

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



## PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

---

Báo cáo môn Phát triển phần mềm mã nguồn mở

# Đề tài: SOCKET VÀ ỨNG DỤNG TRONG GAME CỜ CARO

---

GVHD: Từ Lăng Phiêu  
SV: Huỳnh Anh Quốc - 3121410410  
Hoàng Nhật Hào - 3121410180

TP. HỒ CHÍ MINH, THÁNG 5/2024

# Mục lục

<b>1</b>	<b>Phần mở đầu</b>	<b>2</b>
1.1	Lý do chọn đề tài . . . . .	2
1.2	Đối tượng nghiên cứu . . . . .	2
1.3	Phạm vi nghiên cứu . . . . .	2
1.4	Mục tiêu nghiên cứu . . . . .	2
1.5	Phương pháp nghiên cứu . . . . .	2
<b>2</b>	<b>Phần nội dung</b>	<b>2</b>
2.1	Chương 1: LÝ THUYẾT . . . . .	2
2.1.1	Pygame và cách cài đặt . . . . .	2
2.1.1.a	Khái niệm: . . . . .	2
2.1.1.b	Cách cài đặt pygame trong Python . . . . .	3
2.1.2	Tổng quan về lập trình Socket . . . . .	3
2.1.3	Giới thiệu về trò caro . . . . .	5
2.2	Chương 2: THIẾT KẾ CHƯƠNG TRÌNH . . . . .	7
2.2.1	Tạo Server cho game cờ caro . . . . .	7
2.2.2	Tạo Client cho game cờ caro . . . . .	10
2.2.3	Tạo khung chat cho game cờ caro . . . . .	17
2.2.4	Tìm người chơi thắng . . . . .	18
2.2.5	Giao diện . . . . .	20
2.2.5.a	Giao diện trò chơi . . . . .	20
2.2.5.b	Giao diện client . . . . .	21
<b>3</b>	<b>Kết luận</b>	<b>21</b>
3.1	Ưu điểm . . . . .	21
3.2	Nhược điểm . . . . .	21



# 1 Phần mở đầu

## 1.1 Lý do chọn đề tài

Cờ caro là một trò chơi có từ lâu đời nay, hiện nay nó rất phổ biến trong giới học đường. Từ nhu cầu giải trí đó, đã có rất nhiều hãng đã sản xuất trò chơi caro trên máy tính, giúp cho việc chơi cờ caro được dễ dàng hơn, và bạn có thể đánh cờ với một người khác. Trên cùng mục đích muốn đem trò chơi cờ caro đến với mọi người, làm cho việc chơi cờ trở nên dễ dàng hơn, có thể hỗ trợ cho việc chơi với bạn bè, nhóm em đã chọn chương trình chơi cờ caro làm đề tài của mình.

## 1.2 Đối tượng nghiên cứu

Ngôn ngữ lập trình Python

Game cờ caro nói riêng và các loại game lập trình Python nói chung

## 1.3 Phạm vi nghiên cứu

Ngôn ngữ lập trình Python nói chung và pygame nói riêng

## 1.4 Mục tiêu nghiên cứu

Hiểu rõ hơn và Python, trau dồi kỹ năng làm game với thư viện pygame

Tìm hiểu về lập trình Socket

## 1.5 Phương pháp nghiên cứu

Thu nhập và phân tích những thông tin liên quan đến đề tài để xây dựng ứng dụng

Xác định các yêu cầu để thiết kế ứng dụng

# 2 Phần nội dung

## 2.1 Chương 1: LÝ THUYẾT

### 2.1.1 Pygame và cách cài đặt

#### 2.1.1.a Khái niệm:

Pygame có thể chạy trên nhiều nền tảng và hệ điều hành khác nhau. Với thư viện pygame trong Python, các nhà phát triển có thể sử dụng công cụ và chức năng mở rộng để tạo ra các trò chơi nhập vai ẩn tượng. Bởi vậy, Pygame đang ngày càng phổ biến với nhà phát triển vì tính đơn giản, linh hoạt, dễ sử dụng. **Pygame là một thư viện của ngôn ngữ lập trình Python và là một tập hợp các mô-đun Python được thiết kế riêng để lập trình trò chơi. Pygame được viết bởi Pete Shinnars thay thế cho chương trình PySDL sau khi quá trình phát triển dự án này bị đình trệ. Chính thức phát hành từ năm 2000, Pygame được phát hành theo phần mềm miễn phí GNU Lesser General Public License.**

Pygame có thể chạy trên nhiều nền tảng và hệ điều hành khác nhau. Với thư viện pygame trong Python, các nhà phát triển có thể sử dụng công cụ và chức năng

mở rộng để tạo ra các trò chơi nhập vai ấn tượng. Bởi vậy, Pygame đang ngày càng phổ biến với nhà phát triển vì tính đơn giản, linh hoạt, dễ sử dụng.

