

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----



Bài tập lớn môn học
CÔNG NGHỆ JAVA
Đề tài: Game Cờ Caro

Giảng viên hướng dẫn : Vũ Huấn

Nhóm : 11

Thành viên nhóm : Nguyễn Hoàng Anh - msv: 211201716
Nguyễn Công Hiếu - msv: 211212220

Lớp : CNTT5-K62

Hà Nội, tháng 5 năm 2023

MỤC LỤC

LỜI NÓI ĐẦU	3
I. GIỚI THIỆU TRÒ CHƠI CA-RO.....	4
II. PHÂN TÍCH GIẢI QUYẾT BÀI TOÁN	5
1. <i>Phân tích yêu cầu</i>	5
2. <i>Demo sản phẩm</i>	6
3. <i>Phương pháp giải quyết</i>	6
III. GIAO DIỆN HỆ THỐNG	12
1. <i>Giao diện khi chạy chương trình</i>	12
2. <i>Giao diện Bàn cờ</i>	13
3. <i>Giao diện Người chơi</i>	13
4. <i>Giao diện khi người chơi thắng</i>	13
IV. KẾ HOẠCH PHÂN CÔNG	15
TỔNG KẾT.....	15
Khó khăn và hướng giải quyết	16
Kết Luận.....	16
Hướng phát triển	16
TÀI LIỆU THAM KHẢO	17

LỜI NÓI ĐẦU

Trong thời đại 4.0 hiện nay thì game đóng vai trò quan trọng trong đời sống của mỗi người. Game giúp chúng ta thoải mái tương tác với bạn bè và nâng cao tư duy logic.

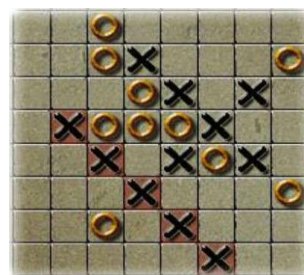
Hiện nay trò chơi dân gian đang bị mai một vào có thể bị lãng quên. Việc phát triển 1 tựa game dân gian là điều cần thiết giúp gìn giữ văn hóa dân tộc vào các trò chơi dân gian không bị lãng quên. Các trò chơi dân gian không gây nguy hiểm với con người. Thậm chí chúng có thể đóng góp vào quá trình phát triển các kỹ năng quan trọng như tư duy sáng tạo, sắp xếp kế hoạch và giải quyết các vấn đề hiệu quả. Chúng còn giúp ta thư giãn sau những giờ làm việc căng thẳng.

Lấy tên gọi là game cờ Caro là sở dĩ chúng được chơi trên một mặt phẳng có các sọc kẻ dọc và ngang tạo thành $N \times M$ ô vuông nhỏ. Nếu bạn từng nghe đến cờ vây thì game cờ Caro cũng có thể xem là một phiên bản khác của kiểu trò chơi này. Theo đó, quân trắng và quân đen sẽ lần lượt được thay thế bằng các ký hiệu X và O. Game cờ Caro gần như trở thành trò chơi quốc dân khi chúng vừa đòi hỏi sự tư duy, một chút khéo léo, tính ý dù luật chơi vô cùng đơn giản. Theo đó, game đánh cờ caro thường dành cho 2 người, mỗi người sở hữu một quân X hoặc O.

I. GIỚI THIỆU TRÒ CHƠI CA-RO

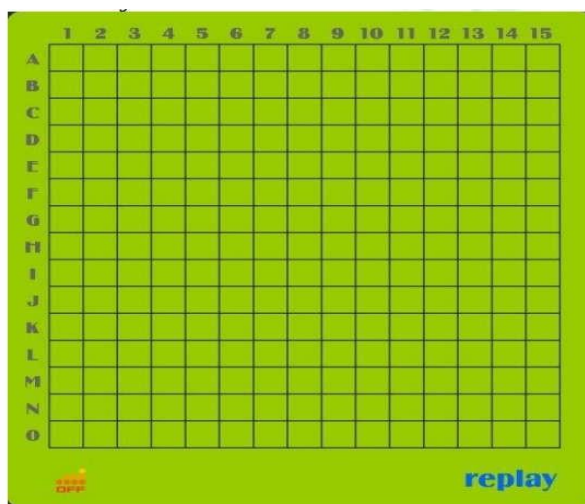
Cờ caro là một loại cờ cổ xưa của người Trung Quốc, một trong những trò chơi logic lâu đời nhất được biết đến trên thế giới.

