



BÁO CÁO ĐỒ ÁN CỜ VUA

MÔN TRÍ TUỆ NHÂN TẠO



Lớp: CS106.J21

GVHD: Phạm Nguyễn Trường An

Danh Sách Sinh Viên:

Nguyễn Duy Minh – 16521735

Trần Quang Khôi – 16521703

Mục Lục

THÔNG TIN VỀ GIẢI VÔ ĐỊCH CỜ VUA THẾ GIỚI	2
1) Đơn vị tổ chức	2
2) Địa điểm và thời gian.....	2
3) Lịch sử.....	2
4) Thể lệ thi đấu của giải Vô địch cờ vua thế giới năm 2018.....	2
a) Đối tượng.....	2
b) Thời gian	3
c) Luật thi đấu	3
Cơ sở lý thuyết	4
CHIẾN LƯỢC TÌM KIẾM MINIMAX.....	4
NGUYÊN LÝ CẮT TỈA ALPHA-BETA	5
Phân tích chương trình	6
Cấu trúc chương trình.....	6
Chức năng và nhiệm vụ của các Class	6
Cấu trúc quân cờ.....	8
Cấu trúc dữ liệu bàn cờ.....	9
Phần các phương thức có trong AI	13
Một số phương thức khác	13
Các vấn đề về giải thuật.....	14
Lấy nước đi hợp lệ	14
Đánh giá thế cờ.....	15
Minh họa các bước chạy của chương trình	15
Ứng dụng.....	17
Giao diện bắt đầu ván đấu.....	17
GIẢI CỜ THẾ.....	17
LƯU VÁN ĐẤU	19
TIẾP TỤC TỪ BẢN GHI VÁN ĐẤU	19
Cài đặt	20
Tài liệu tham khảo.....	20

THÔNG TIN VỀ GIẢI VÔ ĐỊCH CỜ VUA THẾ GIỚI

1) Đơn vị tổ chức

Liên đoàn Cờ vua thế giới (FIDE - Fédération internationale des échecs) được thành lập tại Paris ngày 20 tháng 7 năm 1924

2) Địa điểm và thời gian

Trận đấu tranh cúp vô địch cờ vua thế giới năm 2018 được tổ chức tại The College in Holborn ở thành phố London của Anh Quốc vào ngày 28 tháng 11 năm 2018.

3) Lịch sử

Trong hai năm 1988–89, Hiệp hội các đại kiện tướng (GMA) tổ chức một chuỗi 6 giải đấu quy tụ các kỳ thủ hàng đầu thời gian đó, có tên Cúp cờ vua thế giới, định dạng tương tự như Grand Prix hiện nay, với giải thưởng riêng cho từng giải và chung cho cả chuỗi 6 giải đấu.

Vào các năm 2000 và 2002, Liên đoàn cờ vua thế giới tổ chức các giải đấu mang tên Cúp cờ vua thế giới thứ nhất và thứ hai. Hai giải đấu này riêng biệt, không nằm trong hệ thống Giải vô địch cờ vua thế giới. Cả hai giải đấu này chức vô địch đều thuộc về Viswanathan Anand.

Từ năm 2005, một giải đấu cùng tên nhưng khác thể thức nằm trong hệ thống Giải vô địch cờ vua thế giới. Giải đấu được tổ chức hai năm một lần, gồm 128 kỳ thủ thi đấu theo thể thức loại trực tiếp, cùng thể thức với các giải trước đó như Tilburg trong khoảng thời gian 1992-94, hoặc các Giải vô địch cờ vua thế giới (FIDE) trong giai đoạn 1998–2004.

Các giải đấu năm 2005, 2007, 2009 và 2011 được tổ chức ở Khanty-Mansiysk.

Sau đó FIDE quyết định đơn vị đăng cai Olympiad cờ vua năm sau sẽ tổ chức luôn Cúp cờ vua thế giới vào năm trước.

Các Cúp cờ vua thế giới từ 2005 lấy một số lượng kỳ thủ nhất định vào Giải cờ vua chọn ứng viên của Giải vô địch cờ vua thế giới tiếp theo. Con số cụ thể của từng giải được trình bày ở bảng bên dưới. Hiện tại hai kỳ thủ vào chung kết giành quyền vào Giải cờ vua chọn ứng viên.

4) Thể lệ thi đấu của giải Vô địch cờ vua thế giới năm 2018

a) Đối tượng

Các ứng viên tham gia sẽ thi đấu để chọn ra người giỏi nhất, người đó được gọi là người thách đấu. Sau đó người thách đấu sẽ thi đấu với đương kim vô địch. Nếu người thách đấu thắng thì họ sẽ trở thành tân vô địch của giải. Nếu người thách đấu thua thì đương kim vô địch bảo vệ thành công ngôi vô địch.

b) Thời gian

Theo qui định của Liên đoàn Cờ Vua thế giới (FIDE), thời gian cho một ván cờ tiêu chuẩn là 90 phút mỗi bên, cộng 30 giây cho mỗi nước đi.

c) Luật thi đấu

Khi một ván cờ vua bắt đầu, một người chơi sẽ cầm quân đen và người chơi còn lại cầm quân trắng. Việc chọn lựa ai sẽ cầm quân đen hay trắng phụ thuộc vào thể thức chơi là giải trí hay thi đấu trong hệ thống có tổ chức. Nó có thể là do hai người thỏa thuận (giải trí) hoặc do quyết định của trọng tài trong giải đấu. Bên cầm quân trắng luôn luôn được đi trước và do đó có ưu thế nhỏ so với bên cầm quân đen. Các quân cờ cần phải xếp trên bàn cờ tiêu chuẩn với ô nằm ở hàng cuối cùng bên tay phải người chơi bao giờ cũng có màu nhạt.

Các quân cờ có nước đi khác nhau:

- Xe (ký hiệu quốc tế R - Rook) di chuyển theo các đường thẳng dọc theo cột hay hàng tới ô còn trống mà không có quân nào cản trên đường đi hay tới ô bị quân đối phương chiếm giữ (ăn quân) nhưng không thể vượt qua quân đang đứng ở ô đó. Ngoại lệ duy nhất là trường hợp nhập thành. Khi đó nó có thể nhảy qua quân Vua của mình để đứng cạnh nó. Chỉ có Xe mới có nước đi như thế.
- Tượng (ký hiệu quốc tế B - Bishop) di chuyển theo đường chéo tới ô có cùng màu với nguyên lý tương tự như Xe tới ô còn trống hay ô bị quân đối phương chiếm giữ (ăn quân).
- Hậu (ký hiệu quốc tế Q - Queen) có nước đi là tổ hợp đơn giản của chuyển động của Xe và Tượng. Trong một nước đi nó có thể di chuyển theo đường chéo hoặc đường thẳng dọc theo cột hay hàng, với nguyên lý đi và ăn quân giống như Tượng và Xe.
- Mã (ký hiệu quốc tế N - Knight) có thể di chuyển tới ô còn trống hay ô bị quân đối phương chiếm giữ (ăn quân) theo dạng hình chữ L (hình chữ nhật 3×2 hay 2×3). Quân Mã không bị cản như trong cờ tướng.
- Tốt (không cần ký hiệu) có thể di chuyển thẳng về phía trước chỉ một ô một lần tới ô còn trống (đi mà không ăn quân), nhưng khi di chuyển quân để ăn quân đối phương thì đi chéo. Ví dụ, Tốt trắng tại ô C4 có quyền ăn quân đối phương tại B5 hoặc D5 nếu một trong hai ô này có quân đối phương chiếm hoặc di chuyển xuống ô C5 nếu ô này còn trống, trừ hai trường hợp sau:
 - Nó có thể di chuyển 1 hoặc 2 ô nếu nó đi từ vị trí xuất phát ban đầu tới ô chưa bị chiếm giữ, nhưng không thể nhảy qua một quân khác để tới ô đó. Ví dụ Tốt trắng tại G2 có thể đi tới G3 hoặc G4 nếu đây là nước đi đầu tiên của nó và các ô này chưa bị chiếm giữ, nhưng nó không thể đi tới G4 nếu ô G3 đã có một quân nào đó chiếm giữ.

