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 vua bằng ngôn ngữ Java  
Giảng viên: Tiến sĩ Nguyễn Trọng Phúc

Nhóm thực hiện: Nhóm 26

Danh sách sinh viên tham gia:

Nguyễn Thế Quân – CNTT1 K62

Trịnh Thành Nam – CNTT1 K62

Ngụy Thế Quang – CNTT1 K62

*Hà Nội, tháng 4 năm 2023*

**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.

Cờ vua là một trong những game lâu đời trên thế giới và đến nay vẫn rất được ưa chuộng. Trong game cờ vua, người chơi sẽ đối đầu với nhau và phải sử dụng chiến thuật và kỹ năng để đánh bại đối thủ, cờ vua còn giúp nâng cao khả năng phân tích của người chơi. Từ đó chúng em đã chọn chủ đề cho bài tập lớn môn Java là “lập trình Game cờ vua”. Việc lập trình một game cờ vua bằng Java không chỉ giúp bạn nâng cao kỹ năng lập trình mà còn giúp bạn hiểu rõ hơn về cách hoạt động của game. Với việc sử dụng Java, bạn có thể dễ dàng tạo ra một giao diện đồ họa thân thiện với người dùng và tạo ra một trải nghiệm chơi game thú vị. Bạn có thể sử dụng các thư viện đồ họa như Swing để thiết kế giao diện và xử lý các sự kiện trong game.

**Mục lục**

[I. Đặt vấn đề 4](#_Toc133427165)

[II. Luật chơi 4](#_Toc133427166)

[III. Thiết kế các đối tượng 8](#_Toc133427167)

[IV. Kết quả đạt được 10](#_Toc133427168)

[V. Cải thiện và nâng cấp 21](#_Toc133427170)

[VI. Tài liệu tham khảo 21](#_Toc133427171)

# Đặt vấn đề

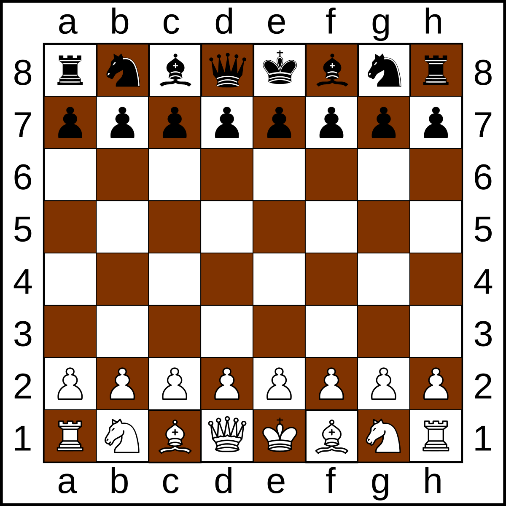
Cờ vua không phải là một trò chơi may rủi, nó dựa thuần túy vào chiến thuật và [chiến lược](https://vi.wikipedia.org/wiki/Chi%E1%BA%BFn_l%C6%B0%E1%BB%A3c). Tuy thế, trò chơi này phức tạp đến mức thậm chí cả những người chơi hay nhất cũng không thể tính hết tất cả mọi phương án: mặc dù chỉ có 64 ô và 32 quân cờ trên bàn cờ nhưng số lượng nước đi có thể được thì còn vượt xa cả số lượng các nguyên tử có trong [vũ trụ](https://vi.wikipedia.org/wiki/V%C5%A9_tr%E1%BB%A5).

Cờ vua là một trong những trò chơi trí tuệ phổ biến nhất thế giới, được hàng triệu người trên toàn thế giới chơi tại nhà, các [câu lạc bộ](https://vi.wikipedia.org/w/index.php?title=C%C3%A2u_l%E1%BA%A1c_b%E1%BB%99_c%E1%BB%9D_vua&action=edit&redlink=1), hay trên các nền tảng [trực tuyến](https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua_tr%E1%BB%B1c_tuy%E1%BA%BFn), nó còn được công nhận là một môn thể thao. Cờ vua đôi khi được nhìn nhận như là [trò chơi chiến tranh](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_chi%E1%BA%BFn_tranh) [trừu tượng](https://vi.wikipedia.org/wiki/Tr%E1%BB%ABu_t%C6%B0%E1%BB%A3ng); cũng như là "các cuộc đấu trí tuệ", và việc chơi cờ vua được coi như là một cách để rèn luyện tư duy và khả năng phân tích của người chơi