Cờ caro được chơi trên toàn thế giới, ở mỗi nơi nó lại có tên gọi khác nhau: ở Nhật là Gomoku, ở Nga và các nước Đông Âu gọi là Five in a row, ở Hàn Quốc là Omok, ở Trung Quốc là Wuziqi, Anh là Connect5... và dĩ nhiên ở Việt Nam là Caro.



Game gồm có 2 người chơi, một người cầm quân X, một người cầm quân O. Hai người chơi sẽ lần lượt đánh quân tương ứng của mình trên mặt bàn cờ M x N ô (thường thì M = N)

Lượt chơi: Quân X là quân đánh trước. Hai người chơi có thể đánh vào bất kỳ ô nào trên bàn cờ, miễn là ô đó chưa được đánh. Trò chơi kết thúc khi người chơi có 5 quân thẳng hàng liên tiếp nhau (ngang, dọc hoặc chéo).



Bàn cờ 15 x 15

Khi các trình độ cờ thủ được nâng cao thì họ nhận ra rằng, nếu chỉ chơi đơn giản trong Gomoku thì đó là lợi thế quá lớn cho bên X (Quy định X đánh trước) vì thế một số cải tiến được đưa ra nhằm cân bằng ưu thế trên như:

- Gomoku. Hiện nay được chơi chính thức với bàn cờ 13x13. Không có hoà. Nếu hết đất thì O thắng. Chưa tìm được chứng minh nào cho thấy X chắc chắn thắng. Tuy nhiên X vẫn có ưu thế rất lớn.
- ProGomoku. Chơi trên bàn 15x15. Nước đầu của X đặt sẵn ô trung tâm. Nước thứ ba (nước thứ hai của X) phải đặt ngoài hình vuông cấm. Hình vuông cấm là hình vuông trung tâm kích thước 5x5. Không có hạn chế cho O. Đã có chứng minh X chắc chắn thắng trong biến thể này.
- Pente. Biến thể này không còn giống Gomoku. Luật bổ sung là có thể ăn quân đối phương. Nước ăn quân được thực hiện bằng cách chặn hai đầu một nước hai quân đối phương và ăn hai quân đó. AI tạo được nước năm hoặc ăn được 5 cặp quân trước thì thắng. Rất phổ biến ở Mỹ. Chơi trên bàn 19x19.

II. PHÂN TÍCH GIẢI QUYẾT BÀI TOÁN

1. Phân tích yêu cầu:

❖ *Mô phỏng bàn cờ*

Bàn cờ bao gồm các ô cờ được đặt trong một mảng hai chiều (kích thước $M \times N$). Trong mỗi ô cờ có thể xác định được:

- Vị trí ô cờ theo hàng, theo cột
- Trạng thái ô cờ: Đang trống (0), nước đi của người chơi 1 (1), nước đi của người chơi 2 (2)
- Độ nguy hiểm của ô cờ tùy theo trạng thái của ô cờ có thể thay đổi được

2. Demo sản phẩm:

- Khi chạy chương trình sẽ hiện 1 jframe gồm các ô xen kẽ nhau hình caro có màu xám và màu trắng ở dưới cùng sẽ có 1 BorderLayout hiển thị nút start , thời gian và người chơi 1,2
- Nếu không ấn nút Start sẽ không chơi được và khi click vào board sẽ không hiện thị
- Khi ấn nút Start sẽ có 1 box hiện ra hỏi “người chơi 1 đi trước đúng không?”. Nếu ấn Yes thì người chơi 1 sẽ là X và đánh trước ngược lại ấn No thì người chơi 2 sẽ là O và đánh trước

