

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

ĐỒ ÁN MÔN HỌC

TRÍ TUỆ NHÂN TẠO

ÁP DỤNG TRÍ TUỆ NHÂN TẠO

TRONG TRÒ CHƠI CỜ VUA

(Đã chỉnh sửa theo yêu cầu giảng viên)



Lớp: CS106.H21.KHTN

GVHD: Huỳnh Thị Thanh Thương

NHÓM THỰC HIỆN:

Cao Quốc Đạt 15520097

Nguyễn Tuyết Nhi 15520585

Võ Hoài Nguyên 15520560

Hoàng Yến 15521042

Tp. Hồ Chí Minh, Tháng 6 năm 2017

Mục lục

LỜI MỞ ĐẦU.....	4
HỢP ĐỒNG NHÓM.....	5
CHƯƠNG 1	9
1.1 NGUỒN GỐC TRÒ CHƠI	9
1.1.1 Nguồn gốc, xuất xứ	9
1.1.2 Hình thành và phát triển	10
1.2 MÔ TẢ TRÒ CHƠI	11
1.2.1 Thể loại.....	11
1.2.2 Luật chơi.....	11
1.3 ỨNG DỤNG CỦA TRÒ CHƠI	15
CHƯƠNG 2	16
CƠ SỞ LÝ THUYẾT.....	16
2.1 THUẬT GIẢI HEURISTIC	16
2.1.1 Nguyên lý vét cạn thông minh	16
2.1.2 Nguyên lý tham lam	16
2.1.3 Nguyên lý thứ tự	16
2.1.4 Hàm Heuristic.....	16
2.2 CHIẾN LƯỢC TÌM KIẾM CÓ ĐỐI THỦ	17
2.2.1 Minimax là gì? Tại sao cần dùng minimax	17
2.2.2 Thủ tục minimax.....	17
2.2.3 Minimax với độ sâu cố định	19
2.2.4 Kỹ thuật cắt tỉa alpha-beta (α - β pruning)	20
CHƯƠNG 3	22
PHÂN TÍCH VÀ THIẾT KẾ.....	22
3.1 PHÂN TÍCH BÀI TOÁN	22
3.2 THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ BIỂU DIỄN TRẠNG THÁI	29
3.2.1 Sơ đồ lớp	29
3.2.2 Mô tả cơ sở dữ liệu dựa trên sơ đồ lớp.....	30
3.2 CÁC VẤN ĐỀ VỀ GIẢI THUẬT	37
3.3.1 Bài toán tổng quát.....	37

Đồ án môn Trí tuệ nhân tạo

3.3.2	Vấn đề 1: Phát sinh các nước đi hợp lệ.....	37
3.3.4	Vấn đề 3: Lựa chọn nước đi tốt nhất.....	47
CHƯƠNG 4 ỨNG DỤNG.....		50
4.1	GIỚI THIỆU CHƯƠNG TRÌNH ỨNG DỤNG	50
4.2	CÀI ĐẶT	55
4.3	KẾT QUẢ CHẠY CHƯƠNG TRÌNH	56
CHƯƠNG 5: KẾT LUẬN		59
5.1	KẾT QUẢ ĐẠT ĐƯỢC	59
5.2	HẠN CHẾ.....	59
5.3	HƯỚNG PHÁT TRIỂN	60

LỜI MỞ ĐẦU

Việc tạo ra những cỗ máy thông minh như con người là một ước mơ cháy bỏng của loài người từ hàng nghìn năm nay. Năng lực máy tính ngày càng mạnh mẽ là một điều kiện hết sức thuận lợi cho trí tuệ nhân tạo. Điều này cho phép những chương trình máy tính áp dụng các thuật giải trí tuệ nhân tạo có khả năng phản ứng nhanh và hiệu quả hơn trước. Sự kiện máy tính Deep Blue đánh bại kiện tướng cờ vua thế giới Casparov là một minh chứng rõ ràng nhất cho sự phát triển của trí tuệ nhân tạo. Mặc dù mục tiêu tối thượng của ngành trí tuệ nhân tạo là xây dựng nên những cỗ máy có năng lực tư duy như con người nhưng khả năng hiện tại của tất cả các sản phẩm trí tuệ nhân tạo vẫn còn hạn chế so với mục tiêu mà nhân loại đã đề ra. Và vì thế, ngành Khoa học máy tính vẫn đang tiến bộ mỗi ngày và chứng minh được sự nỗ lực hết mình của mình trong kỷ nguyên mới-kỷ nguyên của khoa học-kỹ thuật-công nghệ.

Để hoàn thành được đồ án chúng em xin trân trọng cảm ơn giảng viên môn Trí tuệ nhân tạo-cô Huỳnh Thị Thanh Thương đã tận tình giúp đỡ, hỗ trợ, hướng dẫn chúng em trong từng khâu thực hiện đồ án. Vì thời gian và năng lực còn có hạn chế nên không thể tránh khỏi những sai sót trong khi thực hiện đề tài nghiên cứu của mình. Rất mong được sự góp ý bổ sung của cô để đề tài của chúng em ngày càng hoàn thiện hơn. Chúng em xin chân thành cảm ơn !

HỢP ĐỒNG NHÓM

I. Tên nhóm: ***Game Cờ Vua (DYN)***

II. Thành viên:

Họ tên	Mã số sinh viên	Vai trò
Nguyễn Tuyết Nhi	15520585	Nhóm trưởng
Võ Hoài Nguyên	15520560	Thành viên
Cao Quốc Đạt	15520097	Thành viên
Hoàng Yến	15521042	Thành viên

III. Mục tiêu thành lập nhóm

- Cùng hoàn thành nhiệm vụ môn Trí tuệ nhân tạo.
- Cùng trau dồi, hoàn thiện các kỹ năng cần thiết.
- Tạo môi trường làm việc nhóm trong bậc Đại học tạo nền tảng cho công việc nhóm sau này trong công việc thực tế.
- Cùng tạo môi trường cho sinh viên học hỏi, trao đổi; là nơi dễ dàng giúp sinh viên thể hiện khả năng cũng như tư duy phản biện của mình.
- Tạo nên coi hội cho từng thành viên thể hiện khả năng riêng của mình.
- Làm khơi dậy tinh thần đoàn kết giữa các thành viên.

IV. Các quy tắc làm việc nhóm

Để hoàn thành tốt công việc, mỗi thành viên buộc PHẢI tôn trọng những qui định sau:

- Điều 1: Đúng giờ.
- Điều 2: Có trách nhiệm với nhóm, hoàn thành công việc được giao đúng thời hạn.
- Điều 3: Hạn chế sự không nghiêm túc trong quá trình làm việc.
- Điều 4: Tôn trọng và duy trì tình cảm với các thành viên trong nhóm.

Những điều một thành viên thuộc về nhóm KHÔNG ĐƯỢC làm

Đồ án môn Trí tuệ nhân tạo

- Điều 1: Trì hoãn việc của nhóm vì lý do không chính đáng.
- Điều 2: Gây mâu thuẫn, làm ảnh hưởng không tốt đến kết quả chung của nhóm.
- Điều 3: Không tôn trọng nhau trong quá trình làm việc.

Những điều một thành viên thuộc về nhóm NÊN làm (không bắt buộc)

- Điều 1: Cố gắng đoàn kết nhóm để đạt đến mục tiêu chung.
- Điều 2: Có ý thức tự giác, có trách nhiệm đối với công việc của nhóm.
- Điều 3: Biết quan tâm, lắng nghe và chia sẻ.

V. Công việc cụ thể

Thành viên	Công việc
Nguyễn Tuyết Nhi	<ul style="list-style-type: none">- Phân công công việc cho các bạn.- Xây dựng và code phần cấu trúc dữ liệu.- Tìm hiểu, xây dựng và code minimax có cắt tỉa alpha beta.- Cập nhật và liên kết code của tất cả các bạn.- Cài đặt hiệu ứng di chuyển cho các quân cờ.- Thực hiện button new game, quit game, và undo game.- Viết báo cáo chương 3 và 5.
Võ Hoài Nguyên	<ul style="list-style-type: none">- Tìm hiểu và code giao diện bàn cờ.- Tìm hiểu cách cài đặt các button.- Chỉnh sửa hình ảnh cho nhóm.- Viết báo cáo chương 1, 2, 3, 4, 5 và chỉnh sửa cuối cùng.- Chép file vào đĩa.
Cao Quốc Đạt	<ul style="list-style-type: none">- Thêm âm thanh cho bàn cờ.- Chỉnh sửa giao diện cho các button và background của trò chơi.- Tìm và code thuật giải Heuristic.- Tìm hiểu, tham khảo và xây dựng cấu trúc dữ liệu.- Tìm hiểu minimax và debug khi có lỗi xảy ra.- Viết báo cáo chương 3 phần heuristic.- Mua đĩa CD.
Hoàng Yến	<ul style="list-style-type: none">- Xứ lý phần nước đi hợp lệ.- Quay demo game.- Viết báo cáo chương 5.- In báo cáo

Đồ án môn Trí tuệ nhân tạo

VI.Đánh giá kết quả

Nguyễn Tuyết Nhi: 100%

Cao Quốc Đạt: 100%

Võ Hoài Nguyên: 90%

Hoàng Yến: 60%

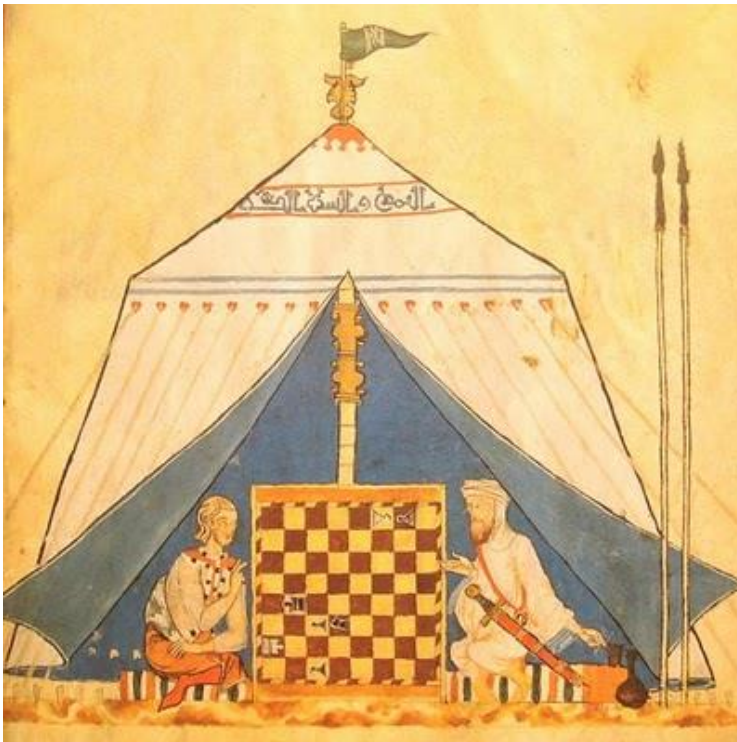
CHƯƠNG 1

GIỚI THIỆU BÀI TOÁN

1.1 NGUỒN GỐC TRÒ CHƠI

1.1.1 Nguồn gốc, xuất xứ

Có nhiều giả thuyết từ các quốc gia khác nhau trên thế giới cho rằng chính họ là người đã phát minh ra cờ vua ở một dạng tiền thân (biến thể) nào đó. Nhưng phổ biến nhất thì người ta tin rằng Ấn Độ mới là cha đẻ của môn cờ này. Giả thuyết đó cho rằng cờ vua ra đời vào khoảng thế kỷ thứ 6 sau Công nguyên. Vào thời đó, người ta gọi trò chơi này là Chaturanga (trò chơi với nhiều loại binh chủng xuất hiện trên bàn cờ gồm bộ binh, kỵ binh, tượng binh và xa binh). Một lý thuyết khác cho rằng cờ vua sinh ra từ trò chơi tương tự của cờ Trung Quốc, hoặc ít nhất là từ tổ tiên của cờ Tướng, là môn đã tồn tại ở Trung Quốc kể từ thế kỷ II trước công nguyên.^[1]



Hình 1.1 Nguồn gốc cờ vua

1.1.2 Hình thành và phát triển

Cờ vua sau đó được phổ biến về phía tây tới châu Âu và về phía đông tới Nhật Bản, sinh ra các biến thể trên đường đi của nó. Từ Ấn Độ nó đã tới Ba Tư, ở đây các thuật ngữ của nó được phiên âm sang tiếng Ba Tư và tên gọi của nó đổi thành Chatrang và những luật lệ đã bắt đầu phát triển cao hơn, đặc biệt trong số đó người ta gọi “Shāh!” khi tấn công vào Vua đối phương, và “Shāh māt!” khi quân vua đó bị tấn công và không có cách nào để thoát khỏi. Có một thuyết cho rằng việc thay đổi tên diễn ra bởi vì trước khi cờ vua tới châu Âu thì các nhà buôn đã tới châu Âu và mang theo các quân vua được trang trí như là các đồ vật hiếm và cùng với chúng là tên gọi shāh, tên gọi này đã bị người châu Âu phát âm sai theo nhiều cách khác nhau. Trò chơi này đã phổ biến trong thế giới Hồi giáo sau khi những người theo đạo Hồi xâm lược Ba Tư. Cờ vua đến Nga theo đường Mông Cổ mà tại đó người ta chơi cờ vua từ đầu thế kỷ 7. Nó đã được người Moor đưa vào Tây Ban Nha trong thế kỷ 10, và đã được miêu tả trong bản viết tay nổi tiếng thế kỷ 13 về cờ vua, cờ thỏ cáo và trò chơi xúc xắc có tên gọi Libro de los juegos. Cờ vua cũng đi theo đường bộ xuyên qua Siberi tới Alaska.^[1]





Ban đầu các quân cờ của người châu Âu có nhiều giới hạn về nước đi. Tượng chỉ có thể đi bằng cách nhảy chính xác qua 2 ô theo đường chéo, Hậu chỉ có thể di chuyển theo đường chéo là một ô, Tốt không thể di chuyển 2 ô trong nước đi đầu tiên của nó và không có nhập thành. Cuối thế kỷ 15, các quy tắc hiện đại đối với các nước đi cơ bản đã được chấp nhận từ Ý: quân Tốt có khả năng đi hai ô nếu đang ở vị trí xuất phát (nhảy) và khả năng bắt quân theo kiểu "bắt tốt qua đường". Tượng có được nước đi như ngày nay và Hậu đã trở thành quân mạnh nhất; do đó cờ vua hiện đại được nói đến như là "cờ của quân Hậu". Trò chơi này kể từ đó đã gần giống như cờ ngày nay. Các quy tắc hiện nay đã được hoàn thiện xong vào đầu thế kỷ 19, ngoại trừ các điều kiện chính xác là một cờ hòa.^[1]

1.2 MÔ TẢ TRÒ CHƠI

1.2.1 Thể loại

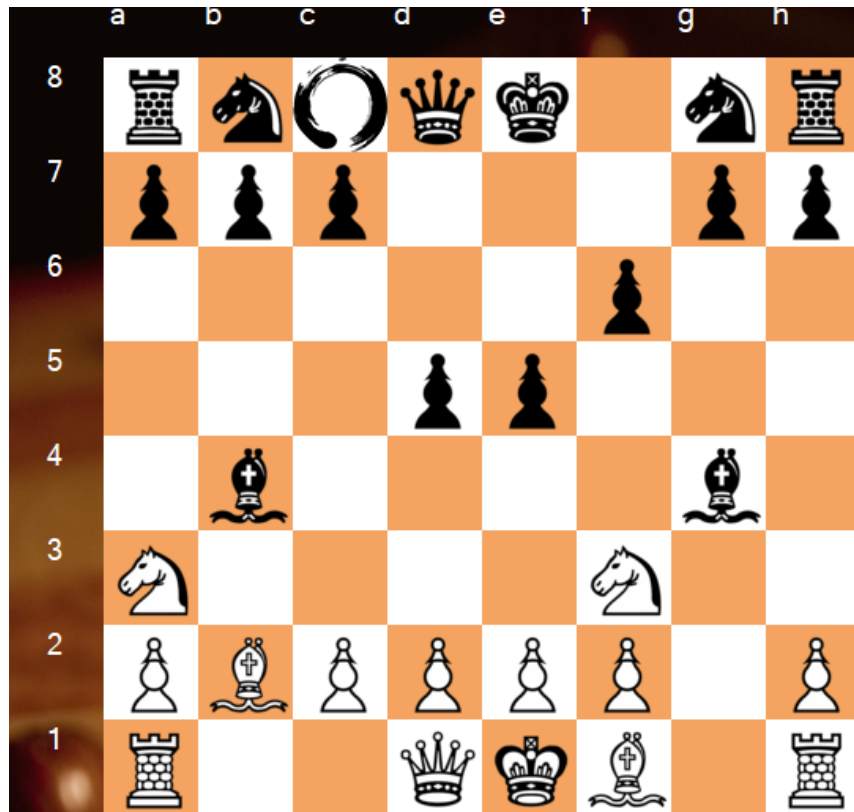
Cờ vua (cờ quốc tế) là một môn thể thao trí tuệ cho 2 người chơi. Trò chơi này diễn ra trên một bảng hình vuông, gọi là bàn cờ, gồm 8 hàng (đánh số từ 1 đến 8) và 8 cột (đánh các chữ cái từ a đến h), tạo ra 64 ô hình vuông với các màu đậm và nhạt xen kẽ nhau, với mỗi người chơi sẽ có ô màu nhạt ở hàng cuối cùng bên tay phải của mình khi ngồi vào bàn chơi cờ. Mỗi người sẽ bắt đầu ván cờ với 16 quân cờ và sẽ lần lượt đi các quân của mình sau khi đối phương đã đi xong một nước.^[1]