Trong thời kì công nghệ thông tin ngày càng phát triển, game cờ vua vẫn chưa bao giờ giảm đi tính trí tuệ nội tại. Máy tính không hề làm trò chơi này mất đi thú vị. Ngược lại máy tính có thể giúp trinh độ của kì thù tăng lên nhanh chóng.

# Luật chơi

1. **Thiết lập quân cờ và bàn cờ.**

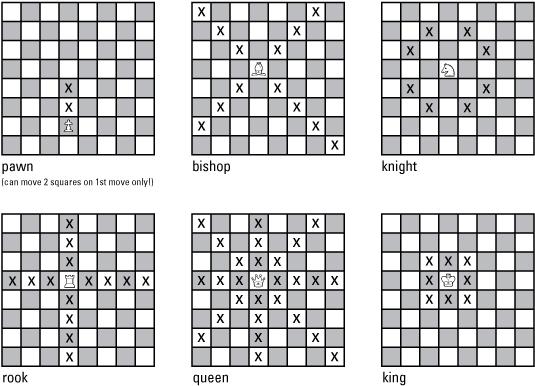
Trò chơi sử dụng một bàn cờ hình vuông chia thành 64 ô vuông với 8 hàng ngang và 8 hàng dọc. Có thể cho 2 người chơi, hoặc một người chơi với máy. Mỗi người chơi sẽ bắt đầu với 16 quân cờ, bao gồm 8 [tốt](https://vi.wikipedia.org/wiki/T%E1%BB%91t_(c%E1%BB%9D_vua)), 2 [mã](https://vi.wikipedia.org/wiki/M%C3%A3_(c%E1%BB%9D_vua)), 2 [tượng](https://vi.wikipedia.org/wiki/T%C6%B0%E1%BB%A3ng_(c%E1%BB%9D_vua)), 2 [xe](https://vi.wikipedia.org/wiki/Xe_(c%E1%BB%9D_vua)), 1 [hậu](https://vi.wikipedia.org/wiki/H%E1%BA%ADu_(c%E1%BB%9D_vua)) và 1 [vua](https://vi.wikipedia.org/wiki/Vua_(c%E1%BB%9D_vua)).

Cách xếp bàn cờ được thực hiện như trong ảnh:

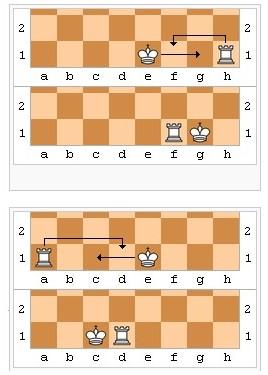
+ Quân trắng sẽ được xếp vào hàng đầu tiên (hàng 1) theo thứ tự từ trái sang phải (từ *a* đến *h*) lần lượt là: xe, mã, tượng, hậu, vua, tượng, mã, xe.

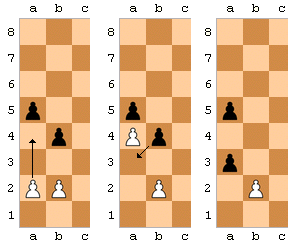
+ Hàng thứ hai được xếp 8 quân tốt trắng. Cách xếp quân đen đối xứng hoàn toàn với quân trắng ở hai hàng cuối cùng.