- Trong trò chơi khi ấn vào các ô sẽ có các tiếng “bụp” đồng thời thời gian cũng sẽ chạy và giá trị của ô đó là “X” hoặc “O” tùy thuộc vào lượt
- Khi người chơi hiện tại đủ 5 ô liên tiếp sẽ hiện Box thông báo người chơi đó thắng cuộc rồi ấn Ok sẽ tự động reset trò chơi
- Khi đang chơi mà muốn chơi lại thì sẽ ấn nút Stop ,cả bàn cờ sẽ được reset và hỏi lại xem ai là người đi trước

3. Phương pháp giải quyết

➤ Tạo các ô vuông nhỏ trong bàn cờ

- Để tạo được Board thì sẽ phải sử dụng 2 vòng lặp for lồng nhau
- x, y tương ứng với hàng và cột trong bàn cờ: x được tính bằng $j * w$, y được tính bằng $i * h$ ta được các khoảng ô
- Gọi 1 biến k tăng theo mỗi vòng lặp nếu k là chẵn thì sẽ ô đó sẽ là màu xám ngược lại là lẻ thì ô đó sẽ màu trắng

public void

```
paint(Graphics g) {
    super.paint(g);
    int w = getWidth()/N;
    int h = getHeight()/M;
    Graphics2D g2 =(Graphics2D)g;

    for (int i=0;i<N;i++) {
        int k=i;
        for (int j=0;j<M;j++) {
            int x=j*w;//0*h,1*h,2*h,...
            int y=i*h;//0*w,1*w,2*w,...

            Color color =k%2 == 0 ? Color.LIGHT_GRAY : Color.white;

            g2.setColor(color);
            g2.fillRect(x, y, w, h);

            k++;
        }
    }
```

```
}  
  
}
```

➤ Lấy tọa độ để xác định ô

- Sử dụng phương thức mousePressed của lớp MouseAdapter giao diện MouseListener

- Gọi các biến cXStart - cXEnd , cYStart – cYEnd để xác định vị trí ô rồi dùng điều kiện để kiểm tra nếu ô đó rỗng thì người chơi hiện tại sẽ đánh còn nếu ô đó có giá trị người chơi X đánh thì lượt tiếp theo sẽ là lượt đánh của người chơi O

```
addMouseListener( new MouseAdapter() {  
  
    @Override  
  
    public void mousePressed(MouseEvent e) {  
  
        super.mousePressed(e);  
  
        int x = e.getX();  
  
        int y = e.getY();  
  
        if(currentPlayer.equals(Cell.EMPTY_VALUE)) {  
  
            return;  
  
            for(int i = 0; i < N; i++) {  
  
                for(int j = 0; j < M; j++) {  
  
                    Cell cell = matrix[i][j];  
  
                    int cXStart = cell.getX();  
  
                    int cYStart = cell.getY();  
  
                    int cXEnd = cell.getWidth() + cXStart;  
  
                    int cYEnd = cell.getHeight() + cYStart;  
  
                    if(x >= cXStart && x <= cXEnd && y >= cYStart && y <= cYEnd) {  
  
                        if(cell.getValue().equals(Cell.EMPTY_VALUE)) {  
  
                            cell.setValue(currentPlayer);  
  
                            repaint();  
  

```

```

    int result = checkWin(currentPlayer,i,j);

    if(eG != null) {

        eG.end(currentPlayer, result);

    }

    if( result == NORMAL) {

        currentPlayer = currentPlayer.equals(Cell.X_VALUE) ? Cell.O_VALUE: Cell.X_VALUE;

    }

}

}

}

}

```

➤ Kiểm tra khi thắng bằng hàm checkWin

- **Kiểm tra từng chiều ngang, dọc:** Sử dụng biến cout để đếm trong vòng lặp for nếu giá trị của người chơi hiện tại cùng giá trị của ô thì sẽ tăng biến cout, nếu cout = 5 thì sẽ đưa ra giá trị WIN ngược lại sẽ reset biến cout = 0.