Các quân cờ của mỗi bên gồm:

Loại quân cờ		Tên quân cờ(kí hiệu quốc tế)	Số lượng(cho mỗi bên)
		King(K)	1
		Queen(Q)	1
		Castle(C)	2
		Knight(N)	2
		Bishop(B)	2
		Pawn(P)	8

Bảng 1.1 Các quân cờ trong cờ vua

1.2.2 Luật chơi



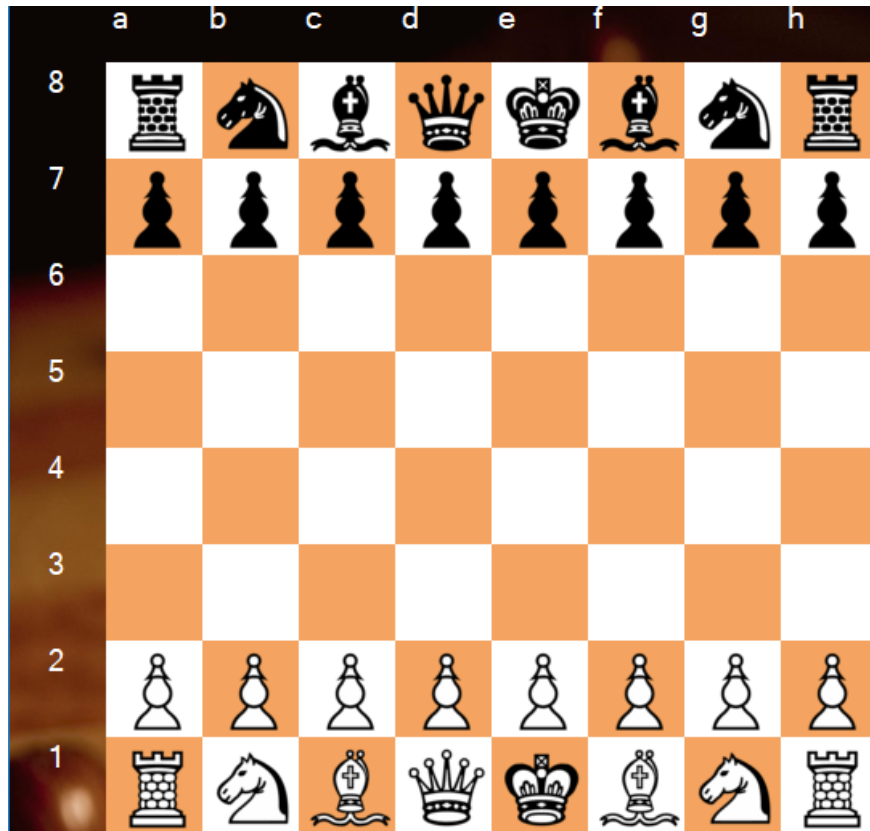
Hình 1.2 Một hình thức khai cuộc

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 một 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.^[1]

Chiến lược và chiến thuật: Khai cuộc là một loạt các nước đi lúc bắt đầu chơi, thông thường là được ghi nhớ sẵn, điều này giúp cho người chơi xây dựng các thế đứng và phát triển quân để chuẩn bị cho giai đoạn trung cuộc. Các thế khai cuộc thông thường được xây dựng trên nguyên tắc chiếm giữ phần trung tâm bàn cờ (gồm 4 ô trung tâm e4, e5, d4 và d5), phát triển quân, bảo vệ vua và tạo ra một cấu

trúc tốt đủ mạnh. Quan điểm của cờ vua hiện đại cho rằng việc kiểm soát trung tâm không chỉ là bằng các Tốt mà còn nhờ sức mạnh của các quân khác. Một cách rất quan trọng để bảo vệ Vua và triển khai nhanh quân Xe là nhập thành nhằm đưa Vua vào vị trí khó bị tấn công, tuy nhiên không phải trong bất kỳ ván cờ nào cũng cần nhập thành.^[1]

Các quân cờ có nước đi khác nhau.



Hình 1.3 Bàn cờ vua

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.^[1]

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. ^[1]

Xe (ký hiệu quốc tế C - Castle) 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. ^[1]

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.

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). ^[1]

Tốt (ký hiệu quốc tế P - Pawn) có thể di chuyển thẳng về phía trước chỉ một ô một lần tới ô còn trống (trừ nước đi đầu tiên Tốt được đi hai ô), nhưng khi di chuyển quân để ăn quân đối phương thì đi chéo. ^[1]

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).

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).

1.3 ỨNG DỤNG CỦA TRÒ CHƠI

Truyền thống của thi đấu cờ đã xuất hiện từ thế kỷ 16. Ngày nay, cờ vua được công nhận là một môn thể thao chính thức bởi Ủy ban Olympic Quốc tế. Nhà vô địch Cờ vua Thế giới đầu tiên, Wilhelm Steinitz đã giành danh hiệu này năm 1886. Hiện nay Magnus Carlsen là đương kim vua cờ.

Các nhà lý thuyết đã sáng tạo ra rất nhiều chiến thuật và chiến lược kể từ khi bắt đầu có cờ vua. Nhiều khía cạnh nghệ thuật cũng được tìm thấy trong cờ thế. ^[1]



Đã từng là trò chơi trí tuệ chỉ dành cho con người, ngày nay cờ vua được cả người lẫn máy tính chơi. Đầu tiên, việc máy tính chơi cờ chỉ là điều hiếm kỳ, nhưng hiện nay các chương trình cờ vua tốt nhất - như Shredder, Fritz v.v. - đã trở nên mạnh hơn con người, đặc biệt là trong cờ nhanh, kể cả khi nó được chạy trên các máy tính thông thường. Garry Kasparov, khi còn là số một thế giới về cờ vua, đã chơi một trận đấu 6 ván với máy tính chơi cờ của IBM có tên gọi là Deep Blue trong tháng 2 năm 1996. Deep Blue đã gây sốc cả thế giới khi thắng ván đầu tiên trong Deep Blue - Kasparov, 1996, Ván 1, nhưng Kasparov đã thắng trận khi thắng 3 và hòa 2 ván tiếp theo. ^[1]

Thất bại của Kasparov trước Deep Blue đã sinh ra một ý tưởng sáng tạo những biến thể cờ vua trong đó trí tuệ con người có thể vượt trội so với khả năng tính toán của máy tính và cố gắng của lập trình viên. ^[1]

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1 THUẬT GIẢI HEURISTIC

Thuật giải Heuristic là một sự mở rộng khái niệm thuật toán. Nó thể hiện cách giải bài toán với các đặc tính sau :

Đặc tính 1: Thường tìm được lời giải tốt (nhưng không chắc là lời giải tốt nhất).

Đặc tính 2: Giải bài toán theo thuật giải Heuristic thường dễ dàng và nhanh chóng đưa ra kết quả hơn so với giải thuật tối ưu, vì vậy chi phí thấp hơn.

Đặc tính 3: Thuật giải Heuristic thường thể hiện khá tự nhiên, gần gũi với cách suy nghĩ và hành động của con người.

Có nhiều phương pháp để xây dựng một thuật giải Heuristic, trong đó người ta thường dựa vào một số nguyên lý cơ sở như sau:

2.1.1 Nguyên lý vét cạn thông minh

Trong một bài toán tìm kiếm nào đó, khi không gian tìm kiếm lớn, ta thường tìm cách giới hạn lại không gian tìm kiếm hoặc thực hiện một kiểu dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.

2.1.2 Nguyên lý tham lam

Lấy tiêu chuẩn tối ưu (trên phạm vi toàn cục) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước (hay từng giai đoạn) trong quá trình tìm kiếm lời giải.

2.1.3 Nguyên lý thứ tự

Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt.

2.1.4 Hàm Heuristic

Trong việc xây dựng các thuật giải Heuristic, người ta thường dùng các hàm Heuristic. Đó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào trạng thái

hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này, ta có thể chọn được cách hành động tương đối hợp lý trong từng bước của thuật giải.

2.2 CHIẾN LƯỢC TÌM KIẾM CÓ ĐỐI THỦ

2.2.1 Minimax là gì? Tại sao cần dùng minimax

Minimax là giải thuật là một thuật toán đệ quy lựa chọn bước đi kế tiếp trong một trò chơi có hai người bằng cách định giá trị cho các Node trên cây trò chơi sau đó tìm Node có giá trị phù hợp để đi bước tiếp theo.

Về cơ bản sẽ có rất nhiều thuật toán đề ứng dụng được trí tuệ nhân tạo trong game như A, Heuristic,... Mỗi thuật toán sẽ phù hợp với từng loại game. Những game đối kháng trong đó người chơi luân phiên đánh như cờ vua, cờ tướng, caro... Khi chơi ta có thể khai triển hết không gian trạng thái nhưng khó khăn chủ yếu là phải tính toán được phản ứng và nước đi của đối thủ như thế nào? Cách xử lý đơn giản là giả sử đối thủ cũng sử dụng kiến thức về không gian trạng thái giống ta. Giải thuật Minimax áp dụng giả thuyết này để tìm kiếm không gian trạng thái của trò chơi. Trường hợp này thuật toán minimax đáp ứng những yêu cầu mà bài toán cần.^[2]

2.2.2 Thủ tục minimax^[2]

Hai đối thủ trong một trò chơi được gọi là MIN và MAX. MAX đại diện cho đối thủ quyết giành chiến thắng hay cố gắng tối đa hóa lợi thế của mình. Ngược lại MIN là đối thủ cố gắng tối thiểu hóa lợi thế của MAX.

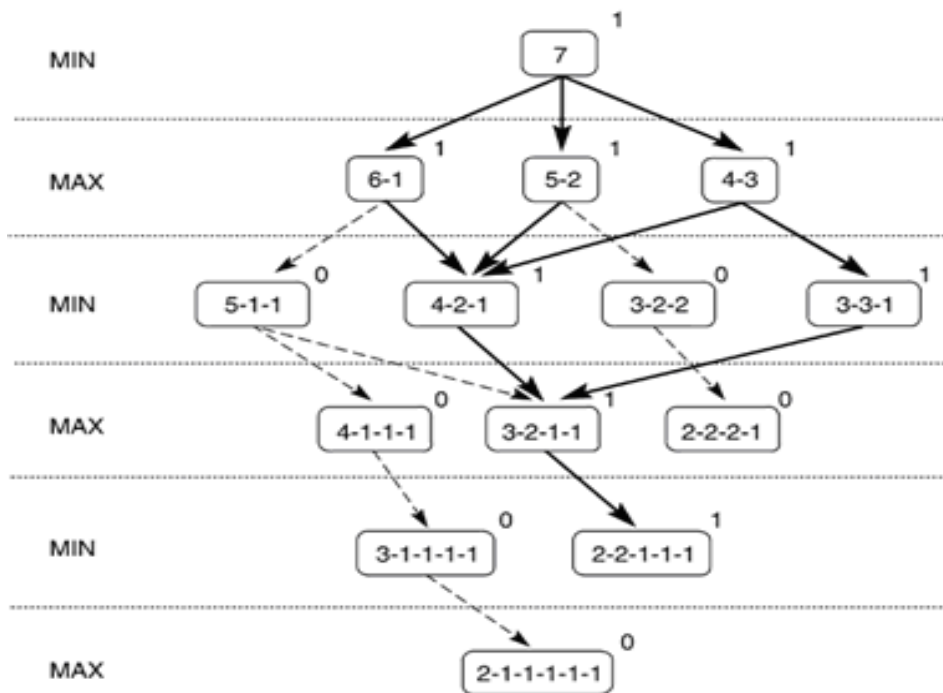
Khi áp dụng thủ tục Minimax, chúng ta đánh dấu luân phiên từng mức trong không gian tìm kiếm phù hợp với đối thủ có nước đi ở mức đó. Ví dụ, MIN được quyền đi trước, từng nút lá được gán giá trị 1 hay 0 tùy theo kết quả đó là thắng cuộc đối với MAX hay MIN. Minimax sẽ truyền các giá trị này lên cao dần trên đồ thị qua các nút cha mẹ kế tiếp nhau theo luật sau:

Nếu trạng thái cha mẹ là nút MAX, gán cho nó giá trị tối đa của các con cháu của nó.

Nếu trạng thái cha mẹ là nút MIN, gán cho nó giá trị tối thiểu của các con cháu của nó.

Xét ví dụ trò chơi Nim dưới đây, Để chơi Nim, một token được đặt giữa bàn đối thủ. Ở mỗi nước đi, người chơi phải chia đóng token thành hai đồng nhỏ có số lượng khác nhau. Ứng với một số token vừa phải, không gian trạng thái này có thể triển khai đến cùng.

Giá trị được gán cho từng trạng thái bằng cách đó sẽ chỉ rõ giá trị của trạng thái tốt nhất mà đối thủ này có thể hy vọng đạt được. Các giá trị này sẽ được dùng để lựa chọn các nước đi có thể có. Kết quả của việc áp dụng Minimax vào đồ thị không gian trạng thái đối với trò chơi được thể hiện như hình bên dưới. Vì tất cả các nước đi đầu tiên có thể xảy ra cho MIN sẽ dẫn đến các nút có giá trị 1 nên đối thủ MAX luôn có thể bắt trò chơi giành thắng lợi cho mình bất kể nước đi đầu tiên của MIN là như thế nào (đường đi thắng lợi của MAX được cho theo mũi tên đậm).



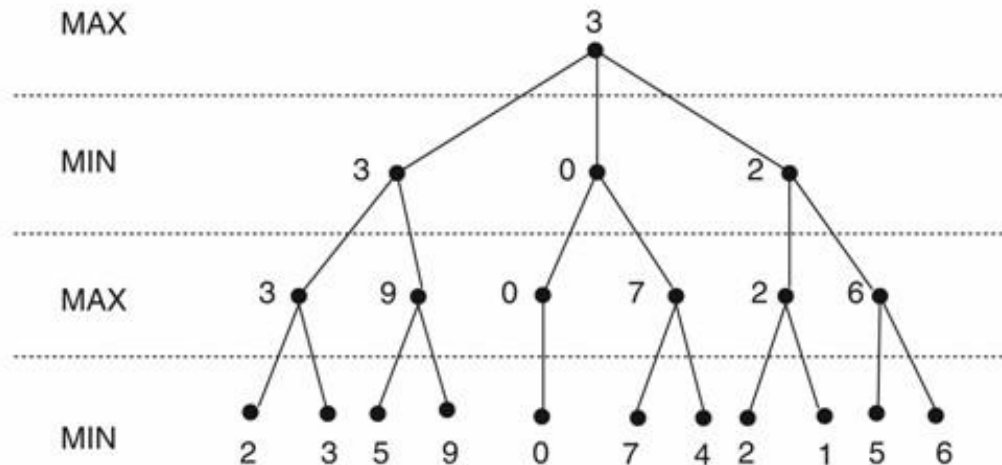
Hình 2.1 Minh họa không gian trạng thái trò chơi Nim

2.2.3 Minimax với độ sâu cố định^[5]

Khi áp dụng Minimax cho các trò chơi phức tạp, việc mở rộng đồ thị không gian trạng thái đến các nút lá tương đối khó khăn. Thay vào đó không gian trạng thái này chỉ có thể được triển khai đến một số mức xác định phụ thuộc tiềm năng về thời gian và bộ nhớ chẳng hạn. Chiến lược này được gọi là *tính trước n nước đi*. Vì giá trị các nút trong đồ thị con này không phải là trạng thái kết thúc của trò chơi nên chúng không phản ánh giá trị thắng cuộc hay thua cuộc. Chúng chỉ có thể được gán một giá trị phù hợp với một hàm đánh giá heuristic nào đó. Giá trị được truyền ngược về nút gốc không cung cấp thông tin thắng cuộc hay thua cuộc mà chỉ là giá trị heuristic của trạng thái tốt nhất có thể tiếp cận sau n nước đi kể từ nút xuất phát. Việc tính trước này sẽ làm tăng hiệu quả của heuristic vì nó được áp dụng vào một phạm vi lớn hơn trong không gian trạng thái.

