**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

bdôca



**CHƠI CỜ CARO QUA MẠNG LAN**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện: | | |
| STT | Họ tên | MSSV |
| 1 | Nguyễn Hà Duy Khang | 19521664 |
| 2 | Hoàng Minh Nhựt | 19521975 |

**TP. HỒ CHÍ MINH – 5/2022**

**Mục lục**

[1. GIỚI THIỆU 1](#_Toc103288375)

[2. NỘI DUNG 2](#_Toc103288376)

[2.1. Phương pháp 2](#_Toc103288377)

[2.1.1. Tạo giao diện cờ caro 2](#_Toc103288378)

[2.1.2. Tạo client và server 2](#_Toc103288379)

[2.1.3 Yêu cầu hệ thống 2](#_Toc103288380)

[2.2. Thiết kế 3](#_Toc103288381)

[2.2.1. Giao diện cờ caro 3](#_Toc103288382)

[2.2.2. Thiết kế client và server 4](#_Toc103288386)

[2.3. Kết quả 6](#_Toc103288387)

[3. KẾT LUẬN 8](#_Toc103288388)

[PHỤ LỤC PHÂN CÔNG NHIỆM VỤ 10](#_Toc103288389)

**Mục lục hình ảnh**

[Hình 2.1. Sơ đồ hoạt động phương thức handlebutton 5](#_Toc103289494)

[Hình 2.2. Tương tác của server và client 6](#_Toc103289495)

[Hình 2.3. Giao diện của cờ caro 7](#_Toc103289496)

[Hình 2.4. IP do server cấp 7](#_Toc103289497)

[Hình 2.5. Thông báo bên server 8](#_Toc103289498)

[Hình 2.6. Client đang kết nối 8](#_Toc103289499)

[Hình 2.7. Server kết nối thành công 8](#_Toc103289500)

[Hình 2.8. Kết quả đạt được 9](#_Toc103289501)

# GIỚI THIỆU

Đề tài này là chơi cờ caro qua mạng lan. Đây là một trò chơi cổ điển mang tên cờ Caro, luật chơi đơn giản đó là chúng ta có 2 người chơi tượng trưng cho ‘X’ và ‘O’, và 2 người chơi lần lượt đánh vào bên trong ô chỉ định còn trống, nếu ai có 5 điểm liên tiếp nhau đầu tiên sẽ giành chiến thắng.

Đồ án này được thực hiện trên công cụ Pycharm với các thư viện như tkinter, threading, socket.

Phương pháp:

* Tìm hiểu thư viện tkinter để thiết kế giao diện cờ caro. Lập trình cờ caro.
* Tìm hiểu thư viện threading, socket. Lập trình socket và threading để tạo kết nối giữa client và server và xử lý đa luồng.

# NỘI DUNG

**2.1. Phương pháp**

**2.1.1. Tạo giao diện cờ caro**

Tạo giao diện cờ caro gồm có 2 frame. Frame 1 gồm có button newgame, undo, phần nhập địa chỉ IP, button connect và makehost. Frame 2 chứa một mảng hai chiều chứa các button để khi nhấn vào sẽ hiện lên “O” hoặc “X”.

Button newgame sẽ có chức năng là tạo một game mới, xóa hết các kí tự hiện trên các button ở frame 2 chứa mảng 2 chiều.

Button undo có chức năng quay lại nước đi, xóa kí tự vừa mới hiện lên trên button vừa nhấn.

Phần nhập địa chỉ IP có chức năng cho người chơi nhập địa chỉ IP vào. Địa chỉ IP đó sẽ được truyền vào nút connect để tiến hành kết nối với người chơi khác.

Button makehost có chức năng hiện ra địa chỉ IP của nguời chơi thứ nhất và người chơi thứ hai sẽ nhập địa chỉ IP đó để kết nối.

**2.1.2. Tạo client và server**

Cấu hình client nhận địa chỉ IP của server và tiến hành kết nối.

Cấu hình server để lấy địa chỉ IP hiện tại và thiết lập lắng nghe client.

**2.1.3 Yêu cầu hệ thống**

**2.1.3.1 Mô tả hệ thống**

Phía máy nào chọn makehost, xuất ra địa chỉ IP sẽ là máy server, còn phía máy

nào nhập địa chỉ IP và tiến hành kết nối sẽ là máy client.

Khi máy server chọn newgame sẽ xóa hết các kí tự ở trong ô, máy client sẽ xóa hết theo máy server và ngược lại.

Khi máy server đánh vào, máy client sẽ hiện ra màn hình theo máy server đã đánh và ngược lại.

