

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

-----o0o-----



**Bài tập lớn môn học**

# CÔNG NGHỆ JAVA

Đề tài :”Lập trình game Cờ CARO bằng ngôn ngữ java”

Giảng viên hướng dẫn: Vũ Huân

Nhóm thực hiện: **TàiLamQuỳnh** - Lớp CNTT5-K60

Danh sách sinh viên tham gia:

Mã SV	Họ tên
191201335	Nguyễn Trung Tài
191203978	Bùi Thị Thanh Lam
191203796	Nguyễn Thị Quỳnh

## LỜI NÓI ĐẦU

Công nghệ thông tin (CNTT) ngày càng có vai trò quan trọng trong cuộc sống hằng ngày của ta. Việc ứng dụng CNTT vào các lĩnh vực trong đời sống giúp công việc được tiến hành nhanh chóng và hiệu quả hơn.

Hiện nay, có rất nhiều công việc mới phát triển song song với sự phát triển của CNTT, một trong những số đó là hiệu ứng game. Lập trình game đang là một lĩnh vực mới thu hút nhiều lập trình viên. Do đó, em chọn đề tài “Lập trình game Cờ CARO bằng ngôn ngữ Java” nhằm tìm hiểu sâu hơn về ngôn ngữ java trong lập trình hướng đối tượng, từ đó viết một ứng dụng cụ thể thử nghiệm làm cơ sở củng cố kiến thức và định hướng, kế hoạch xây dựng những ứng dụng game cụ thể, phát triển theo hướng dịch vụ trong tương lai.

## MỤC LỤC

- I. Mục đích của bài tập lớn
- II. Giới thiệu bài tập lớn:
  - 1. Thông tin về game và luật chơi
  - 2. Thiết kế đối tượng
  - 3. Các hàm tiêu biểu liên quan tới logic
  - 4. Mô tả cơ sở dữ liệu
  - 5. Mô tả giao diện chương trình
  - 6. Các công nghệ và kiến thức được áp dụng
  - 7. Khó khăn gặp phải
  - 8. Một vài mở rộng có thể làm
- III. Kết luận

## **I. Mục đích của bài tập lớn :**

Java là một công nghệ xây dựng các ứng dụng phần mềm có vị trí rất lớn trong những năm cuối thế kỉ 20 đầu thế kỉ 21. Nó được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch 1 lần.

Sun, công ty đã phát minh ra ngôn ngữ java, chính thức ban hành bản Java Development Kit 1.0 vào năm 1996 đã hoàn toàn miễn phí để các nhà phát triển có thể tải về, học java, xây dựng các chương trình, ứng dụng và triển khai chúng trên các hệ điều hành hỗ trợ java. Khi nhắc đến Java người ta không chỉ nhắc đến java như một ngôn ngữ mà nhắc đến như một nền tảng phát triển...

Cũng như các trường đại học khác, Đại Học Giao Thông Vận Tải từ lâu đã đưa java vào chương trình giảng dạy cho sinh viên. Bài tập lớn này nhằm mục đích giúp sinh viên tìm hiểu và bước đầu làm quen với lập trình java, lập trình những ứng dụng đơn giản và có tính hữu dụng thực tế.

## **II. Giới thiệu về bài tập lớn :**

### *1. Thông tin về game và luật chơi:*

Cờ CARO là game rất phổ biến, đặc biệt là trong giới thanh thiếu niên, là trò chơi đối kháng giữa hai người hoặc giữa người và máy tính.

Ván cờ được chơi trên bàn cờ 15x15 giữa hai người và 19x19 giữa người và máy. Hai bên sẽ thay phiên nhau tích vào những ô vuông trên bàn cờ. Ký hiệu mỗi bước đi của từng người là **X** hoặc **O**.

Người chơi sẽ phải dùng chiến thuật và kinh nghiệm để tạo thành một hàng ngang, hàng dọc, hoặc chéo có 5 quân cờ liên tiếp của mình.

Người chiến thắng là người tạo được hàng 5 quân cờ đầu tiên.

Kết thúc ván chơi, kết quả thắng thua sẽ được hiển thị trên màn hình game.

Luật chơi CARO khá đơn giản nhưng không kém phần hấp dẫn. Để chiến thắng, bạn cần tạo ra những nước cờ hiểm và độc đáo. Điểm đặc biệt của CARO là bạn rất dễ bị thua nếu không để ý từng nước đi của đối phương mặc dù bạn có chiến thuật tốt hơn. Với giao diện game hấp dẫn, dễ sử dụng sẽ giúp bạn giải trí sau những giờ làm việc căng thẳng.

Chúc bạn chơi game vui vẻ!

## *2. Thiết kế đối tượng:*

- Game được thiết kế dựa trên mô hình MVC. Trong chương trình game sẽ được tạo nên từ các package sau:

- Controller: Nơi tiếp nhận yêu cầu từ Client (Client có thể là người dùng, hay từ ứng dụng, hệ thống khác) và xử lý các yêu cầu đó.

- Service: Là nơi xử lý logic nghiệp vụ của trò chơi yêu cầu và thường tương tác lấy, thêm, sửa data trong database.

- Model: Là nơi lưu trữ các cấu hình của trò chơi.

- View: Hiện dữ liệu và giao diện của trò chơi mà package model chứa nó.

- Utils: Lưu trữ các tiện ích như là thao tác kết nối với Database. Hay lấy các thông số của trò chơi từ file conf.

- Image: Lưu trữ tất cả các image trong game nhằm để quản lý tìm kiếm và sửa đổi trong quá trình code.

- Minimax: Chứa AI phục vụ cho nghiệp vụ người chơi với máy.

- Valid: để kiểm tra các điều kiện câu chuỗi có chuẩn form không.