- Trong trường hợp khi một quân Tốt nào đó của bên trắng đạt tới hàng 5 (ví dụ tới ô E5) và quân Tốt thuộc một trong hai cột của bên đen nằm ngay bên cạnh cột mà Tốt trắng này đang chiếm giữ (trong trường hợp đã cho là cột D và cột F) đi từ vị trí xuất phát đầu tiên (D7 hay F7) nhảy liên 2 ô tới ô D5/F5 thì Tốt trắng tại vị trí E5 ngay tại nước đi sau đó có quyền ăn Tốt đen tại ô D5/F5 và di chuyển tiếp tới ô D6/F6. Quyền này sẽ tự động mất, nếu tại nước đi ngay sau đó quân trắng di chuyển quân khác.
- Tương tự như vậy cho Tốt đen khi nó đã chiếm giữ hàng 4. Đây là trường hợp mà trong cờ vua người ta gọi là bắt tốt qua đường (en passant).

Tốt còn một đặc điểm nữa là khi nó di chuyển đến hàng cuối cùng thì người chơi có quyền phong cấp cho nó thành bất kỳ quân nặng hay nhẹ nào (Hậu, Xe, Tượng, Mã).

- Vua (ký hiệu quốc tế là K - King) là quân quan trọng nhất, nếu mất Vua thì người chơi thua cuộc. Mỗi lần đi nó có thể ăn quân hoặc di chuyển sang các ô bao quanh ô mà nó hiện tại đang chiếm giữ, nhưng không thể tới ô mà quân của mình đang chiếm giữ hay các ô bị quân đối phương kiểm soát. Ngoại lệ duy nhất là trường hợp nhập thành. Khi đó nó có thể di chuyển qua hai ô đồng thời với việc di chuyển quân Xe của mình để quân Xe đó đứng bên cạnh nó về phía cột trung tâm. Ký hiệu của nhập thành là 0-0 (nhập thành gần) và 0-0-0 (nhập thành xa). Xem thêm nhập thành.

✚ **Lưu ý:** Khi thực hiện nhập thành trên thực tế, theo luật của FIDE, bao giờ cũng phải di chuyển Vua trước và thực hiện bằng một tay duy nhất.

Khi ăn quân đối phương, quân tấn công sẽ di chuyển tới ô đó và thay thế cho quân đối phương tại vị trí này, bắt tốt qua đường (en passant) là ngoại lệ duy nhất. Quân bị ăn được loại ra khỏi bàn cờ. Vua không thể không bảo vệ khỏi nước chiếu, do đó khi bị chiếu thì người chơi phải thực hiện các biện pháp nhằm cứu Vua (di chuyển Vua khỏi vị trí bị chiếu, ăn quân đang chiếu hay dùng quân khác của mình cản đường chiếu nếu có thể). Nếu không thể có nước đi để cứu Vua thì người chơi bị chiếu bí và thua cuộc.

Các ván cờ không phải bao giờ cũng kết thúc bằng chiếu bí. Có thể một bên xin thua, có thể thua do hết giờ hoặc phạm luật chơi. Có thể xảy ra các ván cờ hòa. Một ván cờ vua là hòa khi: do thỏa thuận của hai bên do không bên nào dám mạo hiểm hay khi không đủ lực lượng để chiếu hết, rơi vào trạng thái hết nước đi (stalemate), cả hai bên lặp lại nước đi ba lần hay luật 50 nước đi (perpetual check).

Cơ sở lý thuyết

CHIẾN LƯỢC TÌM KIẾM MINIMAX

Minimax (còn gọi là minmax) là một phương pháp trong lý thuyết quyết định có mục đích là tối thiểu hóa (minimize) tổn thất vốn được dự tính có thể là "tối đa" (maximize). Có thể hiểu ngược lại là, nó nhằm tối đa hóa lợi ích vốn được dự tính là tối thiểu (maximin). Nó bắt nguồn từ trò chơi có tổng bằng

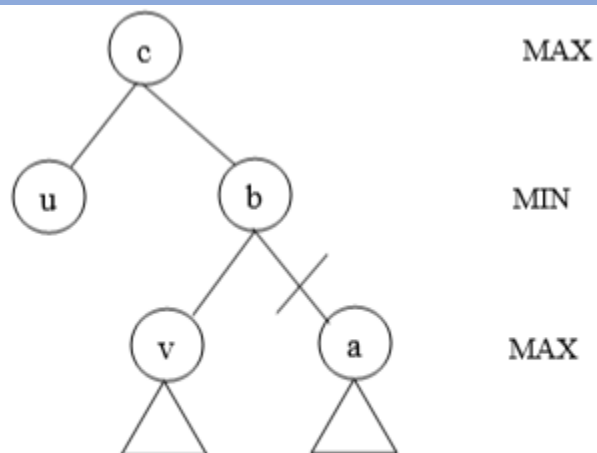
không. Nó cũng được mở rộng cho nhiều trò chơi phức tạp hơn và giúp đưa ra các quyết định chung khi có sự hiện diện của sự không chắc chắn.

Một phiên bản của giải thuật áp dụng cho các trò chơi như tic-tac-toe, khi mà mỗi người chơi có thể thắng, thua, hoặc hòa. Nếu người chơi A có thể thắng trong 1 nước đi, thì "nước đi tốt nhất" chính là nước đi để dẫn đến kết quả thắng đó. Nếu người B biết rằng có một nước đi mà dẫn đến tình huống người A có thể thắng ngay ở nước đi tiếp theo, trong khi nước đi khác thì sẽ dẫn đến tình huống mà người chơi A chỉ có thể, tốt nhất, là hòa thì nước đi tốt nhất của người B chính là nước đi sau.

Ta sẽ nắm rõ, thế nào là một nước đi "tốt nhất". Giải thuật Minimax giúp tìm ra nước đi tốt nhất, bằng cách đi ngược từ cuối trò chơi trở về đầu. Tại mỗi bước, nó sẽ ước định rằng người A đang cố gắng tối đa hóa cơ hội thắng của A khi đến phiên anh ta, còn ở nước đi kế tiếp thì người chơi B cố gắng để tối thiểu hóa cơ hội thắng của người A (nghĩa là tối đa hóa cơ hội thắng của B).