Minimax sẽ hợp nhất tất cả các giá trị của các nút con cháu của một trạng thái thành một giá trị duy nhất cho trạng thái đó.

Trong các đồ thị trò chơi được tìm kiếm bằng mức hay lớp, MAX và MIN luân phiên nhau chọn các nước đi. Mỗi nước đi của một đối thủ sẽ xác định một lớp mới trên đồ thị. Các chương trình trò chơi nói chung đều dự tính trước một độ sâu lớp cố định (thường được xác định bằng các giới hạn về không gian hoặc thời gian của máy tính). Các trạng thái trên mức đó được đánh giá theo các heuristic và các giá trị này sẽ được truyền ngược lên bằng thủ tục Minimax, sau đó thuật toán tìm kiếm sẽ dùng các giá trị vừa nhận được để chọn lựa một nước trong số các nước đi kế tiếp. Bằng cách tối đa hóa cho các cha mẹ MAX và tối thiểu hóa cho các cha mẹ MIN, những giá trị này đi lùi theo đồ thị đến con của trạng thái hiện hành. Sau đó trạng thái hiện hành dùng chúng để tiến hành lựa chọn trong các con của nó. Hình sau trình bày quá trình Minimax trên một không gian trạng thái giả thuyết tính trước bốn lớp.



Hình 2.2 Minimax với một trạng thái giả định

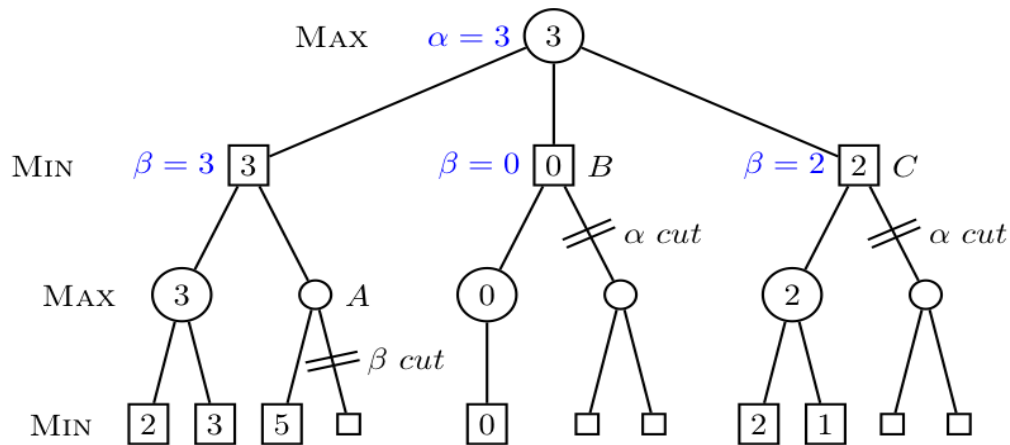
2.2.4 Kỹ thuật cắt tỉa alpha-beta (α - β pruning)^[5]

Minimax yêu cầu phải có sự phân tích qua hai bước đối với không gian tìm kiếm: Bước đầu truyền xuống đến độ sâu của lớp áp dụng Heuristic và bước sau để truyền ngược các giá trị lên cây. Minimax lần theo tất cả các nhánh trong không gian bao gồm cả những nhánh mà một thuật toán thông minh hơn có thể bỏ qua hay tĩa bớt. Các nhà nghiên cứu trong lĩnh vực chơi game đã xây dựng một kỹ thuật tìm kiếm gọi là cắt tỉa α - β nhằm nâng cao hiệu quả tìm kiếm trong các bài toán có tính chất đối kháng.

Kỹ thuật cắt tỉa alpha-beta được hiểu một cách đơn giản là “cắt giảm những thứ không cần thiết”. Ý tưởng của kỹ thuật trên như sau: Thay vì nếu như tìm kiếm toàn bộ không gian đến một độ sâu lớp cố định, tìm kiếm α - β thực hiện theo kiểu tìm kiếm sâu. Có hai giá trị là α và β được tạo ra trong quá trình tìm kiếm. Giá trị α đại diện cho các nút MAX, có khuynh hướng không bao giờ giảm. Ngược lại, các giá trị β đại diện cho các nút MIN, có khuynh hướng không bao giờ tăng.

Kỹ thuật cắt tỉa α - β được tiến hành như sau: Ta đi xuống hết độ sâu của lớp theo cơ chế tìm kiếm sâu, đồng thời áp dụng hàm đánh giá heuristic cho một trạng thái và tất cả các trạng thái “anh em” với nó. Ta có quá trình cắt tỉa β như sau: Giả thuyết tất cả các nút đều là nút MIN. Giá trị tối đa của các nút MIN này sẽ được

truyền ngược lên cho nút “cha” (là một nút MAX). Sau đó giá trị này được gán cho nút “ông” của các nút MIN và gọi đây là giá trị β . Tiếp theo, thuật toán này sẽ đi xuống các nút khác và kết thúc việc tìm kiếm đối với các nút “cha” của chúng nếu gặp bất kỳ một giá trị nào lớn hơn hoặc bằng giá trị β này. Các làm tương tự cũng được thực hiện cho việc cắt tia α đối với các nút “cháu” của một nút MAX.



Hình 2.3 Minh họa cắt tia α - β

CHƯƠNG 3

PHÂN TÍCH VÀ THIẾT KẾ

3.1 PHÂN TÍCH BÀI TOÁN

Chơi cờ là việc luân phiên thay đổi vị trí các quân cờ sao cho đến khi thắng (hoặc thua hoặc hòa). Vậy bản chất của trò chơi cờ vua là đi tìm trạng thái làm cho người chơi thắng. Thế nên ta sẽ giải quyết trò chơi bằng cách đưa trò chơi về bài toán tìm kiếm. Để giải quyết bài toán tìm kiếm trước hết ta cần xây dựng **không gian trạng thái(1)**. Mỗi một bước đi của quân cờ sẽ sinh ra trạng thái mới, việc di chuyển của các quân cờ cũng chính là các **toán tử(2)** làm biến đổi trạng thái. Toán tử ở đây chính là đi tìm tất cả những bước đi hợp lệ của các quân cờ cùng màu trên bàn cờ. Nghĩa là ta có thể chia nhỏ bài toán tìm toán tử của trạng thái thành tìm toán tử của từng loại quân cờ độc lập trên bàn cờ. Vậy ta cần xây dựng **đối tượng các quân cờ(3)** để dễ dàng quản lý và thực hiện phương thức tìm nước đi hợp lý. Rõ ràng ta biết cờ vua là trò chơi có tính đối kháng, mỗi người đều có lượt đi riêng, cả hai đều biết mọi thông tin về tình thế trò chơi. Nên ta áp dụng thuật toán **minimax** và để tối ưu thời gian tìm kiếm ta sẽ dùng **minimax có cắt tỉa alpha và beta(4)**. Vấn đề tiếp theo cần quan tâm là làm sao chúng ta có thể biết được trạng thái nào có lợi cho người đi, trạng thái nào có lợi cho máy đi để máy có thể lựa chọn được nước đi tốt nhất. Để giải quyết vấn đề đó, chúng ta sẽ xây dựng **hàm đánh giá heuristic(5)**. Tới đây, ta cần nghĩ ra một cách thức nào đó để lưu giữ các trạng thái, và giải pháp là **cây trò chơi(6)**. Các trạng thái khác nhau của bàn cờ trong quá trình chơi có thể biểu diễn thành một cây tìm kiếm, ta sẽ tiến hành tìm kiếm trên cây để được nước đi tốt nhất. Ở cây này, các nút là các tình huống khác nhau của bàn cờ, các nhánh nối sẽ cho ta biết từ một tình huống cờ này chuyển sang tình huống cờ khác thông qua một nước đi nào đó. Các nước đi này diễn ra theo cặp do hai đấu thủ lần lượt tiến hành. Độ sâu của cây trò chơi là số tầng của cây.

Đồ án môn Trí tuệ nhân tạo

(1) Không gian trạng thái(KGTT)

KGTT được biểu diễn bằng một mảng 2 chiều kiểu int, mỗi giá trị trong mảng là giá trị của từng loại quân cờ. Ngoài ra, mỗi trạng thái còn có một giá trị biểu diễn độ tốt của bàn cờ.

R \ C	0	1	2	3	4	5	6	7
0	1	2	3	4	5	3	2	1
1	6	6	6	6	6	6	6	6
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	12	12	12	12	12	12	12	12
7	7	8	9	10	11	9	8	7

Bảng 3.1: Trạng thái khai cờ

Các giá trị tương ứng của loại cờ

1: XE đen	7: XE trắng
2: MÃ đen	8: MÃ trắng
3: TƯỢNG đen	9: TƯỢNG trắng
4: HẬU đen	10: HẬU trắng
5: VUA đen	11: VUA trắng

Đồ án môn Trí tuệ nhân tạo

6: TỐT đen	12: TỐT trắng
------------	---------------

Trạng thái đầu là trạng thái khai cờ, trạng thái kết thúc là trạng thái một trong hai vua bị đối thủ chiếu bí. Ta quy ước, người chơi quân trắng, máy chơi quân đen,

R \ C	0	1	2	3	4	5	6	7
0	5	1	0	0	0	0	0	0
1	2	6	0	0	0	0	0	0
2	6	8	6	0	0	0	0	0
3	0	3	6	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	11	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Bảng 3.2: Trạng thái kết thúc (vua đen bị chiếu bởi mã trắng).

Vậy với không gian trạng thái được biểu diễn như trên, bây giờ ta chỉ cần làm việc trên mảng 2 chiều kiểu int. Từ trạng thái ban đầu, dùng các toán tử, càng nhanh càng tốt biến đổi về trạng thái vua trắng bị chiếu bí (trạng thái kết thúc) để giành được chiến thắng trong trò chơi.

(2) Toán tử

Toán tử không gian trạng thái là một tập các bước đi hợp lệ của các loại quân cờ cùng màu trên bàn cờ. Vậy ta sẽ chia nhỏ bài toán từ việc đi tìm toán tử của bàn cờ thành đi

Đồ án môn Trí tuệ nhân tạo

tìm toán tử của từng loại quân cờ trên bàn cờ. Ta dễ thấy một trạng thái có thể sinh ra rất nhiều trạng thái con sau một lần biến đổi trạng thái (dùng toán tử).

- Ta xét ví dụ minh họa sau đây:

+ Dưới là trạng thái ban đầu ta khởi tạo

<div>R \ C</div>	0	1	2	3	4	5	6	7
0	1	2	3	4	5	3	2	1
1	6	6	6	6	6	6	6	6
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	12	12	12	12	12	12	12	12
7	7	8	9	10	11	9	8	7

- Để tìm các trạng thái mới của bàn cờ, ta cần tìm tất cả những bước đi hợp lệ của tổng số cờ đen đang nằm trên bàn cờ.

+ 1 (xe đen) ở vị trí (0, 0) không có bước đi hợp lệ nào.

+ 2 (mã đen) ở vị trí (1, 0) có 2 bước đi hợp lệ là (0, 2) và (2, 2).

+ 3 (tượng đen) ở vị trí (2, 0) không có bước đi hợp lệ.

....

+ 6 (tốt đen) ở vị trí (0, 1) có 2 bước đi hợp lệ là (0, 2) & (0, 3).

Đồ án môn Trí tuệ nhân tạo

Vậy trạng thái trên sinh ra được tổng cộng 20 trạng thái con mới sau một lần thực hiện toán tử.

Vấn đề bây giờ là ta cần xây dựng lớp quân cờ để thực hiện việc di chuyển trên.

(3) Đối tượng các quân cờ

Xây dựng lớp quân cờ là lớp ảo, là lớp được kế thừa từ các lớp từ các lớp xe, mã, tượng, hậu, vua.

Trong lớp sẽ có thuộc tính màu, tọa độ x, y và phương thức tìm tất cả những bước đi thích hợp của quân cờ, và những bước đi đó sẽ được lưu vào một mảng cũng là thuộc tính của quân cờ.

(4) Cây trò chơi

Trọng trò chơi có tính đối kháng như cờ vua, người chơi phải biết phân tích ván cờ, xem xét nước đi nào sẽ mang lợi lại thế cho mình và bất lợi cho đối thủ. Phải đặt giả thiết liệu đi quân cờ như thế này thì đối phương sẽ đi tiếp theo như thế nào. Vậy nếu bên nào nhìn được trước càng nhiều bước thì bên đó sẽ có nhiều cơ hội thắng hơn. Vì thế nên ta cần phải xây dựng cây trò chơi.

Cây trò chơi được xây dựng như sau:

+ Nếu A là trạng thái bất kì nào đó, và đến lượt máy chơi. Áp dụng toán tử cho tất cả quân cờ đen trên bàn cờ, ta sinh ra được các trạng thái B, C, D... Từ B, C, D áp dụng toán tử cho tất cả quân trắng trên bàn cờ, ta lại sinh ra được trạng thái E, F, G... Lặp lại tương tự chu kì như trên cho tới bằng độ sâu mà ta mong muốn.

+ Nếu sinh ra nhiều trạng thái như vậy, máy sẽ biết lấy trạng thái thế nào, lúc này ta cần hàm heuristic để đánh giá độ tốt xấu của bàn cờ.

(5) Hàm đánh giá heuristic

Để đánh giá được độ tốt của một quân cờ ta dựa vào các kinh nghiệm chơi cờ thực tế.

Theo Tomasz Michniewski, để định giá một ván cờ, ta có hai phần: Độ tốt của quân cờ và độ tốt tại vị trí tương ứng của mỗi quân cờ.

Để định giá cho độ tốt của một quân cờ, ta có các kiến thức sau từ những kinh nghiệm chơi cờ thực tế:

1. Tránh trao đổi một Bishop hoặc một Knight với ba Pawn.
Trong một vài tình huống đặc biệt, nhờ vào một vị trí mà Pawn sẽ có độ tốt tốt lớn hơn Bishop hoặc Knight, nhưng thông thường Bishop hoặc Knight sẽ có độ tốt tốt hơn Pawn. Nên ta có bất phương trình:
$$\text{Bishop} > 3\text{Pawn} \text{ và } \text{Knight} > 3\text{Pawn}.$$
2. Khuyến khích hai Bishop cùng một đội luôn đi cùng với với nhau.
Khi chơi cờ, nếu dùng đôi Bishop hoặc đôi Knight hỗ trợ nhau sẽ rất có lợi cho việc phòng thủ và tấn công, nhưng ta thấy được rằng, Bishop có nhiều nước đi hơn Knight vì thế độ tốt của Bishop thường sẽ nhỉnh hơn Knight, nên ta có:
$$\text{Bishop} > \text{Knight}.$$
3. Tránh trao đổi Bishop và Knight với Rook và Pawn của đối phương.
Theo kinh nghiệm thực tế, việc trao đổi một Bishop và một Knight với một Rook và một Pawn của đối phương vì nếu giữ lại một đôi Bishop hoặc một đôi Knight hoặc giữ lại cả hai đôi Bishop và Knight thì chiến lược tấn công và phòng thủ sẽ mạnh hơn đối phương, nếu một bên mất đi một Bishop và một Knight thì việc duy trì đôi Bishop hoặc đôi Knight thất bại dẫn đến việc các quân cờ không còn phối hợp chặt chẽ với nhau sẽ dẫn đến việc phòng thủ và tấn công kém.
Vì vậy ta có:
$$\text{Bishop} + \text{Knight} > \text{Rook} + \text{Pawn}.$$
4. Tránh đổi Queen với Bishop và Rook.
Ta thấy, Queen có thể 2 kiểu đi chéo và đi thẳng và có thể thay đổi cách đi tùy theo tình hình ván cờ nên Queen có tính linh hoạt hơn Bishop và Knight, nên ta có:
$$\text{Queen} = \text{Rook} + \text{Bishop}.$$
 Và ta cũng có:
$$\text{Queen} + \text{Pawn} > \text{Rook} + \text{Bishop}.$$

Đồ án môn Trí tuệ nhân tạo

5. King sẽ lấy giá trị nào đó rất lớn để thể hiện sự kết thúc ván cờ nếu king của một trong hai bên mất trước.

Cộng (4) và (3) ta được: Knight + Queen > 2Rook.