### 3. Các hàm tiêu biểu liên quan tới logic:

- Class MainEvent: Xử lý khi người dùng lựa chọn các tác vụ trong menu
  - Nếu chọn PlayButton: Ứng dụng sẽ yêu cầu người dùng lần lượt nhập tên 2 người chơi (người nhập đầu tiên sẽ được ưu tiên đi trước) và đảm bảo tên người chơi không được để trống hay bị trùng nhau. Sau đó người chơi sẽ được đưa đến giao diện 2 người chơi với nhau và bắt đầu chơi.
  - Nếu chọn PlayWithComputerButton: Ứng dụng sẽ đưa người chơi đến giao diện của người chơi với máy và bắt đầu game với người được ưu tiên đi trước.
  - Nếu chọn HowToPlayButton: Người dùng sẽ được đưa đến giao diện hướng dẫn cách chơi và cách tính điểm.
  - Nếu chọn RankingButton: Người dùng sẽ được nhìn thành tích của 15 game thủ có thành tích tốt nhất từ trước tới hiện tại.
- Class PlayEvent: Xử lý quá trình chơi game của 2 người chơi
  - Hàm tạo: Để xử lý sự kiện của các ô trong bàn cờ và các nút trong game.
  - Hàm checkRow: Để kiểm tra khả năng chiến thắng của người chơi theo từng hàng. Nếu xuất hiện 5 quân cờ liên tiếp của người chơi trên 1 hàng thì trả về giá trị true.
  - Hàm checkCol: Để kiểm tra khả năng chiến thắng của người chơi theo từng cột. Nếu xuất hiện 5 quân cờ liên tiếp của người chơi trên 1 cột thì trả về giá trị true
  - Hàm checkMainCross: Kiểm tra khả năng chiến thắng của người chơi theo đường chéo chính. Nếu xuất hiện 5 quân cờ liên tiếp của người chơi trên 1 đường chéo chính thì trả về giá trị true. Class này sẽ gọi tới hàm NormalMinimax để khởi tạo ván game.

- Hàm checkSubCross: Kiểm tra khả năng chiến thắng của người chơi theo đường chéo phụ. Nếu xuất hiện 5 quân cờ liên tiếp của người chơi trên 1 đường chéo phụ thì trả về giá trị true.
- Class MinimaxEvent: Xử lý các thao tác giữa người và máy trong quá trình chơi. Sau khi kết thúc ván cờ có lựa chọn chơi ván mới tiếp với AI và cập nhật tỷ số của lượt đấu.
- Class NormalMinimax: Nối chứa các hàm ứng dụng thuật toán trong quá trình chơi với máy.

#### 4. Mô tả cơ sở dữ liệu:

- Thông tin của người chơi ta lưu vào bảng table.
- Bảng table gồm có 3 trường dữ liệu:
  - Ten: kiểu dữ liệu char và lưu tên người chơi.
  - Thang: kiểu số nguyên lưu trữ số trận thắng người chơi.
  - Thua: kiểu số nguyên lưu trữ số trận thua người chơi.
- Câu truy vấn tạo bảng.  
create table nguoichoi (  
 Ten varchar(255),  
 Thang int,  
 Thua int  
 )  
 - Câu truy vấn tạo kho dữ liệu mẫu.

```
insert into nguoichoi(Ten, Thang, Thua)
values ('g्रेgr',1, 0),
('tai',5, 1),
('879',1, 1),
('56465',1, 1),
('lam',5, 1),
('Lam',5, 2),
('tai lam',3, 0),
('Quynh',5, 10),
```

('Hao',5, 0),  
 ('harry',5, 8),  
 ('messi',5, 8),  
 ('ronaldo',5, 10),  
 ('anh',2, 0),  
 ('italia',2, 1),  
 ('teayeon',10, 8)

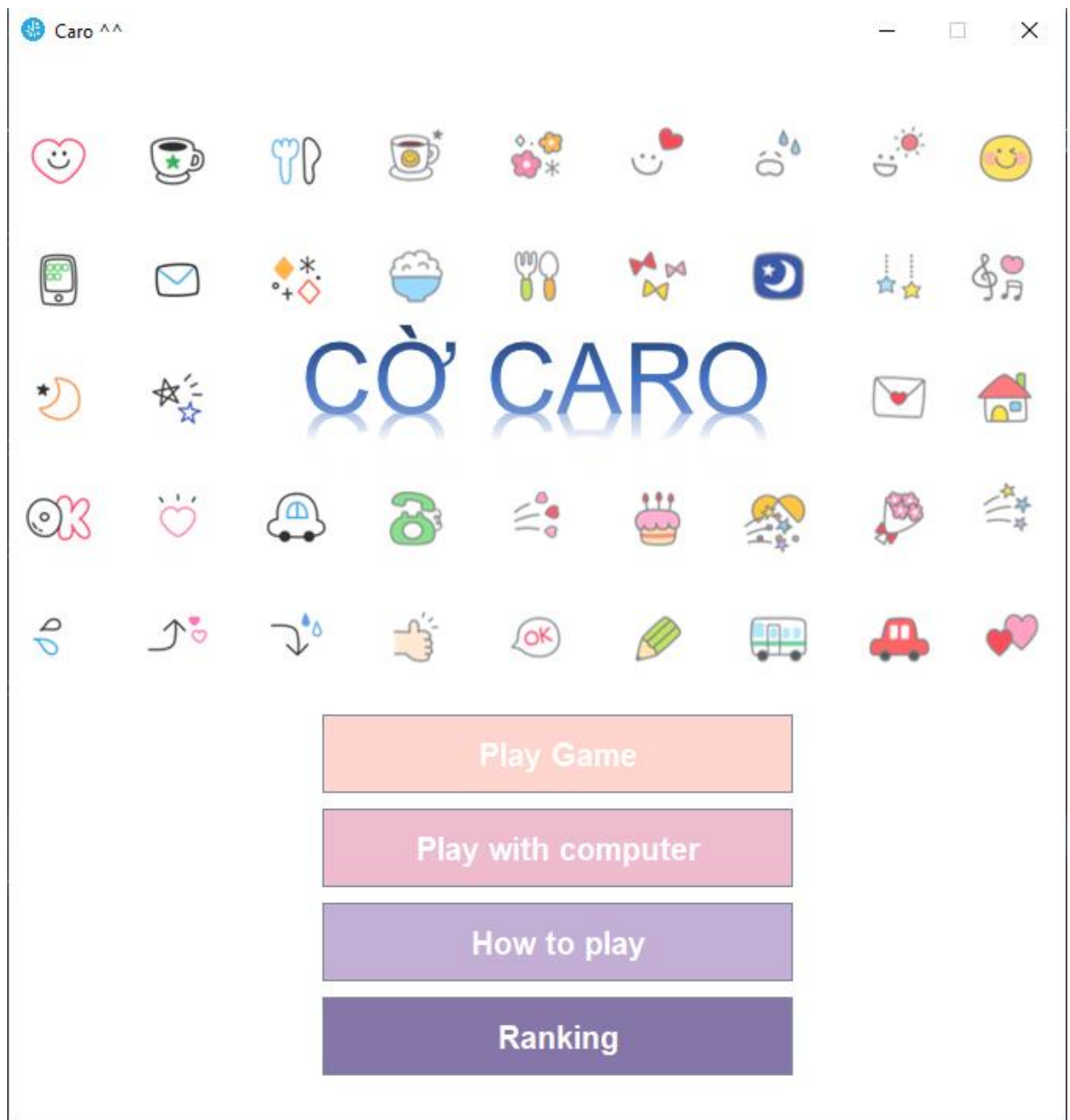
	Ten	Thang	Thua
▶	g nagr	1	0
	tai	5	1
	vdff	3	0
	879	1	1
	56465	1	1
	lam	5	1
	Lam	5	2
	tai lam	3	0
	Quynh	5	10
	Hao	5	5
	harry	5	8
	teayeon	10	8
	messi	5	8
	ronaldo	5	10
	anh	2	0
	italia	2	1

