



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO

NGÔN NGỮ LẬP TRÌNH JAVA

Đề tài : Game Cờ Toán

Giảng viên hướng dẫn:

Ths. Huỳnh Ngọc Tín

Sinh viên thực hiện:

MSSV

Hoàng Xuân Lâm 11520198

Phạm Trung Kiên 11520187

5/30/2014

Mục Lục

LỜI NÓI ĐẦU	2
CHƯƠNG I	3
I.1. Giới thiệu về cờ toán.....	3
I.1.1. Lịch sử game cờ toán.	3
I.1.2. Luật chơi.....	3
I.1.3. Phát biểu bài toán.	7
CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....	9
II.1 Giải thuật Minimax.....	9
II.2. Giải thuật Minimax đến độ sâu lớp cố định.....	10
II.3. Giải thuật cắt tỉa Alpha-Beta.	11
II.3.1. Ý tưởng.....	12
II.3.2. Nguyên tắc Alpha- Beta.....	13
II.3.3. Giải thuật.....	14
CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ.....	16
CHƯƠNG IV: Chương trình xây dựng.....	19
IV.1 Giới thiệu chương trình.....	19
IV.2. Ngôn ngữ sử dụng.....	21
IV.3. Các hàm chính trong chương trình.	22
IV.4. Kết quả chương trình.	23
CHƯƠNG V: TỔNG KẾT ĐÁNH GIÁ	23
V.1. Tổng kết.....	23
V.2. Ưu điểm:	23
V.3 Hạn chế.	23
V.4. Hướng phát triển.	24
TÀI LIỆU THAM KHẢO	25

LỜI NÓI ĐẦU

Cờ toán đã xuất hiện được hơn nửa thế kỷ, và khả năng phát triển tư duy của nó cũng đã được chứng nhận. Tuy vậy, cờ toán vẫn còn khá xa lạ với nhiều người. Một xã hội công nghệ, con người không còn nhiều thời gian để giao tiếp hay chơi trực tiếp với nhau, bởi vậy phát triển game trên nền tảng công nghệ là việc cần thiết.

Không chỉ là một trò chơi, cờ toán còn là 1 cách để luyện tập tư duy hữu hiệu cho trẻ nhỏ học tính toán. Với cách học vừa chơi, vừa học.

Đối tượng tìm hiểu cũng chính là đề tài nhóm đã đăng ký. Phạm vi tìm hiểu xung quanh các vấn đề về tri thức, phương pháp giải quyết chơi đơn giữa hai người và AI, cách biểu diễn chúng trên máy tính qua ngôn ngữ Java.

CHƯƠNG I

GIỚI THIỆU

I.1. Giới thiệu về cờ toán

I.1.1. Lịch sử game cờ toán.

Cờ toán học Việt Nam là sản phẩm sáng tạo của ông Vũ Văn Bảy (Vũ Bảy), một nghệ nhân nặn tượng (gia đình ba thế hệ làm họa sĩ điêu khắc) ở khu Suối Hoa, phường Vũ Ninh, thành phố Bắc Ninh, tỉnh Bắc Ninh, Việt Nam. Cờ toán Việt Nam đã được Cục Bản quyền tác giả văn học nghệ thuật, Bộ Văn hóa - Thông tin (Nay là Bộ Văn hóa, Thể thao và Du lịch) chính thức công nhận sản phẩm trí tuệ vào tháng 5-2005.

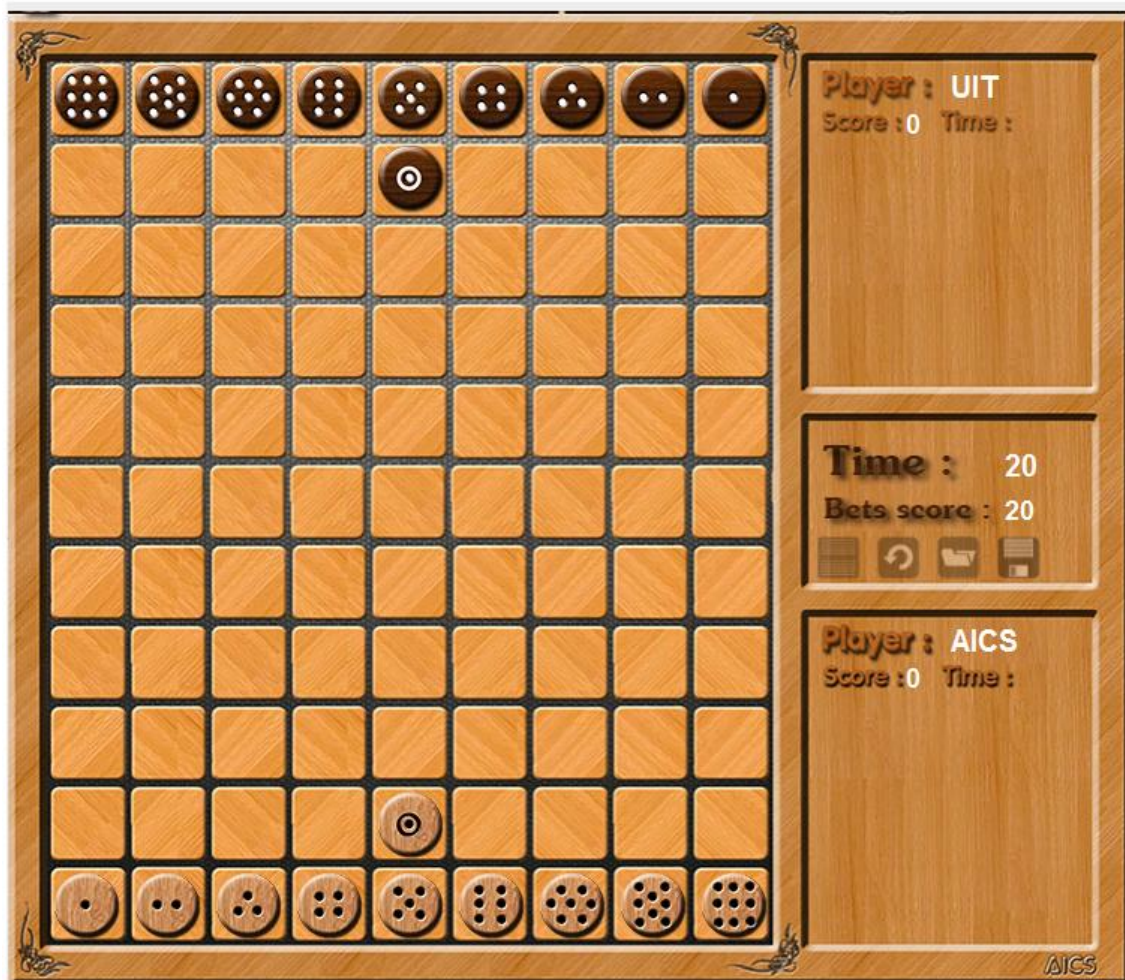
Cờ toán Việt Nam được xem là loại cờ giải trí vừa mang tính chất toán học và triết lý xã hội, khác hẳn so với các loại cờ đã phổ biến như cờ tướng là của người Trung Quốc, cờ vua của người phương Tây. Người chơi phải vận dụng các kiến thức về toán học giản đơn như cộng, trừ, nhân, chia để tính toán các bước đi. Trong cờ vua và cờ tướng, khi mất vua hoặc tướng là bị thua. Tuy nhiên, Cờ toán Việt Nam lại khác cờ tướng, cờ vua ở chỗ: quân số 0 là dân chứ không phải tướng hay vua, và khi đề dân bị đối phương bắt, người chơi sẽ bị thua tuyệt đối. Theo tác giả Vũ Bảy: “Khi chơi cờ toán, nó không chỉ giúp người ta tính toán mà còn dạy cho người ta phải biết lễ sống. Tính cách từng người sẽ được thể hiện qua ván cờ toán. Nếu chỉ thích cộng, thích nhân rồi cứ nhân, cứ cộng mãi người chơi sẽ bị thất bại”.

