý kết nối đó.
- Phân biệt và xử lý dữ liệu từ client: Server nhận dữ liệu từ client và phân biệt xem dữ liệu đó là loại gì (ví dụ: thông điệp văn bản, trạng thái trò chơi, hoặc trạng thái bàn cờ). Dựa vào loại dữ liệu, server sẽ xử lý tương ứng, ví dụ như reset trò chơi, hoặc cập nhật trạng thái bàn cờ và gửi lại cho tất cả client trong cùng một trò chơi.
- Quản lý trạng thái trò chơi và bàn cờ: Server quản lý các trò chơi đang diễn ra và trạng thái của bàn cờ trong mỗi trò chơi. Nó lưu trữ thông tin về các trò chơi trong một từ điển (dictionary), với key là id của trò chơi và value là đối tượng Game.
- Tạo game mới và gán người chơi: Khi một client kết nối, server sẽ xác định xem client đó là người chơi thứ nhất (X) hay thứ hai (O). Nếu đó là người

chơi thứ hai, server sẽ thêm client đó vào một trò chơi có sẵn (nếu có), nếu không sẽ tạo ra một trò chơi mới và chờ đợi người chơi thứ hai kết nối.

- Gửi thông điệp và trạng thái trò chơi cho client: Server gửi thông điệp và trạng thái trò chơi cho tất cả client trong cùng một trò chơi thông qua kết nối của họ.
- Quản lý đa luồng: Server sử dụng đa luồng để xử lý nhiều kết nối từ client cùng một lúc mà không gây ra trễ hoặc đóng kết nối không mong muốn.

Cài đặt code:

```
import socket
from _thread import *
import pickle
from game import Game
import struct

server = "192.168.1.7"
port = 5555

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    s.bind((server, port))
except socket.error as e:
    str(e)

s.listen(2)
print("Waiting for a connection, Server Started")

connected = set()
games = {}
idCount = 0

def threaded_client(conn, p, gameId):
    global idCount
    conn.send(str.encode(str(p)))
    print("Player",p," connected")

    reply = ""
    while True:
        try:
            data = conn.recv(4096)
            game = games[gameId]
            # Giải mã dữ liệu nhận được
            received_object = pickle.loads(data)
            if not received_object:
                break
            else:
```



```

        # Xác định loại đối tượng nhận được và xử lý tương ứng
        if isinstance(received_object, str):
            if received_object == "reset":
                print("reset game")
                game.resetGame()
            elif data != "get":
                pass
            # game.play(p, data)
            games[gameId] = game
            conn.sendall(pickle.dumps(game))
        elif isinstance(received_object, Game):
            games[gameId] = received_object
            conn.sendall(pickle.dumps(received_object))

    except:
        break

    print("Lost connection")
    try:
        del games[gameId]
        print("Closing Game", gameId)
    except:
        pass
    idCount -= 1
    conn.close()

player = ["X", "O"]
while True:
    conn, addr = s.accept()
    print("Connected to:", addr)

    idCount += 1
    p = 0
    gameId = (idCount - 1) // 2
    if idCount % 2 == 1:
        games[gameId] = Game(gameId)
        print("Creating a new game...")
    else:
        games[gameId].ready = True
        p = 1

    start_new_thread(threaded_client, (conn, player[p], gameId))

```

3.2.3 NetWork

Được sử dụng để kết nối và truyền dữ liệu giữa client và server trong trò chơi cờ Caro. Dưới đây là mô tả chi tiết về chức năng của từng phần:

- **init()**: Phương thức khởi tạo của lớp Network. Trong phương thức này, một đối tượng socket được tạo để thiết lập kết nối. Địa chỉ và cổng của server cũng được xác định ở đây.