Hình 1: Mẫu dữ liệu

## 5. Mô tả giao diện chương trình:

### 5.1 Giao diện chính



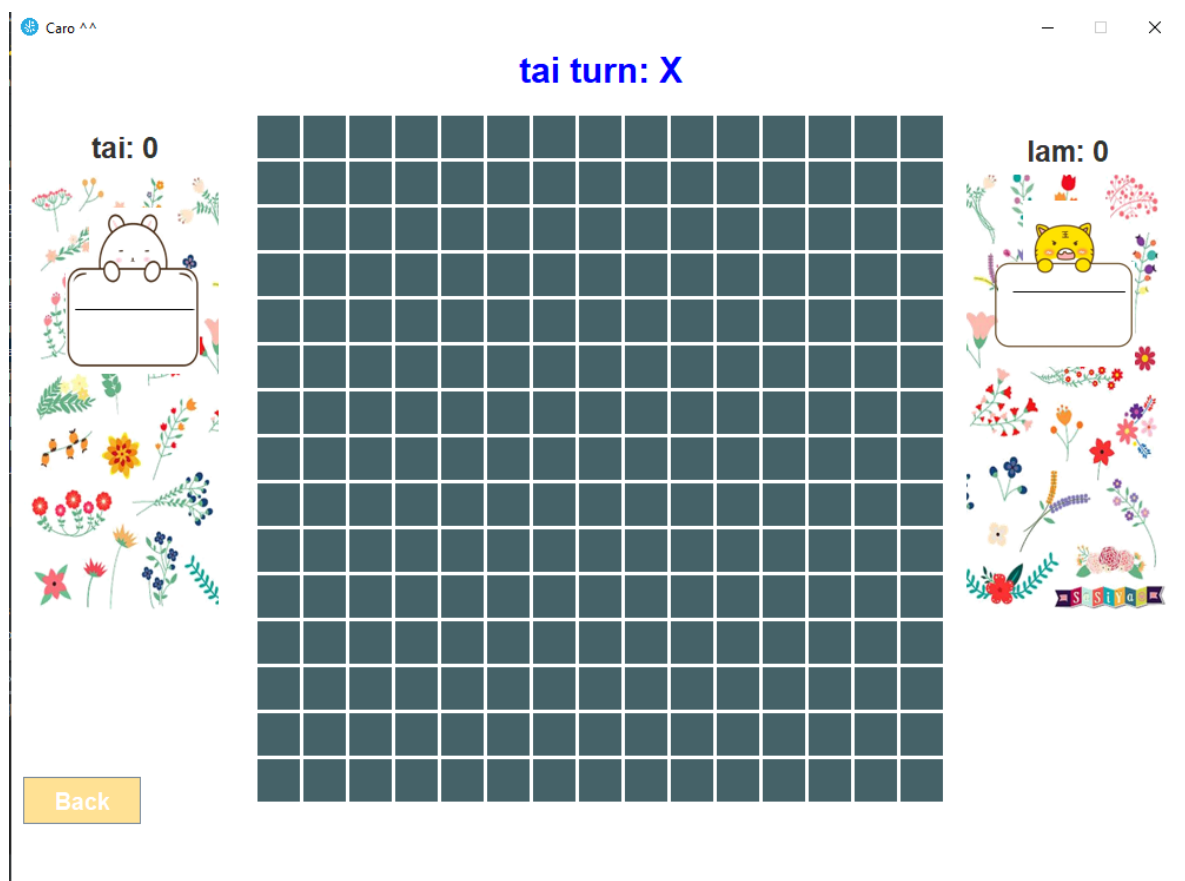


Hình 2: Giao diện chính game

- Giao diện menu trò chơi để người dùng có thể lựa chọn các chế độ chơi, xem thành tích của mình và có thể xem thông tin hướng dẫn về trò chơi.
- Lớp MainFrame: Có chứa một hàm tạo ở đó bao bao gồm hình ảnh background, được add thêm các button gồm PlayGame, PlayWithComputer, HowToPlay và Ranking.
- Nhiệm vụ các button:
  - PlayGame: Yêu cầu người dùng đăng nhập tên 2 người để vào trò chơi.

- PlayWithComputer: Người dùng sẽ được chơi với máy.
  - HowToPlay: Người dùng được hướng dẫn cách chơi game.
  - Ranking: Người dùng sẽ được xem bảng xếp hạng của mỗi người chơi khi đấu với nhau.
- Cách làm: Thêm các button được tạo từ các class bên ngoài mỗi button đều được addActionListener sẽ được MainEvent quản lý. MainEvent sẽ xử lý các logic điều hướng các button.

