

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐỀ TÀI: AI CHƠI CỜ CARO

Môn: Nhập môn trí tuệ nhân tạo

GVHD: TS. Trần Thế Hùng

Nhóm thực hiện: Nhóm 4

Nguyễn Văn Minh	20215092
Hoàng Mạnh Kiên	20215068
Nguyễn Trung Đạt	20215029
Nguyễn Hồ Tấn Tài	20215134
Hoàng Trường Nam	20215096

Hà Nội, tháng 6 năm 2024

I. Tên đề tài, danh sách thành viên

- 1. Tên đề tài : **AI chơi cờ caro**
 - 2. Danh sách thành viên, phân công công việc
- Chung: tìm hiểu và thảo luận cài đặt thuật toán, phân chia các thư viện

Họ và tên	Phân công nhiệm vụ	Phần trăm công việc
Nguyễn Văn Minh	Lập trình thuật toán minimax	25%
Hoàng Trường Nam	Điều chỉnh thuật toán cho luật chơi 4 nước. Làm báo cáo	22.5%
Nguyễn Trung Đạt	Thiết kế, lập trình giao diện bằng Turtle. Làm slide thuyết trình	22.5%
Hoàng Mạnh Kiên	Lập trình kiểm tra trạng thái thắng, thua. Lập trình cho bàn cờ 3x3	15%
Nguyễn Hồ Tấn Tài	Tìm hiểu và thiết kế thuật toán cắt tỉa anpla-beta	15%

II. Giới thiệu đề tài

1. Tổng quan đề tài

Chủ đề báo cáo của chúng em là trò chơi Cờ Caro được viết bằng Python, Cờ Caro là một trò chơi chiến thuật giữa hai người chơi một bàn cờ hình vuông. Chương trình AI cờ caro này không chỉ mang lại sự giải trí mà còn là một công cụ học tập hữu ích cho những ai quan tâm đến thuật toán và trí tuệ nhân tạo.

2. Game cờ Caro

+ Bàn cờ:

Bàn cờ caro thường là một hình vuông với kích thước 15x15 hoặc 19x19 ô vuông. Tuy nhiên, có thể chơi trên bàn cờ với kích thước tùy ý

+ Quân cờ:

Hai người chơi sử dụng hai loại quân cờ khác nhau, thường là X, O hoặc Trắng, Đen

+ Luật chơi:

Hai người chơi lần lượt đặt quân cờ của mình vào các ô trống trên bàn cờ. Người chơi đầu tiên thường là người đi quân X. Mục tiêu là tạo thành một đường thẳng (ngang, dọc, hoặc chéo) gồm 5 quân cờ liên tiếp của mình trước đối phương.

+ Kết thúc ván cờ

Thắng: Người chơi đầu tiên tạo được đường 5 quân cờ liên tiếp sẽ thắng.

Hòa: Nếu toàn bộ bàn cờ được lấp đầy mà không có người chơi nào tạo được đường 5 quân cờ, ván cờ sẽ kết thúc với kết quả hòa.

+ Các biến thể và quy tắc bổ sung:

Cờ caro 4: Trong một số biến thể, người chơi chỉ cần tạo thành một đường 4 quân cờ liên tiếp để thắng.

Cờ caro 3x3 (Tic tac toe): Người chơi cần tạo thành 3 nước cờ liên tiếp trên bàn cờ 3x3 để thắng

3. Mô tả thuật toán Minimax

+ Thuật toán Minimax là một thuật toán tìm kiếm và ra quyết định phổ biến trong các trò chơi hai người chơi, đặc biệt là các trò chơi có tổng bằng không như cờ vua, cờ vây, tic-tac-toe (cờ caro), v.v. Thuật toán này giúp xác định nước đi tối ưu bằng cách giả định rằng đối thủ sẽ luôn chơi nước đi tốt nhất có thể để tối ưu hóa cơ hội chiến thắng của mình. Dưới đây là một mô tả chi tiết về thuật toán Minimax:

+ Cơ bản về thuật toán Minimax:

- Mục tiêu: Tìm nước đi tối ưu bằng cách tối đa hóa giá trị nhỏ nhất có thể đạt được.

- Nguyên tắc hoạt động: Thuật toán xem xét tất cả các nước đi có thể của người chơi và đối thủ, xây dựng một cây tìm kiếm trong đó các nút lá đại diện cho các trạng thái cuối cùng của trò chơi (thắng, thua, hòa), và các nút trung gian đại diện cho các trạng thái trung gian của trò chơi.