Lấy độ tốt của Pawn là 1. Vậy giá trị tối thiểu ta có thể gán cho độ tốt của các con cờ có giá trị như bảng bên dưới:

Pawn= 1.	Rook=5.	Bishop=3.	Knight=3.	Queen= 7.	King= 200.
-----------------	----------------	------------------	------------------	------------------	-------------------

(4) Chiến thuật minimax có cắt tỉa alpha beta

Ta xây dựng minimax dựa trên 2 hàm MAX và MIN mục đích của max là cố gắng làm tối ưu lợi thế của mình, và min cố gắng làm tối thiểu lợi thế của đối phương.

Trước hết ta cần xây dựng những hàm cần thiết để hỗ trợ minimax:

+ Hàm kiểm tra trạng thái kết thúc():

. Input: trạng thái hiện tại của trò chơi

. Output: true hoặc false

+ Hàm heuristic():

. Input: trạng thái hiện tại của bàn cờ và màu của quân cờ

. Output: 1 số kiểu double

+ Hàm sinh con (sinh trạng thái mới):

. Input: trạng thái hiện tại của bàn cờ

. Output: tập hợp những trạng thái kế tiếp sau khi di chuyển tất cả các bước đi có thể của con cờ trên bàn cờ

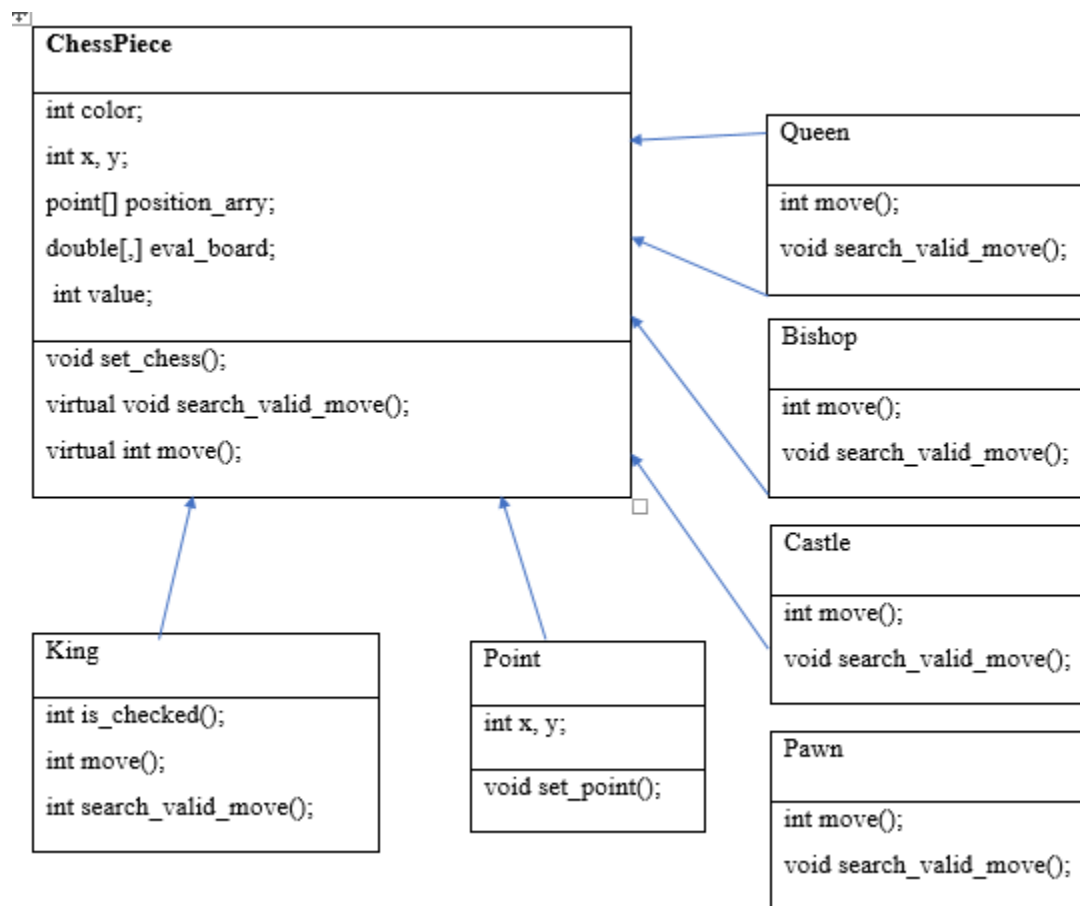
+ Giá trị đánh giá độ tốt max và min: alpha và beta

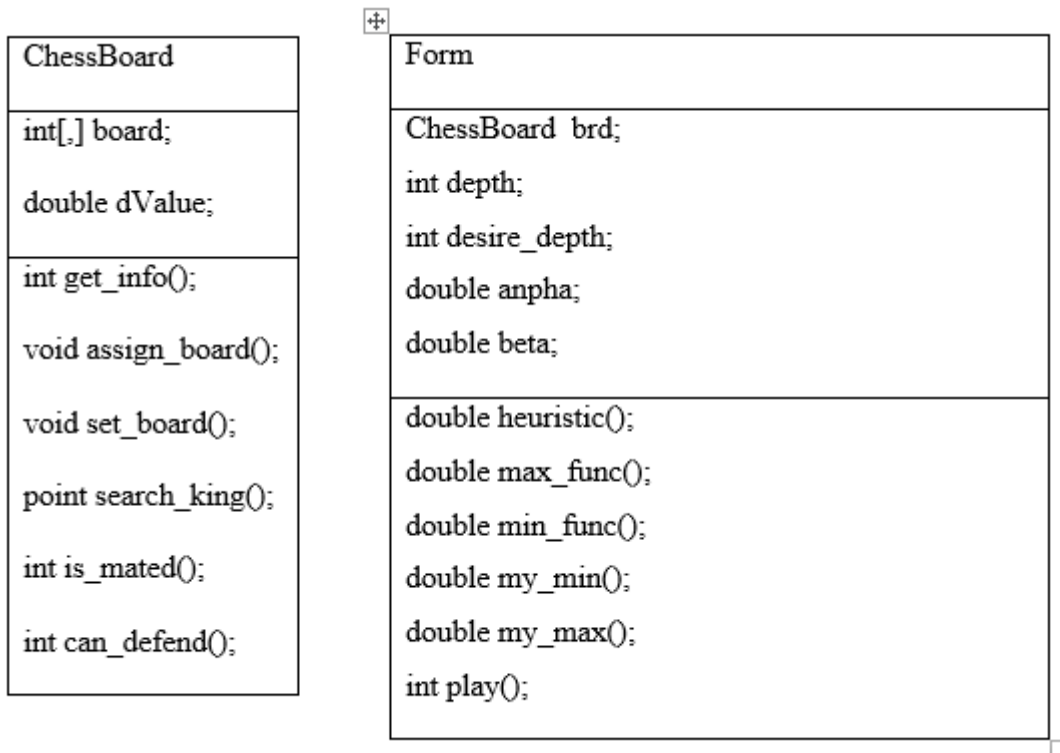
Ban đầu khởi tạo: alpha = -vô cùng, beta = +vô cùng. Alpha sẽ lưu giá trị độ tốt max của cây trò chơi. Beta sẽ lưu giá trị độ tốt min của cây trò chơi.

- Hai hàm Max và Min có input là trạng thái bàn cờ, độ sâu, alpha, beta. Và output là 1 số kiểu double.
- Vậy ta lấy gì để biết được trạng thái tốt nhất được sinh ra từ trạng thái hiện tại sau khi chạy hàm minimax. Ta dùng 1 biến toàn cục để lưu giữ trạng thái này, mỗi lần có sự thay đổi trong hàm minimax, trạng thái này sẽ được cập nhật.

3.2 THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ BIỂU DIỄN TRẠNG THÁI

3.2.1 Sơ đồ lớp





3.2.2 Mô tả cơ sở dữ liệu dựa trên sơ đồ lớp

1. ChessPiece

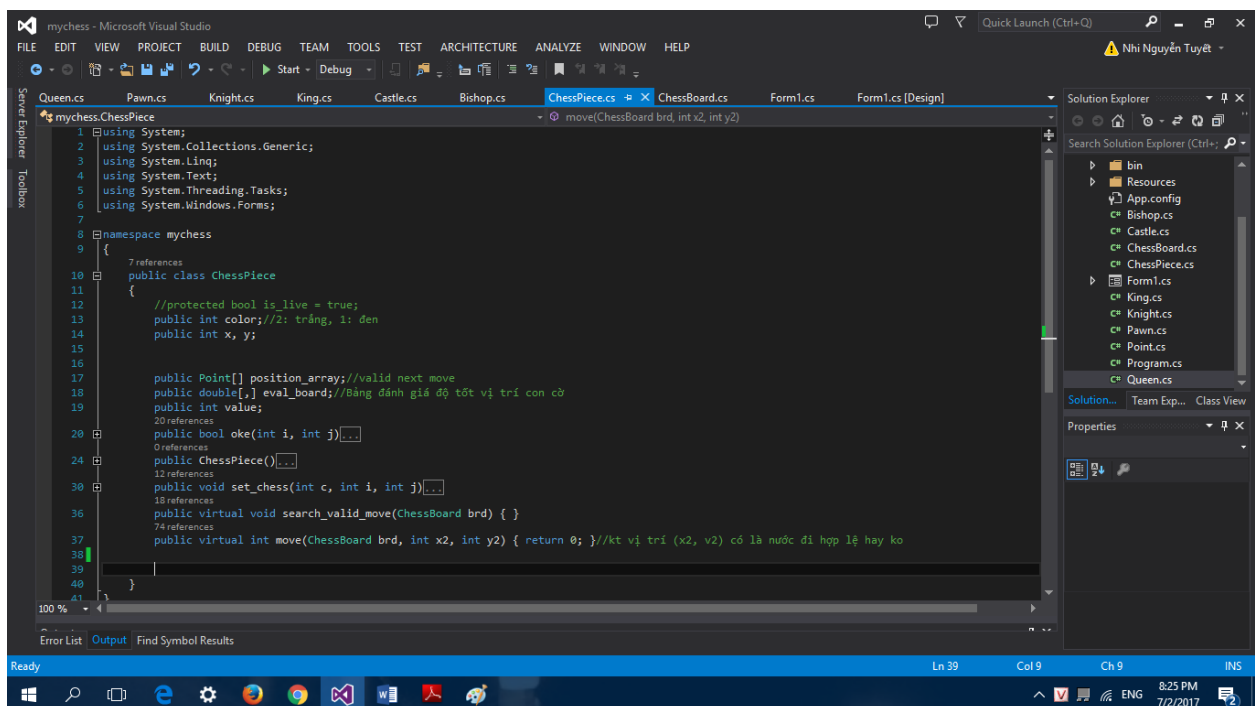
- Thuộc tính:

- + int color: lưu giá trị màu của quân cờ (1: đen, 2: trắng).
 - + int x, y: lưu giá trị lần lượt là hoành độ và tung độ của quân cờ trong bàn cờ ($0 \leq x, y \leq 7$).
 - + Point[] position_array:
 - Mảng 1 chiều kiểu Point, lưu giá trị là tất cả những bước đi hợp lệ của quân cờ khi đứng tại một vị trí bất kì. Số phần tử trong mảng phụ thuộc vào từng loại quân cờ.
- Vd: Tốt: 4 phần tử, vì số nước đi hợp lệ tối đa của tốt là 4.
Vua: 8 phần tử, vì số nước đi hợp lệ tối đa của vua là 8.
- Ban đầu mảng được khởi tạo bằng các giá trị ảo (-1, -1). Và được cập nhật giá trị khi chạy phương thức `search_valid_move()`.
 - Mảng này lưu dữ liệu cho việc sinh trạng thái con trong hàm minimax.

Đồ án môn Trí tuệ nhân tạo

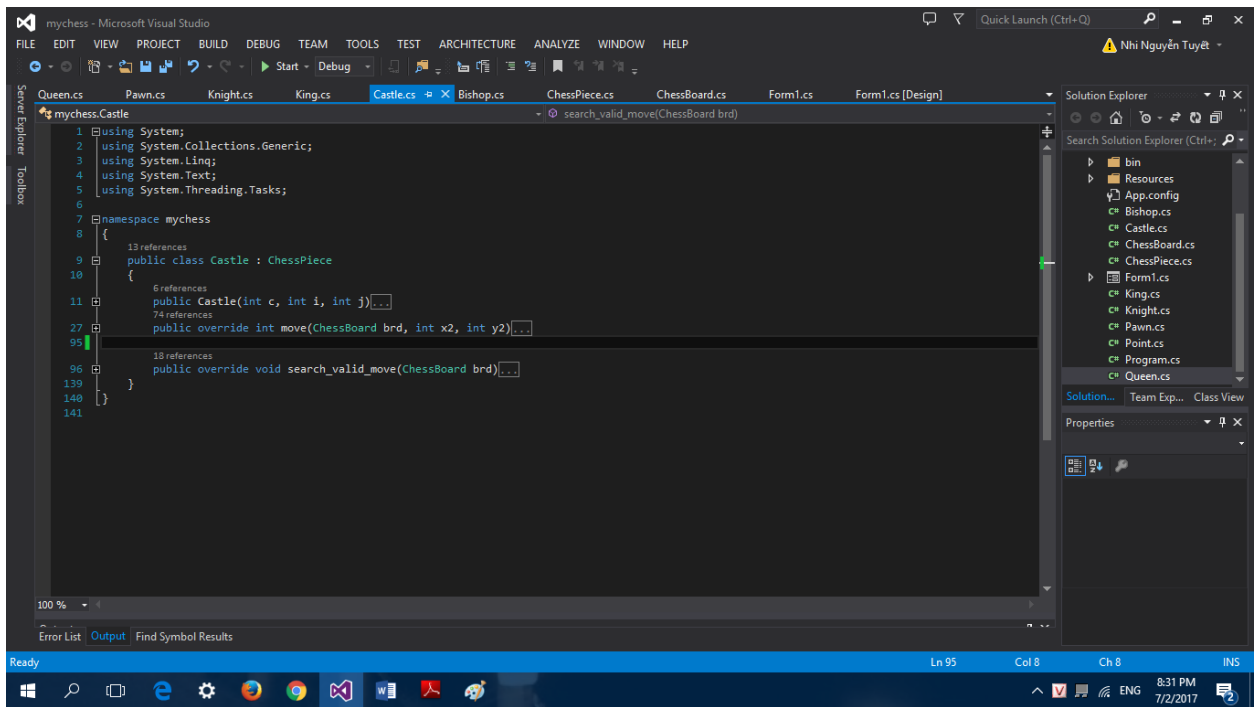
- + double[] eval_board: mảng 2 chiều kiểu double lưu giá trị độ tốt tại vị trí tương ứng của mỗi quân cờ.
- + int value: Độ tốt của quân cờ cờ. Đã được giải thích rõ ràng ở phía trên.
- Phương thức:
 - + void set_chess(int c, int i, int j): gán giá trị màu, tọa độ x y cho quân cờ.
 - + int move(ChessBoard, int i, int j): kiểm tra vị trí mới (i, j) có phải là nước đi hợp lệ không (1: đi được, 0: không đi được). Hàm này nhóm đã tham khảo và viết lại để phù hợp với cấu trúc dữ liệu của nhóm.
 - + void search_valid_move(ChessBoard brd): Hàm liệt kê tất cả các nước đi hợp lệ vào mảng position_array.