- **getP()**: Phương thức này trả về biến p, đại diện cho loại người chơi (X hoặc O) được gửi từ server sau khi kết nối.
- **connect()**: Phương thức này được gọi để kết nối đến server. Nếu kết nối thành công, client nhận được loại người chơi từ server và trả về biến p.
- **send(data)**: Phương thức này được sử dụng để gửi dữ liệu từ client đến server. Dữ liệu được pickle và gửi dưới dạng byte stream. Sau khi gửi dữ liệu, client nhận lại dữ liệu từ server và giải pickle để nhận đối tượng tương ứng. Nếu không nhận được dữ liệu hoặc gặp lỗi socket, phương thức sẽ thông báo lỗi.

Cài đặt Code:

```
import socket
import pickle
from game import Game
import struct

class Network:
    def __init__(self):
        self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server = "192.168.1.7"
        self.port = 5555
        self.addr = (self.server, self.port)
        self.p = self.connect()

    def getP(self):
        return self.p

    def connect(self):
        try:
            self.client.connect(self.addr)
            # print("Player :",self.client.recv(2048).decode())
            return self.client.recv(2048).decode()
        except:
            pass

    def send(self, data):
        try:
            self.client.send(pickle.dumps(data))
            return pickle.loads(self.client.recv(2048*2))
```

```

if not received_object:
    pass
else:
    # Xác định loại đối tượng nhận được và xử lý tương ứng
    if isinstance(received_object, str):
        pass
    elif isinstance(received_object, Game):
        return pickle.loads(self.client.recv(2048*2))
    # elif isinstance(received_object, Message):
    #     print(f"Received message from {received_object.sender}:
{received_object.message}")
except socket.error as e:
    print("socket error ",e)

```

3.2.4 Game

Biểu diễn trạng thái và logic của trò chơi cờ Caro. Dưới đây là mô tả chi tiết về các phương thức và thuộc tính của lớp Game:

- **init()**: Phương thức khởi tạo của lớp Game. Trong phương thức này, các thuộc tính cơ bản của trò chơi được khởi tạo như số hàng và cột của bàn cờ, lượt chơi hiện tại, trạng thái sẵn sàng, id của trò chơi, người chiến thắng, tin nhắn trong trò chơi và bàn cờ.
- **connected()**: Phương thức kiểm tra xem trò chơi có sẵn sàng để bắt đầu hay không. Nó trả về giá trị của thuộc tính ready.
- **bothWent()**: Phương thức kiểm tra xem cả hai người chơi đã thực hiện nước đi của mình chưa. Trong đoạn mã đã đưa ra, dường như thiếu một số thông tin liên quan đến các biến p1Went và p2Went.
- **check_player_win(mark)**: Phương thức kiểm tra xem người chơi có đánh chiến thắng hay không. Nó kiểm tra các hàng, cột và đường chéo của bàn cờ để xác định xem có 3 ô liên tiếp giống nhau và không trống hay không. Phương thức này nhận đối số mark để chỉ định người chơi cần kiểm tra.
- **checkTie()**: Phương thức kiểm tra xem trò chơi có kết thúc hòa hay không. Nó đếm số ô đã được đánh trên bàn cờ và trả về True nếu bàn cờ đã được điền đầy.
- **check_win()**: Phương thức kiểm tra xem có người chiến thắng hay không. Nó gọi **check_player_win()** cho cả hai người chơi và xác định người chiến thắng dựa trên kết quả.
- **changeTurn()**: Phương thức thay đổi lượt chơi từ người chơi hiện tại sang người chơi kế tiếp.
- **resetGame()**: Phương thức reset lại trò chơi để chuẩn bị cho một ván mới. Tất cả các thuộc tính của trò chơi được thiết lập lại vào giá trị ban đầu.