III. Mô tả chương trình

1. Mô tả tổng quán

@Nguyễn Trung Đạt, @@@@

2. Thiết kế giải thuật

- Các hàm sử dụng trong caro 5

+ **forward(x, y, dx, dy, len)**

Mô tả: hàm tịnh tiến một điểm $A(x, y)$ qua một vector độ dài len, hướng (dx, dy)

Giá trị trả về là một tuple $(x2, y2) = (x, y) + length * (dx, dy)$. Nếu $(x2, y2)$ nằm ngoài đồ thị thì tịnh tiến ngược lại theo hướng $(-dx, -dy)$ để đưa trở lại đồ thị.

+ **get_score_sum(score_dict)**

Mô tả: đầu vào là một dictionary chứa các danh sách các nước đi 2 liên tiếp, 3 liên tiếp, 4 liên tiếp trên tất cả các hàng. Sau đó tính tổng lại bên X/ O có bao nhiêu nước 2 liên tiếp, 3 liên tiếp, 4 liên tiếp

+ **score_section(begin_point, dx, dy, end_point, bw)**

Trả về một list với mỗi phần tử đại diện cho số điểm của 5 ô liên tiếp từ begin_point đến end_point

+ **score_point(x, y, bw)**

Trả về điểm số của 4 đoạn thẳng xoay quan điểm (x, y) , đoạn thẳng theo hàng dọc, hàng ngang, hàng chéo thuận, hàng chéo nghịch

+ **win_situation(score_sum)**

Tìm kiếm các nước đi buộc phải chặn như người chơi có nước thắng luôn, có 4 nước liên tiếp trống 2 đầu. Hay có 2 nước 3 hoặc 1 nước 3, 1 nước 4 trống 2 đầu

+ **AI_calc_score(x, y)**

Giả sử AI tấn công ở vị trí (x, y) , sau đó tính số điểm lợi thế bên AI đạt được. Sau đó lại giả sử người chơi tấn công ở vị trí đó, nếu nước đó giúp người chơi có nhiều

điểm thì phải chặn

+ **get_expect_move()**

Từ những ô đã đánh, lấy toạ độ x_min -> x_max, y_min -> y_max của hình chữ nhật bao quanh các ô đã đánh. Sau đó mở rộng phạm vi hình chữ nhật rộng hơn 4 ô mỗi chiều. Từ hình chữ nhật thu được lấy ra m

+ **best_move()**

Trả lại ô có điểm số cao nhất trong những ô được đề xuất từ hàm get_expect_move()

. Các hàm sử dụng caro 4

- Các hàm sử dụng

Giải thuật khá tương đồng với code caro trên tuy nhiên có 1 vài khác biệt như

+ **get_score_sum(score_dict)**

Mô tả: đầu vào là một dictionary chứa các danh sách các nước đi 2 liên tiếp, 3 liên tiếp, trên tất cả các hàng. Sau đó tính tổng lại bên X/ O có bao nhiêu nước 2 liên tiếp, 3 liên tiếp

+ **AI_calc_score(x, y):** cố gắng di chuyển x,y trả về điểm số tượng trưng lợi thế

+ **get_expect_move():** Lấy những ô đã đánh, lấy toạ độ x_min -> x_max, y_min -> y_max của hình chữ nhật bao quanh các ô đã đánh, trả về các ô có thể đi trong phạm vi rộng hơn 4 ô mỗi chiều của hình chữ nhật đó, hàm này có 1 số tùy chỉnh về tấn công và phòng thủ

+ **TF34score(score2, score3):** hàm này trả về các tình huống mà chiến thắng chắc chắn xảy ra