### 5.2 Giao diện người chơi với người

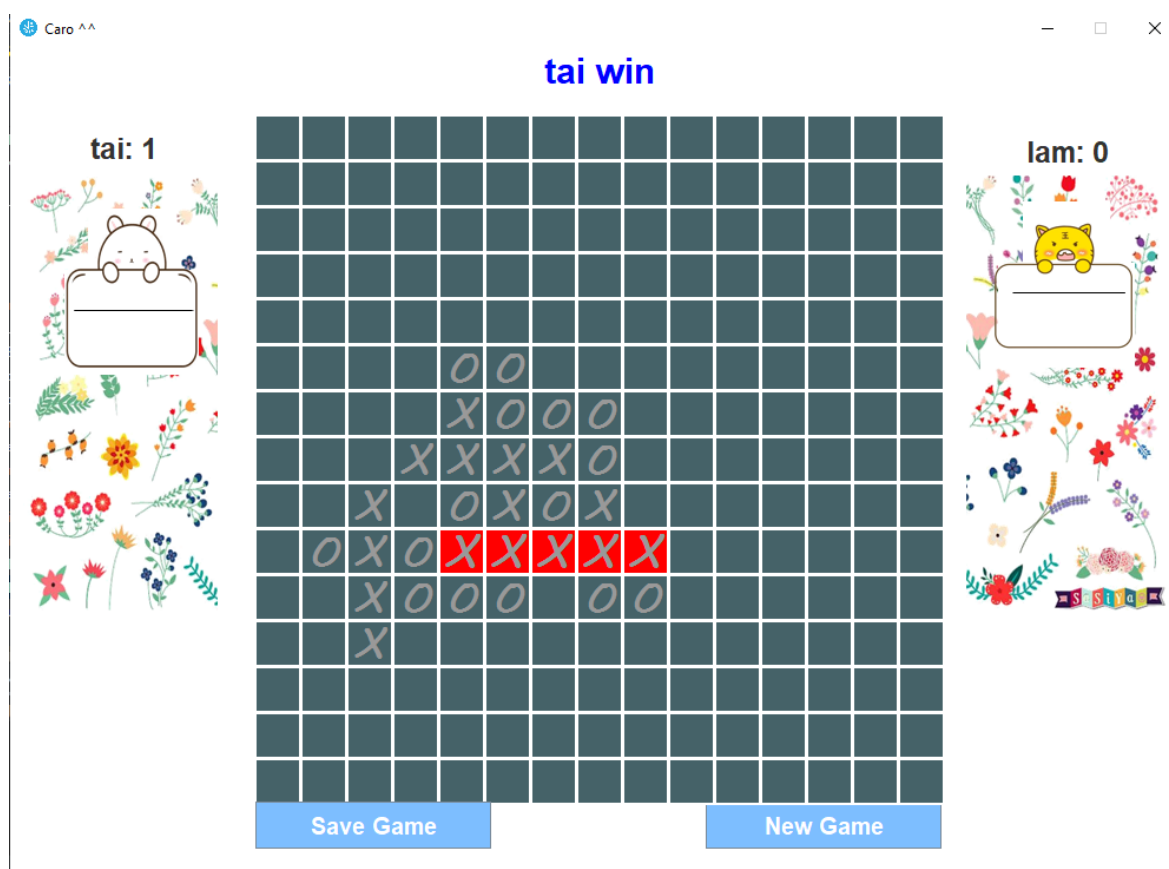


Hình 3: Giao diện game người chơi với người

- Giao diện người chơi gồm: Bàn cờ caro kích thước 15x15, có phần thông báo lượt của mỗi người chơi và thông báo kết quả người chiến thắng sau mỗi ván đấu. Hai bên là tên mỗi người chơi

nhập vào trước khi bắt đầu trò chơi và mỗi bên chứa tỷ số mỗi người chơi. Nút Back để quay lại màn hình chính của game.

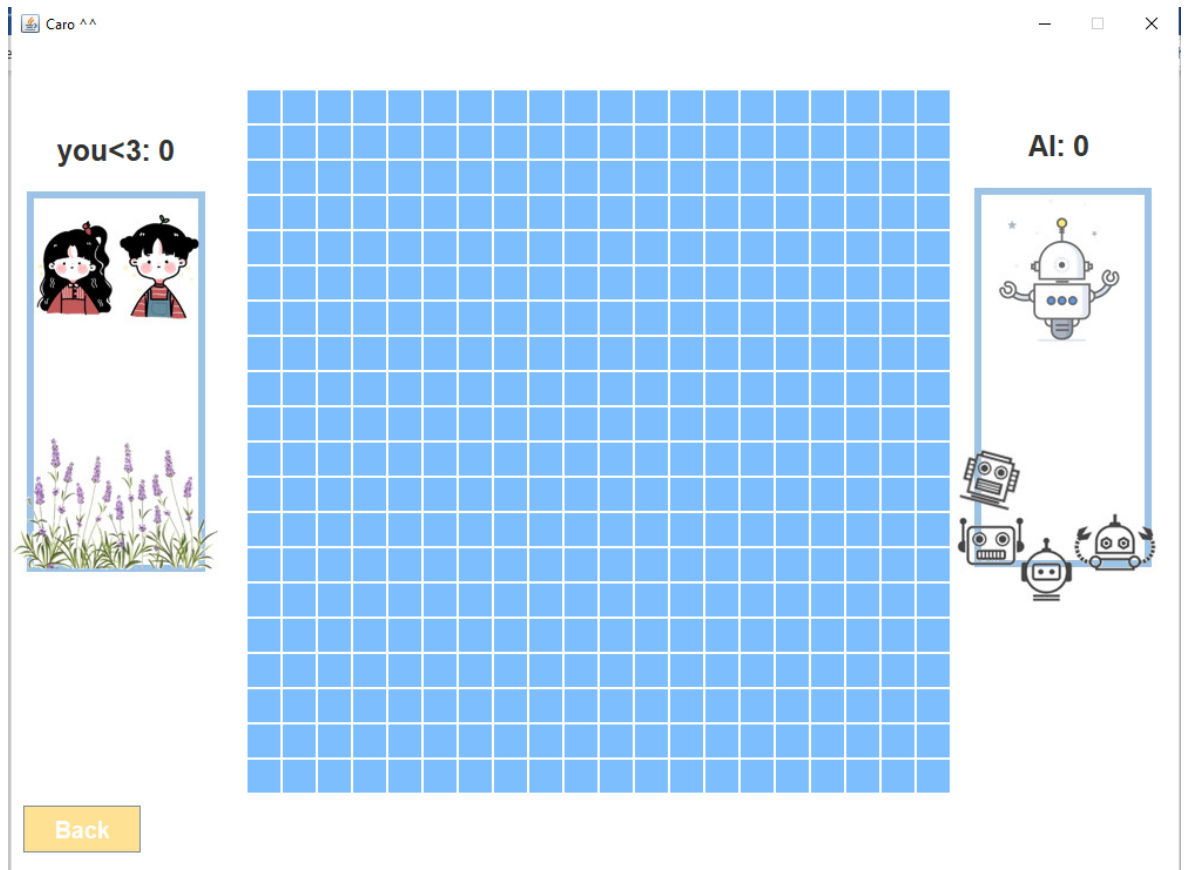
- Sau mỗi ván chơi sẽ hiện giao diện như bên dưới. Chọn new game người chơi sẽ tiếp tục ván chơi với và cập nhật tỉ số mỗi khi kết thúc ván. Chọn Save Game nếu người dùng muốn kết thúc lượt chơi của 2 người và sẽ được lưu lại ở database. Sau khi ấn Save Game người dùng sẽ được đưa về màn hình menu.