Code minh họa lớp ChessPiece:

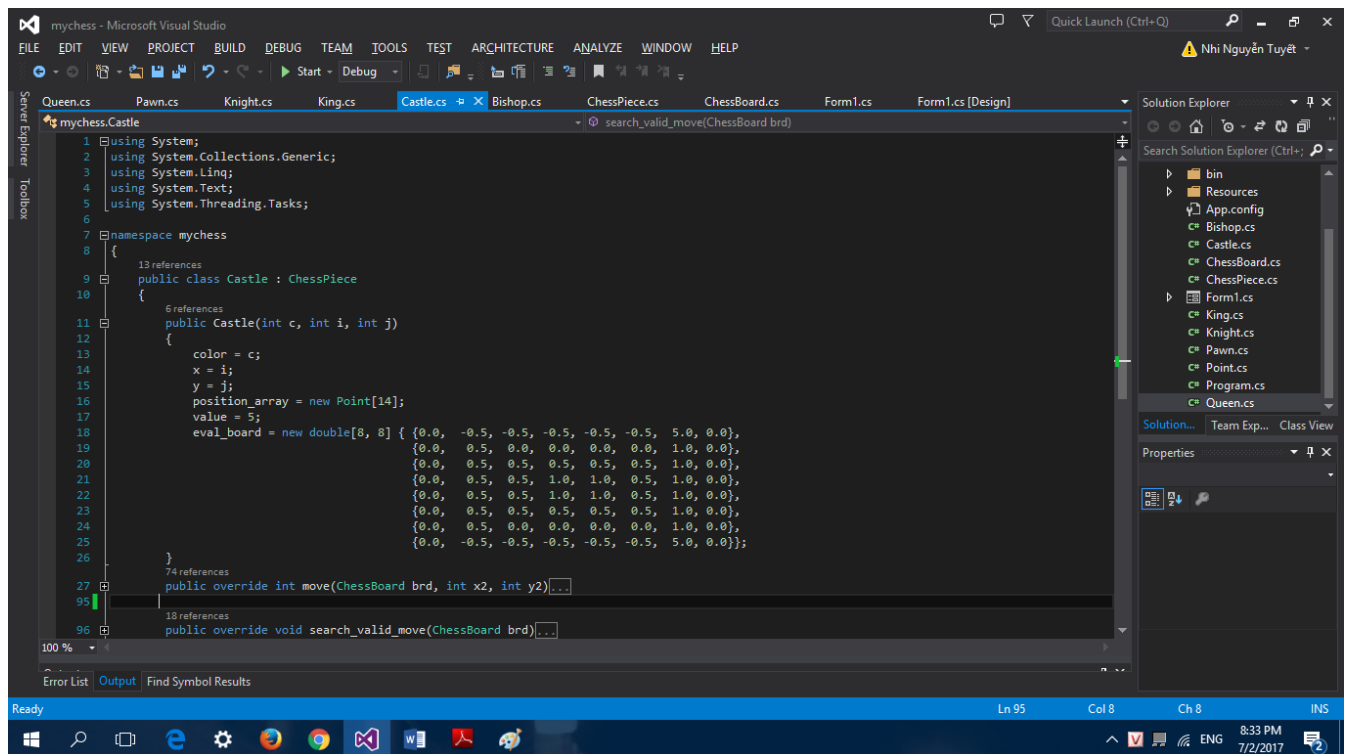


Lấy Quân xe làm ví dụ minh họa cho lớp ChessPiece

Đồ án môn Trí tuệ nhân tạo



+ Giá trị khởi tạo ban đầu của xe là:



Đồ án môn Trí tuệ nhân tạo

+ Phương thức move() của quân xe sẽ được mô tả như sau:

- . x và y là vị trí xe đang đứng.
- . x2, y2 là vị trí truyền vào để kiểm tra.
- . Mã giả:

Nếu ((x2, y2) nằm ở hướng bắc) thì

{

 Duyệt các vị trí từ (x, y) tới (x2, y2)

 Nếu giữa 2 vị trí này có quân cờ thì return 0;

}

Nếu ((x2, y2) nằm ở hướng đông) thì

{

 Duyệt các vị trí từ (x, y) tới (x2, y2)

 Nếu giữa 2 vị trí này có quân cờ thì return 0;

}

Nếu ((x2, y2) nằm ở hướng nam) thì

{

 Duyệt các vị trí từ (x, y) tới (x2, y2)

 Nếu giữa 2 vị trí này có quân cờ thì return 0;

}

Nếu ((x2, y2) nằm ở hướng tây) thì

{

 Duyệt các vị trí từ (x, y) tới (x2, y2)

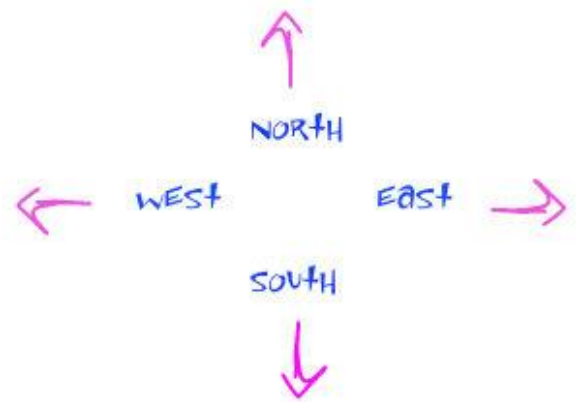
 Nếu giữa 2 vị trí này có quân cờ thì return 0;

}

Nếu (x2, y2) trùng với (x, y) thì return 0;

Nếu (x2, y2) nằm ở vị trí đường chéo thì return 0;

Nếu (x2, y2) là vị trí rỗng thì return 1;

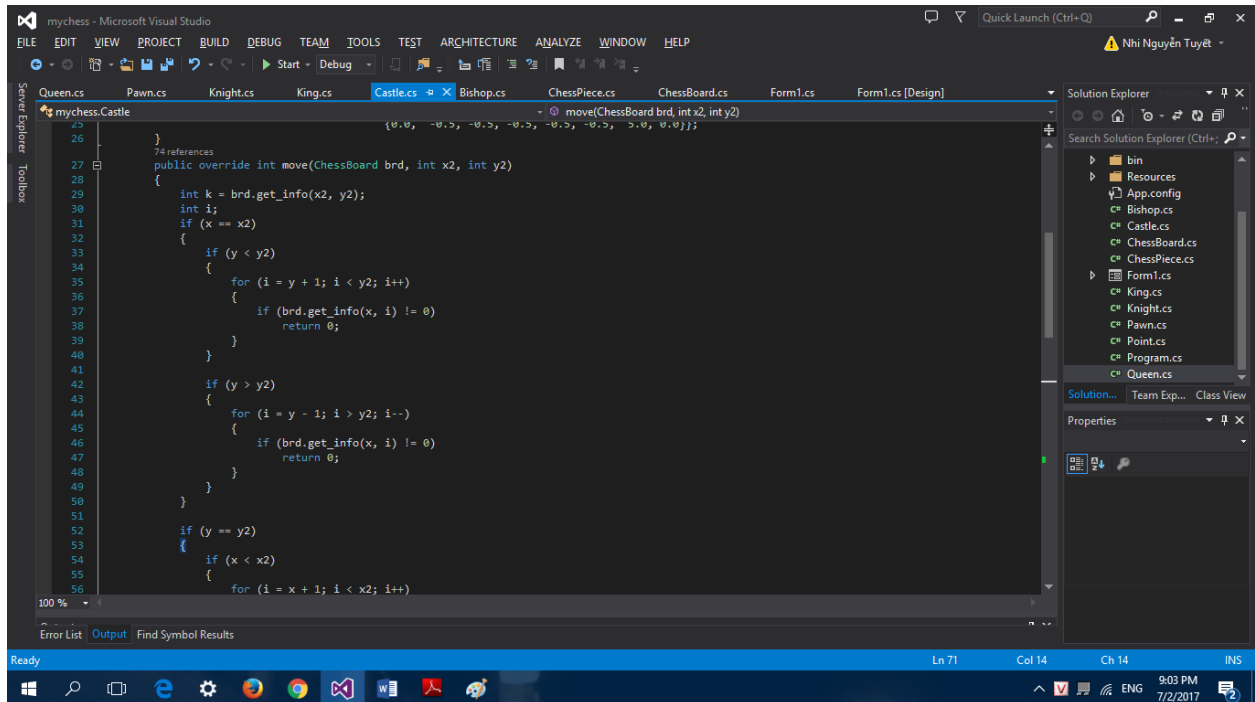


Đồ án môn Trí tuệ nhân tạo

Nếu (x2, y2) là vị trí quân đen và quân xe đang xét là quân trắng thì return 1;

Nếu (x2, y2) là vị trí quân trắng và quân xe đang xét là quân đen thì return 1;

Code minh họa



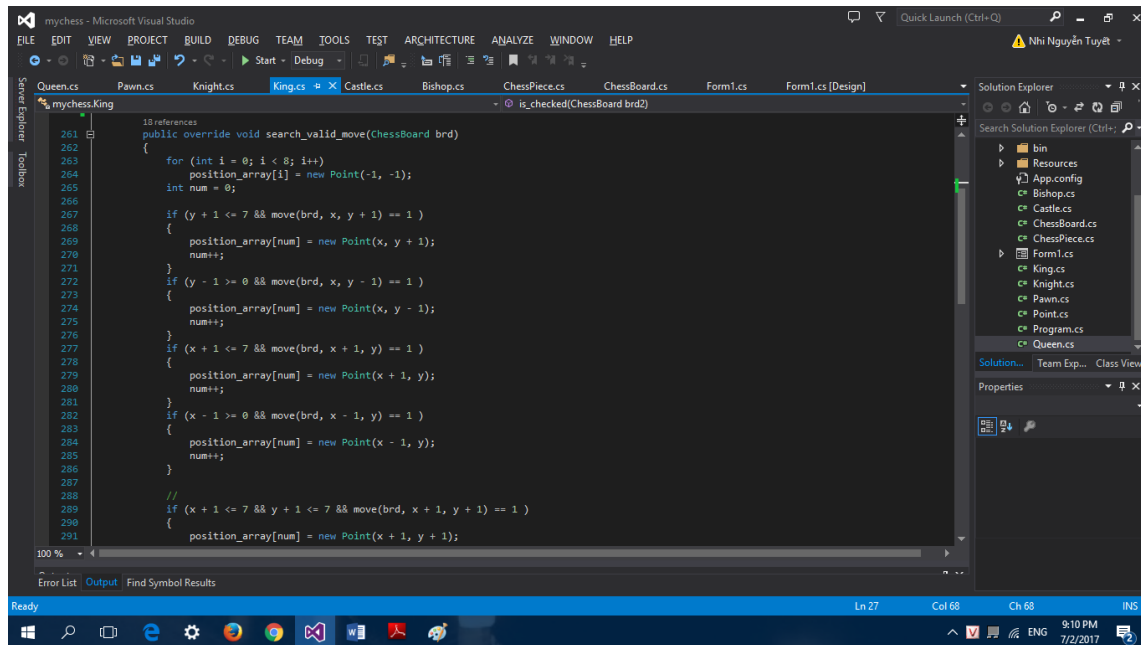
+ Phương thức search_valide_move():

- Mã giả

Duyệt tất cả các ô trên bàn cờ

Nếu (move(brd, ô đang duyệt) == 1) thì position_array[biến đếm] = ô đang duyệt;

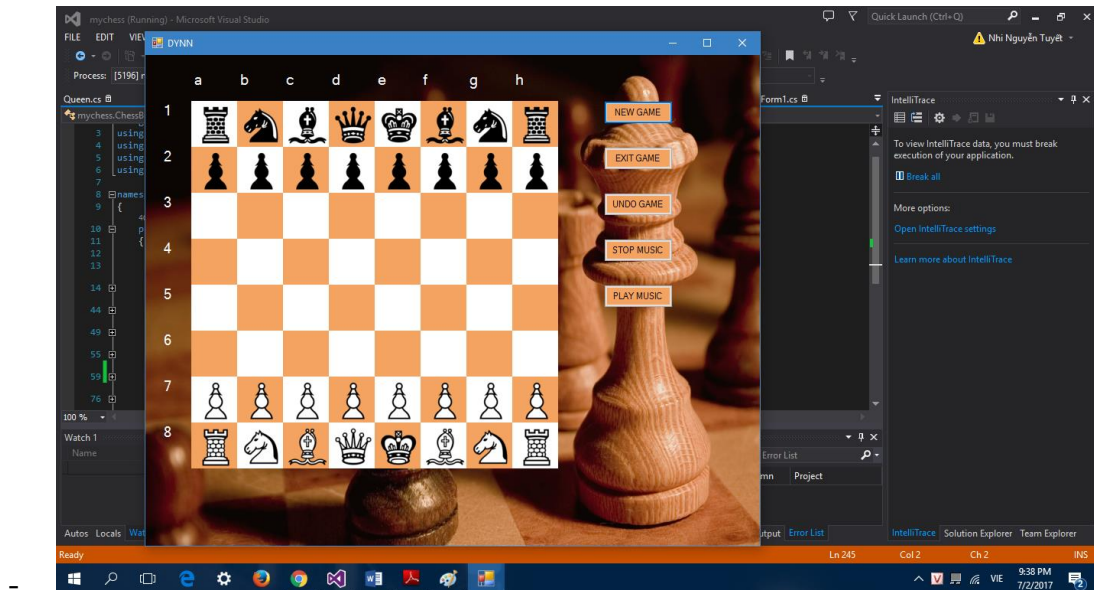
- Code minh họa



2. ChessBoard

- Đây là lớp để biểu diễn không gian trạng thái.
- Thuộc tính:
 - + int [,] board: mảng 2 chiều kiểu int lưu vị trí các quân cờ.
 - + double dValue: lưu giá trị độ tốt của trạng thái.
- Phương thức:
 - + int get_info(int i, int j): lấy giá trị tại vị trí (i, j).
 - + void assign_board(ChessBoard b): gán giá trị của b cho đối tượng gọi hàm.
 - + void set_board(int value, int i, int j): đặt lại giá trị cho vị trí hàng j cột i.
 - + Point searchking(int color): tìm vị trí con vua màu color.
 - + int is_mated(int color): kiểm tra con vua màu color có bị chiếu bí hay không.
 - + int can_defend(int color): kiểm tra con vua màu color có thể được cứu bởi quân nào hay không.
- Ví dụ minh họa:

Đồ án môn Trí tuệ nhân tạo



+ Trạng thái khai cờ được biểu diễn trong máy tính bằng mảng 2 chiều kiểu in như sau

R \ C	0	1	2	3	4	5	6	7
0	1	2	3	4	5	3	2	1
1	6	6	6	6	6	6	6	6
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	12	12	12	12	12	12	12	12
7	7	8	9	10	11	9	8	7

3.2 CÁC VẤN ĐỀ VỀ GIẢI THUẬT

3.3.1 Bài toán tổng quát

Từ khi start game đến khi kết thúc game, chương trình phải trải qua những giai đoạn xử lý sau đây:

B1: Khởi tạo.

B2: Kiểm tra người đi (tức là quân trắng đi) có hợp lệ không và cập nhật bàn cờ sau khi người đi. Kiểm tra có phải là trạng thái kết thúc chưa, nếu chưa thì thực hiện B3, ngược lại thực hiện B4.

B3: Thực hiện minimax có cắt tỉa alpha beta và cập nhật bàn cờ sau khi máy đi chuyển. Kiểm tra trạng thái hiện tại có phải là trạng thái kết thúc không, nếu không thì quay lại B2, ngược lại thực hiện B4.

B4: Xuất kết quả ra màn hình và kết thúc trò chơi.

3.3.2 Vấn đề 1: Phát sinh các nước đi hợp lệ

Phát biểu bài toán: Từ trạng thái hiện tại, sinh ra tất cả các trạng thái con mà hình thành do sự thay đổi vị trí của các quân cờ trên bàn cờ.

- Input: Trạng thái hiện tại

- Output: Tập hợp các trạng thái con

Ý tưởng giải quyết: Từ trạng thái ban đầu, duyệt tất cả các phần tử trên bàn cờ, với mỗi quân cờ tìm tất cả các bước đi hợp lệ của quân cờ đó và lưu lại vào trong mảng position_array của quân cờ. Như vậy ta sẽ có tất cả các bước đi hợp lệ của các quân cờ.

Thuật giải: Thuật giải khá đơn giản, không có gì phức tạp. Trước hết ta cần các dữ liệu sau:

- Hàm move(int i, int j) trong lớp ChessPiece: kiểm tra vị trí (i, j) có phải là nước đi hợp lệ hay không.

- Hàm search_valid_move(ChessBoard b) trong lớp ChessPiece: Tìm tất cả các nước đi hợp lý của quân cờ và lưu lại trong mảng position_array.