I.1.2. Luật chơi.

I.1.2.a. Bàn cờ.

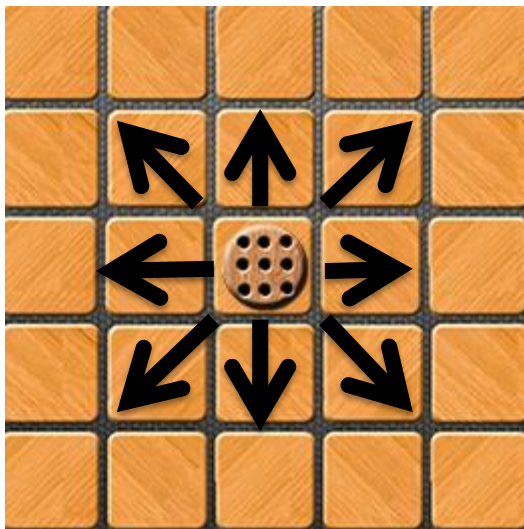
Bàn cờ hình chữ nhật, gồm 99 ô (9 ô hàng ngang và 11 ô hàng dọc). Ô thứ 5 ở hàng thứ 2 của mỗi bên có đường chéo là vị trí cố định của quân số 0 (tức trong khi chơi, số 0 không được di chuyển). Quân cờ hình tròn, mỗi bên một màu khác nhau, trên mỗi quân có các dấu chấm tròn thể hiện các số thứ tự 1-9 (ví dụ, quân số 1 có một chấm tròn; quân số 9 có chín chấm tròn).

Như vậy, mỗi bên có chín quân cờ và một quân số 0. Hai bên cùng xếp quân vào hàng ngang dưới cùng theo thứ tự 1-9 (theo chiều từ trái qua phải, tức tăng dần).



1.1.2.b. Cách đi quân.

Ngoại trừ quân số 0 không được phép di chuyển ra khỏi vị trí, các quân còn lại 1-9 đều được đi theo tám hướng (trong đó bốn hướng đi thẳng ra bốn phía ngang, dọc và bốn hướng đi chéo theo đông - tây - nam - bắc). Mỗi ô trống trên bàn cờ là một bước đi. Số bước đi được thực hiện theo trị số riêng của từng quân cờ. Chẳng hạn, số 2 có thể đi 1-2 ô trống, số 9 có thể đi 1-9 ô trống tùy mục đích của người chơi.



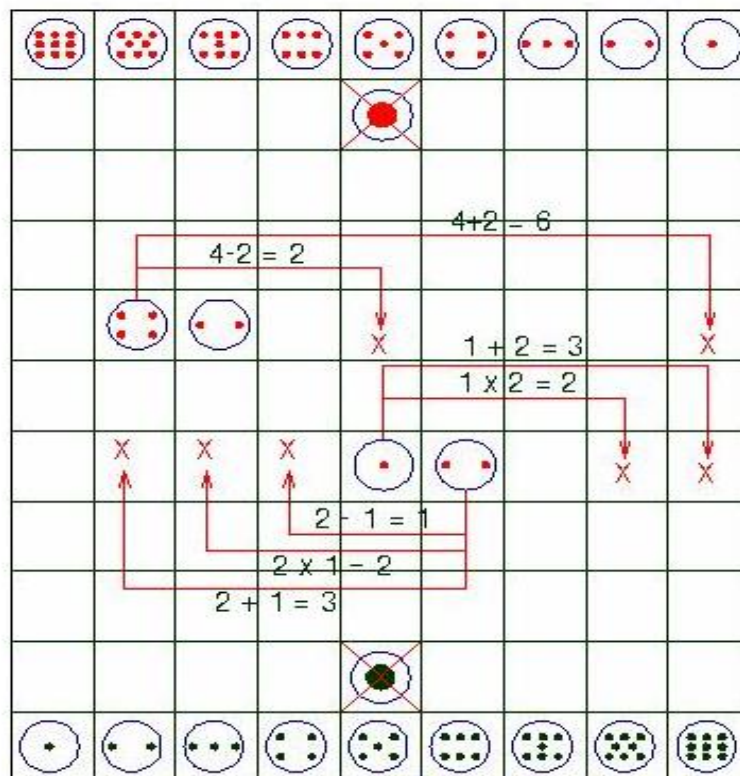
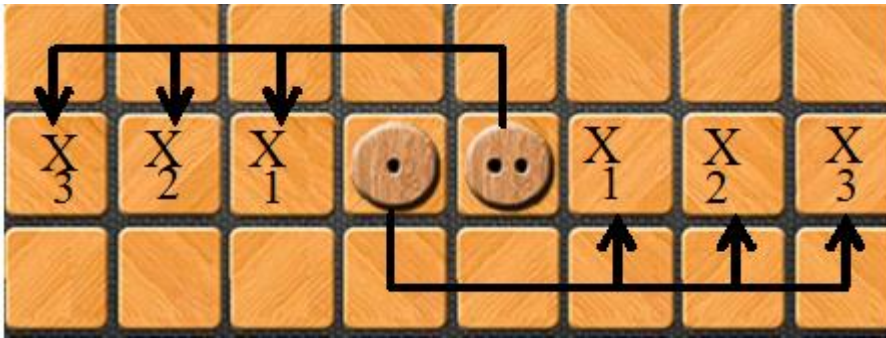
1.1.2.c. Cách bắt quân.