Hình 4: Giao diện 2 người chơi sau khi kết thúc 1 ván

- Cách làm: Tạo một Frame chứa chứa các nút tương tác là bảng trò chơi đã được tạo từ trước.

### 5.3 Giao diện chơi với máy



Hình 5: Giao diện người chơi với máy

- Giao diện phần người chơi với máy. Khá tương tự như giao diện người chơi với người. Gồm có bảng cờ caro và hai bên hiện tỷ số người với máy. Ô cờ cỡ 19x19
- Ở phần lần được cài đặt người luôn được ưu tiên đánh trước và chọn X.

#### *5.4 Giao diện hướng dẫn cách chơi*



Hình 6: Giao diện hướng dẫn chơi

- Giao diện hướng dẫn sơ lược cách chơi dùng để hướng dẫn người dùng sơ qua cách chơi của trò caro và hướng cách tính điểm xếp hạng của ứng dụng mà người dùng đang sử dụng.
- Lớp giao diện gồm:
  - Hàm tạo không đối: trong đó sẽ add button back, bảng hướng dẫn chơi game và 2 ảnh icon.
  - Xét event cho các nút.
- Cách làm: Xét tuyệt đối các thành phần và sử dụng `<html>` `<br>` để viết văn bản.

### 5.5 Giao diện bảng xếp hạng

TOP 15 HIỆU SỐ CAO NHẤT				
Id	Player name	Win	Lose	Different
1	lam	5	1	4
2	tai	4	1	3
3	vdff	3	0	3
4	Lam	5	2	3
5	tai lam	3	0	3
6	teayeon	10	8	2
7	anh	2	0	2
8	gregr	1	0	1
9	italia	2	1	1
10	879	1	1	0
11	56465	1	1	0
12	Hao	5	5	0
13	harry	5	8	-3
14	messi	5	8	-3
15	Quynh	5	10	-5

Back

Hình 7: Giao diện bảng xếp hạng người chơi

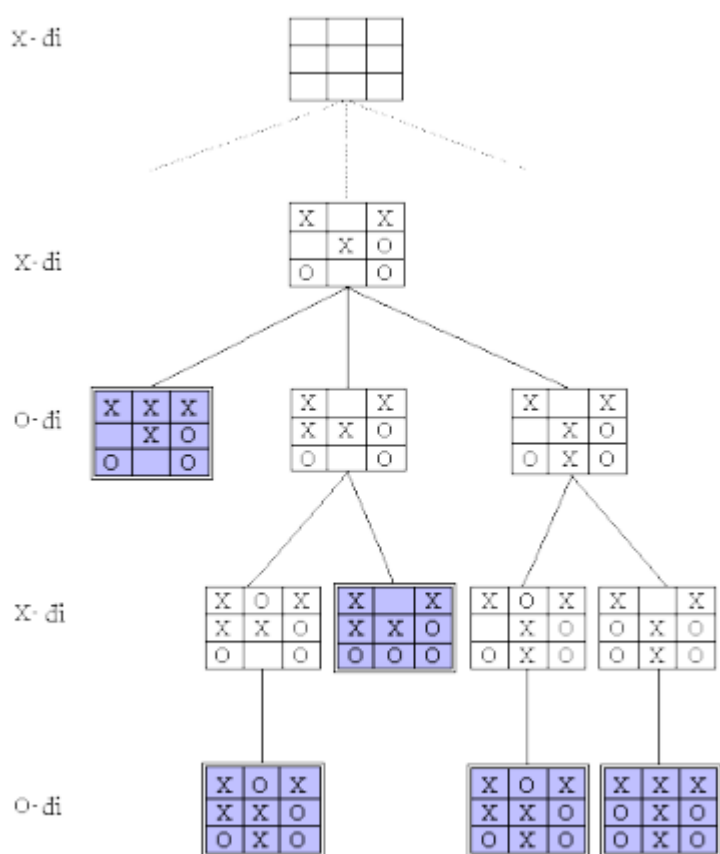
- Bảng xếp hạng top 15 những người đã từng tham gia chơi game có thứ hạng cao nhất.
- Thông tin được bảng được lấy từ database sau mỗi lượt thi đấu của mỗi cặp đấu.
- Cách làm: tạo một class kế thừa từ Jtable lưu trữ thông tin của tất cả những người chơi sau khi đọc được từ databse. Sau đó sẽ thêm bảng đó vào trong Frame của bảng xếp hạng.