#### 2.1.1.b Cách cài đặt pygame trong Python

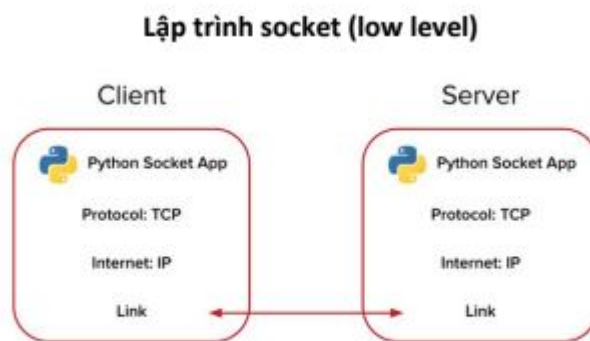
Để cài đặt thư viện Pygame trong Python bạn thực hiện theo các bước dưới đây:

**Bước 1:** Trước khi cài đặt thư viện Pygame trong Python, hãy đảm bảo rằng bạn đã cài Python trên máy tính của mình. Sau đó truy cập Python.org và tải xuống phiên bản tương thích với hệ điều hành của mình.

**Bước 2:** Chạy file “Python-version” trong terminal **Bước 3:** Để cài đặt lập trình Pygame, bạn cần sử dụng trình quản lý gói pip (thường đi kèm với các bản cài đặt Python). Chạy thiết bị đầu cuối và thực hiện lệnh “pip install pygame”. Lệnh sẽ tải xuống và cài đặt phiên bản Pygame mới nhất từ Python Package Index (PyPI)

**Bước 4:** Tạo một file mới bằng Python và nhập mô-đun Pygame bằng cách thêm dòng “Nhập pygame”, lưu tập có đuôi “.py” và bắt đầu trải nghiệm thư viện Pygame

#### 2.1.2 Tổng quan về lập trình Socket



Socket là một điểm cuối (endpoint) của một liên kết giao tiếp 2 chiều giữa 2 chương trình chạy trên hệ thống mạng. Một socket được gắn với một cổng (PORT) để tầng TCP có thể nhận diện ứng dụng nào đã gửi dữ liệu đến.