Khi muốn bắt quân của đối phương, điều kiện là bên mình phải có hai quân đứng trong hai ô liền nhau theo chiều dọc hoặc ngang hoặc chéo (để tạo thành một phép tính và phía trước không có quân của đối phương đứng cản). Sau đó dùng các phép hoặc cộng (+), hoặc trừ (-), hoặc nhân (x), hoặc chia (:) với nhau để ra đáp số. Đáp số của mỗi phép tính là điểm có thể bắt được quân của đối phương.

Chẳng hạn, bên mình có quân 8 và quân 5 đứng liền nhau (8 đứng dưới, 5 đứng trên) theo hàng dọc thì có thể lấy $8-5 = 3$ hoặc $8+5 = 13$ để bắt quân đang đứng ở ô thứ 3 của đối phương (tính từ ô của quân 5 đứng trước) nếu muốn đánh tiến. Sau đó, lấy quân số 8 thế vào vị trí mà quân số của đối đối phương bị bắt. Còn nếu muốn bắt lùi thì lấy $5+8 = 13$ và bắt quân đang đứng ở ô thứ 3 bắt đầu tính từ quân số 8.

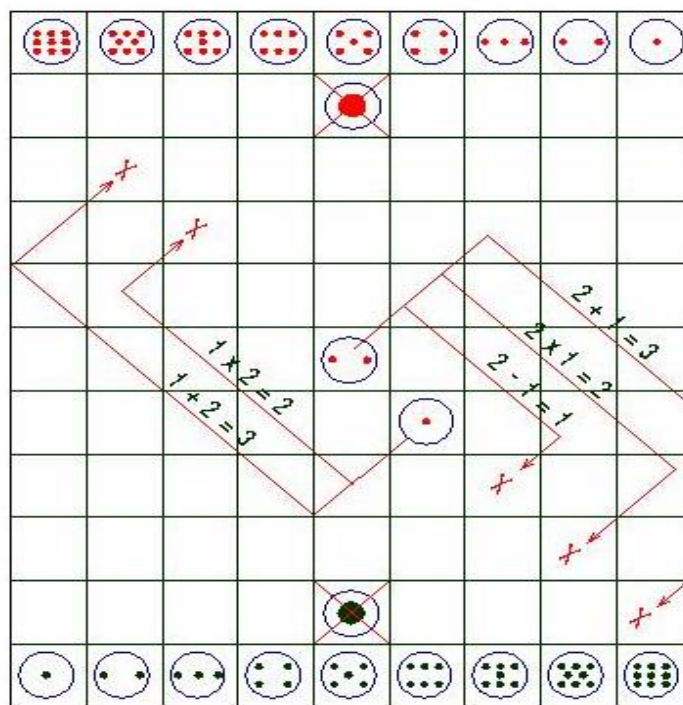
Chú ý, nếu kết quả của phép cộng hoặc nhân mà lớn hơn 10 thì chỉ lấy số của hàng đơn vị để tính điểm bắt quân. Chẳng hạn $5+8 = 13$ thì 3 là điểm để bắt quân của đối phương. Nếu là phép chia có dư thì được tính cả đáp số và số dư để bắt quân. Chẳng hạn, lấy quân 8 chia cho quân 5 bằng 1 dư 3 thì 1 hoặc 3 là ô cờ được bắt quân của đối phương. Không thể lấy $5 \times 8 = 40$ vì điểm 0 là không có giá trị.

Trong khi tính toán, nếu phía trước có quân của đối phương đứng cản thì không thể bắt được quân của đối phương. Chẳng hạn $8+5 = 13$ thì có thể bắt được quân số bất kỳ của đối phương (1, 2, 3, 4...) đang đứng ở ô thứ 3 tính từ quân số 5 của bên mình, nhưng nếu ở ô thứ 1, 2 có quân đối phương đang đứng thì không thể bắt được quân ở ô thứ 3 kể trên.



Hình 1 : Cách ăn quân theo hướng ngang

(Ô đánh dấu X là những vị trí bắt quân đối phương.)



Hình 2: Cách ăn quân theo hướng chéo

(Ô đánh dấu X là những vị trí bắt quân đối phương.)

1.1.2.d. Kết thúc ván cờ.

- Thắng tuyệt đối

Trong quá trình chơi bất kể khi nào, cứ bắt được quân số 0 của đối phương là thắng tuyệt đối.

- Thắng điểm.