//Chiều ngang

```

int count = 0;

for(int col = 0; col < M; col++){

    Cell cell = matrix[i][col];

    if (cell.getValue().equals(currentPlayer)) {

        count++;

        if(count == 5){

            System.out.println("Ngang");

            return WIN;

        }

    }

}

}

```



```

        count = 0;

    }

}

//Chiều dọc

count = 0;

for(int row = 0; row < N; row++){

    Cell cell = matrix[row][j];

    if (cell.getValue().equals(currentPlayer)) {

        count++;

        if(count == 5){

            System.out.println("Doc");

            return WIN;

        }

    }else {

        count = 0;

    }

}

```

- Kiểm tra đường chéo trái:

- + Đầu tiên tìm min của i và j.
- + Gọi biến TopI và TopJ để xác định hàng đầu tiên và cột đầu tiên của đường chéo bằng công thức $TopI = i - min$, $TopJ = j - min$.
- + Sử dụng vòng lặp for để xác định ô của đường chéo đây TopI và TopJ tăng dần 1 đơn vị.
- + Xét biến cout tương tự hàng và cột.

```

//Chéo trái

int min = Math.min(i, j);

int TopI = i - min;

int TopJ = j - min;

```

```

count = 0;

for(;;TopI < N && TopJ < M; TopI++, TopJ++){

    Cell cell = matrix[TopI][TopJ];

    if (cell.getValue().equals(currentPlayer)) {

        count++;

        if(count == 5){

            System.out.println("Cheo trai");

            return WIN;

        }

    }else {

        count = 0;

    }

}

```

-Kiểm tra đường chéo phải:

- + Đầu tiên tìm min của i và j.
- + Gọi biến TopI và TopJ để xác định hàng đầu tiên và cột đầu tiên của đường chéo bằng công thức $TopI = I - \min$, $TopJ = J + \min$.
- + Nếu mà TopJ vượt quá cột M thì ta phải gọi thêm 1 biến dư và dư sẽ được tính theo công thức $dư = TopJ - (M-1)$. Và TopI mới bằng TopI cũ + dư , TopJ mới bằng M-1.
- + Sử dụng vòng lặp for để xác định ô của đường chéo đấy TopI tăng dần 1 đơn vị và TopJ giảm dần 1 đơn vị
- + Xét biến cout tương tự hàng và cột.

//Chéo phải

```

min = Math.min(i, j);

TopI = i - min;

TopJ = j + min;

count = 0;

if(TopJ >= M){

```

```

    int du = TopJ - (M - 1);

    TopI = TopI + du;

    TopJ = M - 1;

}

for(;TopI < N && TopJ >= 0; TopI++, TopJ--){

    Cell cell = matrix[TopI][TopJ];

    if (cell.getValue().equals(currentPlayer)) {

        count++;

        if(count == 5){

            System.out.println("Cheo phai");

            return WIN;

        }

    }else {

        count = 0;

    }

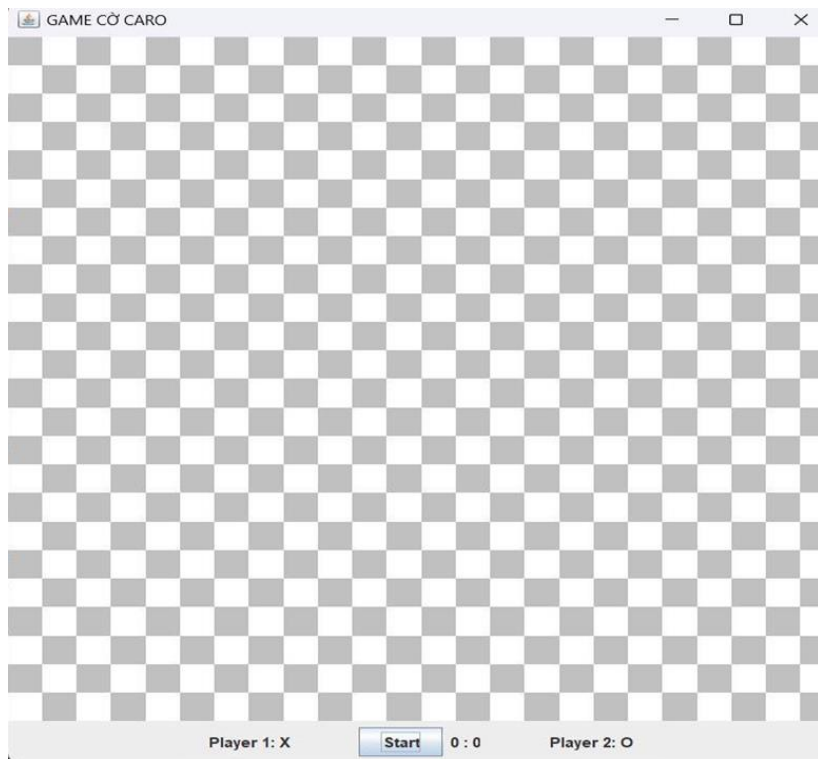
}