NGUYÊN LÝ CẮT TỈA ALPHA-BETA

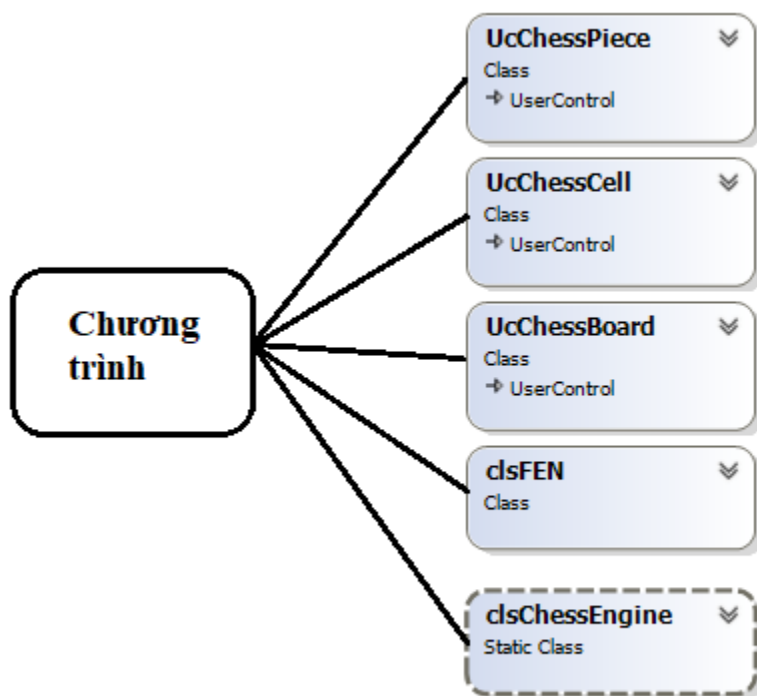
Trong chiến lược tìm kiếm minimax, để tìm kiếm nước đi tốt nhất cho Trắng tại trạng thái u , cho dù ta hạn chế không gian tìm kiếm trong phạm vi cây trò chơi gốc u với độ cao h , thì số đỉnh của cây trò chơi này cũng còn rất lớn $h > 3$. Chẳng hạn trong cờ vua, nhân tố nhánh trong cây trò chơi trung bình khoảng 35, thời gian đòi hỏi phải đưa ra nước đi là 150 giây, với thời gian này trên máy tính thông thường chương trình của bạn chỉ có thể xem xét các đỉnh có độ sâu 3 hoặc 4. Một người chơi thông thường có thể tính trước được 5,6 bước hoặc hơn nữa và do đó chương trình của bạn mới đạt trình độ người mới tập chơi. Khi đánh giá đỉnh u với độ sâu h , một thuật toán minimax đòi hỏi ta phải đánh giá tất cả các đỉnh của cây gốc u tới độ sâu h . Song ta có thể giảm bớt số đỉnh cần phải đánh giá mà không mà vẫn không ảnh hưởng gì đến sự đánh giá đỉnh u . Phương pháp cắt tỉa alpha-beta cho phép người ta cắt bỏ các nhánh không cần thiết cho sự đánh giá đỉnh u . Tư tưởng của kỹ thuật cắt tỉa alpha-beta là như sau : Nhớ lại rằng, chiến lược tìm kiếm minimax là chiến lược tìm kiếm theo độ sâu. Giả sử Trong quá trình tìm kiếm ta đi xuống đỉnh a là đỉnh trắng đỉnh a có người anh em v đã được đánh giá. Giả sử cha của đỉnh a là b và b có người anh em u đã được đánh giá, và giả sử cha của b là c (Xem hình 1). Khi đó ta có giá trị đỉnh c (đỉnh trắng) ít nhất là giá trị của u , giá trị của đỉnh b (đỉnh đen) nhiều nhất là giá trị của v . Do đó, nếu $eval(u) > eval(v)$, ta không cần đi xuống để đánh giá đỉnh a nữa mà vẫn không ảnh hưởng gì đến đánh giá đỉnh c . Hay nói cách khác ta có thể cắt bỏ cây con gốc a . Lập luận tương tự cho trường hợp a là đỉnh đen, trong trường hợp này nếu $eval(u) < eval(v)$ ta có thể cắt bỏ cây con gốc a . Để cài đặt kỹ thuật cắt tỉa alpha-beta, đối với các đỉnh nằm trên đường đi từ gốc tới đỉnh hiện thời, ta sử dụng tham số α để ghi lại giá trị lớn nhất trong các giá trị của các đỉnh con đã đánh giá của một đỉnh trắng, còn tham số β ghi lại giá trị nhỏ nhất trong các đỉnh con đã đánh giá của một đỉnh đen. Giá trị của α và β sẽ được cập nhật trong quá trình tìm kiếm. α và β được sử dụng như các biến địa phương trong các hàm $MaxVal(u, \alpha, \beta)$ (hàm xác định giá trị của đỉnh trắng u) và $MinVal(u, \alpha, \beta)$ (hàm xác định giá trị của đỉnh đen u). [



Hình 1

Phân tích chương trình

Cấu trúc chương trình



Chức năng và nhiệm vụ của các Class

UcChessPiece Xử lý các sự kiện tại một ô cờ

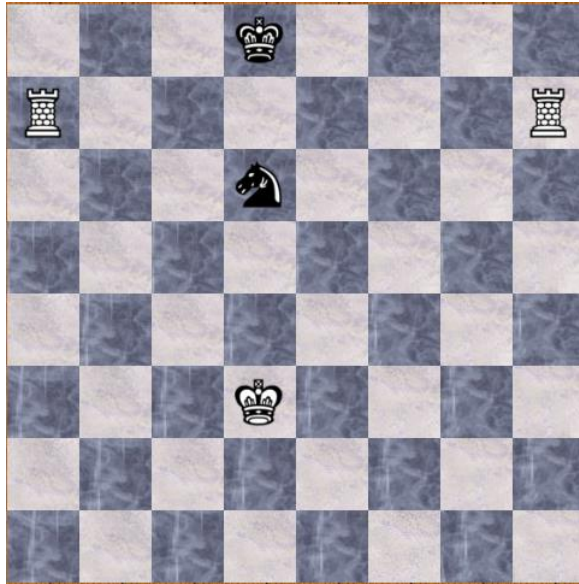
- Lấy loại cờ tại ô cờ `public ChessPieceType Type`
- Load ô cờ `UcChessPiece_Load`, `UcChessPiece_Paint`

- Các sự kiện click chuột UcChessPiece_MouseDown, UcChessPiece_MouseUp, UcChessPiece_MouseMove. Dùng để di chuyển con cờ, hiển thị các nước có thể di chuyển.

clsFEN Forsyth Kiem Edwards Notation (FEN) là một ký hiệu chuẩn để mô tả một vị trí bàn cờ cụ thể của một ván cờ. Mục đích của FEN là cung cấp tất cả các thông tin cần thiết để khởi động lại một trò chơi từ một vị trí cụ thể.

Một FEN gồm có 2 trường:

- Piece placement: dùng để xác định vị trí của các quân cờ trên bàn cờ
 pawn = "P", hiệp sĩ = "N", bishop = "B", rook = "R", queen = "Q" và vua = "K"
 chữ hoa biểu thị cho quân trắng và thường biểu thị cho quân đen
 Vd về FEN: 3k4/R6R/3n4/8/8/3K4/8/8 w - - có bàn cờ như dưới



- Active color: lượt đi hiện hành thuộc về bên nào white or black

clsChessEngine chứa các thao tác xử lý khi đánh cờ như

- `bool` CheckInsufficientMaterial: Kiểm tra trường hợp hòa cờ do cả 2 bên không đủ quân để chiếu bí đối phương
- `GameStatus` GetGameStatus: Kiểm tra Chiếu Bí, Hòa Cờ
- `bool` LegalMoveAvaiable: kiểm tra 1 phe có còn Nước Đi Hợp Lệ hay không
- chứa phần lõi AI sử dụng trong chế độ đánh giữa người và máy, hàm Successors(Hàm Tìm Tất Cả Các Nước Đi Hợp Lệ), ...
- Evalute(hàm đánh giá thế cờ).

```
private static int Evaluate(int[,] BoardState, ArrayList arrFEN) {...}
/*
 * Speelman considers that endgames are positions in which each player has 13
 * or fewer points in material (not counting the king)- wikipedia
 *
 * Khi giá trị quân cờ của mỗi đối thủ nhỏ hơn hoặc bằng 13(khoảng 1350 điểm)=>EndGame
 */
```
- Hàm kiểm tra kết thúc ván cờ `bool` IsEndGame

- ...