- Mã giả: Xét trạng thái hiện tại là b kiểu ChessBoard

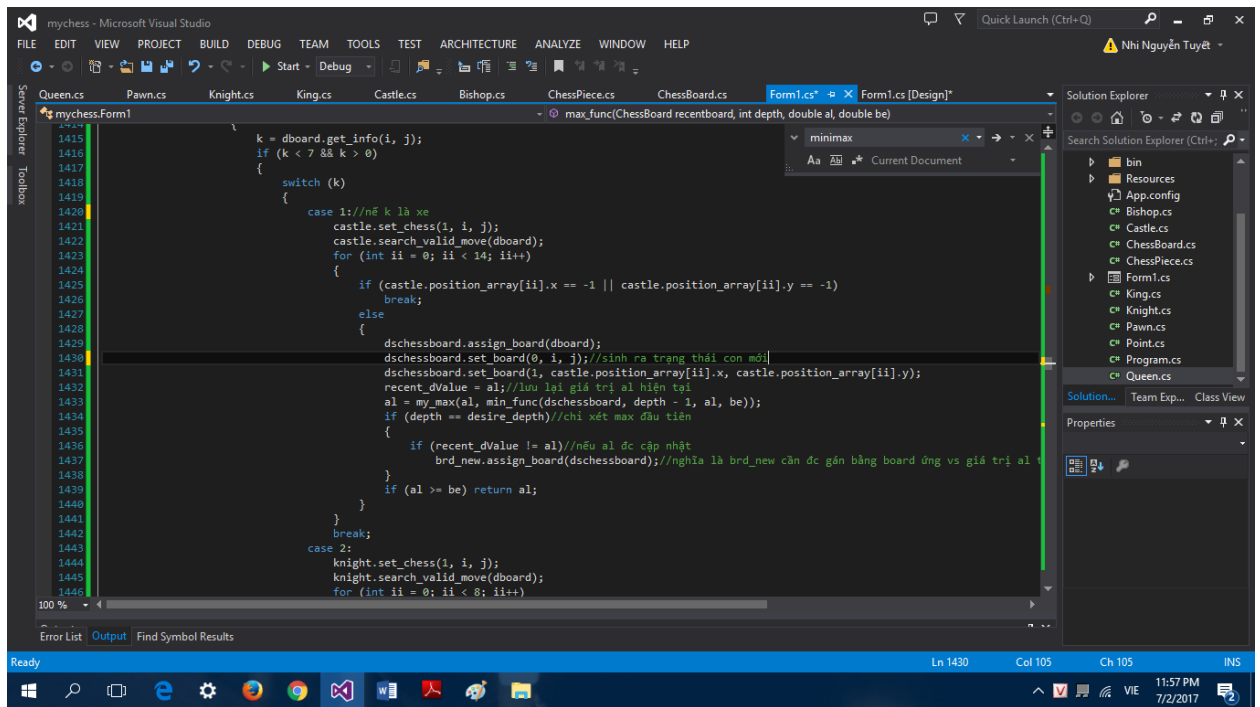
Đồ án môn Trí tuệ nhân tạo

```
For(int i = 0; i < 8; i++)
    For (int j = 0; j < 8; j++){
        K = giá trị trong bảng b hàng j cột i
        If (k là quân xe){
            Thì xe.search_valide_move();//hàm này thực
            hiện thì mảng position_array lưu tất cả các nước đi hợp lệ của quân cờ
            For (int ii = 0; ii < 14; ii++)
                Sinh ra bàn cờ b1 với 2 vị trí bị thay đổi
                là (i, j) và (position_array[ii].x, position_array[ii].y)
            }

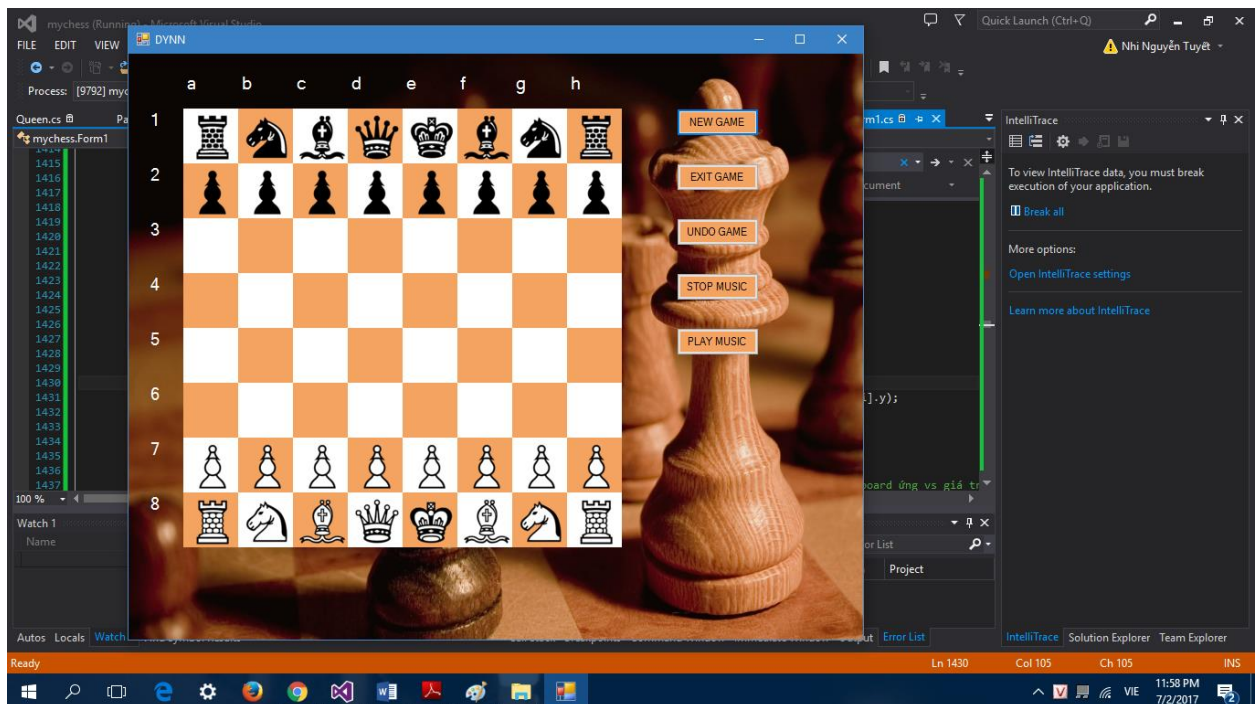
            Thì ma.search_valid_move();
            //Thực hiện tương tự như quân xe
        }
        //Làm tương tự cho tượng, hậu, vua và tốt
    }
```

- Code minh họa:

Đồ án môn Trí tuệ nhân tạo



- Vd minh họa:



Đồ án môn Trí tuệ nhân tạo

Đầu tiên là quân trắng di chuyển. Tiếp theo tới lượt máy đi. Vậy với không gian trạng thái như hiện tại thì có bao nhiêu trạng thái con được sinh ra. Ta dễ dàng nhận ra có đúng 20 trạng thái con được sinh ra.

Đầu tiên là xe, mảng position_array của xe(0, 0) không có giá trị hợp lệ nào.

Tiếp theo là mã, mảng position_array của mã(1, 0) có 2 giá trị là (0, 2) và (2, 2).

Tiếp theo là tốt, mảng position_array của tốt(0, 1) có 2 giá trị là (0, 2) và (0, 3).

Tương tự cho các quân cờ khác

3.3.3 Vấn đề 2: Lượng giá độ tốt của trạng thái

Ý tưởng:

Ban đầu, hàm Heuristic1 có thuật giải sau:

Giá trị của hàm Heuristic trả về một số thực bằng tổng giá trị từng quân cờ đen hoặc trắng còn trên bàn cờ và cộng cho độ chênh lệch số lượng quân cờ của hai bên. Nhưng vì thuật giải quá đơn giản, khiến cho một số ván cờ, quân cờ bên đen (do hàm Heuristic và kỹ thuật Minimax có cắt tia alpha-beta điều khiển) đi một số nước đi rất tệ không thể hiện được tinh thần “trí tuệ” cần có của một thuật giải. Vấn đề phát sinh ra các nước đi không tốt do thiếu đi tính đánh giá khác rất quan trọng đó là độ tốt vị trí con cờ và thành phần số lượng quân cờ hai bên. Nên hàm Heuristic ban đầu đã bị đã bị loại bỏ và thay thế bằng một hàm Heuristic khác:

Tổng quát:

- Input: lớp chessboard và biến color có kiểu giá trị int cho biết lượt đi của quân nào (color = 1: quân đen, color = 2: quân trắng).
Output: kết quả trả về kiểu double cho biết độ tốt một ván cờ đối với quân đen hoặc quân trắng.

Đồ án môn Trí tuệ nhân tạo

- Nếu quân đen còn đôi Bishop thì được cộng thêm 2 điểm vào hàm Heuristic. Còn nếu đối phương còn đôi Bishop thì sẽ bị trừ thêm 2 điểm bởi vì duy trì đôi Bishop sẽ rất có lợi cho bên có đầy đủ hai Bishop.
- Khuyến khích con tốt tiến về phía bên đối phương (quân cờ trắng) và tiến về vị trí hàng cuối cùng để được phong hậu.
- Độ tốt của một ván cờ ngoài độ tốt của từng quân cờ sẽ được cộng thêm độ tốt vị trí mà quân cờ cùng màu đang đứng và trừ đi độ tốt quân cờ và độ tốt vị trí của đối thủ.
- Trong class các loại quân cờ, sẽ có bảng đánh giá độ tốt vị trí quân cờ gọi là eval_board là mảng hai chiều có kích thước 8x8 gồm 64 ô, mỗi ô có các giá trị thể hiện độ tốt của vị trí mà quân cờ đang đứng. Lưu ý, mỗi loại quân cờ có một bảng eval_board đánh giá riêng và biến int đánh giá độ tốt của quân cờ gọi là value.

Xét các bảng đánh giá độ tốt vị trí: (số càng lớn là những vị trí nên đi, số càng âm là vị trí càng tệ)

Pawn:

	A	B	C	D	E	F	G	H
0	0	-1	1	2	1	2	2	6
1	0	-1	1	1	1	2	2	6
2	0	-1	2	1	1	2	2	6
3	0	-2	1	2	2	2	2	6
4	0	-2	1	2	2	2	2	6
5	0	-1	2	1	1	2	2	6
6	0	-1	1	1	1	2	2	6
7	0	-1	1	2	1	2	2	6

Tại B3 và B4 Pawn nên chuyển đến vị trí C3 và C4 vì khi Pawn nằm đó sẽ ngăn cản nước đi của Queen và king.

Đồ án môn Trí tuệ nhân tạo

Tại D3, D4 và E3, E4 là những vị trí phòng thủ, Pawn nên nằm tại đó và dần dần tiến về phía quân địch và tại hàng cuối cùng Pawn được phong hậu nên giá trị tại cột cuối cùng lớn.

Rook:

	A	B	C	D	E	F	G	H
0	0	-1	-1	-1	-1	-1	-1	0
1	0	1	1	1	1	1	1	0
2	0	1	1.5	1	1	1.5	1	0
3	0	1	1	2	2	1	1	0
4	0	1	1	2	2	1	1	0
5	0	1	1.5	1	1	1.5	1	0
6	0	1	1	1	1	1	1	0
7	0	-1	-1	-1	-1	-1	-1	0

Tại các vị trí B0, C0, D0, E0, F0, G0, B7, C7, D7, E7, F7, G7, con xe có ít nước đi hơn nên mang giá trị âm.

Tại D3, D4 và E3, E4 là những vị trí mà xe có nhiều nước đi nhất nên độ tốt vị trí tại đó có giá trị là 2.

Bishop:

	A	B	C	D	E	F	G	H
0	-2	-2	-1	-1	-1	-1	-2	-2
1	-1.5	-1	1	1	1	1	-1	-1.5
2	-1	0	1.5	1.5	1.5	1.5	0	-1
3	-0.5	0.5	1.5	2	2	1.5	0.5	-0.5
4	-0.5	0.5	1.5	2	2	1.5	0.5	-0.5
5	-1	0	1.5	1.5	1.5	1.5	0	-1
6	-1.5	-1	1	1	1	1	-1	-1.5
7	-2	-2	-1	-1	-1	-1	-2	-2

Đồ án môn Trí tuệ nhân tạo

Càng nằm giữa bàn cờ ,Bishop càng nắm giữ nhiều vị trí chủ chốt của bàn cờ nên càng ra giữa độ tốt vị trí càng lớn.

Knight:

	A	B	C	D	E	F	G	H
0	-2	-1	-1	-1	-1	-1	-1	-2
1	-1.5	-1	0	1	1	0	-1	-1.5
2	-1	1	1.5	1.5	1.5	1.5	1	-1
3	-1	1	1.5	2	2	1.5	1	-0.5
4	-1	1	1.5	2	2	1.5	1	-0.5
5	-1	1	1.5	1.5	1.5	1.5	1	-1
6	-1.5	-1	0	1	1	0	-1	-1.5
7	-2	-1	-1	-1	-1	-1	-1	-2

Knight tại các vị trí A0, A7, H0, H7 mang giá trị âm do hạn chế nước đi và không làm chủ được bàn cờ, do vậy, càng vào giữa bàn cờ, giá trị độ tốt vị trí càng cao.

Và cao nhất là tại vị trí D3, D4, E3, E4 nằm giữa bàn cờ thì Knight sẽ có nhiều vị trí di chuyển và kiểm soát bàn cờ tốt hơn.

Queen:

	A	B	C	D	E	F	G	H
0	-2	-1	-1	-1	-0.5	-1	-1	-2
1	-1	0	0	1	1	0	0	-1
2	-1	1	1	1.5	0.5	0.5	0	-1
3	-0.5	0	1	0.5	0.5	0.5	0	-0.5
4	-0.5	0	1	0.5	0.5	0.5	0	-0.5
5	-1	1	1	1.5	0.5	0.5	0	-1
6	-1	0	0	1	1	0	0	-1
7	-2	-1	-1	-1	-0.5	-1	-1	-2

Đồ án môn Trí tuệ nhân tạo

Queen càng tiến tới phía đối thủ thì độ tốt vị trí càng giảm do Queen có hai cách đi nên bán kính hoạt động rộng vì thế ở gần quân của mình để phòng thủ sẽ tốt hơn tấn công.

King:

	A	B	C	D	E	F	G	H
0	2	1	-1	-1	-1.5	-1.5	-1	-2
1	1	1	-1	-1	-1.5	-1.5	2	2
2	1	1	-1	-1.5	-1.5	-1.5	-1.5	-1.5
3	0	0	-1	-1.5	-2	-2	-2	-2
4	0	0	-1	-1.5	-2	-2	-2	-2
5	1	1	-1	-1.5	-1.5	-1.5	-1.5	-1.5
6	1	1	-1	-1	-1.5	-1.5	2	2
7	2	1	-1	-1	-1.5	-1.5	-1	-2

King càng tiến về hướng bên đối phương thì độ tốt vị trí càng giảm do không nằm trong vùng an toàn, nhưng tại vị trí H1, G1, H6, G6, là số dương lớn vì khi gần đến trạng thái gần kết thúc, giá trị hàm Heuristic có thể sẽ chênh lệch lớn, khi đó, nếu ưu thế không nghiêng về bên đối thủ thì king sẽ tiến dần về bên đối thủ, tiến gần hơn king của đối thủ để chiếu.

Mã giả hàm Heuristic:

```
public double heuristic(int[, ] brd, int color)
{
    double cost = 0; //giá trị return cho hàm heuristic
    double countB = 0; //đếm số lượng quân ở trắng
    double countW = 0; //đếm số lượng quân cờ đen
    double countBBishop=0; //đếm số lượng Bishop đen
    double countWBishop=0; //đếm số lượng Bishop trắng
```

Đồ án môn Trí tuệ nhân tạo

```
for (int i = 0; i < 8; i++)
    for (int j = 0; j < 8; j++)
    {
        if (brd[i, j] < 7 && brd[i, j] > 0 && brd[i, j] là Bishop đen)
            countBBishop = countBBishop + 1;
        else if (brd[i, j] >= 7 && brd[i, j] == 9)
            countWBishop = countWBishop + 1;
        else
            { countB = countB + 0; countW = countW + 0; }
    }
```

```
for (int i = 0; i < 8; i++)
    for (int j = 0; j < 8; j++)
    {
        if (tại ô cờ đang xét có giá trị lớn hơn bé hơn 7 và lớn hơn 0 )
            countB = countB + 1;
        else if (tại ô cờ đang xét có giá trị lớn hơn 7)
            countW = countW + 1;
        else
            { countB = countB + 0; countW = countW + 0; }
    }
```

```
if (lượt đi của quân đen: color==1)
{
    for (int i = 0; i < 8; i++)
        for (int j = 0; j < 8; j++)
            { cost=cost+ độ tốt vị trí quân đen đang xét+độ tốt quân đen đang xét
```

```
        - độ tốt vị trí quân địch – độ tốt quân địch}
if (nếu quân đen có đủ 2 Bishop và đối thủ không đủ 2 Bishop)
    cost = cost + 2;
if (nếu quân đen có đủ 2 Bishop và đối thủ đủ 2 Bishop)
    cost = cost + 0;
if (nếu quân đen không có đủ 2 Bishop và đối thủ có đủ 2 Bishop)
    cost = cost - 2;
if (nếu quân đen không đủ 2 Bishop và đối thủ không đủ 2 Bishop)
    cost = cost + 0;
return cost + (countB - countW);
}

if (lượt đi của quân trắng: color==2)
{
    for (int i = 0; i < 8; i++)
        for (int j = 0; j < 8; j++)
            { cost=cost+ độ tốt vị trí quân trắng đang xét+độ tốt quân trắng đang xét
              - độ tốt vị trí quân địch – độ tốt quân địch}
    if (nếu quân trắng có đủ 2 Bishop và đối thủ không đủ 2 Bishop)
        cost = cost + 2;
    if (nếu quân trắng có đủ 2 Bishop và đối thủ đủ 2 Bishop)
        cost = cost + 0;
    if (nếu quân trắng không có đủ 2 Bishop và đối thủ có đủ 2 Bishop)
        cost = cost - 2;
    if (nếu quân trắng không đủ 2 Bishop và đối thủ không đủ 2 Bishop)
        cost = cost + 0;
    return cost + (countW – countB);
}
```

else return 0;

ví dụ cách tính độ tốt một ván cờ: cho bên quân cờ trắng

	a	b	c	d	e	f	g	h
0			black bishop	black king				
1					black bishop			
2								
3								
4								
5			white castle					
6								
7				white king				

$$\begin{aligned} \text{Cost} &= \text{value}(\text{white king}) + \text{eval_board}(\text{white king}) \\ &+ \text{value}(\text{white Rook}) + \text{eval_board}(\text{white Rook}) \\ &- \text{value}(\text{black king}) - \text{eval_board}(\text{black king}) \\ &- \text{value}(\text{black Bishop}) - \text{eval_board}(\text{black Bishop}) \\ &- \text{value}(\text{black Bishop}) - \text{eval_board}(\text{black Bishop}) \\ &- 2 \text{ (do quên đen còn 2 con Bishop)} \\ &+ \text{độ chênh lệch về số quân cờ} \\ &= (200 + 0) + (5 + 1.5) - (200 + 0) - (3 + (-1)) - (3 + 1) - 2 + (2 - 3) = -2.5 \end{aligned}$$

Vậy độ tốt của ván cờ khi đến lượt đi quân cờ trắng là -2.5

3.3.4 Vấn đề 3: Lựa chọn nước đi tốt nhất

Phát biểu bài toán: Tìm nước đi tốt nhất ở đây có nghĩa là tìm nước đi dẫn đến trạng thái thắng nhanh nhất trong tập hợp những nước đi tốt nhất của ta và nước đi tệ nhất của đối phương. Vậy làm sao để chọn ra được đâu là nước đi tốt nhất trong số rất nhiều những bước đi của ta và của đối thủ.