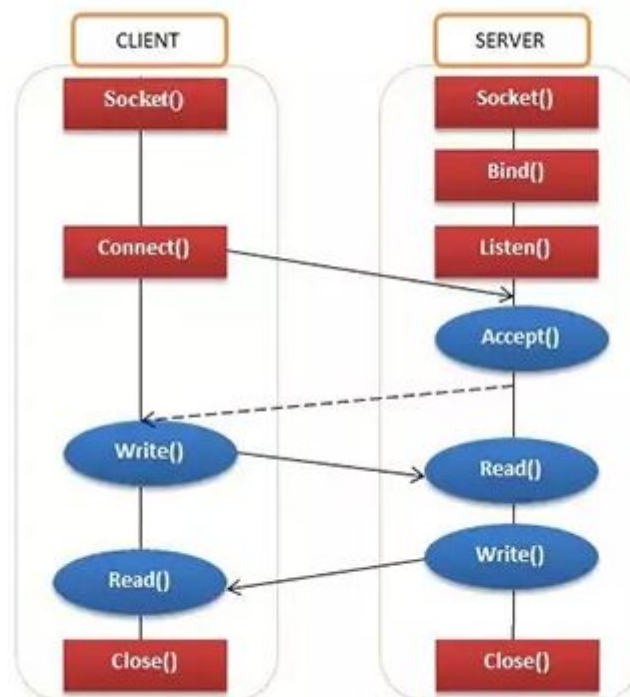
## •Cơ chế hoạt động của socket



1. Client sẽ mở một kết nối TCP và cố gắng kết nối với server qua một PORT quy định.



2. Nếu kết nối thành công, server chấp nhận kết nối nó sẽ mở ra một PORT và duy trì kết nối này



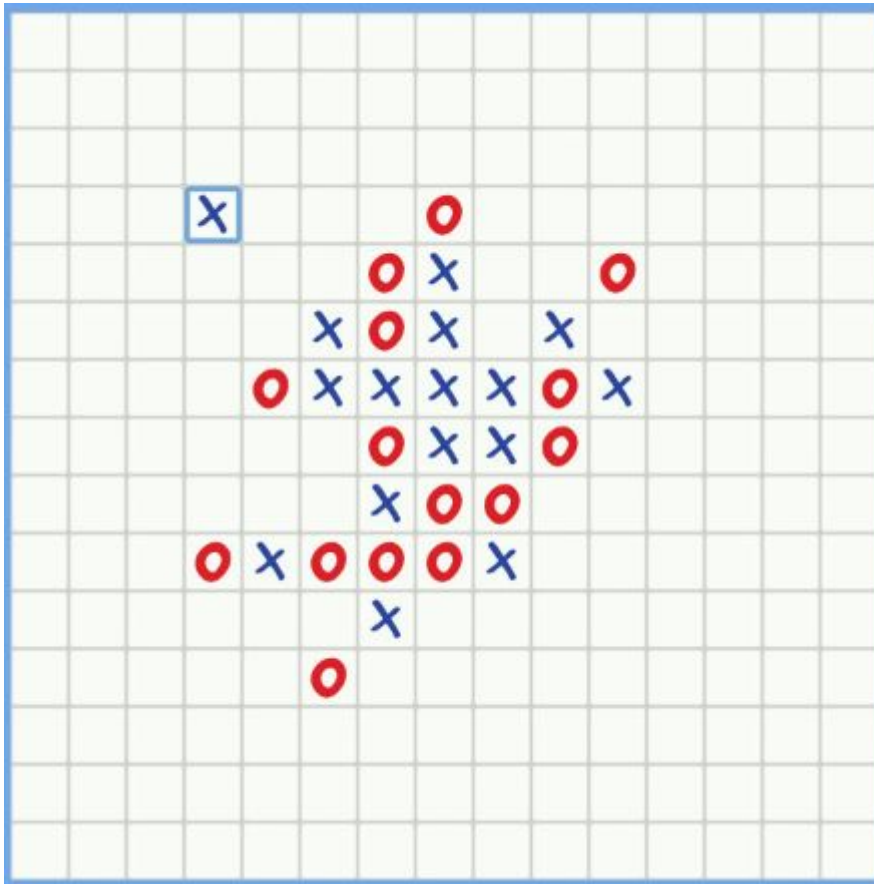
•Ví dụ: tạo ứng dụng gửi nhận dữ liệu client –server với socket

```
# chương trình server
# import thư viện
import socket
host = '127.0.0.1' # ip server
port=21451 #portserver
s = socket.socket() # tạo socket
s.bind((host, port)) # bind to ip and port
s.listen(1) # lắng nghe connect từ client
print('Đang lắng nghe client kết nối tới')
while 1:
    conn, addr = s.accept() # chấp nhận kết nối
    print ('Kết nối từ máy có địa chỉ: ', addr)
    # chuyển chuỗi thành mảng byte utf-8
    msg=bytes('Cảm ơn bạn đã kết nối tới','utf-8')
    # gửi về client
    conn.send(msg)
    # đóng kết nối
    conn.close()
```

```
# chương trình client
# import thư viện
import socket
host = '127.0.0.1' # ip server
port=21451 #portserver
s = socket.socket() # tạo socket
s.connect((host, port)) # connect tới server
# nhận dữ liệu từ server và chuyển thành chuỗi
data=s.recv(1024).decode('utf-8')
# in dữ liệu
print (data)
# đóng socket
s.close()
```

### 2.1.3 Giới thiệu về trò caro

Cờ caro là một loại trò chơi khá phổ biến vì luật chơi đơn giản và không đòi hỏi nhiều về công cụ chơi, bạn chỉ cần một tờ giấy có kẻ ô và bút là đủ. Chính vì vậy trò chơi này lại càng phổ biến trong giới học sinh sinh viên. Cờ caro là trò chơi đối kháng 2 người: Trò chơi đối kháng (twoagent, conflicting game) : Gồm 2 người chơi, đối thủ này sẽ tìm cách đánh chiến thắng trước đối thủ kia trong một số hữu hạn nước đi, mỗi nước đi được tạo ra dựa từ 1 trạng thái bất kỳ của trận đấu. Nếu sau 1 số giới hạn nước đi, nếu chưa ai đánh chiến thắng thì xem như hoà. Ngoài ra, thông tin về trận đấu là hoàn toàn biết được (perfect information) đối với cả 2 đối thủ.