Cài đặt Code:


```

class Game:
    def __init__(self, id):
        self.BOARD_ROWS = 3
        self.BOARD_COLS = 3
        self.playerTurn = 0
        self.ready = False
        self.id = id
        self.winner = 0
        self.messages = []
        self.board = [['' for _ in range(self.BOARD_COLS)] for _ in
range(self.BOARD_ROWS)]

    def connected(self):
        return self.ready

    def bothWent(self):
        return self.p1Went and self.p2Went

    def check_player_win(self, mark):
        for row in range(self.BOARD_ROWS):
            if self.board[row][0] == self.board[row][1] == self.board[row][2] ==
mark:
                return True

        for col in range(self.BOARD_COLS):
            if self.board[0][col] == self.board[1][col] == self.board[2][col] ==
mark:
                return True

        if self.board[0][0] == self.board[1][1] == self.board[2][2] == mark:
            return True

        if self.board[0][2] == self.board[1][1] == self.board[2][0] == mark:
            return True

        return False

    def checkTie(self):
        temp = 0
        for row in range(self.BOARD_ROWS):
            for col in range(self.BOARD_COLS):
                if self.board[row][col] == 'X' or self.board[row][col] == 'O':
                    temp += 1

```

```

        if temp == 9:
            return True
        return False

    def check_win(self):
        if self.check_player_win("X"):
            self.winner = 0
            return True
        if self.check_player_win("O"):
            self.winner = 1
            return True
        return False

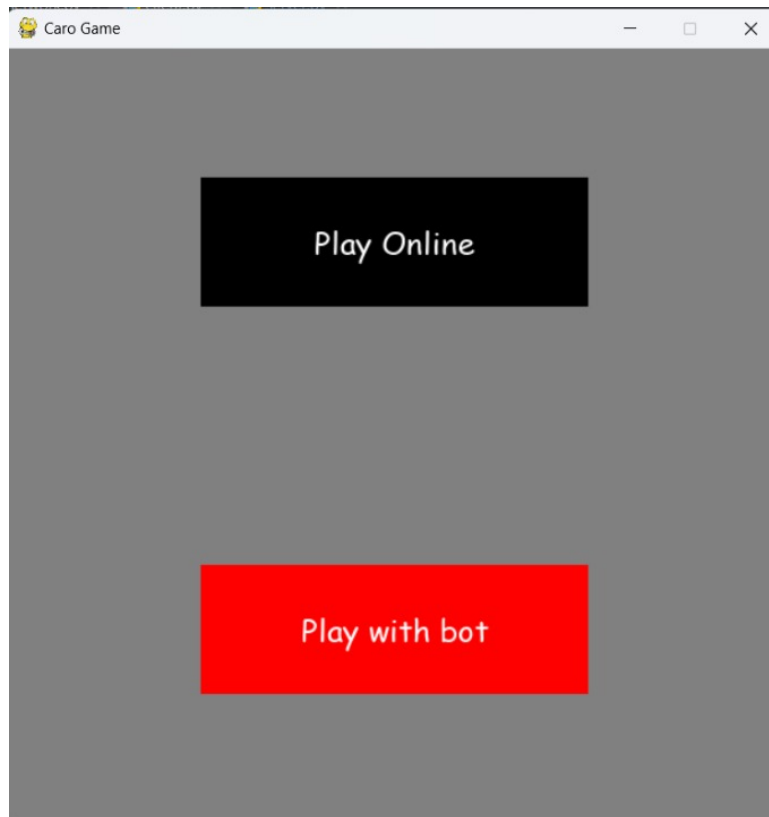
    def changeTurn(self):
        if(self.playerTurn == 0):
            self.playerTurn = 1
        else:
            self.playerTurn = 0

    def resetGame(self):
        self.BOARD_ROWS = 3
        self.BOARD_COLS = 3
        self.playerTurn = 0
        self.ready = True
        self.winner = 0
        self.messages = []
        self.board = [[' ' for _ in range(self.BOARD_COLS)] for _ in
range(self.BOARD_ROWS)]