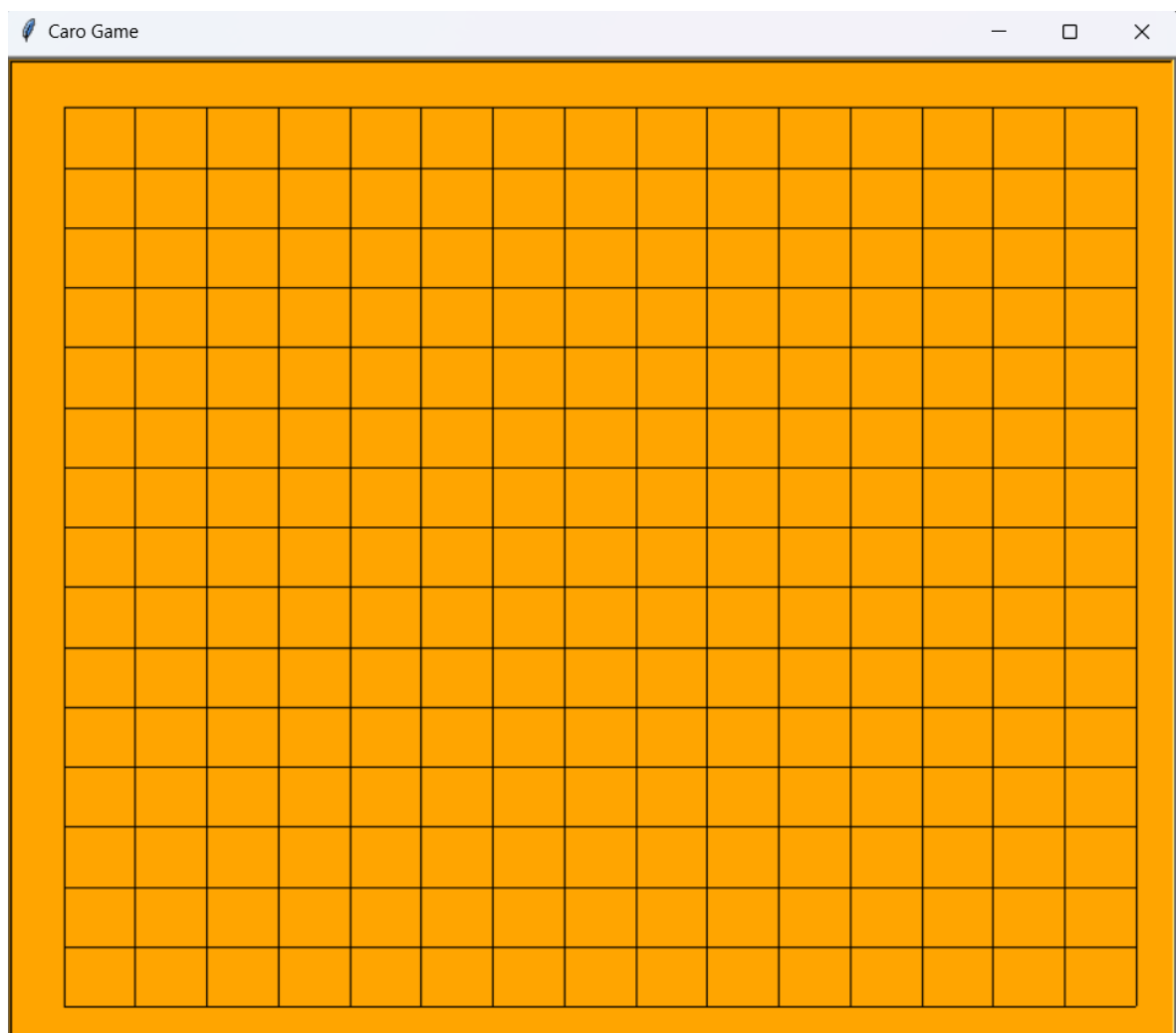
+ **win_situation(score_sum):** Tìm kiếm các nước đi buộc phải chặn như người

chơi có nước thắng luôn. Hàm này gọi hàm TF34 để biết được các tình huống chiến thắng