Hình ảnh minh họa

## 2.2 Chương 2: THIẾT KẾ CHƯƠNG TRÌNH

### 2.2.1 Tạo Server cho game cờ caro

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

server = "192.168.26.4"
port = 5552

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)))
```



```
reply = ""
while True:
    try:
        data = conn.recv(4096).decode()

        if gameId in games:
            game = games[gameId]

            if not data:
                break
            else:
                if data == "reset":
                    game.resetWent()
                elif data != "get":
                    if data.startswith("chat|"):
                        # Trích xuất tin nhắn chat
                        chat_message = data.split("|")[1].strip()
                        game.chat(chat_message)
                    elif data.startswith("name|"):
                        # Trích xuất tin nhắn chat
                        player_name = data.split("|")[1].strip()
                        game.name(p, player_name)

                    else:
                        game.play(p, data)
```

```
        conn.sendall(pickle.dumps(game))
    else:
        break
except:
    break

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

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)
```

### 2.2.2 Tạo Client cho game cờ caro

```
import pygame
from network import Network
import random
from pygame.locals import Rect
import pickle
pygame.font.init()
pygame.init()
# Thiết lập một số hằng số
WIDTH_MAIN = 900
WIDTH, HEIGHT = 600, 600

# Tạo màn hình
screen = pygame.display.set_mode((WIDTH_MAIN, HEIGHT))

line_1 = 3
markers = []
pos = []
winner = 0
playerv = 1
clicked = False
green = (0, 255, 0)
gray = (117, 135, 167)
red = (255, 0, 0)
gray_light = (200, 200, 200)
blue = (0, 0, 255)
```

```
font = pygame.font.SysFont(None, 40)
again_rect = Rect(WIDTH + 30, HEIGHT - 115, 140, 50)
font_max = pygame.font.SysFont(None, 40)
font_pl = pygame.font.SysFont(None, 35)
font_pl1 = pygame.font.SysFont(None, 26)
fontx = pygame.font.SysFont(None, 25)
text_part = ""
input_text = ""

class Chat:
    def __init__(self, text):
        self.text = text

chat_box = Chat("")
Name = Chat("")

def draw_grid(screen, game):
    bg = (255, 255, 255)
    grid = (128, 128, 128)
    screen.fill(bg)
    for x in range(1, 10):
        pygame.draw.line(screen, grid, (0, x * 60), (WIDTH, x * 60), line_1)
        pygame.draw.line(screen, grid, (x * 60, 0), (x * 60, HEIGHT), line_1)
    for cell in game.find_winning_cells():
        x, y = cell
        pygame.draw.rect(screen, blue, (x * 60, y * 60, 60, 60), 3)
```

```
pygame.draw.line(screen, (128,128,128), (600, 0), (600, 600), 2)
pygame.draw.line(screen, (128,128,128), (0, 600), (800, 600), 2)
def redrawWindow(win, game, p, WIDTH, HEIGHT):
    bg = (255, 255, 255)
    win.fill(bg)

    if not(game.connected()):
        original_image = pygame.image.load("Images/loading.png")
        background_image = pygame.transform.scale(original_image, (900, 600))
        win.blit(background_image, (0, 0))
        font = pygame.font.SysFont(None, 35)
        text = font.render("Waiting Other Player...", True, (255,255,255))
        win.blit(text, (320,380))
    else:
        draw_grid(win, game)
        font = pygame.font.SysFont("comicsans", 60)
        font1 = pygame.font.SysFont("comicsans", 35)