Khi máy server chọn undo sẽ lui lại nước đi lần trước, xóa đi kí tự ở ô vừa lúc nãy, máy client sẽ xóa theo kí tự ở ô đó và ngược lại.

**2.1.3.2 Ý tưởng hệ thống**

Khi máy server và máy client có những thao tác gì, đều sẽ gửi dữ liệu qua lại với nhau. Máy server thao tác, phía máy server sẽ gửi dữ liệu gồm loại thao tác, vị trí của ô đó. Phía máy client sẽ nhận dữ liệu đó và tách dữ liệu đó ra thành loại thao tác, vị trí của ô và tiến hành loại thao tác tương tự bên server và ngược lại.

Ví dụ: máy server đánh vào ô 3x4 thì sẽ gửi dữ liệu gồm (“hit” | 3| 4). Sau đó phía máy client sẽ nhận dữ liệu đó và tách dữ liệu theo dấu | thành mảng và tạo 3 biến để gán 3 giá trị vào. Nếu loại thao tác là “hit”, bên máy client sẽ thực hiện thao tác tương tụ bên phía server và ngược lại.

**2.2. Thiết kế**

**2.2.1. Giao diện cờ caro**

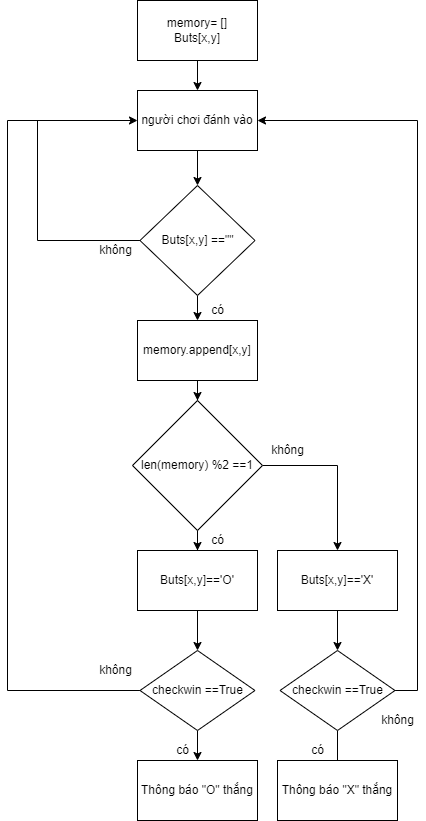
Đầu tiên, chúng tôi cài đặt các thư viện cần thiết bao gồm tkinter để sử dụng các layer có sẵn như button, khung nhập input,… .

Tạo một đối tượng cờ caro gồm các thuộc tính như:

* Memory: lưu lại những nước đi.
* Buts: tạo một mảng 2 chiều chứa button.
* Threading\_socket: sử dụng class của threading\_socket để gọi phương thức.

Và các phương thức như:

* Showframe: hiển thị giao diện cờ caro gồm có button newgame, button undo, text\_input, button connect, button makehost và một mảng 2 chiều chứa các button.
* Notification: Hàm này sẽ sử dụng messagebox của thư viện tkinter để xuất thông báo khi kết nối server thành công, khi “O” hoặc “X” chiến thắng.
* Undo: quay lại nước đi trước bằng cách xóa đi kí tự thêm cuối cùng ở memory.
* Checkwin: hàm này sẽ kiểm tra các trường hợp thắng của “O” hoặc “X” bằng cách kiểm tra xem hàng ngang, hàng dọc, hàng chéo phải, hàng chéo trái của “O” hoặc “X” có đủ 5 kí tự không, nếu đủ xuất thông báo chiến thắng.
* Newgame: xóa hết tất cả các kí tự trên các button ở Buts.
* Handlebutton: khi bấm các button sẽ thêm kí tự. Mặc định lượt “O” sẽ được đi trước. Đầu tiên sẽ kiểm tra ô đó có đang trống không, nếu không trống sẽ không cho người chơi chọn vào, nếu có thì xem lượt đi này có phải bước chẵn không. Nếu là bước chẵn, đánh sẽ ra kí tự “O”, nếu lẻ sẽ đánh ra kí tự “X”. Sau đó kiểm tra qua hàm checkwin sau khi đánh vào có hay không, nếu có thì xuất ra thông báo và kết thúc.



Hình .. Sơ đồ hoạt động phương thức handlebutton



### Thiết kế client và server

**2.2.2.1. Tạo socket kết nối giữa server và client**

Tạo một luồng kết nối để kết nối server và client với nhau bằng cách tạo một class threading\_socket gồm các thuộc tính như:

* Datareceive: nhận và lưu lại dữ liệu gửi qua lại của server và client.
* GUI: được gọi từ class caro.
* Name: lưu tên server hoặc client.
* Conn: tạo connect và cũng để gửi dữ liệu.