UcChessCell Khai báo các ô cờ trên bàn cờ

UcChessBoard

- Khởi tạo bàn cờ(từ FEN hoặc bàn cờ mới)
- Đặt các quân cờ lên bàn cờ. `void InitBoardState()`
- Tạo bàn cờ từ các quân cờ `void CreateChessBoard()`
- Các chức năng di chuyển quân cờ `void DoMove(UcChessPiece Piece, clsMove move)`
- Chức năng Undo
- ComputerMove

Cấu trúc quân cờ

Các class: `clsKing`, `clsBishop`, `clsRook`, `clsQueen`, `clsKinght`, `clsPawn`.

Có chung các đặc điểm như

- ValueTable bảng giá trị của một con cờ tại vị trí trên bàn cờ.
- Hàm `FindAllPossibleMove` trả về tất cả các nước di chuyển khả thi.
- Hàm `GetPositionValue` trả về giá trị của quân cờ tại một vị trí trên bàn cờ được lấy từ bảng `valueTable`.

Class **clsKing**

- Bảng giá trị `KingTableEndGame`
- Hàm xử lý nhập thành `AddCastlingPoint`
- Hàm kiểm tra quân vua của 1 phe có đang bị chiếu hay không
`bool IsChecked(int[,] arrState, ChessSide eSide)`
- Hàm Kiểm Tra 1 ô có bị khống chế hay không
`bool IsChecked(int[,] arrState, ChessSide eSide, Point pos)`

Trong class **clsPawn** có phương thức phong được cài đặt hàm `bool Promotion`

```

public static bool Promotion(UcChessPiece UcPawn, ChessPieceType PromoteTo)
{
    if ((UcPawn.PositionY == 8) || (UcPawn.PositionY == 1))
    {
        Bitmap queen = clsImageProcess.GetChessPieceBitMap(UcPawn.Side, ChessPieceType.Queen, UcPawn.Style);
        Bitmap root = clsImageProcess.GetChessPieceBitMap(UcPawn.Side, ChessPieceType.Rook, UcPawn.Style);
        Bitmap knight = clsImageProcess.GetChessPieceBitMap(UcPawn.Side, ChessPieceType.Knight, UcPawn.Style);
        Bitmap bishop = clsImageProcess.GetChessPieceBitMap(UcPawn.Side, ChessPieceType.Bishop, UcPawn.Style);

        if (PromoteTo == ChessPieceType.Null)
        {
            Form f = new frmPromotion(queen, root, knight, bishop);
            f.ShowDialog();
            UcPawn.Type = frmPromotion.Type;
        }
        else
        {
            UcPawn.Type = PromoteTo;
        }
        Bitmap image = clsImageProcess.GetChessPieceBitMap(UcPawn.Side, UcPawn.Type, UcPawn.Style);
        UcPawn.Image = image;
        return true;
    }
    return false;
}

```

Hàm Promotion kiểm tra quân cờ là tốt tại vị trí dòng thứ 8 hoặc dòng thứ 1

Tiếp tục kiểm tra ChessPieceType bằng null thì thực hiện mở form frmPromotion



Click để lựa chọn quân cờ cần phong trả về ChessPiece.Type

Nếu chơi với máy thì máy sẽ chọn quân hậu và không hiển thị form chọn.

Cấu trúc dữ liệu bàn cờ

Class UcChessBoard trong chương trình là class xử lý chính có khối lượng dữ liệu lớn và nhiều phương thức, gồm những phần chính sau :

- Phần dữ liệu: gồm một số thành phần cơ bản sau:
 - Một mảng hai chiều để lưu trạng thái bàn cờ

```

public int[,] _BoardState = new int[10, 10]; // Lưu trạng thái bàn cờ
/*
 * Qui ước:
 * + Cạnh bàn cờ: -1
 * + Không có quân cờ: 0
 * + Quân Tốt Đen: 11, Quân Tốt Trắng: 12
 * + Quân Tượng Đen: 21, Quân Tượng Trắng: 22
 * + Quân Mã Đen: 31, Quân Mã Trắng: 32
 * + Quân Xe Đen: 41, Quân Xe Trắng: 42
 * + Quân Hậu Đen: 51, Quân Hậu Trắng: 52
 * + Quân Vua Đen: 61, Quân Vua Trắng: 62
 */

```

- Một mảng hai chiều **UcChessCell [9,9]** để lưu các ô cờ
- **GameMode** chế độ chơi
- Mảng lưu trạng thái Stack **stkState**
- Mảng lưu FEN ArrayList **arrFEN**
- Mảng lưu các nước đi chuyển của quân trắng/đen: Stack **stkWhiteMoves/ Stack stkBlackMoves**
- Biến kiểm tra tới lượt trắng hay đen đi **bool _WhiteToMove**
- Biến kiểm tra quân sở hữu trắng/đen là người hay máy **ChessSide _OwnSide**
- Phần các phương thức trong bàn cờ: gồm một số phương thức cơ bản sau:
 - Constructor :
Chơi với người:

```

public UcChessBoard(ChessBoardStyle eBoardStyle, ChessPieceStyle ePieceStyle, ChessSide eOwnSide, GameMode eGameMode, int CellSize, int PieceSize, bool bPlaySound, string strFEN)

```

 Chơi với máy:

```

public UcChessBoard(ChessBoardStyle eBoardStyle, ChessPieceStyle ePieceStyle, ChessSide eOwnSide, GameMode eGameMode, GameDifficulty eDifficulty, int CellSize, int PieceSize, bool bPlaySound, string strFEN)

```
 - Các phương thức đặt các quân cờ lên bàn cờ

```

0 references
private void InitBoardState()...

2 references
private void AddChessPiece(ChessPieceStyle ePieceStyle, int[,] BoardState)
{
    //*****Add Tot*****
    //Quân Tốt Trắng nằm ở dòng 2
    //Quân Tốt Đen nằm ở dòng 7
    //*****
    if (this._PieceSize == 0)
    {
        this._PieceSize = this._CellSize;
    }

    for (int y = 1; y <= 8; y++)
        for (int x = 1; x <= 8; x++)
        {
            if (BoardState[x, y] > 0)
            {
                ChessPieceType eType = (ChessPieceType)(BoardState[x, y] / 10);
                ChessSide eSide = (ChessSide)(BoardState[x, y] % 10);
                arrChessCell[x, y].ChessPiece = new UcChessPiece(eSide, eType, ePieceStyle, this._CellSize, this._PieceSize, x, y, arrChessCell[x, y]);
            }
        }
}

```

- "Tạo bàn cờ từ các ô cờ"
 Góc dưới bên trái của bàn cờ vua là một ô đen
 Quân Hậu Trắng sẽ nằm trên ô Trắng và Quân Hậu Đen sẽ nằm trên ô đen

Bàn cờ được đánh tọa độ như sau :

Từ A->>I tính từ trái sang phải

Từ 1->8 tính từ dưới lên bắt đầu từ dòng đầu tiên của quân trắng

Nếu x+y là số lẻ thì đó là ô Trắng

Nếu x+y là số chẵn thì đó là ô Đen

- HighlightPossibleMoves/ UnHighlightMoves Tô màu(hủy tô) các nước có thể di chuyển của quân cờ

```
public void HighlightPossibleMoves()
{
    if (arrMove.Count == 0)
        return;

    foreach (Point p in arrMove)
    {
        UcChessCell cell = this.arrChessCell[p.X, p.Y];
        cell.HighlightPossibleMove();
    }

    this.DrawToBitmap bmpBackImage, new Rectangle(0, 0, this.Width, this.Height));
    Thread t = new Thread(new ThreadStart(SetBackGround));
    t.Start();
}

public void UnHighlightMoves()
{
    if (arrMove.Count == 0)
        return;
    foreach (Point p in arrMove)
    {
        UcChessCell cell = this.arrChessCell[p.X, p.Y];
        cell.UnHighlightMove();
    }
    this.HighlightLastMove();
}
```