1. **Di chuyển quân cờ.**



* Quân Vua (K): Mỗi lượt, Vua có thể di chuyển tới một trong 8 ô xung quanh.
* Quân Hậu (Q): Mỗi lượt, Hậu có thể di chuyển theo đường chéo hoặc đường thẳng mà không giới hạn số ô.
* Quân Xe (R): Mỗi lượt, Xe có thể di chuyển theo hàng dọc hoặc ngang mà không giới hạn số ô
* Quân Tượng (B): Mỗi lượt, Tượng có thể di chuyển theo đường mà không giới hạn số ô
* Quân Mã (N): Mỗi lượt, Mã có thể di chuyển sao cho từ ô đích có thể vẽ 1 hình chữ L gồm 4 ô tới ô hiện tại đang đứng, đồng thời cũng không bị cản bởi quân đồng minh
* Quân Tốt (P): Ở lượt di chuyển đầu tiên được đi tối đa 2 ô thẳng về phía trước, các lượt còn lại chỉ được đi một ô. Tốt ăn theo đường chéo tới 1 ô ở trước mặt

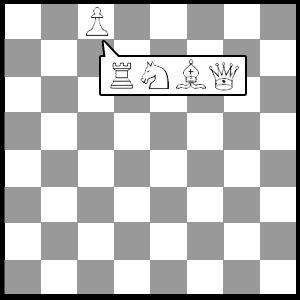


* **Nước đi đặc biệt**
* **Nhập thành**: Nếu như xe và vua chưa đi nước đi đầu tiên, đồng thời giữa vua và xe không có bất kì quân nào ngăn cản thì có thể tiến hành nước đi nhập thành. Có thể là nhập thành ngắn (như hình trên), hoặc nhập thành dài (như hình dưới), khi nhập thành phải dùng đi vua trước, đi xe sau.
* **Bắt tốt qua đường** “en passant”: 

.Về Cách ăn quân: Nếu tốt trắng nhảy 2 ô từ hàng 2 lên hàng 4 thì tốt đen ở cột bên cạnh nhưng cùng hàng với tốt trắng có thể ăn chéo theo cách mà nó ăn tốt trắng nếu tốt trắng tiến 1 ô.

.Về thời hạn hiệu lực: Nước bắt Tốt qua đường chỉ có thể thực hiện liền sau nước di chuyển Tốt của đối phương. Nếu thực hiện một nước đi khác thay vì bắt Tốt qua đường thì các nước đi tiếp sau sẽ không được bắt tốt qua đường quân đó nữa.

.Có thể áp dụng tương tự với việc tốt đen đi từ hàng 7 lên hàng 5.



* **Phong cấp**:

+ Khi con tốt trắng đi được đến hàng 8 hoặc con tốt đen đi được đến hàng 1, chúng sẽ được phong cấp thành bất cứ quân cờ nào trong 4 quân (Hậu, Xe, Tượng, Mã).