```

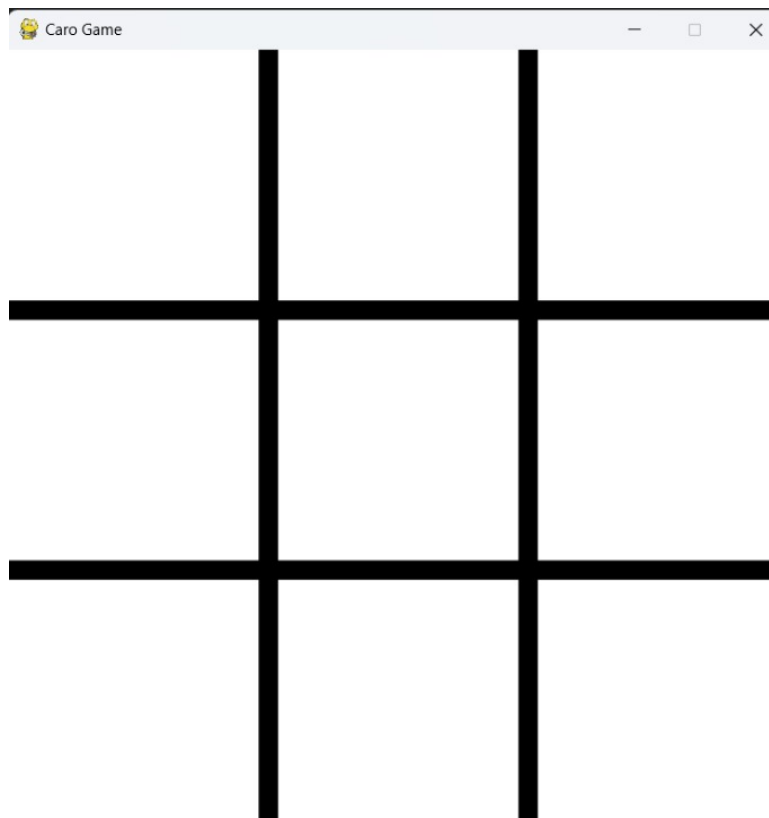
3.3 Giao diện game cờ Caro

Màn hình chính khi vào game, đây là nơi người chơi quyết định chế độ mà mình sẽ chơi đó alf:



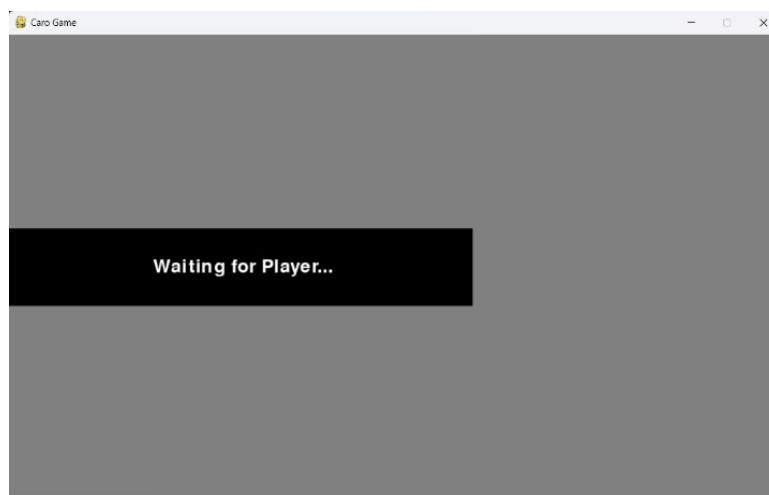
Hình 4: Màn hình chính

Màn hình chơi với máy, đây là chế độ cho phép người chơi đấu với bot với tốc độ phản hồi nhanh hơn so với người thường và do không có trí thông minh nên đây là chế độ rất dễ dàng dành được chiến thắng:



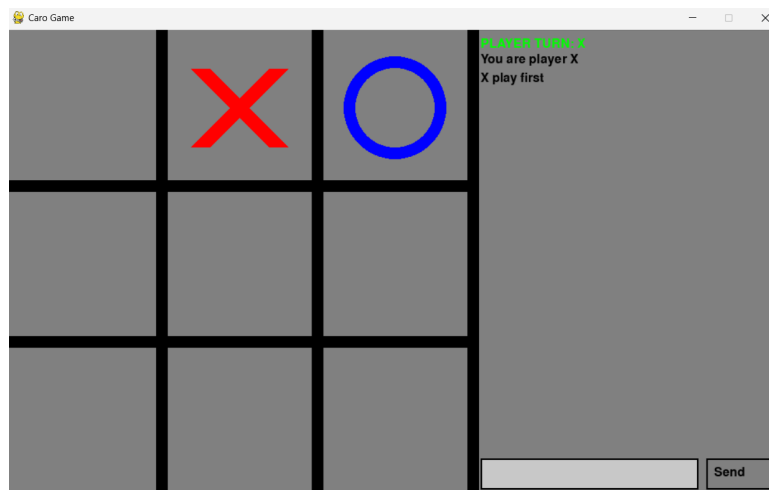
Hình 5: Màn hình chơi với máy

Màn hình chơi online khi đang đợi người chơi khác, đây là cơ chế trong game để cho người dùng biết được rằng hệ thống đang tìm người chơi chứ không phải đang bị đơ:



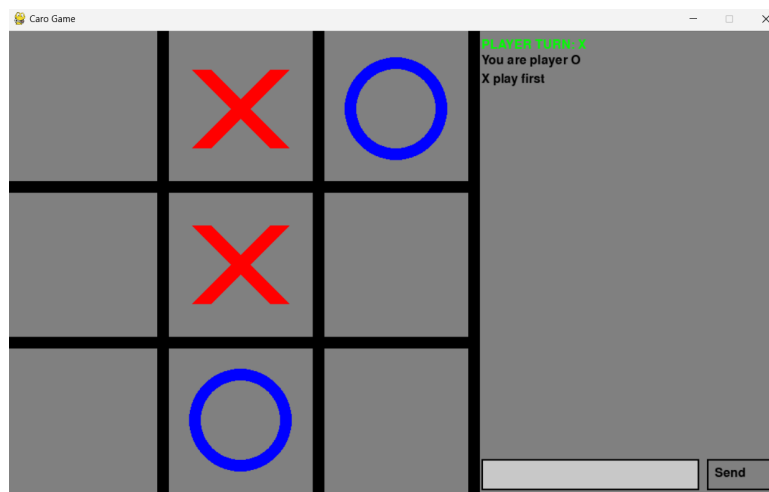
Hình 6: Màn hình chơi online khi đang đợi người chơi khác

Màn hình chơi online khi vào game ở bên X, bên X là bên bắt đầu trò chơi nên bên X sẽ chơi trước:



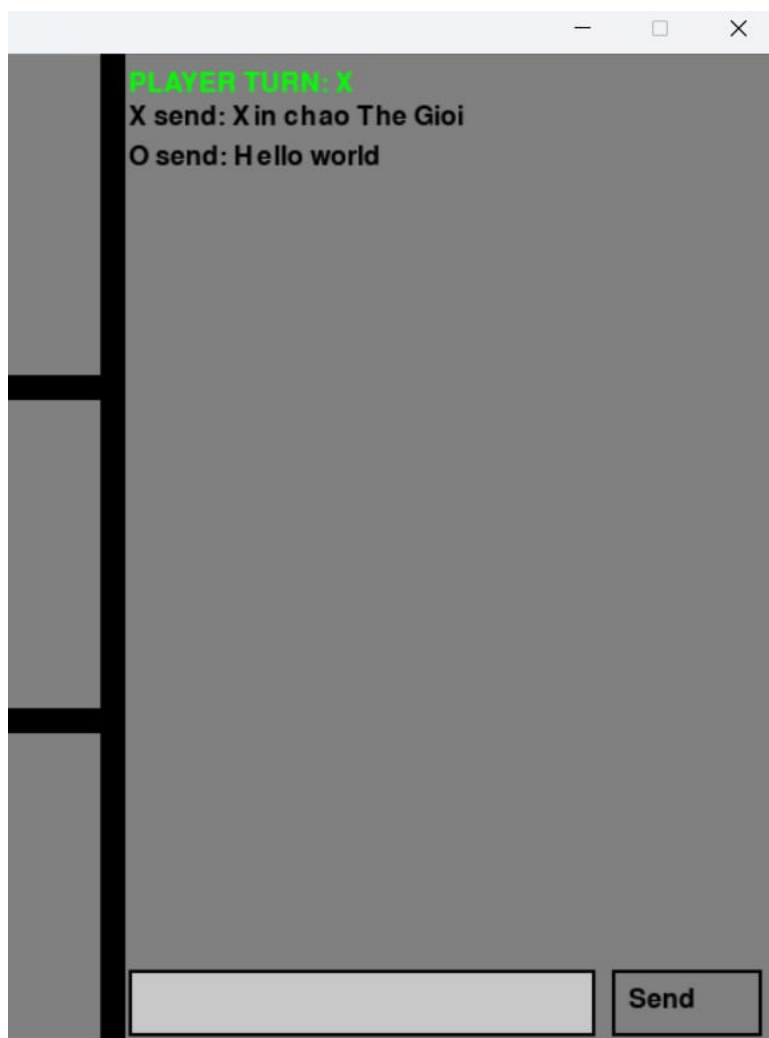
Hình 7: Màn hình chơi online ở phía X

Màn hình chơi online khi vào game ở bên O:



Hình 8: Màn hình chơi online ở phía O

Màn hình ở mục chat, game cho phép người chơi giao tiếp với nhau thông qua chat:



Hình 9: Màn hình ở mục chat

3.4 Hướng dẫn cài đặt game

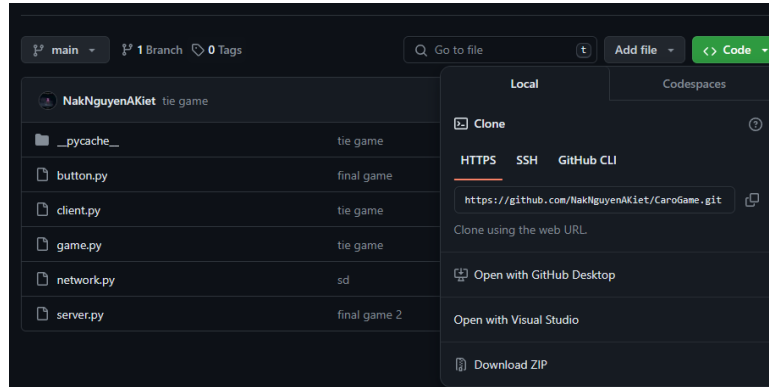
3.4.1 Trên Window

Cách 1:

Bước 1: Truy cập đường dẫn sau để đến file trên GitHub:

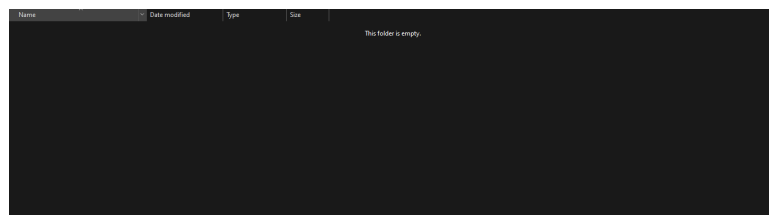
<https://github.com/NakNguyenAKiet/CaroGame.git>

Bước 2: Copy đường dẫn trên hoặc tham khảo hình dưới.



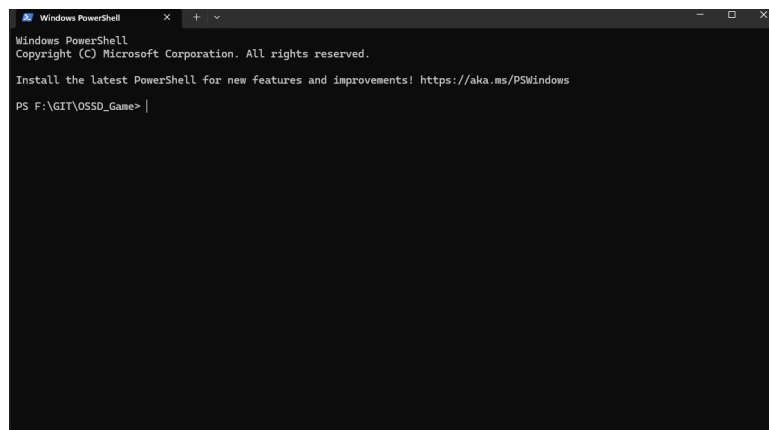
Hình 10: Đường dẫn để tải game

Bước 3: Chọn thư mục mà bạn muốn đặt game vào.