#### 6. Các công nghệ và kiến thức được áp dụng:

- Giải thuật ứng dụng kỹ thuật Mimax và Alpha Beta.

- Vấn đề đặt ra: Trong trò chơi Caro, cứ sau mỗi nước cờ, mỗi đối thủ sẽ chọn ra từ những ô trống để đi, do đó, sau 1 mỗi nước đi thì số ô trống còn lại sẽ giảm. Như vậy, việc tìm nước đi tiếp theo cho trạng thái có sẵn chỉ là việc tìm kiếm những ô trống còn lại, đồng thời, không gian tìm kiếm sẽ thu hẹp theo số nước đi đã tạo. Không gian chọn nước đi từ mỗi trạng thái ban đầu là hữu hạn, nhưng không gian tìm kiếm 1 nước đi dẫn đến chiến thắng là rất lớn. Do đó ta không thể vét sạch không gian tìm kiếm nước đi này mà ta phải giới hạn không gian tìm kiếm. Một không gian tìm kiếm có thể hiện theo 1 cây đa phân và được gọi là cây tìm kiếm hay cây trò chơi.

Ví dụ:



Hình 3-10: Một phần của cây trò chơi caro 9 ô

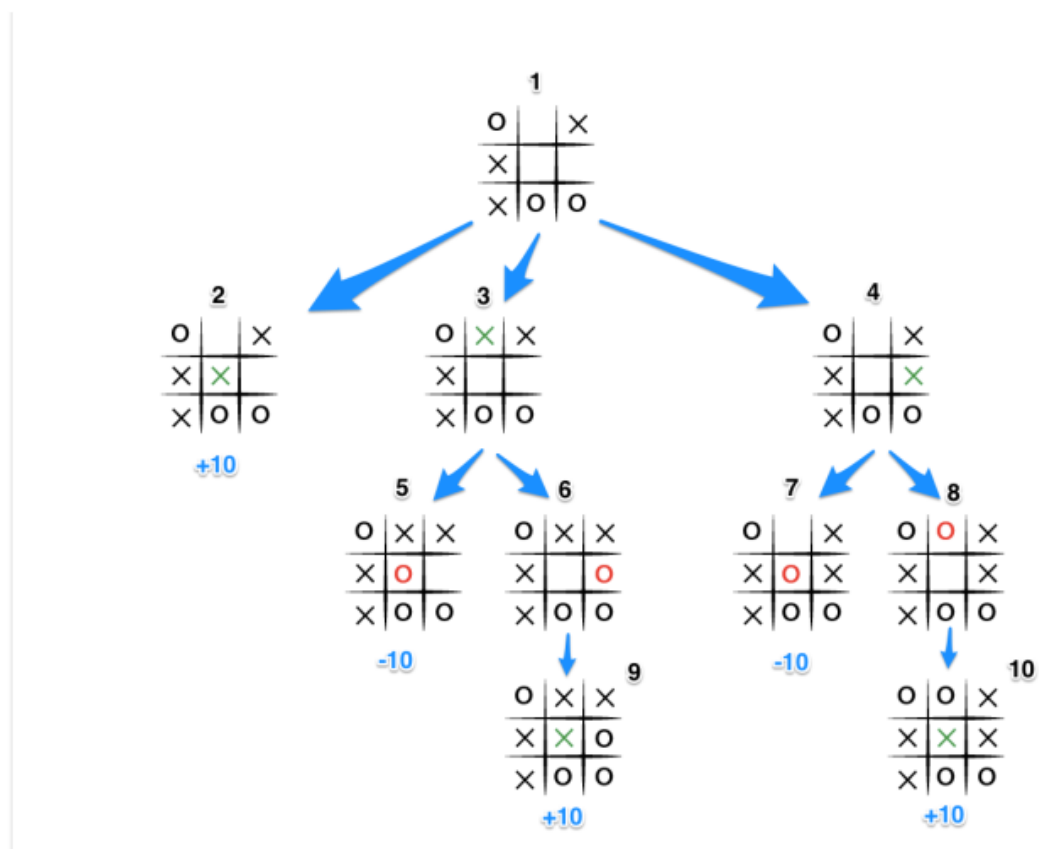
Hình 8: Ví dụ cây trò chơi

- Dựa vào cây trò chơi đã định nghĩa ở trên ta có thể thấy muốn chọn 1 nước đi tốt thì nếu chỉ dựa vào thế cờ hiện tại là chưa đủ, mà phải biết thông tin của những thế cờ sau khi chọn nước này để đi. Ta cùng tìm hiểu thuật toán Minimax để giải quyết vấn đề trên.

### 6.1 Thuật toán Minimax

- Giới thiệu: Minimax là 1 mô hình cây tìm kiếm và dùng thuật toán quay lui. Minimax được sử dụng trong việc ra quyết định và lý thuyết trò chơi để tìm ra nước đi tối ưu cho 1 người chơi, giả sử rằng đối thủ của bạn cũng sẽ chơi tối ưu nhất. Maximizer sẽ cố gắng đạt điểm cao nhất còn minimizer thì sẽ cố gắng đạt điểm số thấp nhất.

- Minh họa thuật toán cho game caro.



Hình 9: Minh họa thuật toán Minimax

- Giải thuật minimax



```

function minimax (node, depth, maximizingPlayer)
  if node is “End Node” or depth = 0
    return value(node)
  if maximizingPlayer:
    Mx = -  $\infty$ 
    foreach child of node
      Mx = max(Mx, minimax(child, depth-1, False))
    return Mx
  else:
    Mn =  $\infty$ 
    foreach child of node
      Mn := min(Mn, minimax(child, depth-1, True))
    return Mn