1. **Người chơi và thắng thua**

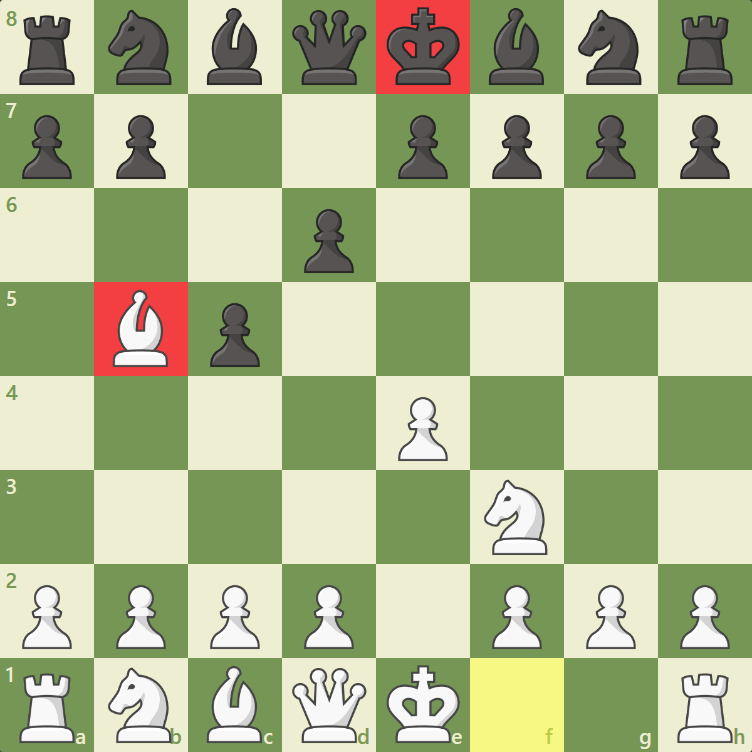
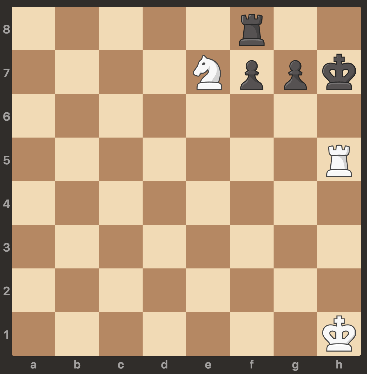
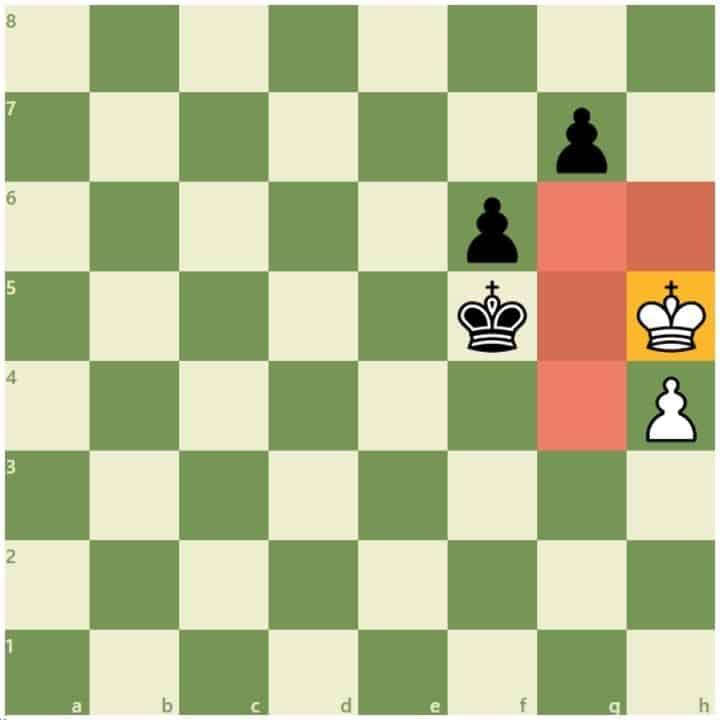
* Cờ vua là 1 trò chơi đối kháng gồm 2 người, lần lượt đi luân phiên nhau. Quân trắng sẽ đi trước. Người chơi dành chiến thắng khi chiếu hết được đối phương (**CHECKMATE**), người thua sẽ là người bị chiếu hết. Trong một số trường hợp đặc biệt có trường hợp hai bên hòa nhau, ví dụ như trạng thái hết nước đi (**STALEMATE**).

**+ Chiếu:** Tình huống mà vua của người chơi bị đe dọa bị bắt trong tình huống tiếp theo.

**+ CHECKMATE:** trạng thái quân vua bị chiếu mà không có cách nào thoát ra.

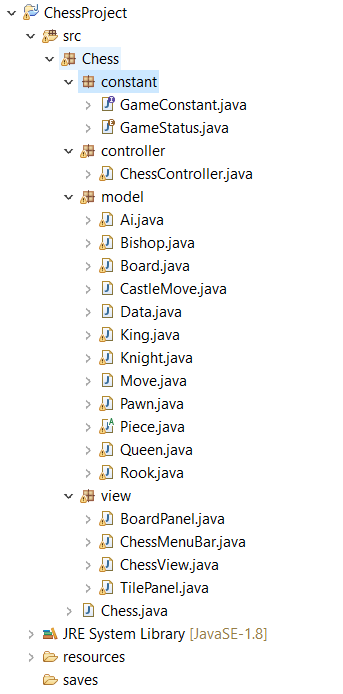
**+ STALEMATE:** một bên không thể có một nước đi hợp lệ dù không bị chiếu.