Nếu không bắt được quân số 0 vẫn có thể tính việc thắng - thua bằng cách dựa theo số điểm. Mỗi quân cờ có số điểm tính theo trị số của nó. Chẳng hạn quân số 1 là 1 điểm, số 2 là 2 điểm... Trước khi chơi, hai bên có thể thỏa thuận thang điểm cho mỗi ván là 10-15-20 điểm... Bên nào bắt được số quân của bên kia có tổng điểm bằng hoặc cao hơn thang điểm thỏa thuận trước thì sẽ thắng. Chẳng hạn nếu thang điểm là 10 thì chỉ cần ăn được quân 5 và quân 6 của đối phương (tổng điểm 11) là thắng.

- Thắng thời gian.

Nếu hết thời gian suy nghĩ mà vẫn chưa phát sinh nước đi thì phần thắng sẽ dành cho đối phương.

1.1.3. Phát biểu bài toán.

Đây là bài toán áp dụng trên trò chơi có tổng bằng 0 (zero-sum).

Trò chơi có tổng bằng 0 là trò chơi có tổng giá trị kết quả (mà người thắng được hưởng) là cố định. Bất cứ bên nào thắng (+1) cũng làm cho bên kia thua cuộc (-1), tương ứng với tình huống ganh đua thuần túy, cuối cùng dẫn tới tổng

$(+1 -1) = 0$.

Cờ toán là một trò chơi có tổng bằng 0 bởi không thể có trường hợp cả hai bên đều thắng hoặc đều thua. Nếu một bên thắng thì bên kia nhất định là thua và ngược lại [1]. Thể thao là những ví dụ điển hình nhất của trò chơi có tổng bằng 0. Nhà vô địch chỉ có thể đạt được vinh quang khi toàn bộ các đối thủ khác đều thua cuộc. Trong một giải bóng đá tổng số trận thắng luôn bằng tổng số trận thua cũng là bởi cái tính chất tổng bằng 0 ấy.

“Được ăn cả, ngã về không”, việc đầu cơ chứng khoán cũng chính là một trò chơi có tổng bằng 0, ở đó, kẻ đầu cơ có thể mất trắng hoặc thắng lớn, nhưng lợi nhuận mà anh ta thu được có thể đổi bằng cả gia tài, đôi khi mạng sống của những nhà đầu tư tài chính khác.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

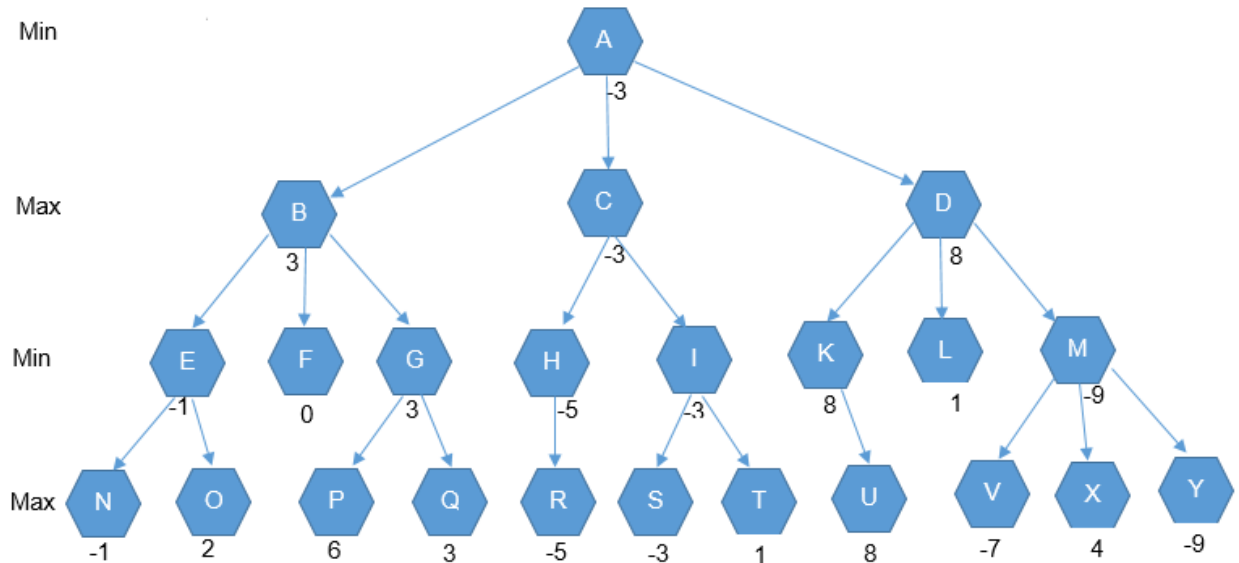
II.1 Giải thuật Minimax.

Hai đối thủ trong một trò chơi được gọi là MIN và MAX. MAX đại diện cho đối thủ quyết giành thắng lợi hay cố gắng tối đa hóa ưu thế của mình. Ngược lại MIN là đối thủ cố gắng tối thiểu hóa điểm số của MAX. Ta giả thiết MIN cũng dùng cùng những thông tin như MAX.

- Một trò chơi như vậy có thể được biểu diễn bởi một cây trò chơi :
 - Mỗi một nút của cây biểu diễn cho một trạng thái.
 - Nút gốc biểu diễn cho trạng thái bắt đầu của cuộc chơi.
 - Mỗi nút lá biểu diễn cho một trạng thái kết thúc của trò chơi (trạng thái thắng, thua hoặc hòa)
 - Nếu trạng thái x được biểu diễn bởi nút n thì các con của n biểu diễn cho tất cả các trạng thái kết quả của các nước đi có thể xuất phát từ trạng thái x .
 - Do hai đấu thủ luân phiên nhau đi nước của mình nên các mức (lớp) trên cây trò chơi cũng luân phiên nhau là MAX và MIN.
 - Các nút ứng với trạng thái mà từ đó người chơi MAX chọn nước đi sẽ thuộc lớp MAX, các nút ứng với trạng thái mà từ đó người chơi MIN chọn nước đi sẽ thuộc lớp MIN.
- Chiến lược MiniMax thể hiện qua quy tắc định trị cho các nút trên cây trò chơi như sau:
 - Nếu nút là nút lá gán cho nút đó một giá trị để phản ánh trạng thái thắng thua hay hòa của các đấu thủ.
 - Sử dụng giá trị của các nút lá để xác định giá trị của các nút ở các mức trên trong cây trò chơi theo quy tắc :
 - Nút thuộc lớp MAX thì gán cho nó giá trị lớn nhất của các nút con của nút đó.
 - Nút thuộc lớp MIN thì gán cho nó giá trị nhỏ nhất của các nút con của nút đó.