        win_img = font_max.render("CARO", 1, (0, 0, 255))
        win.blit(win_img, (600 + 120, 600 - 570))
        win_x = font_pl1.render("10x10", 1, (0, 0, 255))
        win.blit(win_x, (670+60, 600 - 540))
        win_xpl = font_pl.render("Player turn:", True, blue)
        screen.blit(win_xpl, (WIDTH + 90, HEIGHT - 430))

        text1 = font1.render("", 1, (0,0,0))
```

```
text2 = font1.render("", 1, (0, 0, 0))
font2 = pygame.font.SysFont("comicsans", 19)

if game.p1Went and p == 0:
    text1 = font_pl1.render("Turn player 2", 1, (0, 0, 0))
elif game.p1Went:
    text1 = font_pl1.render("Your turn", 1, (0, 0, 0))

if game.p2Went and p == 1:
    text2 = font_pl1.render("Turn player 2", 1, (0, 0, 0))
elif game.p2Went:
    text2 = font_pl1.render("Your turn", 1, (0, 0, 0))

# original_image6 = pygame.image.load("Images/boxchat.png")
# background_image3 = pygame.transform.scale(original_image6, (280, 250))
# win.blit(background_image3, (610, 320))
pygame.draw.rect(screen, (0, 0, 0), (610, 320, 280, 250), 1)
original_image1 = pygame.image.load("Images/chat.png")
background_image2 = pygame.transform.scale(original_image1, (260, 50))
win.blit(background_image2, (620, 504))

fontchat = pygame.font.SysFont("comicsans", 15)
text_surface1 = fontchat.render(chat_box.text, True, (0, 0, 0))
screen.blit(text_surface1, (680, 518))
```

```
text_surface2 = fontchat.render(game.input_text, True, (0, 0, 0))
screen.blit(text_surface2, (640, 410))
text_surface3 = fontchat.render(game.text_part, True, (0, 0, 0))
screen.blit(text_surface3, (640, 370))

textp1 = font_pl1.render("You:", 1, (0, 0, 255))
win.blit(textp1, (WIDTH + 130, HEIGHT - 500))
textp1 = font_pl1.render("Opponent name: ", 1, (0, 0, 255))
win.blit(textp1, (WIDTH + 60, HEIGHT - 460))

if p == 1:
    win.blit(text2, (600 + 105, 600 - 400))
    win.blit(text1, (600 + 110, 600 - 400))
    textp1 = font_pl1.render("O", 1, (255, 0, 0))
    win.blit(textp1, (WIDTH + 167, HEIGHT - 500))
    textp1x = font_pl1.render(game.namep1, 1, (0, 0, 0))
    win.blit(textp1x, (WIDTH + 207, HEIGHT - 460))

else:
    win.blit(text1, (600 + 105, 600 - 400))
    win.blit(text2, (600 + 110, 600 - 400))
    textp1 = font_pl1.render("X", 1, (0, 255, 0))
    win.blit(textp1, (WIDTH + 167, HEIGHT - 498))
```



```
textpix = font_pl1.render(game.namep2, 1, (0, 0, 0))
win.blit(textpix, (WIDTH + 207, HEIGHT - 460))

x_pos = 0
for x in range(len(game.markers)):
    y_pos = 0
    for y in range(len(game.markers[x])):
        if game.markers[x][y] == 1:
            if (x, y) == game.last_selected:
                pygame.draw.rect(screen, (253, 222, 167), (x_pos * 60 + 2, y_pos * 60 + 2, 57, 57))
                pygame.draw.line(screen, green, (x_pos * 60 + 10, y_pos * 60 + 10),
                                   (x_pos * 60 + 45, y_pos * 60 + 45), 3)
                pygame.draw.line(screen, green, (x_pos * 60 + 10, y_pos * 60 + 45),
                                   (x_pos * 60 + 45, y_pos * 60 + 10), 3)
            else:
                pygame.draw.line(screen, green, (x_pos * 60 + 10, y_pos * 60 + 10),
                                   (x_pos * 60 + 45, y_pos * 60 + 45), line_1)
                pygame.draw.line(screen, green, (x_pos * 60 + 10, y_pos * 60 + 45),
                                   (x_pos * 60 + 45, y_pos * 60 + 10), line_1)
        if game.markers[x][y] == -1:
            if (x, y) == game.last_selected:
                pygame.draw.rect(screen, (253, 222, 167), (x_pos * 60 + 2, y_pos * 60 + 2, 57, 57))
                pygame.draw.circle(screen, red, (x_pos * 60 + 30, y_pos * 60 + 30), 24, 3)
            else:
                pygame.draw.circle(screen, red, (x_pos * 60 + 30, y_pos * 60 + 30), 24, line_1)
```