**Chiếu CHECKMATE STALEMATE**

1. **Thiết kế các đối tượng**

- Game được thiết kế dựa trên mô hình MVC. Cấu trúc của chương trình:



* Package **constant** (chứa những hằng số và config của trò chơi)

+ Interface **GameConstant**: Lưu các kích thước của bàn cờ.

+ Enum **GameStatus**: Lưu các trạng thái của game;

* Package **controller (**Lấy tương tác người dùng để lấy dữ liệu của **model** rồi gửi vào **view** để nhận lại giao diện)

**+** Class **chessController**: Lấy tương tác người dùng để lấy dữ liệu của **model** rồi gửi vào **view** để nhận lại giao diện. Cập nhật dữ liệu của bàn cờ, điều khiển trận đấu. Điều khiển các chức năng thanh menu.

* Package **model** (Lưu trữ toàn bộ cấu trúc dữ liệu, logic game)

+ Class **Ai:** Lưu các thông tin về đối tượng Ai, thuật toán, các phương thức lấy nước đi và di chuyển quân của nó.

+ Abstract class **Piece:** Làm khung của các quân cờ.

**+** Class **King:** Lưu trữ thông tin về quân vua. (extends **Piece**)

+ Class **Pawn:** Lưu trữ thông tin về quân tốt. (extends **Piece**)

+ Class **Rook:** Lưu trữ thông tin về quân xe. (extends **Piece**)

+ Class **Knight:** Lưu trữ thông tin về quân mã. (extends **Piece**)

+ Class **Bishop:** Lưu trữ thông tin về quân tượng. (extends **Piece**)

+ Class **Bishop:** Lưu trữ thông tin về quân hậu. (extends **Piece**)

+ Class **Move**: Lưu trữ thông tin của nước đi.

+ Class **CastleMove**: Lưu trữ thông tin về nước đi nhập thành   
(extends **Move**)

+ Class **Board**: Lưu trữ thông tin về mọi đối tượng trên bàn cờ (quân cờ, nước đi), bàn cờ trước nó và lượt của người chơi.

+ Class **Data**: Lưu trữ phương thức lưu và đọc bàn cờ.

* Package **view**: (Chịu trách nhiệm xử lý giao diện)

+ Class **TilePanel:** Xử lý hình ảnh cho mỗi ô trên bàn cờ.

+ Class **BoardPanel**: Xử lý giao diện cho cả bàn cờ.

+ Class **ChessMenu**: Tạo thanh menu chức năng.

+ Class **ChessView**: Xử lý giao diện của cả trò chơi.

* Folder **saves**: Lưu trữ các game dưới dạng file .CSV
* Folder **resources**: Lưu trữ tất cả hình ảnh sử dụng trong trò chơi
* Class **Chess**: Khởi động trò chơi.

# Kết quả đạt được

* Game cờ vua với giao diện đơn giản. Có thể chơi hai người hoặc một người chơi với máy.
* Game bao gồm những luật chơi cơ bản của cờ vua.
* Trên thanh Menu:

+ “Game” gồm: New AI game, New game, Undo, Close.

+ “File” gồm: Save, Load.

**4.1. Giao diện**



**4.2. Hệ thống sẽ giúp hiển thị những nước đi khả thi**



**4.3. Các nước đi đặc biệt**

**4.3.1. Nhập thành**





**4.3.2. Bắt tốt qua đường**



**4.3.3. Phong cấp**

* 1. **Undo**

* 1. **Làm việc với tệp tin qua các chức năng Save và Load.**





Làm việc và lưu trữ các tệp tin dưới dạng .CSV. Cho phép người dùng có thể lưu lại ván đấu mà mình đang chơi và có thể chơi lại nếu muốn.