```

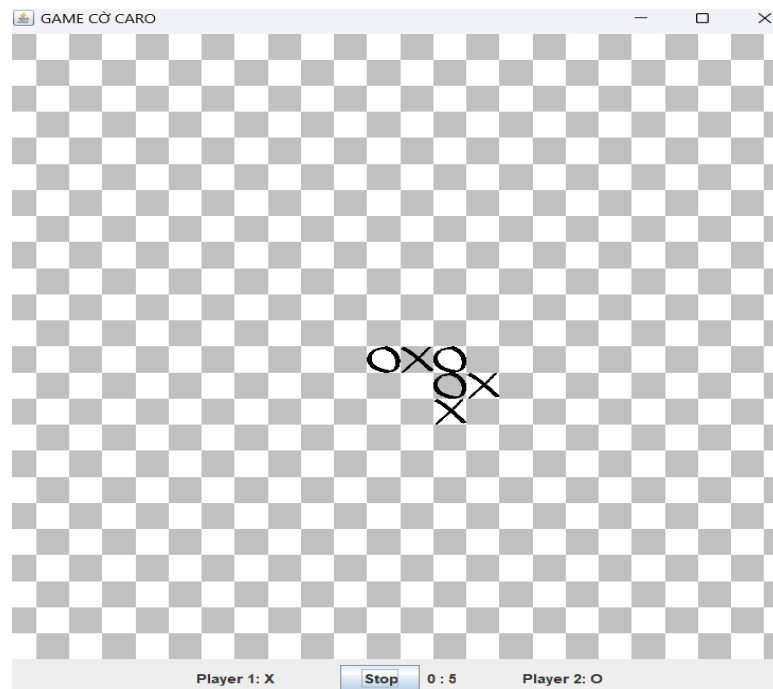
III. GIAO DIỆN HỆ THỐNG

1. *Giao diện khi chạy chương trình*

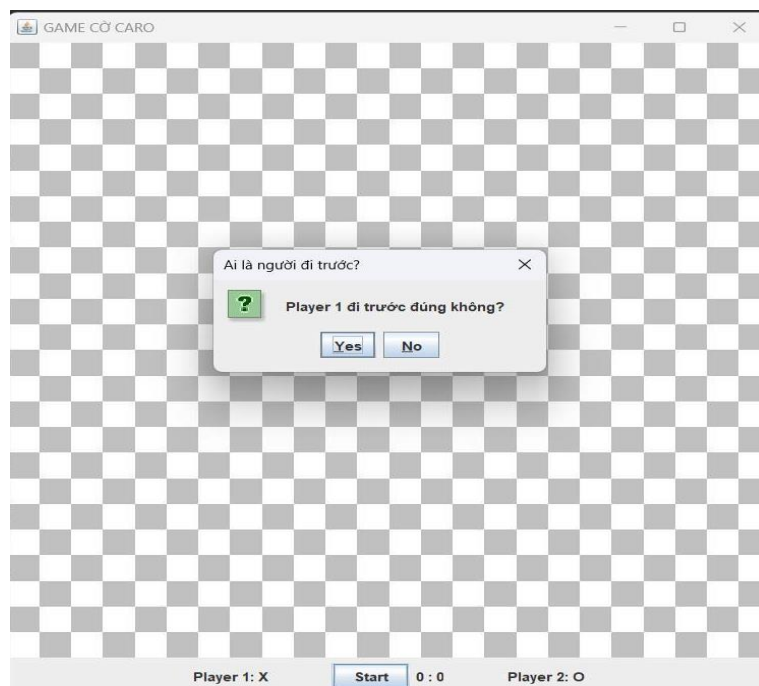
- Start: Chơi ván mới
- Player 1: Mặc định là X
- Player 2: Mặc định là O