Giá trị được gán cho từng trạng thái theo quy tắc trên chỉ rõ giá trị của trạng thái tốt nhất mà mỗi đối thủ có thể hy vọng đạt được. Người chơi sẽ sử dụng các giá trị này để lựa chọn các nước đi cho mình. Đối với người chơi MAX khi đến lượt đi, người chơi này sẽ chọn nước đi ứng với trạng thái có giá trị cao nhất trong các trạng thái con, còn với người chơi MIN khi đến lượt sẽ chọn nước đi ứng với trạng thái có giá trị nhỏ nhất trong các trạng thái con.

Ví dụ trạng thái của trò chơi đang ở mức A, nó sẽ được phân tích thành các mức tiếp theo như hình :



Hình: Cây biểu diễn trạng thái của giải thuật Minimax.

- Ta gán cho mỗi nút lá một giá trị để phản ánh trạng thái thắng thua hay hòa của các đấu thủ. Chẳng hạn ta gán cho nút lá các giá trị như sau:

$$\left\{ \begin{array}{l} < 0 \text{ nếu tại đó người đi X đã thắng} \\ > 0 \text{ nếu tại đó người đi X đã thua} \\ = 0 \text{ nếu hai đấu thủ đã hòa nhau} \end{array} \right.$$

- Như vậy từ một trạng thái bất kỳ, đến lượt mình, người đi MIN sẽ chọn cho mình một nước đi sao cho dẫn đến trạng thái có giá trị nhỏ nhất (trong trường hợp này là C). Ta nói MIN chọn nước đi Min, nút mà từ đó MIN chọn nước đi của mình được gọi là nút Max.
- Người đi MAX đến lượt mình sẽ chọn một nước đi sao cho dẫn đến trạng thái có giá trị lớn nhất (trong trường hợp này là I).
- Đến lượt MIN đi. MIN chọn S, ván đấu kết thúc vì không còn trạng thái nào được sinh ra, MIN thắng, MAX thua.

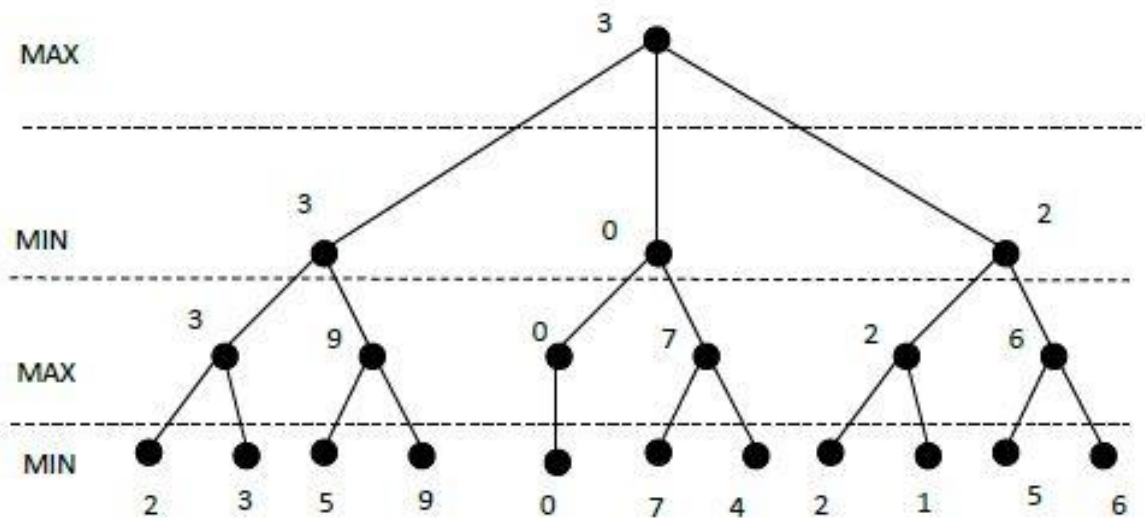
II.2. Giải thuật Minimax đến độ sâu lớp cố định.

Khi áp dụng Minimax cho các trò chơi phức tạp, hiếm khi có khả năng mở rộng đồ thị không gian trạng thái đến các nút lá. 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 (n – move lookahead).

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 cây 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 cây. 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.



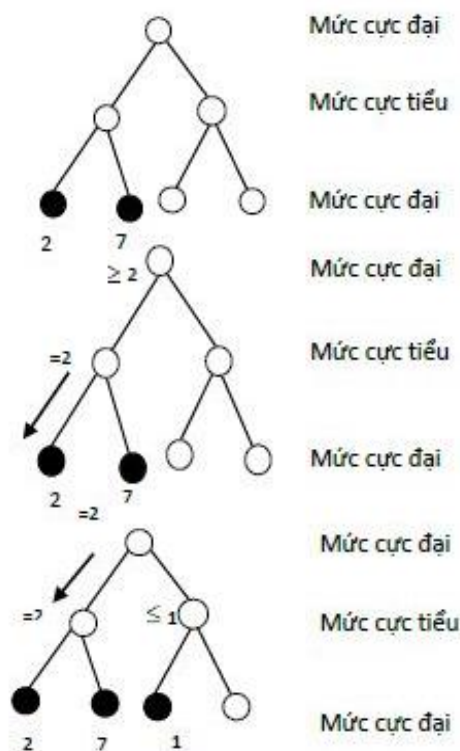
Hình 1. Minh họa thuật giải Minimax với chiều cao cố định

Ở đây sử dụng một heuristic phức tạp hơn, nó cố đo mức độ tranh chấp trong trò chơi. Heuristic chọn một trạng thái cần đo, tính tất cả các đường thắng mở ra cho MAX, rồi trừ đi tổng số các đường thắng mở ra cho MIN. Giải thuật tìm kiếm sẽ cố gắng tối đa hóa sự chênh lệch (hiệu số) đó. Nếu có một trạng thái bắt buộc thắng cuộc cho MAX, nó sẽ được đánh giá là $+\infty$, còn với trạng thái bắt buộc thắng cuộc cho MIN thì được đánh giá là $-\infty$.