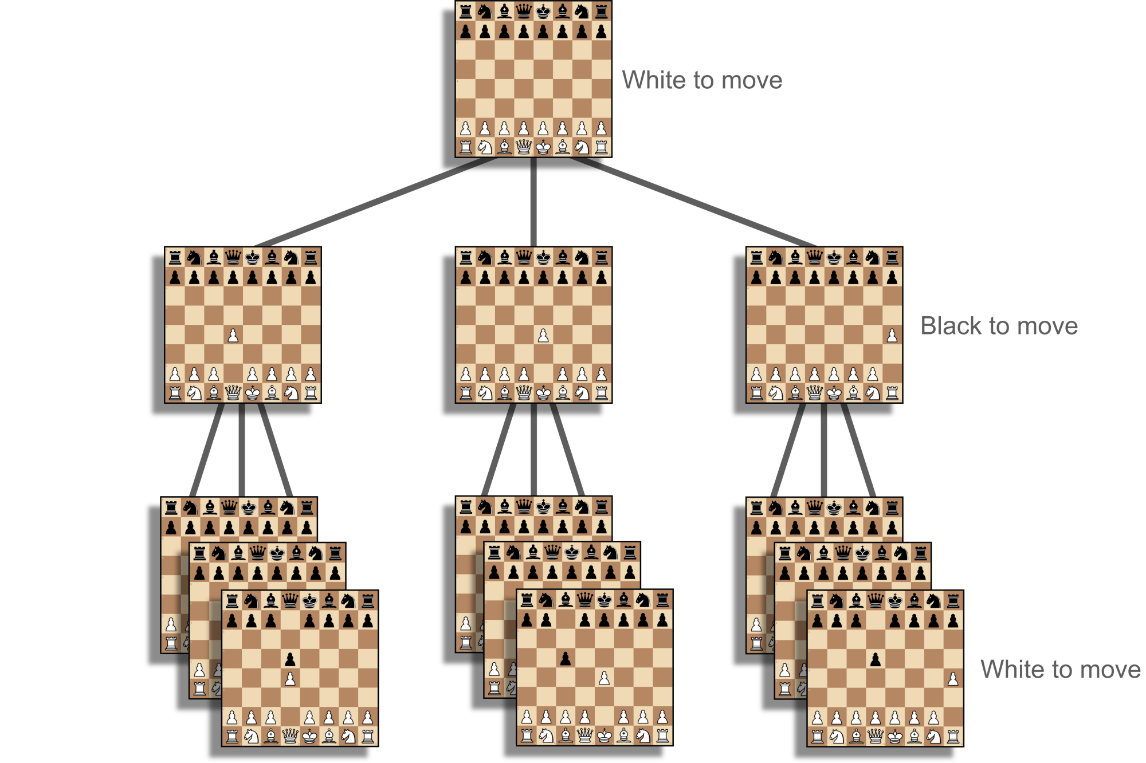
Hạn chế: Chứa thiết kế để kết hợp với CSDL

* 1. **AI với nhiều độ khó dùng giải thuật cắt tỉa Alpha-Beta trong thuật toán Minimax**



AI có 3 mức Easy, Normal, Hard, tương ứng với các độ sâu 2, 3, 4.

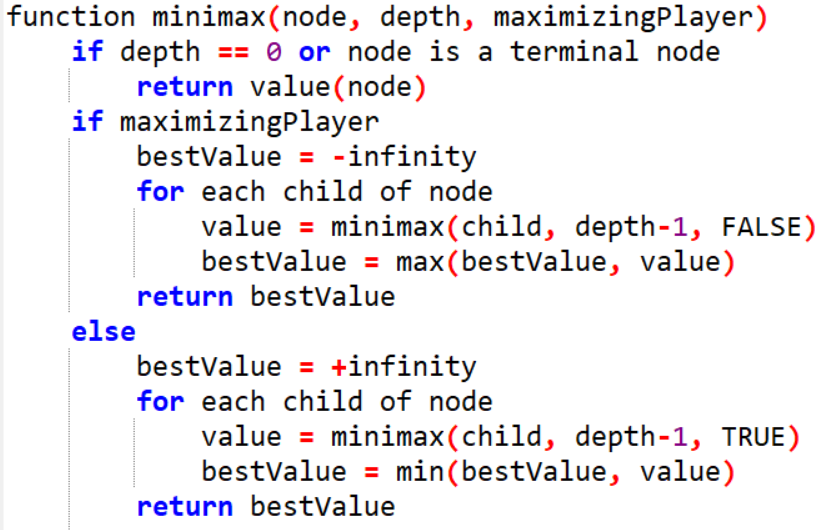
- *Vấn đề đặt ra*: Trong trò chơi Cờ vua, tại mỗi nước cờ sẽ có thể có rất nhiều thế cờ khác nhau được tạo ra (tùy vào quân nào di chuyển và di chuyển tới đâu). Mỗi thế cờ sẽ là một nút trên cây tìm kiếm đa phân. Nút con được sinh ra từ nút cha sau khi thực hiện một nước đi hợp lệ. Ta có thể coi không gian tìm kiếm các nước đi chính là một cây đa phân