- Hàm DoMove (UcChessPiece Piece, clsMove move)
Di chuyển quân cờ

```

public void DoMove(UcChessPiece Piece, clsMove move)
{
    if ((Piece.Side == ChessSide.Black && this._WhiteToMove == true) || (Piece.Side == ChessSide.White && this._WhiteToMove == false))
        return;

    Point CurPos = move.CurPos;
    Point NewPos = move.NewPos;

    UcChessCell ct = (UcChessCell)this.arrChessCell[NewPos.X, NewPos.Y];
    //*****Kiểm tra nước đi*****
    //Kiểm tra Xem Quân Cờ có thể di chuyển từ CurPos đến NewPos hay không
    if (clsChessEngine.CanMove(arrMove, NewPos) == true)
    {
        UnHighlightLastMove();
        PushState();
        string strLastCell = Convert.ToChar(CurPos.X + 64) + CurPos.Y.ToString();
        string strNewCell = Convert.ToChar(NewPos.X + 64) + NewPos.Y.ToString();

        this._BoardState[CurPos.X, CurPos.Y] = 0; //Đánh dấu vị trí cũ không chứa quân cờ
        this._BoardState[NewPos.X, NewPos.Y] = (int)Piece.Type * 10 + (int)Piece.Side;
        //Kiểm tra xem quân vua có bị chiếu không
        //*****Chưa tính trường hợp chiếu bí và Hòa*****

        bool bCapture = false;

        //Lấy vị trí Bắt tốt qua đường nếu có
        if (Piece.Type == ChessPieceType.Pawn) ...
        else ...
        //Nếu đã có quân cờ đặt tại ô đến
        if (ct.ChessPiece != null && ct.ChessPiece.Side != Piece.Side) ...

        Piece.UConCell.ChessPiece = null; //Ô cờ trước đó ko còn chứa quân cờ này nữa
        //Dat Quan Co Tai Vi Tri Moi
        ct.ChessPiece = Piece;
        Piece.UConCell = ct;
    }
}

```

Đầu tiên hàm kiểm tra có thể di chuyển tới vị trí newPos sau đó đặt quân cờ tại vị trí mới sau đó cập nhật trạng thái hiện hành vào bảng trạng thái quân cờ.

Thực hiện kiểm tra endGame nếu endGame!= null thì chuyển lượt đi.

- Phương thức Undo

Gồm 3 hàm xử lý chính

PushMove(ChessSide side, string strMove)

Push các bước di chuyển vào stkWhiteMoves/ stkBlackMoves. Được gọi ở hàm

DoMove

void PushState()

Push các trạng thái của bàn cờ vào stkState khi thực hiện DoMove

void UndoMove() gồm các bước xử lý chính sau đây:

Lần lượt kiểm tra 2 stack stkWhiteMoves và stkBlackMoves nếu khác null thì thực hiện lấy phần tử cuối cùng của hai mảng.

Sau đó xóa hai bước đó và vẽ lại bàn cờ nếu chơi với máy

Nếu chơi với người xét lượt đi hiện tại white/black nếu lượt đi hiện tại là white thì sẽ xóa lượt đi trước đó là black và ngược lại

Và vẽ lại bàn cờ

```

public void UndoMove()
{
    string strMoves = "";
    string strLastMove = "";
    if (stkBlackMoves.Count == 0 && this.GameMode == GameMode.VsComputer && this.OwnSide == ChessSide.Black)
        return;
    if (stkWhiteMoves.Count > 0 && this.WhiteToMove == false)
    {
        strMoves = stkWhiteMoves.Pop().ToString();

        UnHighlightLastMove();
        if (stkBlackMoves.Count > 0)
            strLastMove = stkBlackMoves.Peek().ToString();
    }
    if (stkBlackMoves.Count > 0 && this.WhiteToMove == true)
    {
        strMoves = stkBlackMoves.Pop().ToString();

        UnHighlightLastMove();

        if (stkWhiteMoves.Count > 0)
            strLastMove = stkWhiteMoves.Peek().ToString();
    }

    if (strLastMove != "")
    {

```

Phần các phương thức có trong AI

- Hàm `int Evaluate(int[,] BoardState, ArrayList arrFEN)` đánh giá thế cờ
- Hàm `int AlphaBeta(int[,] BoardState, ArrayList arrFEN, int depth, ChessSide eSide, int Alpha, int Beta)`
- Hàm Tạo Trạng Thái Bàn Cờ Từ 1 Nước Đi Hợp Lệ
`void TryMove(int[,] State, clsMove Move)`
- Successors (Hàm Tìm Tất Cả Các Nước Đi Hợp Lệ)
`ArrayList Successors(int[,] BoardState, ChessSide eSide)`
- Hàm sắp xếp các nước đi để tăng hiệu suất hàm alpha beta
`void Sort(ArrayList arrMoves, ChessSide eSide)`
- Hàm Tạo Nước Đi Hợp Lệ Ngẫu Nhiên
`clsMove RandomMove(int[,] BoardState, ChessSide eSide)`
- Hàm tạo nước đi
`ArrayList GenerateMove(int[,] BoardState, ArrayList arrFEN, ChessSide eSide, ref clsMove Move, GameDifficulty eDifficult)`

Một số phương thức khác

- Lưu lại bàn cờ đang chơi
 - Bàn cờ được lưu dưới định dạng xml gồm có các trường

```

<SaveFile>
  <GameMode>2</GameMode>
  <GameDifficulty>2</GameDifficulty>
  <TimeLimit>10</TimeLimit>
  <TimeBonus>5</TimeBonus>
  <mRemain1>10</mRemain1>
  <sRemain1>13</sRemain1>
  <mRemain2>10</mRemain2>
  <sRemain2>9</sRemain2>
  <OwnSide>2</OwnSide>
  <FEN>rn1qkbnr/pppb1ppp/3p4/4p3/2P1P1P1/8/PP1P1P1P/RNBQKBNR w KQkq e6 0 4</FEN>
  <MoveList>E2-E4 D7-D6 C2-C4 C8-D7 G2-G4 E7-E5</MoveList>
</SaveFile>

public void SaveToFile(string path)
{
    CreateNewSaveFile(path);
    DataTable tbl = clsXMLProcess.GetTable(path);
    DataRow r = tbl.Rows[0];
    r["GameMode"] = (int)this._GameMode;
    r["GameDifficulty"] = (int)this._Diff;
    r["TimeLimit"] = this._Limit;
    r["TimeBonus"] = this._Bonus;
    r["mRemain1"] = this._mRemain1;
    r["sRemain1"] = this._sRemain1;
    r["mRemain2"] = this._mRemain2;
    r["sRemain2"] = this._sRemain2;
    r["OwnSide"] = (int)this._OwnSide;
    r["FEN"] = this._FEN;
    r["MoveList"] = this._MoveList.Trim();

    tbl.WriteXml(path);
}

```

Các vấn đề về giải thuật

Lấy nước đi hợp lệ

Từ trạng thái hiện tại của bàn cờ, ta thực hiện việc lấy danh sách các nước đi hợp lệ của cả đội quân trắng hoặc đen bằng phương thức Successors trong phần AI.