```
pygame.display.update()
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        run = False
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_BACKSPACE:
            Name.text = Name.text[:-1]
        elif event.key == pygame.K_RETURN:
            text = Name.text
            if len(Name.text) > 0:
                # n.send("name|" + text)
                run = False

        else:

            if event.unicode:
                if len(Name.text + event.unicode) <= 14:
                    Name.text += event.unicode
            # if event.type == pygame.MOUSEBUTTONDOWN:
            #     run = False
    main(screen)
while True:
    menu_screen()
```

### 2.2.3 Tạo khung chat cho game cờ caro

```
import tkinter as tk

def send_message():
    message = input_text.get()
    print("User sent message:", message)
    input_text.delete(0, tk.END)

root = tk.Tk()
root.title("Chat GUI")

# Tạo khung hiển thị tin nhắn
message_frame = tk.Frame(root)
message_frame.pack(pady=10)

message_label = tk.Label(message_frame, text="Enter message:")
message_label.pack(side=tk.LEFT)

input_text = tk.Entry(message_frame, width=50)
input_text.pack(side=tk.LEFT)

send_button = tk.Button(message_frame, text="Send", command=send_message)
send_button.pack(side=tk.LEFT)

root.mainloop()
```



Activa

#### 2.2.4 Tìm người chơi thắng

Kiểm tra xem ai đi được 5 ô trước

```
# Kiểm tra người chiến thắng
def check_winner2(self):
    # Kiểm tra hàng và cột
    for row in self.markers:
        for i in range(len(row) - 4):
            if sum(row[i:i + 5]) == 5:
                self.winner = 1
                self.game_over = True
            elif sum(row[i:i + 5]) == -5:
                self.winner = 2
                self.game_over = True

    for col in range(len(self.markers[0])):
        for i in range(len(self.markers) - 4):
            if sum(self.markers[row][col] for row in range(i, i + 5)) == 5:
                self.winner = 1
                self.game_over = True
            elif sum(self.markers[row][col] for row in range(i, i + 5)) == -5:
                self.winner = 2
                self.game_over = True

    # Kiểm tra đường chéo chính
    for i in range(len(self.markers) - 4):
        if sum(self.markers[i + j][i + j] for j in range(5)) == 5:
            self.winner = 1
            self.game_over = True
```

```
elif sum(self.markers[i + j][i + j] for j in range(5)) == -5:
    self.winner = 2
    self.game_over = True

# Kiểm tra đường chéo phụ
for i in range(len(self.markers) - 4):
    if sum(self.markers[i + j][len(self.markers) - 1 - i - j] for j in range(5)) == 5:
        self.winner = 1
        self.game_over = True
    elif sum(self.markers[i + j][len(self.markers) - 1 - i - j] for j in range(5)) == -5:
        self.winner = 2
        self.game_over = True