II.3. Giải thuật cắt tỉa Alpha-Beta.

○ Thuật toán Minimax yêu cầu phải có sự phân tích qua hai bước đối với không gian tìm kiếm:

- Bước 1 : Truyền xuống đến độ sâu của lớp áp dụng heuristic.
- Bước 2 : Truyền ngược các giá trị trên 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 Alpha-beta nhằm nâng cao hiệu quả tìm kiếm trong các bài toán trò chơi hai đối thủ.
 - Thuật toán Alpha-beta là một cải tiến của thuật toán Minimax nhằm tĩa bớt nhánh của cây trò chơi, làm giảm số lượng nút phải sinh và lượng giá, do đó có thể tăng độ sâu của cây tìm kiếm. Giả sử hình sau là một thế cờ mà hai nút đầu tiên đã được lượng giá. Nếu thực hiện thủ tục Minimax đối với các nút đó sẽ cho thấy người chơi cực đại đã được đảm bảo nếu đi nước bên trái sẽ được ít nhất là 2 điểm dù là các lượng giá của các nút khác cho kết quả như thế nào đi nữa.



Hình 2. Thuật toán Alpha-Beta cắt tỉa nhánh

- Bây giờ, ta lại giả sử nút tiếp theo được lượng giá và cho kết quả là 1. Nếu đi vào nhánh này thì đối phương sẽ đảm bảo làm điểm của người chơi cực đại không thể vượt quá được giá trị 1 dù là các lượng giá của các nút khác cho kết quả như thế nào đi nữa. Do đó đến đây, nước đi tốt nhất là chọn nước đi bên trái với đảm bảo là ít nhất đạt được 2 điểm. Và do đó, hoàn toàn không cần thiết phải lượng giá nút còn lại.

II.3.1. Ý tưởng.

Ý tưởng của tìm kiếm Alpha-beta rất đơn giản: Thay vì nếu như tìm kiếm toàn bộ không gian đến một độ sâu lớp cố định, tìm kiếm Alpha-beta thực hiện

theo kiểu tìm kiếm sâu. Có hai giá trị, gọi là alpha và beta được tạo ra trong quá trình tìm kiếm:

- Giá trị alpha liên quan với các nút MAX và có khuynh hướng không bao giờ giảm.
- Ngược lại giá trị beta liên quan đến các nút MIN và có khuynh hướng không bao giờ tăng.

Để bắt đầu thuật toán tìm kiếm Alpha-beta, ta đi xuống hết độ sâu lớp theo kiểu tìm kiếm sâu, đồng thời áp dụng đánh giá heuristic cho một trạng thái và tất cả các trạng thái anh em của nó. Giả thuyết tất cả đều là nút MIN. Giá trị tối đa của các nút MIN này sẽ được truyền ngược lên cho nút cha mẹ (là một nút MAX). Sau đó giá trị này được gán cho ông bà của các nút MIN như là một giá trị beta kết thúc tốt nhất. Tiếp theo thuật toán này sẽ đi xuống các nút cháu khác và kết thúc việc tìm kiếm đối với nút cha mẹ của chúng nếu gặp bất kỳ một giá trị nào lớn hơn hoặc bằng giá trị beta này. Quá trình này gọi là cắt tia Beta (β cut). Cách làm tương tự cũng được thực hiện cho việc cắt tia Alpha (α cut) đối với các nút cháu của một nút MAX.

- Hai luật cắt tia dựa trên các giá trị alpha và beta là :
 - Luật 1 : Quá trình tìm kiếm có thể kết thúc bên dưới một nút MIN nào có giá trị beta nhỏ hơn hoặc bằng giá trị alpha của một nút cha MAX bất kỳ của nó.
 - Luật 2 : Quá trình tìm kiếm có thể kết thúc bên dưới một nút MAX nào có giá trị alpha lớn hơn hoặc bằng giá trị beta của một nút cha MIN bất kỳ của nó.

Việc cắt tia Alpha-beta như vậy thể hiện quan hệ giữa các nút ở lớp n và các nút ở lớp $n+2$ và do quan hệ đó toàn bộ các cây con bắt nguồn ở lớp $n+1$ đều có thể loại khỏi việc xem xét.

Chú ý rằng giá trị truyền ngược thu được hoàn toàn giống như kết quả Minimax, đồng thời tiết kiệm được các bước tìm kiếm một cách đáng kể.

II.3.2. Nguyên tắc Alpha- Beta

Nếu biết điều đó thật sự tồi thì đừng mất thời gian tìm hiểu nó sẽ tồi tệ đến đâu.