Như đã trình bày, mỗi quân cờ có đều có phương thức FindAllPossibleMove, phương thức này trả về danh sách các nước đi hợp lệ của quân cờ tương ứng, và việc cài đặt phương thức này khác nhau đối với mỗi quân cờ khác nhau, phức tạp nên ta không trình bày tại đây.

Input: trạng thái bàn cờ

Output: danh sách các nước đi hợp lệ(arrMove).

Mã giả:

Bước 1: Tạo hàm Successors với kiểu trả về là danh sách với đối số Team cho biết các quân cờ cần xét thuộc quân trắng hay quân đen.

Bước 2: Tạo một ArrayList arrMoves để chứa các instance thuộc class clsMove.

Bước 3 : Với mỗi quân cờ trong mảng BoardState nếu là quân thuộc team eSide thì thêm arrMoves

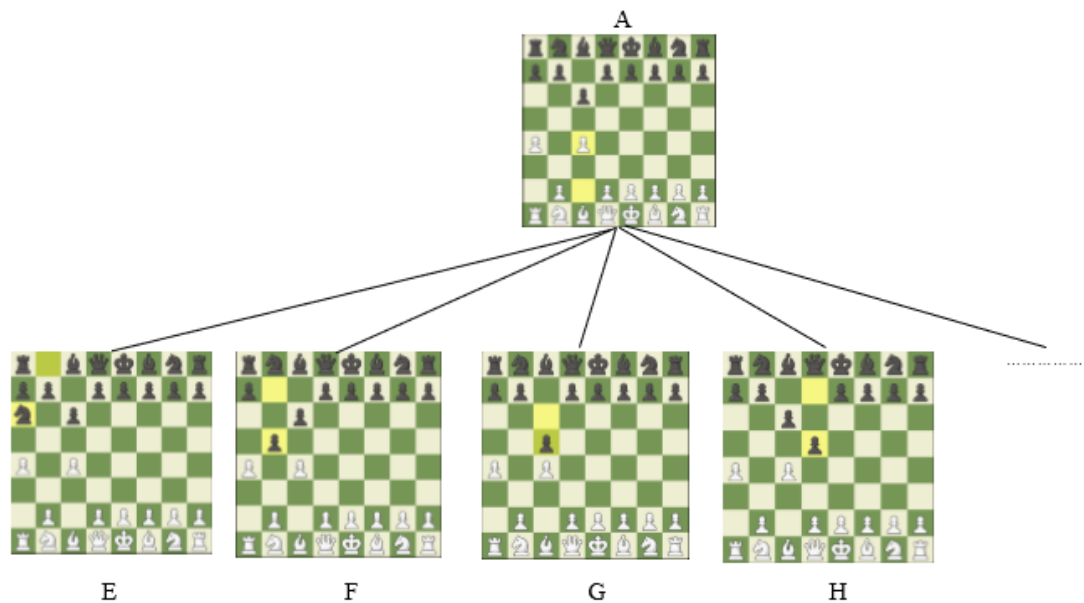
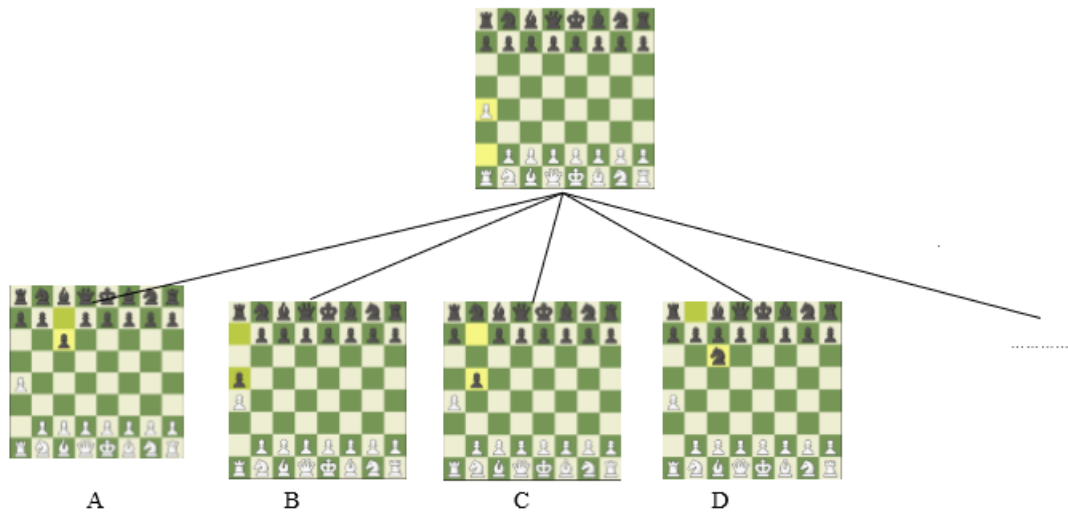
Bước 4: trả về giá trị arrMoves cho hàm Successors