Đồ án môn Trí tuệ nhân tạo

Input: Trạng thái bàn cờ hiện tại, độ sâu, alpha, beta.

Output: 1 số kiểu double.

Ý tưởng: Dùng thuật giải minimax

Thuật giải: Để dùng minimax, ta xây dựng 2 hàm max_func và min_func.

Max_func để cập nhật trạng thái có độ tốt cao nhất, min_func để cập nhật trạng thái có độ tốt thấp nhất. Để làm được việc này, ta cần khai báo biến toàn cục alpha và beta. Sau nữa, để giới hạn độ sâu ta cần thêm biến depth. Cuối cùng là dùng 1 biến toàn cục kiểu ChessBoard b_new để lưu lại trạng thái được cập nhật mới nhất.

- Mã giả: Xét trạng thái hiện tại là ChessBoard b, độ sâu là depth, alpha = -vô cùng, beta = +vô cùng. Output là 1 số kiểu double

+ max_func

If(depth == 0 || b là trạng thái kết thúc)

Thì return heuristic(b);

Else {

Duyệt tất cả các giá trị trong bàn cờ

Nếu là quân đen thì sinh con

Duyệt tất cả các con vừa mới tìm được

Alpha = max(alpha, min_func(b, depth - 1, alpha, beta))

If (alpha được cập nhật và depth == độ sâu mong muốn)

Cập nhật trạng thái cần tìm = trạng thái con đang xét

If (alpha >= beta) cắt tỉa

Return alpha

}

+ min_func

If(depth == 0 || b là trạng thái kết thúc)

Thì return heuristic(b);

Else {

Duyệt tất cả các giá trị trong bàn cờ

Đồ án môn Trí tuệ nhân tạo

Nếu là quân trắng thì sinh con

Duyệt tất cả các con vừa mới tiềm được

$Beta = \min(beta, \max_func(b, depth - 1, alpha, beta))$

If ($alpha \geq beta$) cắt tỉa

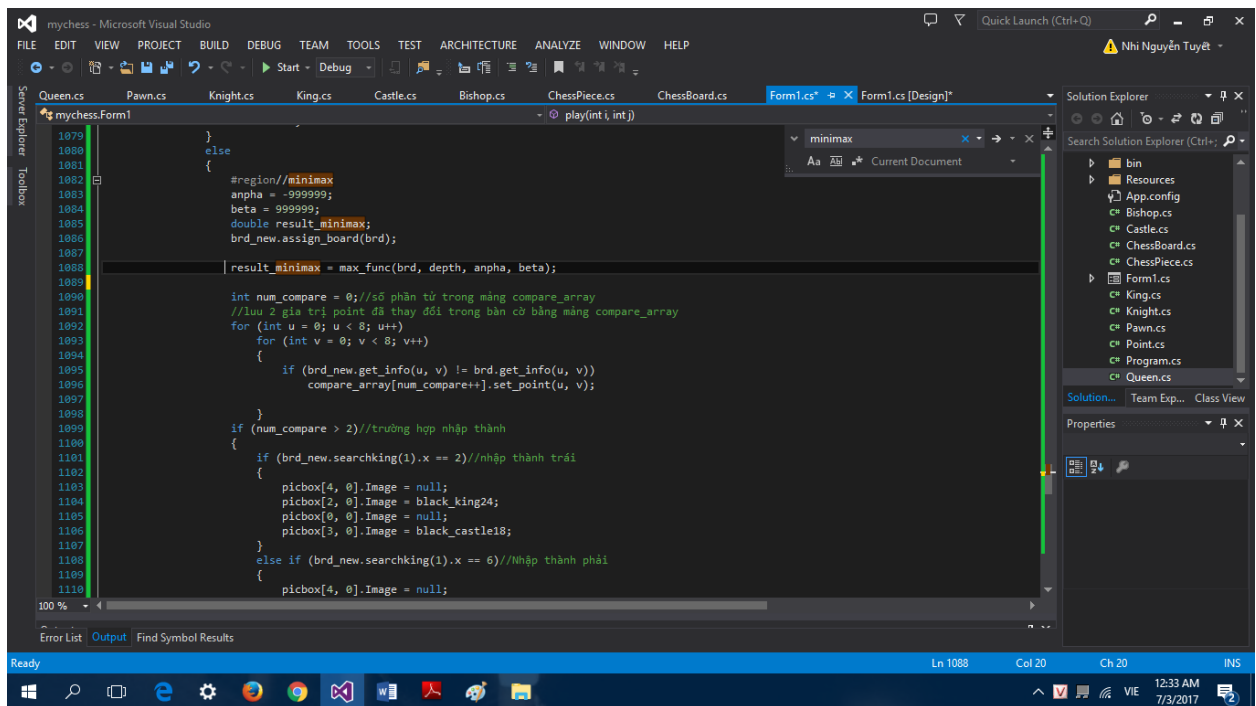
Return beta

}

+ minimax

Trạng thái cần tìm = $\max_func(b, depth \text{ mong muốn}, alpha, beta)$

+ code minh họa:



CHƯƠNG 4 ỨNG DỤNG

4.1 GIỚI THIỆU CHƯƠNG TRÌNH ỨNG DỤNG

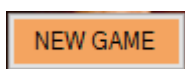
Chương trình ứng dụng được xây dựng và hoàn thành như một game cờ vua kiểu mẩu, phục vụ được nhu cầu vui chơi, giải trí cũng như sở thích đánh cờ của mọi người.

Ứng dụng thể hiện được phần nào các chức năng cơ bản của trò chơi như: NEW GAME, EXIT GAME, UNDO GAME, STOP MUSIC, PLAY MUSIC, ngoài ra ứng dụng còn đáp ứng được nhu cầu giải trí thư giãn của người chơi ở đầu cuộc bằng bản Romance No.2 của Beethoven.

Giao diện trò chơi gồm 2 khu vực thông tin chính. Thứ nhất là cụm các nút chức năng với các chức năng cơ bản của trò chơi. Thứ hai là bàn cờ. Các nút chức năng bao gồm:



NEW GAME: Được sử dụng trong trường hợp bắt đầu ván cờ hoặc trong các trường hợp muốn chơi lại cả ván cờ khi người chơi đi một vài nước sai lầm dẫn đến gặp khó khăn trong các bước tiếp theo thậm chí dẫn đến việc thua cuộc. Nhìn chung việc sử dụng nút

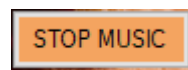
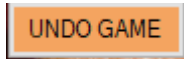


Newgame giữa cuộc chơi khá là không công bằng, tuy nhiên việc này vẫn đáp ứng một số nhu cầu chơi lại của một số người chơi.

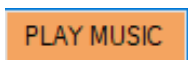
EXIT GAME: Được sử dụng trong trường hợp muốn thoát game kết thúc trò chơi.



UNDO GAME: Được sử dụng trong những trường hợp cụ thể khi thấy nước đi của mình không mang lại lợi thế hoặc có nguy cơ dẫn đến thua cuộc người chơi có thể “Undo” để cải thiện lại tình thế ván cờ. Cũng như nút “Newgame”, việc sử dụng nút “Undo” tương đối không công bằng trong cuộc chơi nhưng để đáp ứng một số nhu cầu “cải thiện tình thế” người chơi vẫn áp dụng biện pháp này.

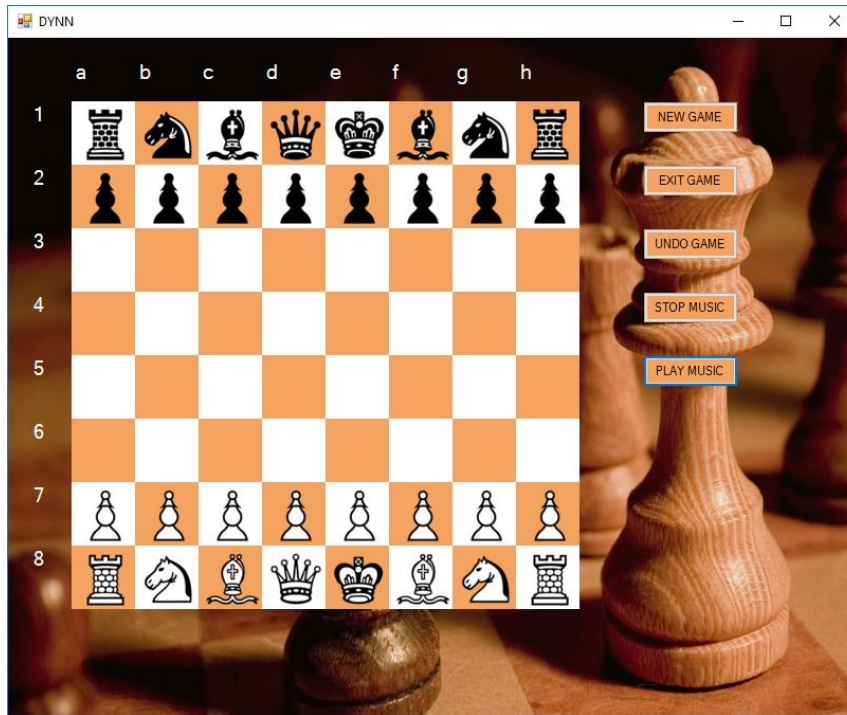


Ngoài ra, ứng dụng còn có 2 chức năng mang tính giải trí là chơi với nhạc nền (PLAY MUSIC) và chơi không nhạc nền (STOP MUSIC). Người chơi có thể tùy chọn 2 chức năng này để bật hoặc tắt âm thanh tùy ý trong lúc chơi tương tự như các trò chơi phổ thông khác.



Giao diện trò chơi bắt đầu với một bàn cờ $8 \times 8 = 64$ ô, được tô màu đen-trắng xen kẽ với vị trí cuối cùng góc bên phải màu trắng.

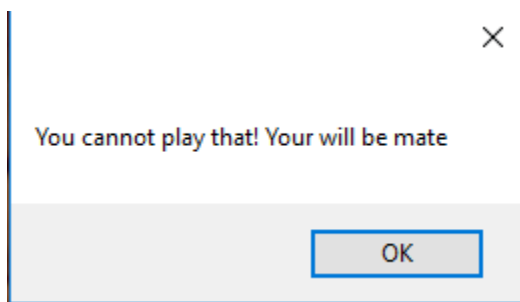
Các quân cờ được sắp xếp như trò chơi cờ vua quốc tế (vì là mô phỏng nên giao diện trò chơi của ứng dụng hoàn toàn phù hợp với chuẩn quốc tế) với các vị trí quân cờ được mô phỏng bên dưới:



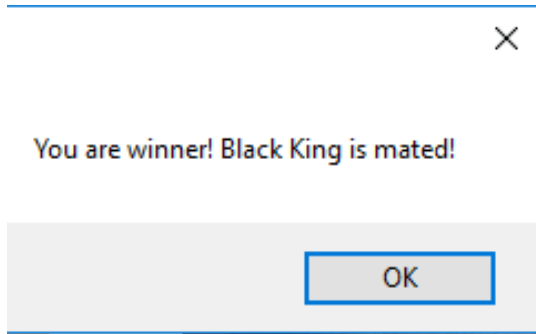
Hình 4.1 Giao diện trò chơi

Theo cài đặt của ứng dụng thì lượt chơi mở màn thuộc về người chơi, tiếp theo đến lượt chơi của máy. Tùy thuộc vào nước đi của máy mà người chơi có nước đi phù hợp để không bị rơi vào trạng thái khó khăn trong trò chơi, ngược lại tùy thuộc vào nước đi của người mà máy sẽ tính toán để đi được nước tiếp theo tốt nhất.

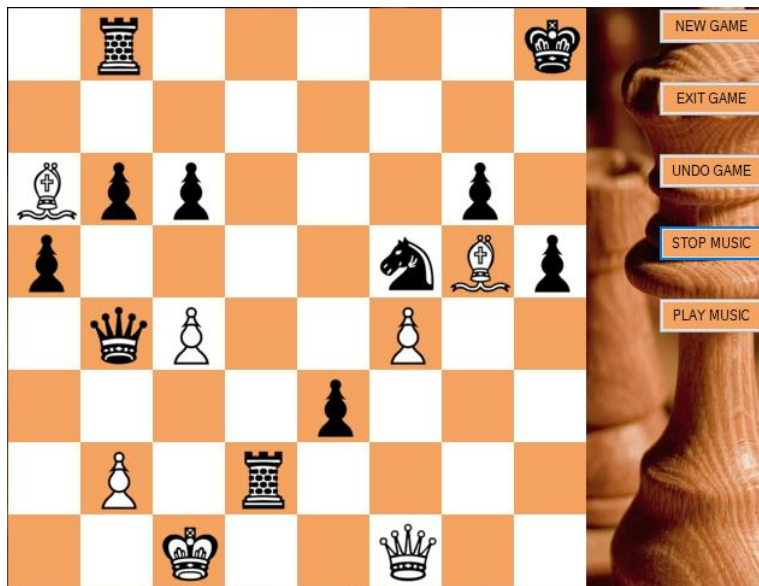
Luật chơi tương tự như luật chơi của Cờ vua quốc tế. Khi một trong 2 bên chiếu bên còn lại hoặc việc không được đi nước tiếp theo do đang ở trong trạng thái bị chiếu đều sẽ có thông báo từ hệ thống.



Hình 4.2 Thông báo bị chiếu từ chương trình



Hình 4.3 Thông báo thắng cuộc từ chương trình



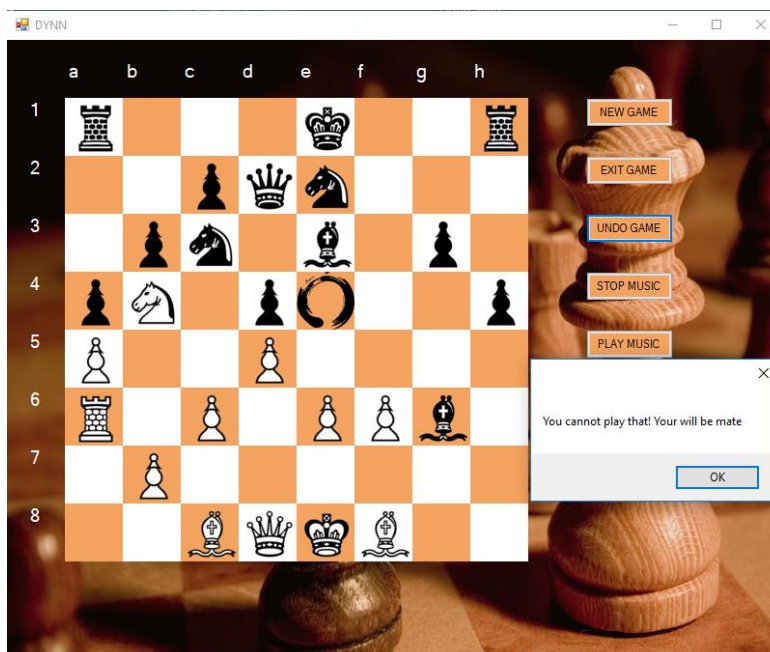
Hình 4.4 Một thế cờ sắp bị chiếu

Tuy nhiên, ứng dụng được cài đặt theo hình thức một ván, tức sau khi kết thúc một ván, người chơi muốn bắt đầu lại phải chọn nút NEWGAME chứ hoàn toàn không theo hình thức đấu 3 ván, nhưng nhìn chung luật chơi vẫn được 2 bên thống nhất trong lúc thỏa thuận.