**4.6.1. Thuật toán Minimax**

- Minimax là 1 mô hình cây tìm kiếm sử dụng 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.

* *Giải thuật minimax*



Hình 10: Thuật toán Minimax

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

*Kỹ thuật cắt tỉa Alpha Beta*

- Giới thiệu:

+ Do mỗi lượt chơi có nhiều cách thực hiện nước đi và mỗi nước đi sẽ dẫn tới việc 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 → dẫn tới thời gian chạy thuật toán rất lâu.

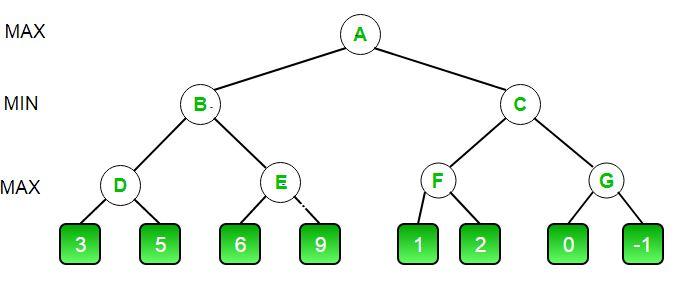
+ Kỹ thuật cắt tỉa Alpha Beta được áp dụng để cắt bỏ các nhánh trong cây tìm kiếm khi mà chắc chắn có 1 nước đi tốt hơn.

- Xác định 2 tham số Alpha và Beta:

+ Alpha là giá trị tốt nhất của maximizer (giá trị khởi tạo là –inf)

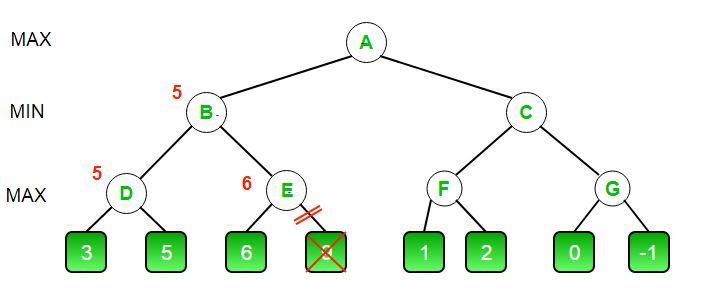
+ Beta là giá trị tốt nhất của minimizer (giá trị khởi tạo là +inf)

- Minh họa thuật toán:



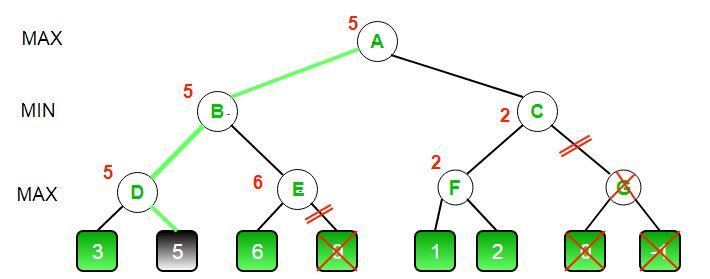
Hình 11: Khi chưa dùng Alpha Beta.

* 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 <= D → B <= 5
* Tại E, nút con bên trái = 6 → Alpha = max (-inf, 6) → giá trị nút E sẽ >= 6 mà nút B <= 5 → Ta sẽ cắt tỉa và không xét đến nút 9 nữa.