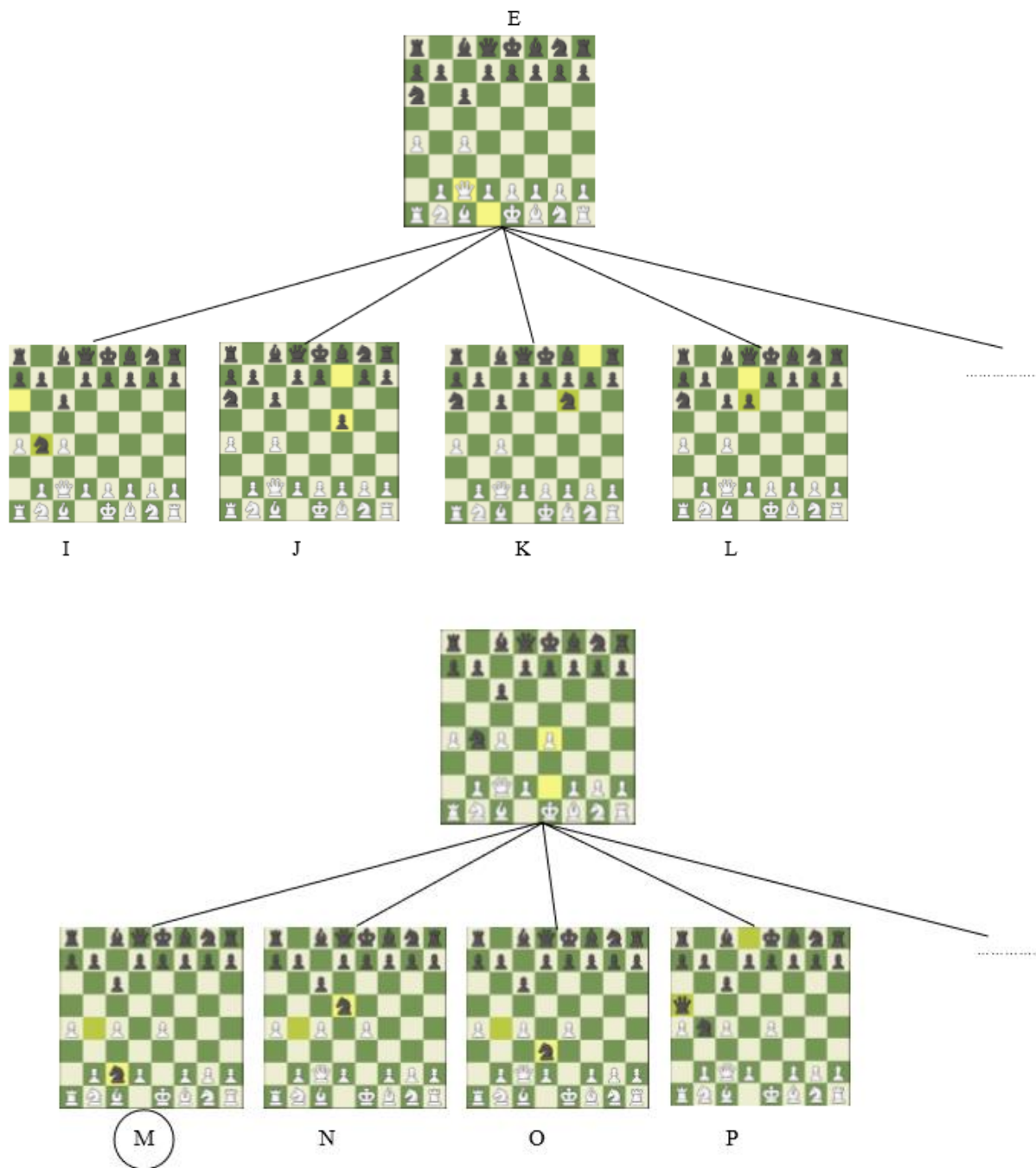
Đánh giá thế cờ

Với mỗi quân cờ ta gán một giá trị Heuristic ban đầu tương ứng với sự ảnh hưởng của quân cờ ấy đối với ván cờ tương ứng với hàm GetPositionValue () của mỗi quân cờ. Hàm đánh giá sẽ trả về một giá trị. Nếu giá trị càng lớn thì càng có lợi có Max và ngược lại tương ứng với Min. Ta chọn Max đại diện cho máy và Min đại diện cho người.

```
int Evaluate(int[,] BoardState, ArrayList arrFEN)
```

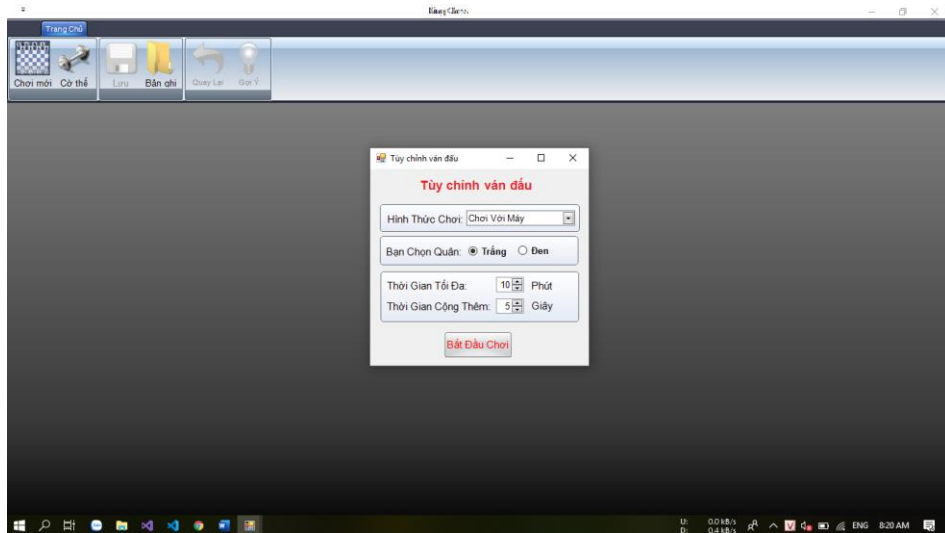
Minh họa các bước chạy của chương trình





Ứng dụng

Giao diện bắt đầu ván đấu

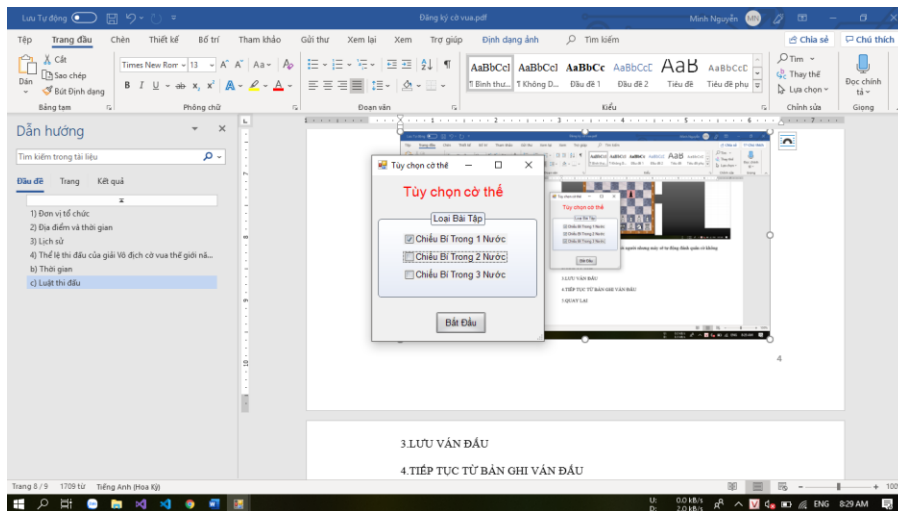


*diễn giải

- Hình thức chơi: Cho phép chơi với người và với máy
 - Chọn quân: Cho phép chọn quân trắng và quân đen
 - Thời gian tối đa/Thời gian thêm: Tùy chỉnh thời gian trận đấu
- => cho phép tạo ván đấu mới với người và máy