```

Hình 10: Thuật toán Minimax

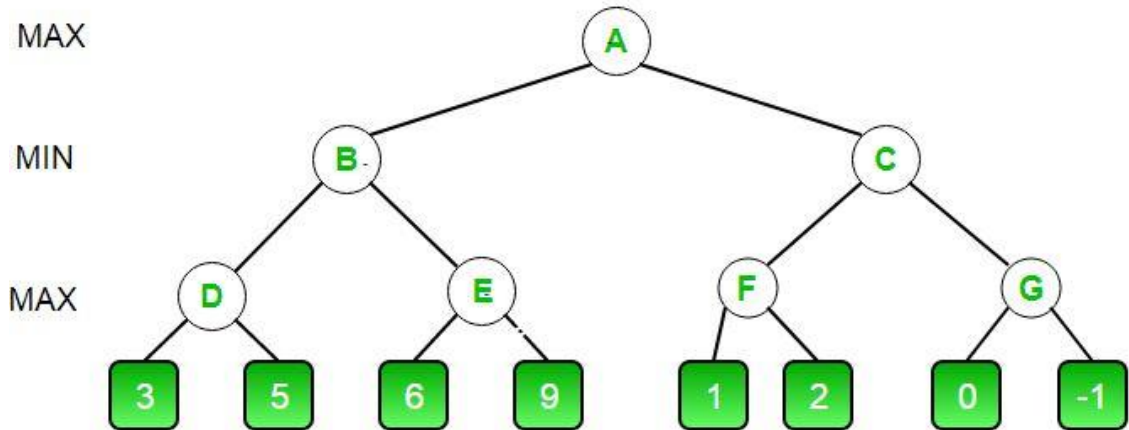
- Tuy nhiên Giải thuật tìm kiếm Minimax vấp phải vấn đề bùng nổ (mức hàm mũ) các khả năng nước đi cần phải xét  $\rightarrow$  không phù hợp với nhiều bài toán trò chơi thực tế. Vậy chúng ta cần phải loại đi những nhánh xét khi không xét đến mà không ảnh hưởng tới kết quả. Chúng ta có thể cắt tỉa (bỏ đi – không xét đến) một số nhánh tìm kiếm trong cây biểu diễn trò chơi.

## 6.2 Kỹ thuật cắt tỉa Alpha Beta

- Giới thiệu: Do lượt chơi sẽ có nhiều nước đi và mỗi nước đi sẽ dẫn tới có rất nhiều và sẽ sinh ra nhiều nút con vì vậy khi tìm kiếm nước đi bằng thuật toán Minimax nếu độ sâu của cây tìm kiếm càng sâu thì sẽ dẫn tới càng nhiều hướng di chuyển nên sẽ dẫn tới thời gian chạy thuật toán rất lâu. Kỹ thuật cắt tỉa Alpha Beta được áp dụng để cắt bỏ các nhánh trong cây tìm kiếm khi mà chắc chắn có 1 nước đi tốt hơn.
- Xác định 2 tham số Alpha và Beta: Alpha là giá trị tốt nhất của maximizer giá trị khởi tạo ban đầu là  $-\infty$ . Beta là giá trị tốt nhất của minimizer giá trị khởi tạo ban đầu là  $+\infty$ .
- Alpha là giá trị của nước đi tốt nhất đối với MAX (giá trị tối đa) tính đến hiện tại đối với nhánh tìm kiếm. Nếu  $v$  là giá trị tồi hơn

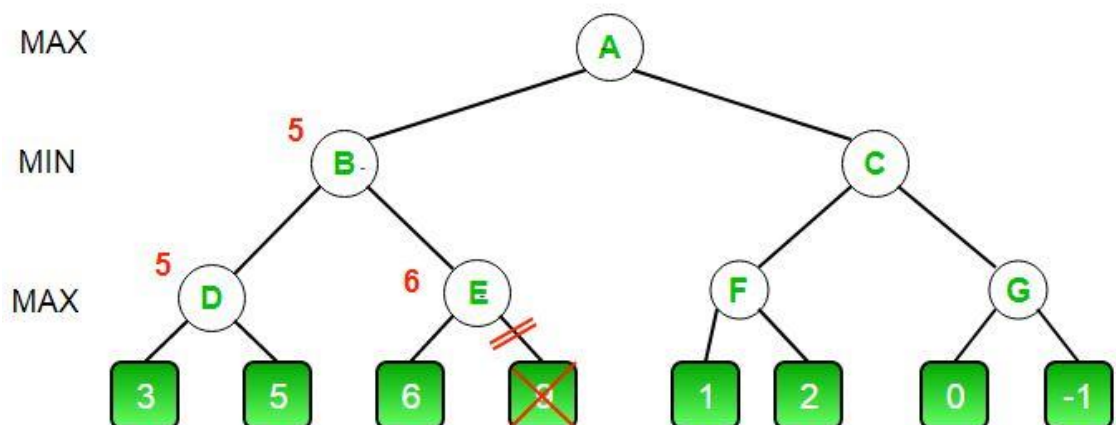
Alpha, MAX sẽ bỏ qua nước đi ứng với v  $\rightarrow$  Cắt tỉa nhánh ứng với v Beta được định nghĩa tương tự đối với MIN.

- Minh họa thuật toán:



Hình 11: Khi chưa dùng Alpha Beta.

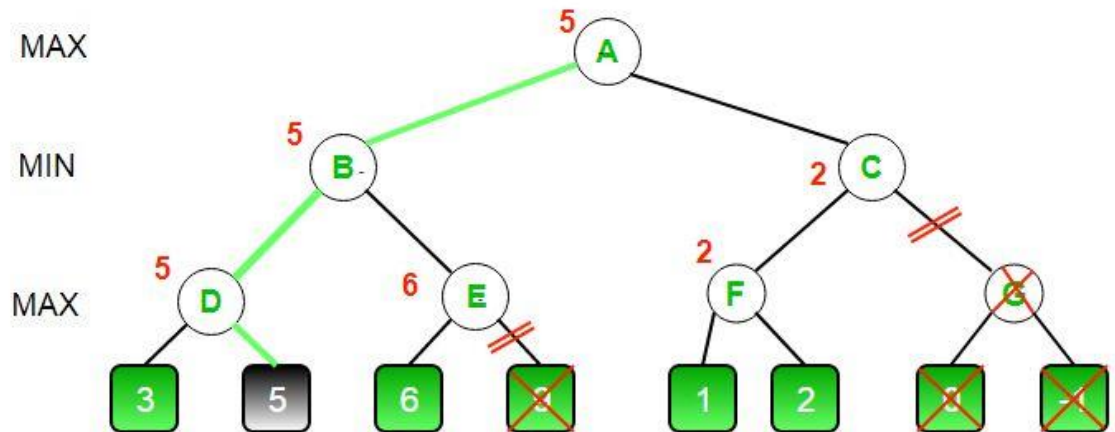
- A sẽ là max của B và C, B là min của D và E, C là min của F và G.
- Tại D Alpha = max (3, 5) nên nút D = 5.
- Tại B Beta = min (+inf, D) nên nút B có giá trị  $\leq 5$ .
- Tại E nút con bên trái = 6 nên Alpha = max (-inf, 6) nên giá trị của nút E phải lớn hơn hoặc bằng 6 mà nút B giá trị lớn nhất chỉ có thể là 5 vì vậy ta sẽ cắt tỉa và không xét đến nút 9 nữa.