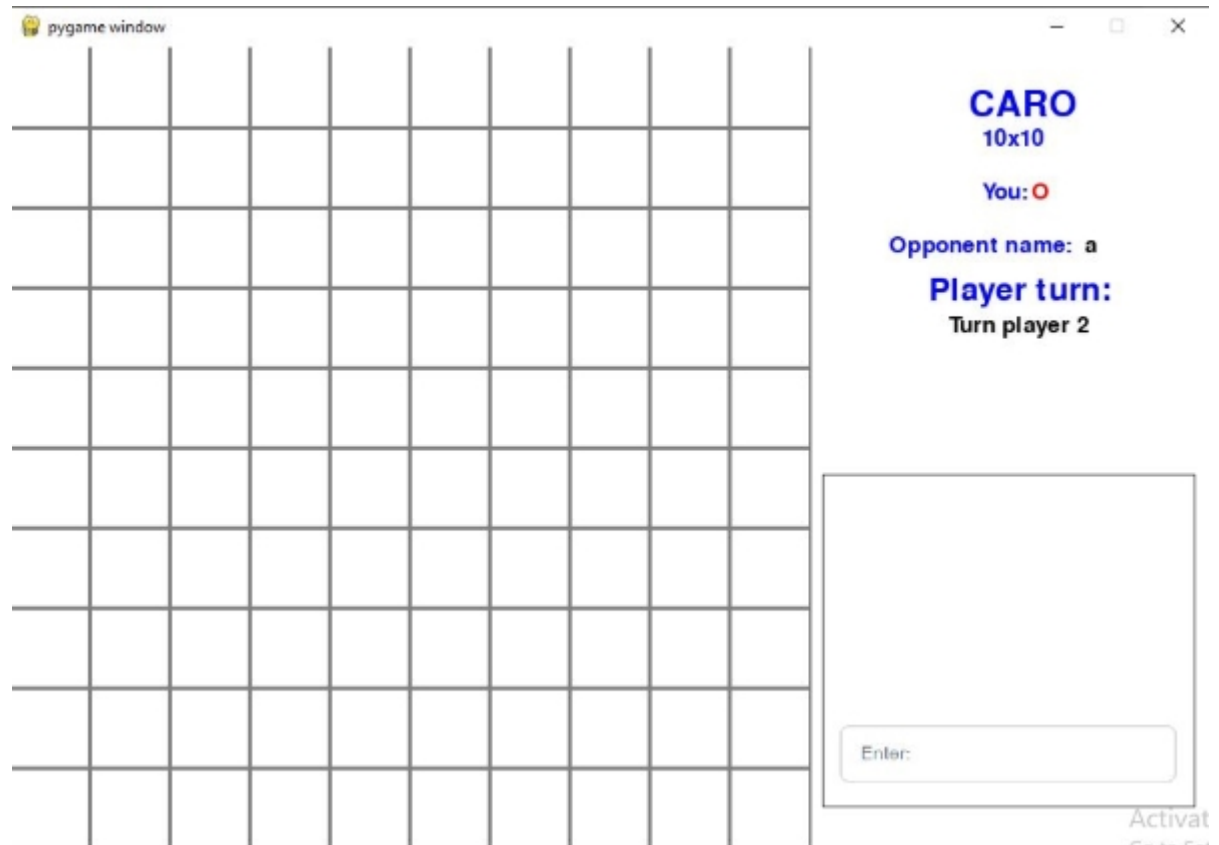
# Kiểm tra các đường chéo còn lại
for i in range(len(self.markers) - 4):
    for j in range(len(self.markers) - 4):
        if sum(self.markers[i + k][j + k] for k in range(5)) == 5:
            self.winner = 1
            self.game_over = True
        elif sum(self.markers[i + k][j + k] for k in range(5)) == -5:
            self.winner = 2
            self.game_over = True

    if sum(self.markers[i + k][len(self.markers) - 1 - j - k] for k in range(5)) == 5:
        self.winner = 1
        self.game_over = True
```



## 2.2.5 Giao diện

### 2.2.5.a Giao diện trò chơi



### 2.2.5.b Giao diện client



## 3 Kết luận

### 3.1 Ưu điểm

Sử dụng Socket cho phép người chơi tương tác trực tiếp với nhau thông qua mạng, tạo ra trải nghiệm chơi game chân thực và hấp dẫn hơn.

Với việc sử dụng Socket, game Caro có thể hỗ trợ nhiều người chơi cùng tham gia vào trò chơi, tạo ra một môi trường trò chơi đa người chơi trực tuyến.

Viết game Caro use Socket là một cách tốt để rèn luyện kỹ năng lập trình mạng trong Python, bao gồm thiết lập kết nối, truyền dữ liệu và xử lý gói tin.

Socket cho phép tạo ra các tính năng tương tác như trò chuyện trực tiếp giữa người chơi, bổ sung thêm tính năng phong phú và hấp dẫn cho trò chơi.

### 3.2 Nhược điểm

Do ảnh hưởng của mạng internet và tình trạng mạng của người chơi, có thể xảy ra lỗi mạng hoặc đứt trong quá trình truyền dữ liệu, ảnh hưởng đến trải nghiệm trò chơi.

Việc truyền dữ liệu qua mạng sử dụng Socket có thể gặp vấn đề liên quan đến bảo



mật. Dữ liệu có thể được đánh cắp hoặc can thiệp nếu không áp dụng các biện pháp bảo mật phù hợp.