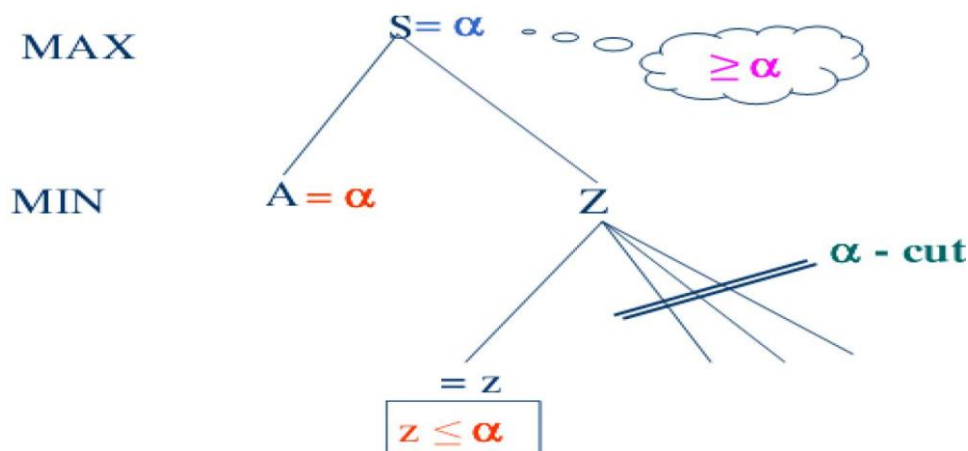
- Ý tưởng này được gọi là nguyên tắc Alpha-beta do nó dùng trong thủ tục Alpha-beta (ta sẽ xét dưới đây). Hai tham số của thủ tục này được gọi là alpha và beta được dùng để theo dõi các triển vọng - chúng cho biết các giá trị nằm ngoài khoảng $[\alpha, \beta]$ là các điểm "thật sự tồi" và không cần phải xem xét nữa. Khoảng $[\alpha, \beta]$ còn được gọi là cửa sổ alpha, beta.
- Trong ngữ cảnh của các trò chơi, nguyên tắc Alpha-beta nói rằng, mỗi khi xem xét một nút bất kì, nên kiểm tra các thông tin đã biết về các nút cha, ông của nó. Có thể do có đủ thông tin từ cha, ông nên không cần phải làm bất cứ việc gì nữa cho nút này. Do đó, nguyên tắc này cũng giúp chỉnh sửa hoặc xác định chính xác giá trị tại nút cha, ông nó [Error! Reference source not

found.]. Như trên nói, một cách dễ tiện theo dõi quá trình tính toán là dùng các tham số alpha và beta để ghi lại các thông tin theo dõi cần thiết. Thủ tục Alpha-beta được bắt đầu tại nút gốc với giá trị của alpha là $-\infty$ và beta là $+\infty$. Thủ tục sẽ tự gọi đệ quy chính nó với khoảng cách giữa các giá trị alpha và beta ngày càng hẹp hơn.

II.3.3. Giải thuật.

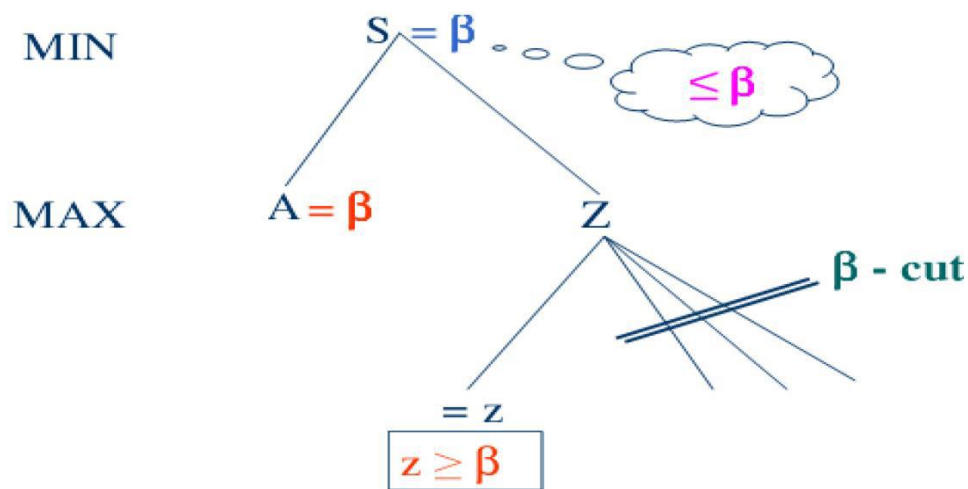
- Thuật toán MiniMax áp dụng Alpha –Beta.
 - Nếu mức đang xét là đỉnh (gốc cây), đặt giá trị của alpha là $-\infty$ và beta là $+\infty$.
 - Nếu như đạt đến giới hạn tìm kiếm (đến tầng dưới cùng của cây tìm kiếm, nút lá), tính giá trị tĩnh của thế cờ hiện tại ứng với người chơi ở đó. Ghi lại kết quả.
 - Nếu như mức đang xét là của người chơi cực tiểu (MIN), thực hiện các công việc sau cho đến khi tất cả các con của nó đã được xét với thủ tục Alpha-beta hoặc cho đến khi alpha là bằng hoặc lớn hơn beta.
 - Áp dụng thủ tục Alpha-beta với giá trị alpha và beta hiện tại cho một con. Ghi nhớ lại kết quả.
 - So sánh giá trị ghi nhớ với giá trị beta, nếu giá trị đó nhỏ hơn thì đặt beta bằng giá trị mới này. Ghi nhớ lại beta (thu hẹp khoảng [alpha, beta] bằng cách giảm giá trị beta).
 - Nếu như mức đang xét là của người chơi cực đại (MAX), thực hiện các công việc sau cho đến khi tất cả các con của nó đã được xét với thủ tục Alpha-beta hoặc cho đến khi alpha là bằng hoặc lớn hơn beta.
 - Áp dụng thủ tục Alpha-beta với giá trị alpha và beta hiện tại cho một con. Ghi nhớ lại kết quả.
 - So sánh giá trị ghi nhớ với giá trị alpha, nếu giá trị đó lớn hơn thì đặt Alpha bằng giá trị mới này. Ghi nhớ lại alpha (thu hẹp khoảng [alpha, beta] bằng cách tăng giá trị alpha).