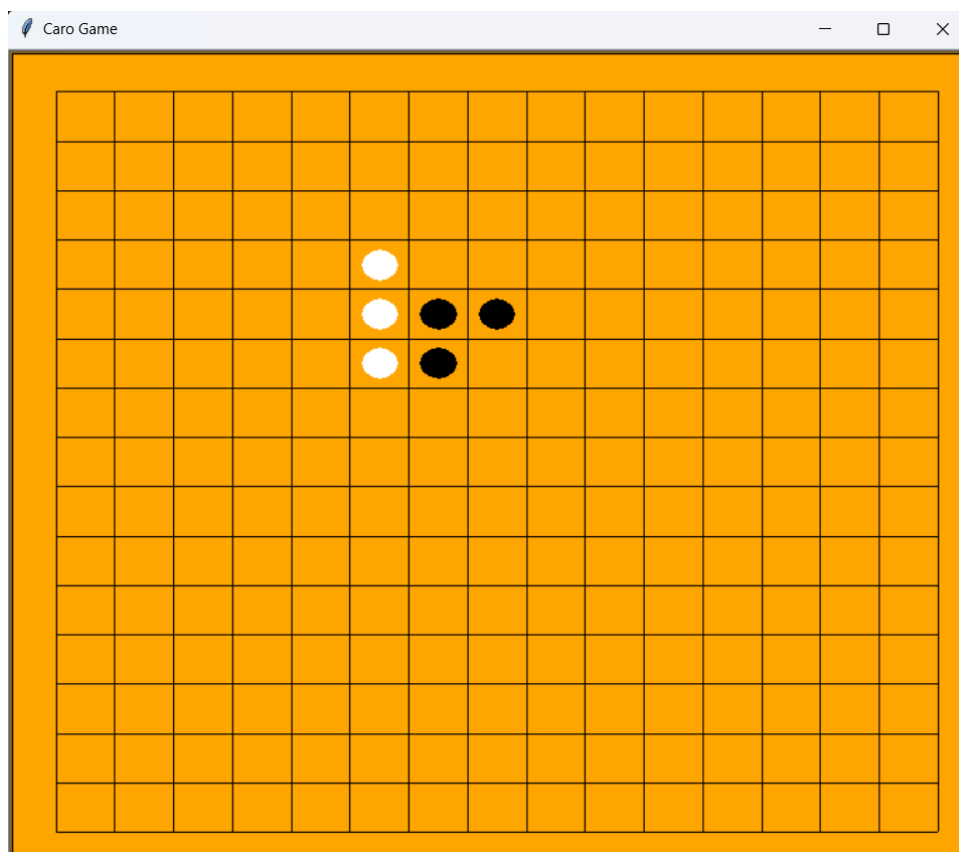
+ Biến win trong setting được đặt =4

- Interface

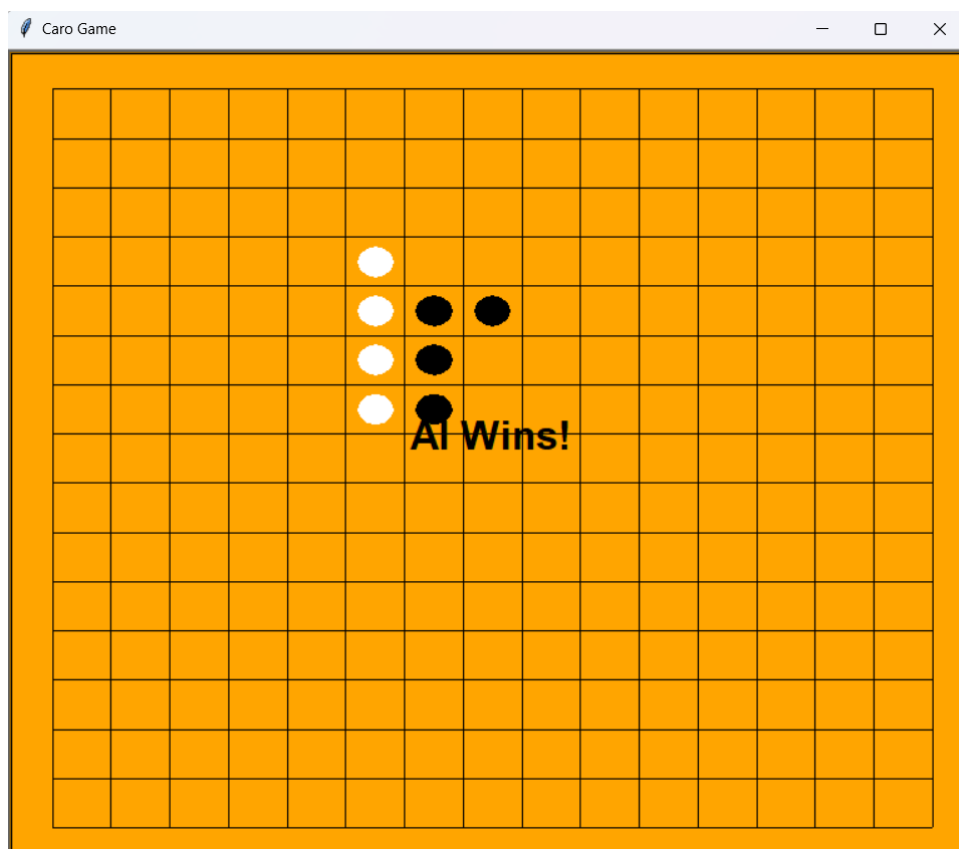
- file **Interface_Turtle.py** là file giao diện người dùng của trò chơi.
- Giao diện người chơi gồm có bảng(có thể thay đổi kích cỡ), hai ký hiệu cho 2 người chơi (đen và trắng)
- Giải thích các hàm
 - **display_winner(winner_text):** Hiển thị tên người chiến thắng lên màn hình
 - **click(x, y):** hàm xử lý click chuột:
 - **def draw_stone(x, y, bw):** hàm vẽ quân cờ của người chơi lên bàn cờ
 - **init():** Hàm khởi tạo trò chơi
- Giao diện:
 - Khi bắt đầu trò chơi:



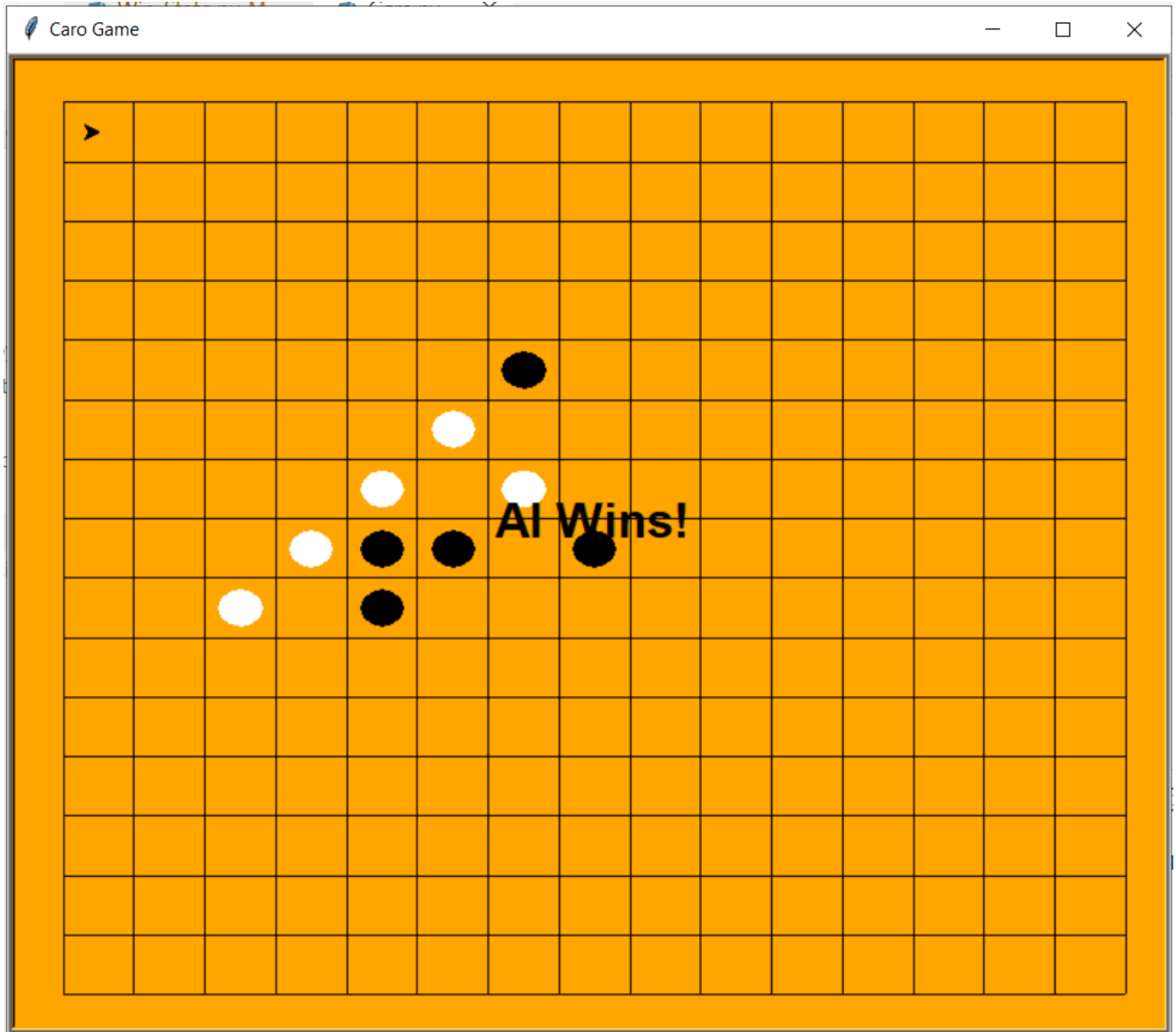
- Thực hiện các nước đi:

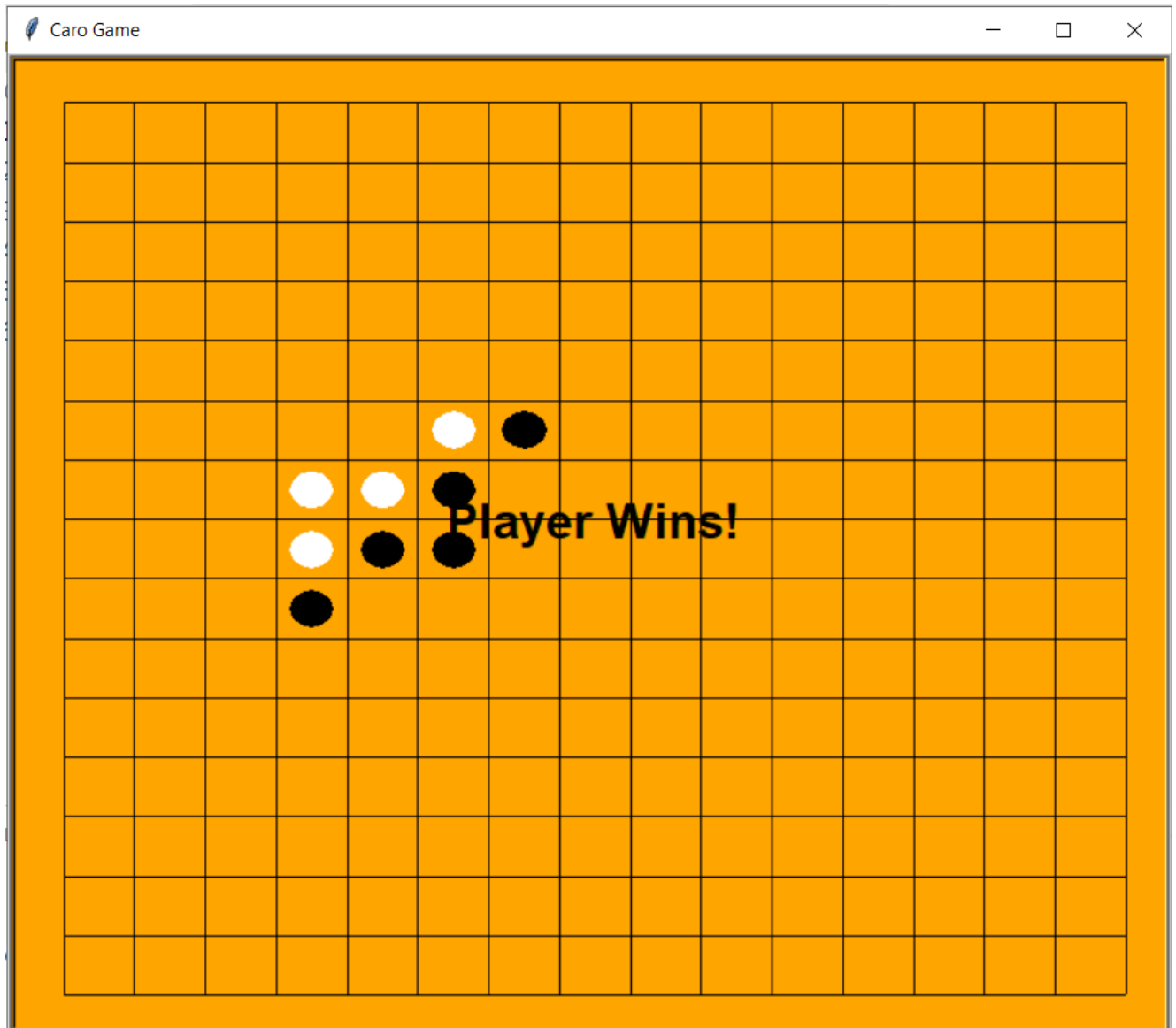


- Hiện thị người chiến thắng:



- Hình ảnh caro 4:





- Setting

- Setting.py là file lưu các giá trị cài đặt các tham số cho game như
 - Board size: kích cỡ bàn cờ
 - First hit: tham số chỉ người chơi nào đi trước
 - Player color: chỉnh màu cho người chơi