Đầu tiên, chúng tôi tự khởi tạo địa chỉ cho server và dùng chung một cổng PORT để có thể truyền qua lại được với nhau, sau đó khởi tạo cấu hình socket, nơi có thể nhận và truyền dữ liệu cho server và client, tiếp theo chúng tôi kết nối với server cũng như tạo luồng chạy qua client.

Tuy nhiên chúng ta mới có được ‘hành động’ của client chứ chưa có client, client cần phải hiểu được server đã gửi những gì để có thể xuất ra màn hình một cách chính xác.

Tiếp theo, khởi tạo server, nơi client kết nối đến, lúc này chúng ta cần lập một địa chỉ để bên client có thể kết nối chung một luồng, cũng thiết lập chung một cổng PORT, chung một socket, và chỉ cho phép thiết lập tối đa một kết nối cùng một lúc.

Khác với ‘hành động’ của client, ngoài có thể đọc được nội dung client được gửi đến thì ‘hành động’ của server cần phải có thông tin địa chỉ của client thì mới có thể đọc được nội dung client gửi đến và cuối cùng là đóng được socket để thoát trò chơi.

**2.2.2.2. Tạo luồng chạy cho server và client**

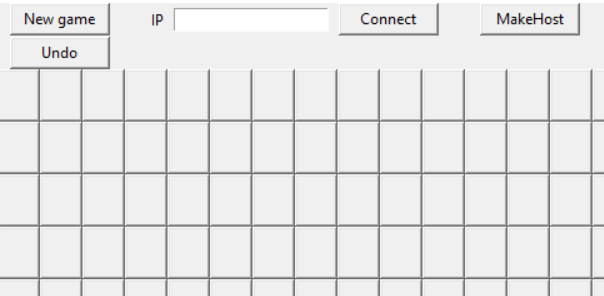
Khi phía client đánh vào, ở hàm handlebutton bên class caro sẽ gửi lên conn một bộ dữ liệu gồm (client, hit, x, y), hit là thao tác nhận biết client đánh vào, x và y là vị trí đánh vào. Lúc đó phía server sẽ nhận bộ dữ liệu đó và tách bộ dữ liệu đó thành 4 giá trị. Nếu người gửi là client và thao tác là hit thì sẽ thực hiện handlebutton ở vị trí x,y tương tự phía client và ngược lại. Ngoài ra còn các thao tác như undo, newgame.

A picture containing table

Description automatically generated

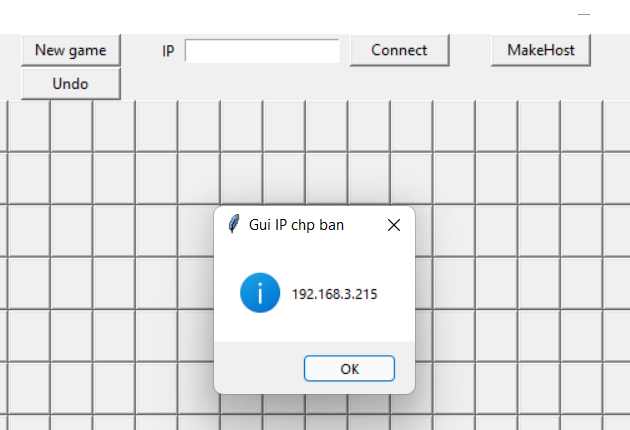
Hình .. Tương tác của server và client

## Kết quả



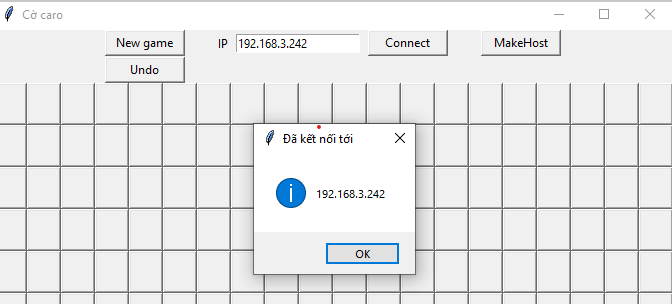
Hình .. Giao diện của cờ caro

Sau khi chạy chương trình, nếu là máy Host thì người chơi cần phải gửi IP cho client.



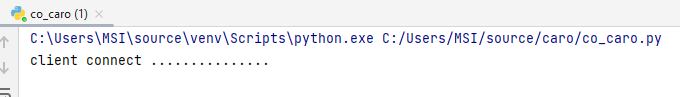
Hình .. IP do server cấp

Sau khi client đăng nhập đúng với địa chỉ IP thì phía bên host sẽ nhận được thông báo “ Đã kết nối tới ”.