2. Giao diện bàn cờ

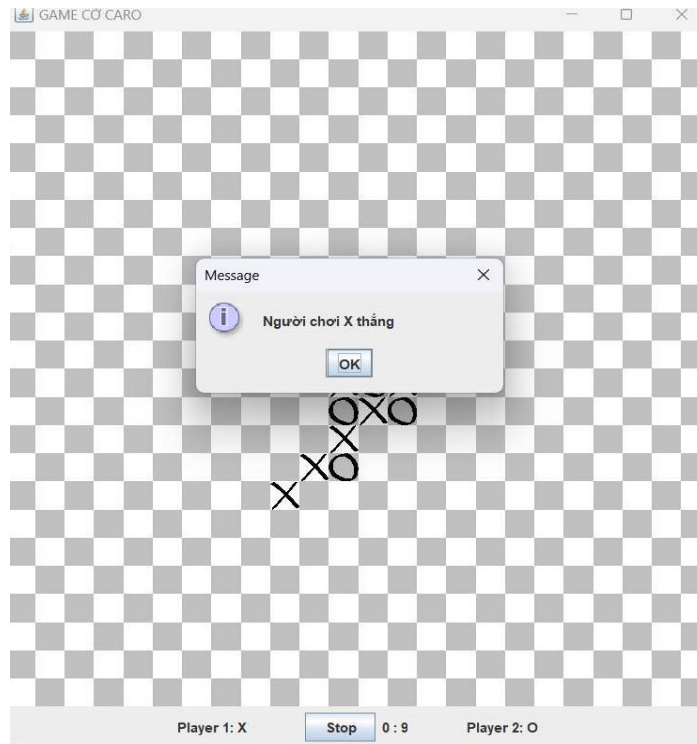


3. Giao diện Người chơi



 Được phép lựa chọn Player đi trước

4. Giao diện khi người chơi thắng



IV. KẾ HOẠCH PHÂN CÔNG

Nhiệm vụ của các thành viên trong nhóm như sau:

Nguyễn Hoàng Anh	Nguyễn Công Hiếu
<ul style="list-style-type: none">- Tìm kiếm đề tài- Phân tích cấu trúc chương trình- Thiết kế giao diện- Viết mã nguồn xây dựng chương trình- Thuyết trình- Viết báo cáo phân tích và hoàn thiện báo cáo	<ul style="list-style-type: none">- Tìm kiếm đề tài- Bổ sung giao diện- Kiểm tra lỗi chương trình- Đóng góp xây dựng chức năng chương trình

TỔNG KẾT

Khó khăn và hướng giải quyết

Khó khăn: Chưa hiểu nhiều về các thư viện , thời gian tiếp cận ngôn ngữ mới chưa được lâu dài

Hướng giải quyết: Đọc lại slide , xem thêm trên youtube và các trang web

Kết Luận

Qua quá trình thực hiện đề tài này chúng em đã hiểu hơn về ngôn ngữ java. Những điều thu được trong quá trình thực hiện:

- Biết thêm về thuật toán checkWin từng hàng, cột, đường chéo trái, phải chứ không cần xét cả ma trận
- Áp dụng được thuật toán trên vào game Caro
- Hiểu và xây dựng được game bằng ngôn ngữ java

Mặc dù có nhiều cố gắng nhưng chắc chắn không tránh khỏi thiếu sót và hạn chế trong quá trình cài đặt chương trình. Chúng em mong nhận được sự góp ý của thầy để có thể hoàn thiện và phát triển chương trình hơn.

Hướng phát triển

Xây dựng thêm một số tính năng

- + Giới hạn thời gian đánh cho người chơi
- + Gợi ý nước đi cho người chơi
- + Lưu, load ván cờ
- + Âm nhạc trong lúc đánh, lúc thắng, lúc hòa
- + Máy đánh với máy

Tài liệu tham khảo

- Internet
- Slide bài giảng
- Chat GPT