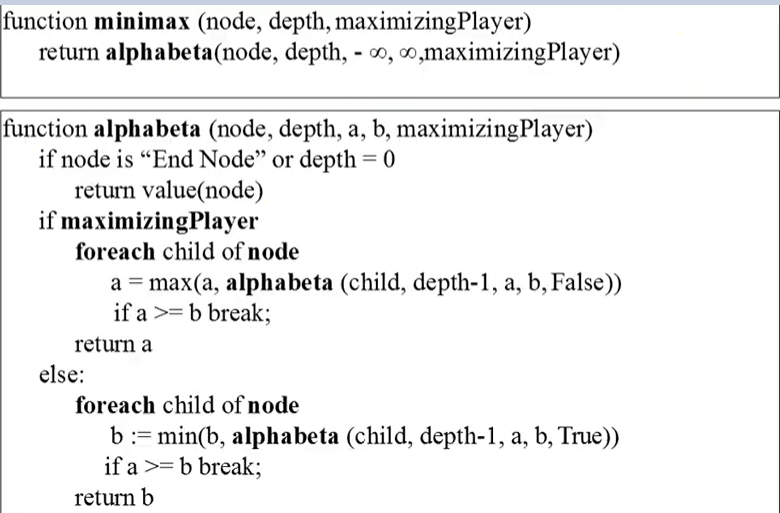
Hình 12: Sau khi cắt tỉa Alpha Beta.

* Khi B có giá trị là 5 thì tại A có Alpha = max (-inf, 5). Khi đó, A >= 5.
* Ta có F = max (1, 2) → F = 2.
* Tại C, Beta = min (+inf, 2) → C <= 2.
* Do nút A = max(B, C) mà A >= 5, nhưng ở trên ta tính được C <= 2 → A lấy luôn giá trị của B là 5 và không cần xét nút G nữa



Hình 13: Ứng dụng Alpha Beta

* *Kỹ thuật cắt tỉa Alpha và Beta*



Hình 14: Kỹ thuật cắt tỉa

* Tuy nhiên, đối với các trò chơi có không gian trạng thái lớn, thì phương pháp cắt tỉa α-β vẫn không phù hợp. Không gian tìm kiếm (kết hợp cắt tỉa) vẫn lớn.

# Cải thiện và nâng cấp

1. Vấn đề hình ảnh  
   - Giao diện được thiết kế đơn giản, chỉ bao gồm một bàn cờ, và thanh menu. Khi phong cấp, chiếu hết,… thì chỉ có hộp thoại xuất hiện, chưa có những hiệu ứng như mở bảng quân phong cấp,... Có thể nâng cấp thêm.
2. Vấn đề cơ sở dữ liệu  
   - Chưa thiết kế được cơ sở dữ liệu, chỉ đọc và lưu dữ liệu trên tệp.
3. Một số tính năng có thể phát triển thêm:

* Bộ đếm thời gian
* Hiệu ứng âm thanh khi: di chuyển, ăn quân, kết thúc trận đấu,…

1. Cải thiện trí tuệ nhân tạo

- Cách tính điểm cho mỗi nước đi nên thay đổi sang việc tính toán dựa trên trạng thái của bàn cờ tại thời điểm đánh

- AI mặc dù được sử dụng thuật toán cắt tỉa Alpha-Beta trong giải thuật Minimax nhưng nếu tính toán với độ sâu >= 5 thì máy sẽ chạy tương đối lâu.

- AI không bao gồm dữ liệu về khai cuộc cờ nên những nước đi ban đầu không hiệu quả mặc dù ở chiều sâu lớn.

1. **Tài liệu tham khảo**

Slide bài giảng của giảng viên **Nguyễn Trọng Phúc** – ĐH GTVT.

<https://www.geeksforgeeks.org/design-a-chess-game/?ref=lbp>

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/>

<https://www.chess.com/vi/cach-choi-co-vua>

https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm

<https://www.youtube.com/watch?v=CAEI_J50B18&t=1636s>.