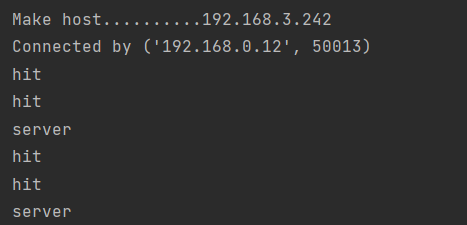
Hình .. Thông báo bên server

Lúc này đây bên dưới Pycharm của bên client sẽ hiện lên “client connet........” nhằm ám chỉ rằng đang kết nối qua server.



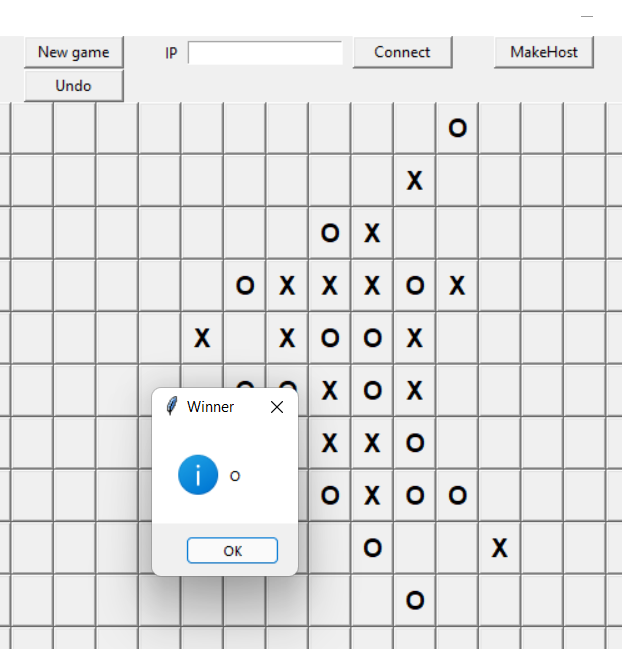
Hình .. Client đang kết nối

Lúc này đây bên phía server sau một hồi chậm trễ do kết nối sẽ kết nối thành công và sẽ lưu lại những kết quả mà client và server đều đánh.



Hình .. Server kết nối thành công

Và đây là kết quả cuối cùng khi cùng một lúc cả 2 máy đều chơi được qua mạng LAN.



Hình .. Kết quả đạt được

# KẾT LUẬN

Đồ án cần vận dụng kiến thức về tkinter và theading\_socket là chính, cũng như là phải có kinh nghiệm trong việc sử dụng hai thư viện này, điều quan trọng nhất trong đồ án lần này đó là làm sao để cả hai máy đều có thể kết nối chung 1 luồng để đọc dữ liệu qua lại lẫn nhau mà không làm sai kết quả trả về. Kết quả đã cho ra đúng với mong đợi, tuy nhiên đôi lúc một máy không thể phản hồi được khi bắt đầu kết nối chung một server làm cho cả hai máy cùng một lúc không thể kết nối chung một luồng, chúng tôi cho rằng còn tùy vào mạng mạnh hay yếu mà có thể ảnh hưởng đến tốc độ phản hồi của cả hai máy hoặc cũng có thể là do độ trễ của hai máy. Tuy nhiên, nhìn tổng thể đồ án đã thực hiện tốt vai trò của nó.

**TÀI LIỆU THAM KHẢO**

1. Giới thiệu Python Tkinter <https://viettuts.vn/python-tkinter> (28/4/2022)
2. Python interface to Tcl/Tk. <https://docs.python.org/3/library/tkinter.html> (28/4/2022)
3. [Tkinter - Python] Bài 5: Button trong tkinter <https://www.youtube.com/watch?v=jLdHhqQopYw> (28/4/2022)
4. Lập trình socket bằng Python <https://viblo.asia/p/lap-trinh-socket-bang-python-jvEla084Zkw> (4/5/2022)
5. Socket Programing in Python (Guide) <https://realpython.com/python-sockets/> (5/5/2022)
6. Lập trình socket trong Python <https://codelearn.io/sharing/lap-trinh-socket-voi-tcpip-trong-python> (5/5/2022)

**PHỤ LỤC PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành viên** | **Nhiệm vụ** |
| 1 | Nguyễn Hà Duy Khang | Làm word, tìm hiểu đồ án, xây dựng thuật toán |
| 2 | Hoàng Minh Nhựt | Viết code, làm word, tìm hiểu đồ án, xây dựng thuật toán |
| 3 |  |  |