Hình 12: Sau khi cắt tỉa Alpha Beta.

- Khi B có giá trị là 5 thì tại A có Alpha = max (-inf, 5) khi đó nút A có giá trị  $\geq 5$ .

- Tại F ta có  $F = \max(1, 2)$  thì giá trị nút F = 2.
- Tại C Beta =  $\min(+\infty, 2)$  khi đó nút C có giá trị  $\leq 2$ .
- Do nút A là max của B và C mà nút A có giá trị  $\geq 5$  mà nút C lại có giá trị  $\leq 2$  nên A lấy luôn giá trị của B là 5 và sẽ cắt tỉa ko xét nút G nữa.



Hình 13: Ứng dụng Alpha Beta

- Kỹ thuật cắt tỉa Alpha và Beta

```
function minimax (node, depth, maximizingPlayer)
    return alphabeta(node, depth, -  $\infty$ ,  $\infty$ , maximizingPlayer)
```

```
function alphabeta (node, depth, a, b, maximizingPlayer)
    if node is "End Node" or depth = 0
        return value(node)
    if maximizingPlayer
        foreach child of node
            a = max(a, alphabeta (child, depth-1, a, b, False))
            if a >= b break;
        return a
    else:
        foreach child of node
            b := min(b, alphabeta (child, depth-1, a, b, True))
            if a >= b break;
        return b
```

Hình 14: Kỹ thuật cắt tỉa

- Đối với các trò chơi có không gian trạng thái lớn, thì phương pháp cắt tỉa  $\alpha$ - $\beta$  vẫn không phù hợp. Không gian tìm kiếm (kết hợp cắt tỉa) vẫn lớn. Tuy nhiên với game caro không gian tìm kiếm cũng không quá lớn. Vậy nên chúng em dừng lại mức độ ứng dụng kỹ thuật cắt tỉa Alpha và Beta. Vẫn còn có 1 vài thuật toán giới hạn tìm kiếm hơn nữa và làm cho AI tron game trở nên “thông minh” hơn.
- Một vài cách chiến lược tìm kiếm tiên tiến hơn nữa
  - + Tri thức để cho phép đánh giá mỗi trạng thái của trò chơi.
  - + Tri thức bổ sung (heuristic) này đóng vai trò tương tự như là hàm ước lượng  $h(n)$  trong giải thuật tìm kiếm  $A^*$

#### *7. Khó khăn gặp phải:*

- Chúng em bàn với nhau để cùng nhau thống nhất về giao diện nên có nhiều chức năng hơn nữa hay là thiết kế một các đơn giản đẹp mắt. Cuối cùng chúng em đã thống nhất thiết kế game đơn giản thân thiện những em nhỏ cũng có thể sử dụng được.
- Trong quá trình nghiên cứu thuật toán người đánh với máy, chúng em đã phải tìm hiểu rất nhiều nguồn khác nhau về các thuật toán và nhiều lần sửa đổi để máy ngày càng thông minh hơn.
- Trong quá trình làm game chúng em cũng đã tìm hiểu thêm về socket với mục đích phục vụ đánh online. Tuy nhiên do kiến thức còn hạn chế em chúng em vẫn chưa hoàn thiện hết được để đưa vào ứng dụng lần này. Chúng em mới hoàn thiện được 3x3 nhưng khi áp dụng mẫu bảng to hơn lại sinh ra quá nhiều code lặp và điều phát sinh.

#### *8. Một vài mở rộng có thể làm:*

- Phân chia độ khó cho chế độ chơi với máy. Và cải thiện máy ngày càng thông minh hơn.

- Thêm tính năng lựa chọn ai là người được đi trước và mỗi người sẽ có 1 khoảng thời gian suy nghĩ nhất định.
- Thêm phần chơi online giữa 2 người mà chúng em chưa kịp hoàn thiện.
- Deploy game lên web và app di động.

### **III. Kết luận:**

Hiện nay, lập trình game đang là một lĩnh vực mới thu hút nhiều lập trình viên. Việc xây dựng các ứng dụng vui chơi giải trí là rất cần thiết cho sự phát triển công nghệ. Trong phạm vi đề tài, chúng em chỉ trình bày những phần công nghệ cơ bản có liên quan đến ứng dụng như lập trình giao diện người dùng với Swing,...

Nhìn chung game tạo ra đáp ứng nhu cầu giải trí của người chơi. Giao diện thân thiện, đơn giản, dễ sử dụng và thao tác. Tuy nhiên vẫn còn một số lỗi như lỗi font chữ chưa được khắc phục, hình ảnh còn hạn chế, chưa đẹp mắt.

Tuy vậy, do những hạn chế về trình độ, thời gian cũng như thiết bị nên em không thể tránh khỏi những vướng mắc, sai sót trong quá trình tìm hiểu và nghiên cứu. Em rất mong nhận được những đánh giá và chỉnh sửa từ thầy.

Em xin chân thành cảm ơn!

### **VI. Tài liệu tham khảo**

- Tài liệu và bài giảng của thầy Vũ Huân.
- Website: Geekforgeek (Thuật toán Minimax và Alpha Beta).
- Youtube: Brocode (Tham khảo thêm về JavaSwing).

- Một vài tài liệu thuật toán khác của các trường đại học trong và ngoài nước.