GIẢI CỜ THẾ

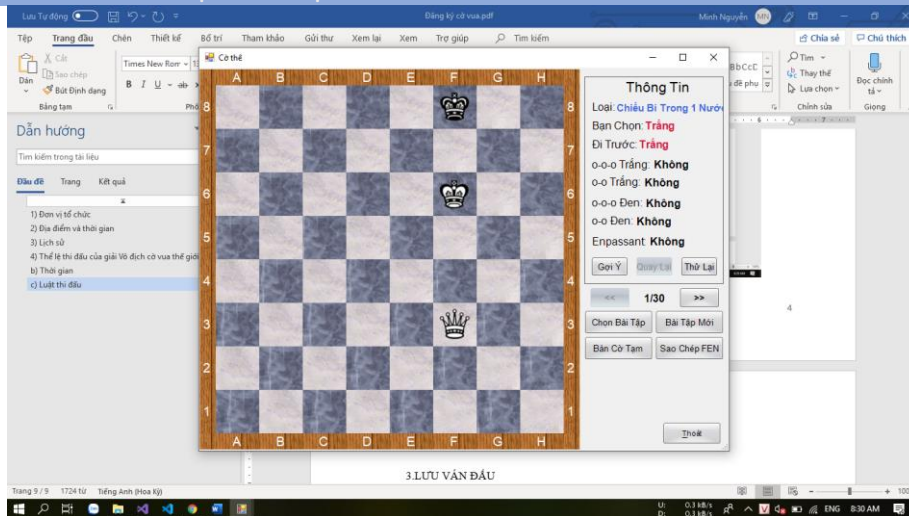
*Giao diện



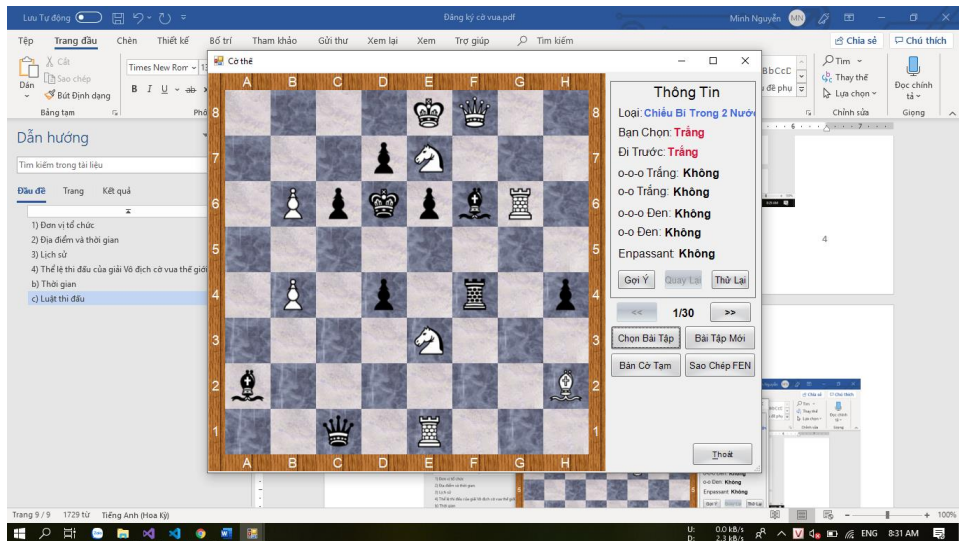
*Diễn giải:Cho phép chọn bài tập giải cờ thế từ kho lưu trữ có sẵn

a.Giải cờ thế 1 nước

BÁO CÁO ĐỒ ÁN TRÍ TUỆ NHÂN TẠO GAME CỜ VUA



b. Cờ thế 2 nước



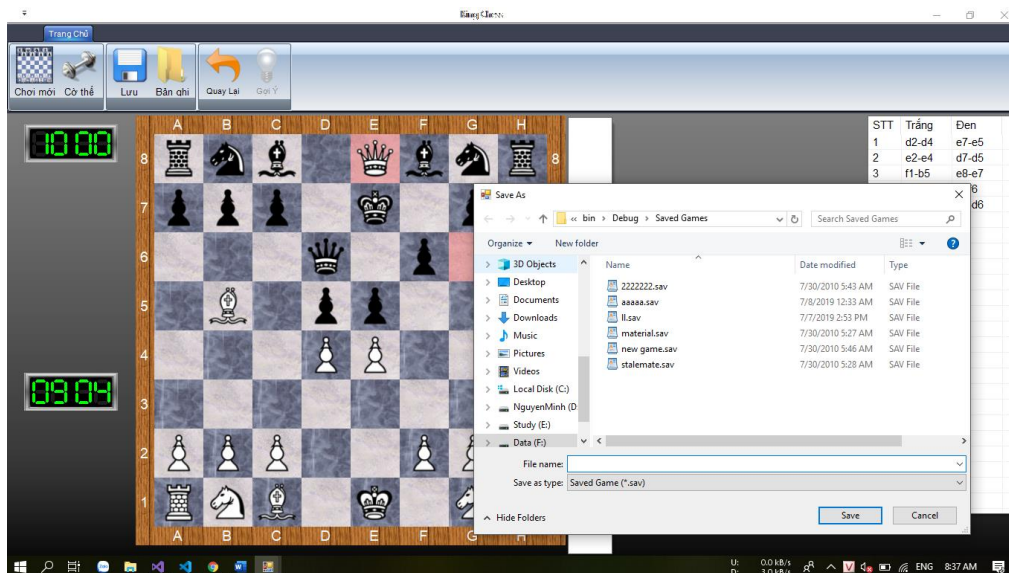
c. Cờ thế 3 nước

BÁO CÁO ĐỒ ÁN TRÍ TUỆ NHÂN TẠO GAME CỜ VUA



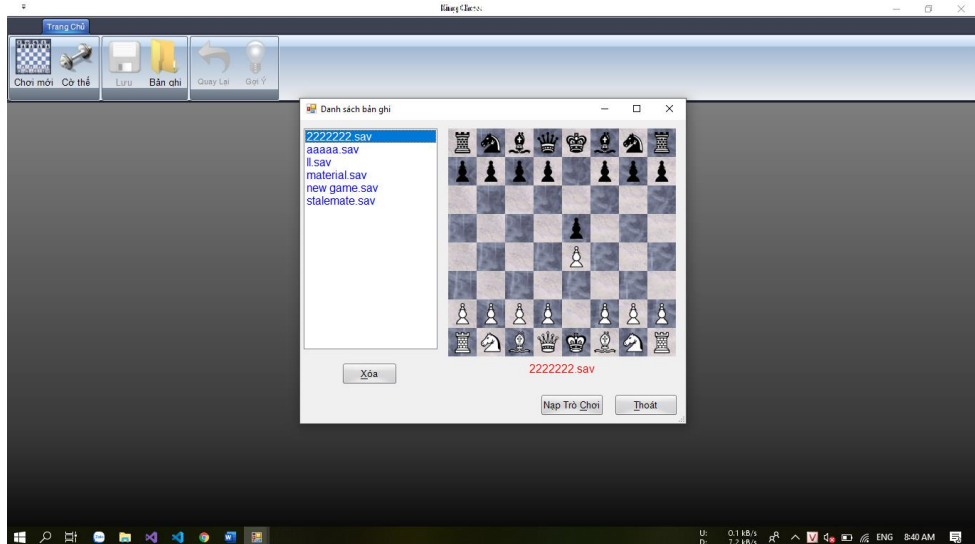
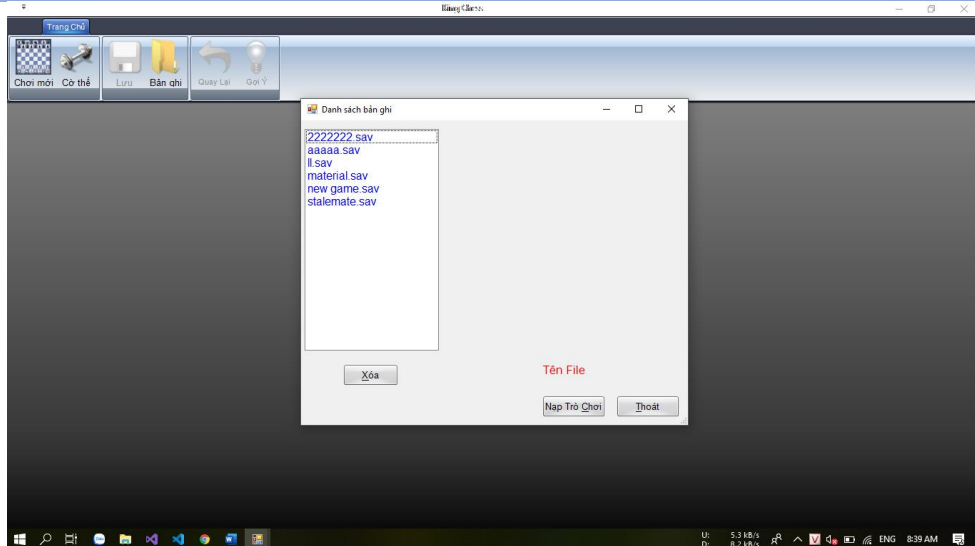
LƯU VÁN ĐẤU

-Khi đang thi đấu có thể lưu ván đấu



TIẾP TỤC TỪ BẢN GHI VÁN ĐẤU

-Cho phép tải ván đấu từ một bản lưu



Cài đặt

Ứng dụng được viết bằng ngôn ngữ C# visual studio 2015

Tài liệu tham khảo

http://www.gtryfon.demon.co.uk/bcc/Computer_chess/Graham/formcmp.htm

<https://www.ocf.berkeley.edu/~yosenl/extras/alphabeta/alphabeta.html>

HẾT