Hình 11: Tạo thư mục chứa game

Bước 4: Trong thư mục đó bấm chuột phải, chọn open in Terminal.



Hình 12: Mở Terminal

Bước 5: Trong Terminal bấm lệnh sau `git init`

```
PS F:\GIT\OSSD_Game> git init
Initialized empty Git repository in F:/GIT/OSSD_Game/.git/
PS F:\GIT\OSSD_Game> |
```

Hình 13: Lệnh git init

Bước 6: Tiếp theo bấm lệnh `git clone` và thêm đường dẫn

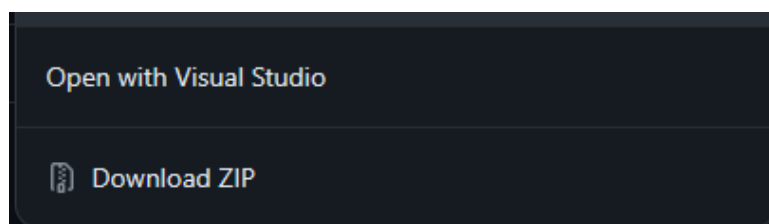
```
PS F:\GIT\OSSD_Game> git clone https://github.com/NakNguyenAKiet/CaroGame.git
Cloning into 'CaroGame'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 41 (delta 19), reused 34 (delta 12), pack-reused 0
Receiving objects: 100% (41/41), 14.61 KiB | 277.00 KiB/s, done.
Resolving deltas: 100% (19/19), done.
PS F:\GIT\OSSD_Game> |
```

Hình 14: Lệnh git clone

Cách 2:

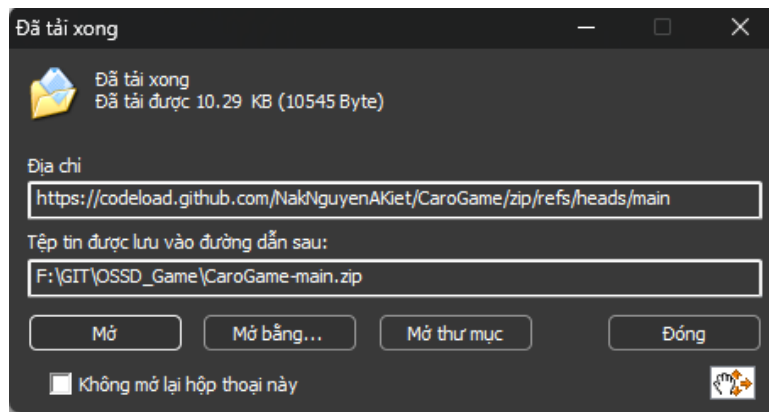
Bước 1: Tương tự như trên.

Bước 2: Chọn download file .zip



Hình 15: Chọn download file zip

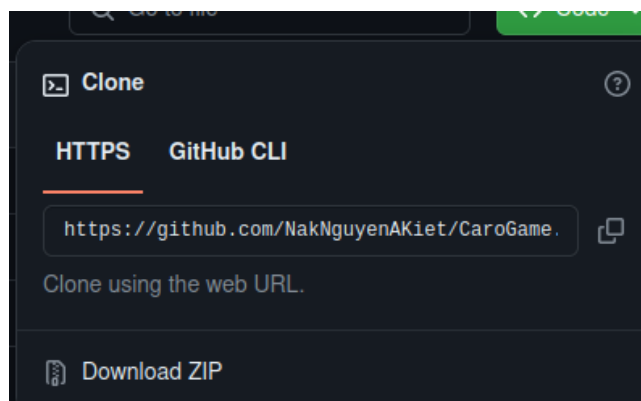
Bước 3: Tải hoàn thành, giải nén.