Hình 4.5 mô phỏng một thế cờ

Ván cờ được kết thúc khi một trong 2 bên bị chiếu bí. Đồng nghĩa với việc bên còn lại không còn khả năng chống trả, cho dù đi nước nào đi chăng nữa thì việc thua cuộc cũng đã xác định.



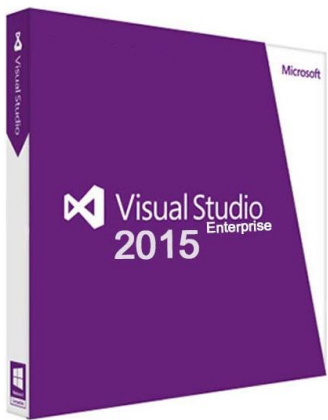
Hình 4.6 Thông báo bị chiếu từ chương trình

Ngoài ra trong quá trình chơi một số trường hợp đặc biệt có thể xảy ra như phong Hậu, nhập thành diễn ra làm tình thế ván cờ thay đổi.

4.2 CÀI ĐẶT

Chương trình ứng dụng được cài đặt ngôn ngữ C# bằng công cụ Visual Studio, ngoài ra còn sử dụng các phần mềm khác như Photoshop CS6 portable, MS Paint để hỗ trợ, phục vụ cho giao diện, nội dung cũng như hoàn thành sản phẩm.

Visual Studio



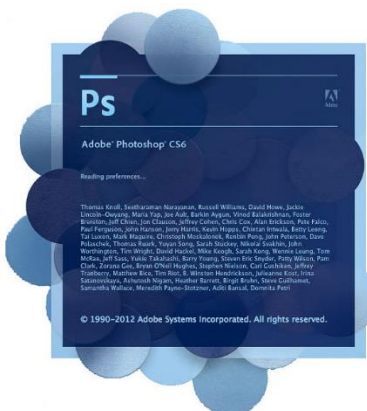
Được sử dụng gần như toàn bộ trong quá trình tạo nên ứng dụng vì đây là công cụ để code. Tạo nên nội dung lẫn giao diện cho ứng dụng.

Bằng ngôn ngữ C#, nhóm đã xây dựng nên ứng dụng từ những phần cơ bản nhất đến phức tạp dần. C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ.

Microsoft phát triển C# dựa trên C++ và Java. C#, theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến .NET Framework mà tất cả các chương trình .NET chạy, và nó phụ thuộc mạnh mẽ vào Framework này. Mọi dữ liệu cơ sở đều là đối tượng và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception, v.v, phản ánh rõ ràng những đặc trưng của .NET runtime.

(wikipedia)

Photoshop CS6 Portable



Công cụ này được sử dụng để chỉnh sửa, thiết kế hình ảnh, nền, các quân cờ.

Adobe Photoshop (thường được gọi là Photoshop) là một phần mềm chỉnh sửa đồ họa được phát triển và phát hành bởi hãng Adobe Systems ra đời vào năm 1988 trên hệ máy Macintosh. Photoshop được

đánh giá là phần mềm dẫn đầu thị trường về sửa ảnh bitmap và được coi là chuẩn cho các ngành liên quan tới chỉnh sửa ảnh. *(theo wikipedia)*

Microsoft Paint



Cùng với Photoshop CS6 Portable thì Microsoft Paint là công cụ hỗ trợ thiết yếu trong việc điều chỉnh các chi tiết về hình ảnh các quân cờ, giao diện cho ứng dụng.

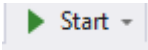
Microsoft Paint, thường được gọi là Paint, là một ứng dụng đồ họa máy tính đơn giản được bao gồm trong tất cả các phiên bản của Microsoft Windows. Ứng dụng này chủ yếu mở và

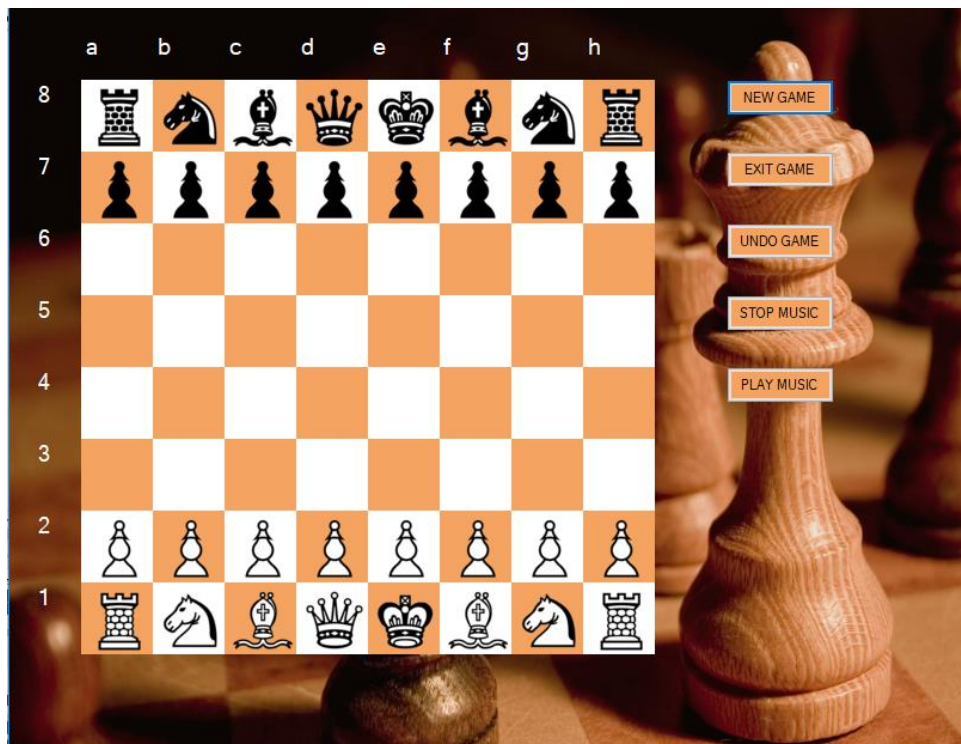
lưu các tệp như Windows bitmap. *(theo wikipedia)*

4.3 KẾT QUẢ CHẠY CHƯƠNG TRÌNH

Kết quả chạy chương trình được đánh giá là cơ bản hoàn chỉnh. Giao diện đẹp, gần gũi, thân thiện với người dùng. Với vị trí bàn cờ nằm bên trái và các nút chức năng nằm bên phải giúp người chơi thuận tiện trong việc chơi cũng như sử dụng các nút chức năng. Vị trí, hình dạng các nút chức năng được sắp xếp, thiết kế đơn giản, dễ hiểu tạo sự thoải mái và không rối mắt cho người chơi. Ngoài ra, ứng dụng còn thêm các chức năng như chơi với nhạc nền tùy chỉnh có hoặc không tạo độ giải trí, thư giãn cho trò chơi.

Kết quả của trò chơi là chính xác trong nhiều trường hợp, đáp ứng được như cầu vui chơi, giải trí cho người chơi. Dù xảy ra nhiều trường hợp như các nước đi khác nhau thì kết quả của trò chơi vẫn đúng, các trường hợp chiếu, chiếu bí vẫn chính xác.

Chương trình có thể được chạy trực tiếp bằng phím  trên IDE Visual Studio 2015 hoặc bằng tổ hợp phím `ctrl+f5`

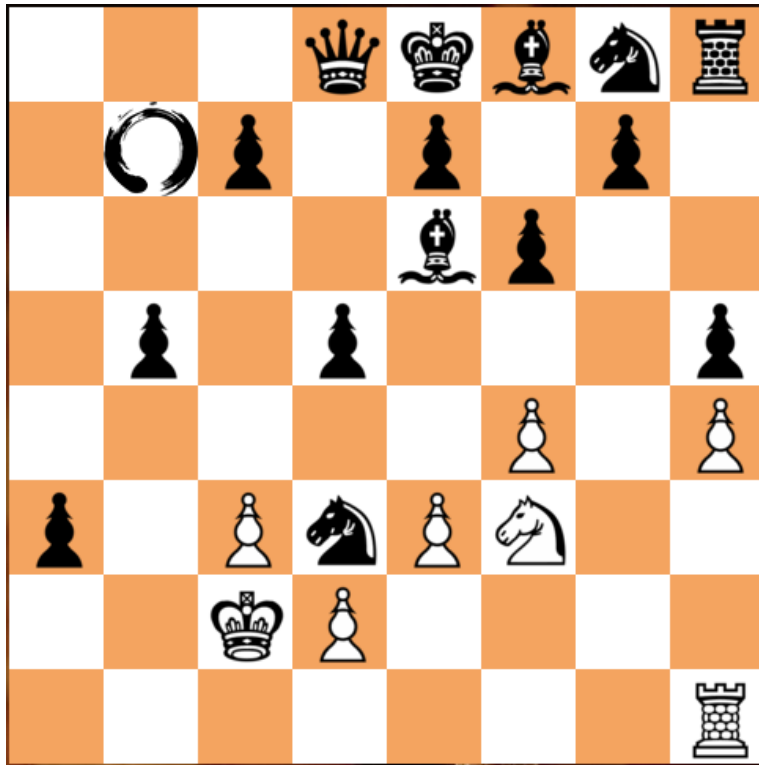


Hình 4.7 Hình ảnh lúc khởi động trò chơi



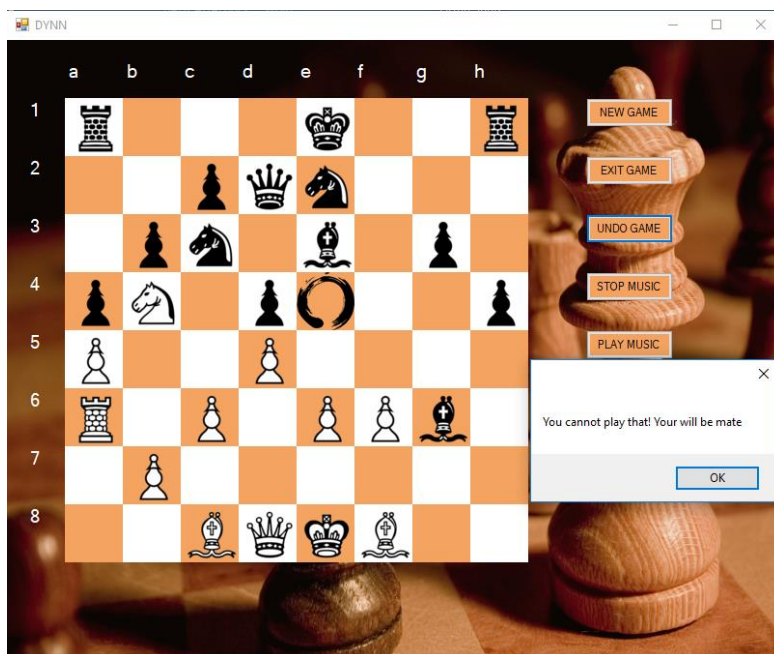
Hình 4.8 Một hình thức khai cuộc

Ưu điểm được thể hiện trong việc ứng dụng đã đáp ứng nhu cầu giải trí cơ bản của người chơi cờ với giao diện đẹp, thân thiện, gần gũi.



Hình 4.9 Một thế trận cuối trung cuộc

Khuyết điểm: Vẫn còn một số nút chức năng muốn thực hiện nhưng còn hạn chế về khả năng lẫn thời gian. Giao diện vẫn chưa thực sự bắt mắt, chỉ tạo sự gần gũi cho người chơi.



Hình 4.10 Một thế cờ bị chiếu

CHƯƠNG 5: KẾT LUẬN

5.1 KẾT QUẢ ĐẠT ĐƯỢC

Đồ án là sự phối hợp thực hiện của các thành viên trong nhóm. Từ những khâu ban đầu như tìm hiểu, đọc hiểu tài liệu đến những bước hiện thực ý tưởng lập trình cũng như việc viết báo cáo.

Sau khi hoàn thành đồ án nhóm nhìn nhận được nhiều vấn đề trong các bước thực hiện. Hiểu rõ cấu trúc một bài báo cáo là như thế nào, đồng thời hiểu rõ hơn những kiến thức đã học ở phần lí thuyết để áp dụng vào đồ án và việc code. Ngoài ra nhóm còn học được việc sử dụng ngôn ngữ C# trong lập trình giao diện, cách tổ chức dữ liệu, các hàm, các biến.

Nhóm còn học được cách phân công công việc cho các thành viên trong nhóm, và cách làm việc nhóm như thế nào cho hiệu quả nhất trong một khoảng thời gian có hạn. Bên cạnh đó nhóm còn biết thêm cách sử dụng các phần mềm về chỉnh sửa hình ảnh, tuy chỉ sử dụng ở mức độ sơ cấp nhưng vẫn nhìn thấy được sự cố gắng nỗ lực của từng thành viên.

5.2 HẠN CHẾ

Vấn đề còn hạn chế đầu tiên là sản phẩm tuy hoàn thành nhưng thật sự chưa hoàn hảo và đáp ứng kì vọng mà nhóm mong muốn.

- + Chưa xét được tất cả những trường hợp đặc biệt như tốt qua đường, phong cấp tùy theo ý muốn người chơi,...

- + Vẫn còn xuất hiện lỗi trong ván cờ như chưa đạt trạng thái kết thúc mong muốn mà đã xuất ra màn hình thông báo thắng hoặc thua và bắt buộc người chơi kết thúc ván cờ. Nhóm vẫn chưa tìm được lỗi xảy ra ở đâu để sửa lại cho hoàn hảo.

- + Nút undo còn hạn chế, chỉ undo được 1 bước thay vì có thể undo được nhiều bước.

- + Vẫn còn nhiều chức năng khác mà nhóm chưa thực hiện được như redo, lưu ván cờ, chức năng 2 người chơi, chức năng gợi ý,...

+ Cách các quân cờ di chuyển chưa đẹp và còn hơi khó theo dõi nếu người chơi không chú ý. Dễ dẫn tới tình trạng máy vừa đi và người chơi không biết máy vừa đi quân nào.

+ Về mặt giao diện lẫn thuật toán tuy tốt nhưng nhóm nghĩ vẫn thật sự chưa tốt nhất.

Ngoài ra còn hạn chế trong việc phối hợp thực hiện giữa các thành viên trong nhóm. Nhóm chưa thực sự sử dụng hết nguồn lực của tất cả các thành viên trong nhóm và thời gian mà nhóm có được suốt một học kì.

5.3 HƯỚNG PHÁT TRIỂN

Phát triển thêm khoảng 2 – 3 hàm Heuristic hỗ trợ cho việc tìm kiếm nước đi tốt nhất cho mỗi trạng thái. Cùng với đó, cải thiện cấu trúc dữ liệu nhằm tối ưu hóa việc sử dụng bộ nhớ, rút ngắn thời gian thực thi chương trình, từ đó cho ra đời một ứng dụng chơi cờ vua vừa “thông minh” hơn lại vừa “tốc độ” hơn.

Tạo thêm nhiều chức năng hỗ trợ người chơi như lưu lại ván cờ đang chơi, mở lại ván cơ đã chơi, tùy chỉnh nhiều cấp độ chơi, hỗ trợ chơi hai người, chơi trực tuyến, tính thời gian, bảng xếp hạng người chơi, v...v...

Nếu có điều kiện về thời gian nhóm thấy ứng dụng được cải thiện thì sẽ hoàn hảo hơn, vừa đáp ứng nhu cầu về mặt thẩm mỹ vừa đáp ứng nhu cầu về giải trí cho người chơi.

TÀI LIỆU THAM KHẢO

- [1] https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua
- [2] http://www4.hcmut.edu.vn/~huynhqlinh/TinhocDC/THDC13/Bai03_6.htm
- [3] <https://chessprogramming.wikispaces.com/>
- [4] <https://medium.freecodecamp.org/simple-chess-ai-step-by-step-1d55a9266977>
- [5] <https://thanhsonnguyen4796.wordpress.com/category/artificial-intelligence/>

Đồ án môn Trí tuệ nhân tạo

Nội dung chỉnh sửa theo yêu cầu của giảng viên

1. Canh chỉnh nội dung, các chỉ mục
2. Thêm input, output cho mã giả hàm heuristic
3. Thêm input, output cho tìm trạng thái tốt nhất trang 48