Hình 16: Tải hoàn thành

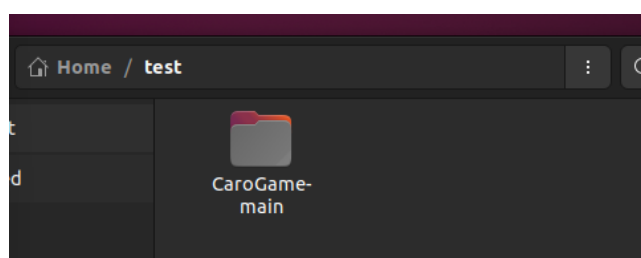
3.4.2 Trên Linux

Bước 1: Tải file zip



Hình 17: Download trên linux

Bước 2: Chọn giải nén vào folder mong muốn.



Hình 18: Extract To

3.4.3 Để chơi game:

Do game có cơ chế chơi online nên chúng ta cần thay đổi một bài thông số trong game để có thể sử dụng hết tính năng của game.

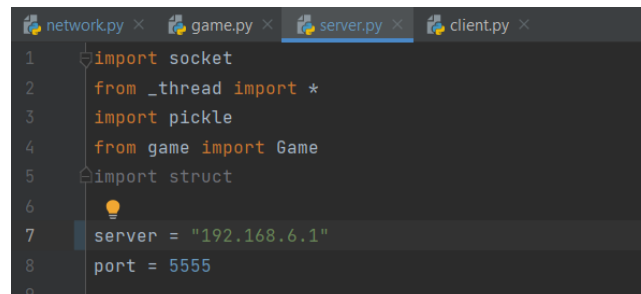
- Vào Python và mở thư mục chứa game
- Mở Terminal nhấn **ipconfig**

Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix  . :  
Link-local IPv6 Address . . . . . : fe80::f101:a3aa:703b:9c2a%8  
IPv4 Address. . . . . : 192.168.1.6  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.1.1
```

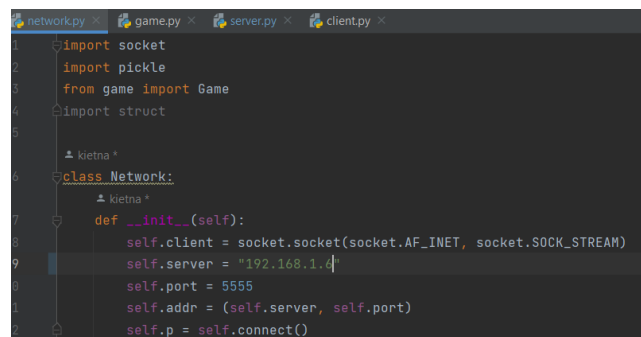
Hình 19: Địa chỉ Ipv4 cần thay đổi

- Copy địa chỉ IPv4 và thay đổi vào biến **server** ở file **server.py** và **network.py**



```
1 import socket  
2 from _thread import *  
3 import pickle  
4 from game import Game  
5 import struct  
6  
7 server = "192.168.6.1"  
8 port = 5555  
9
```

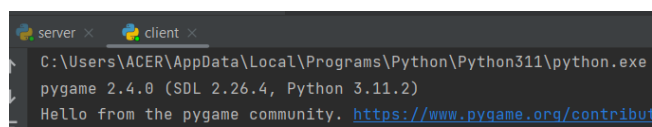
Hình 20: Đổi file server



```
1 import socket  
2 import pickle  
3 from game import Game  
4 import struct  
5  
6 class Network:  
7     def __init__(self):  
8         self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
9         self.server = "192.168.1.6"  
10        self.port = 5555  
11        self.addr = (self.server, self.port)  
12        self.p = self.connect()
```

Hình 21: Đổi file network

- Để 2 thiết bị có thể chơi online được với nhau cần thay đổi địa chỉ server trên máy bằng địa chỉ server của máy chủ đang mở server
- Khi muốn chơi game cần chạy file sever, sau đó mở file client để chơi game.



Hình 22: Chạy file server trước rồi mới chạy file client