Cắt tỉa α

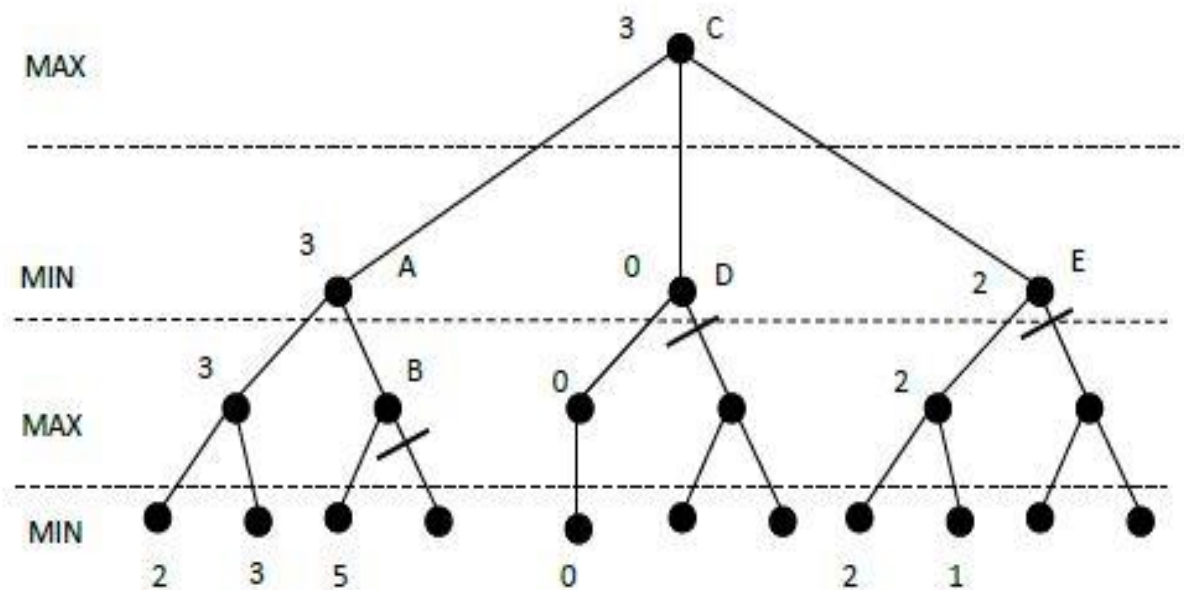


Hình 3. Cắt tỉa Alpha

Cắt tỉa β



Hình 4. Cắt tỉa Beta



Hình 5. Cắt tỉa Alpha-Beta trên cây

A có $\beta = 3$ (Giá trị nút A sẽ không lớn hơn 3). B bị cắt tỉa β , vì $5 > 3$. C có $\alpha = 3$ (Giá trị nút C sẽ không nhỏ hơn 3). D bị cắt tỉa α , vì $0 < 3$. E bị cắt tỉa α , vì $2 < 3$.

CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ

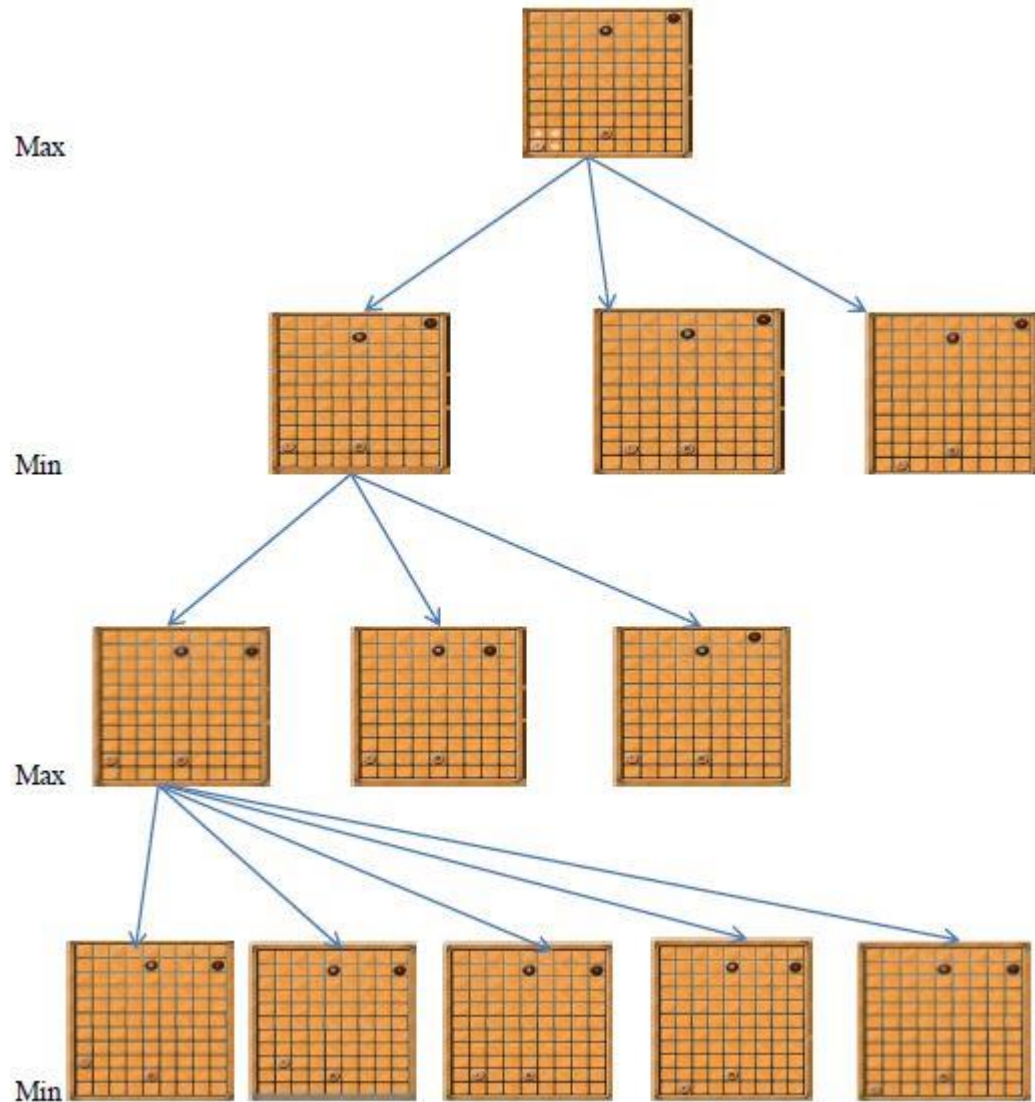
Bài toán áp dụng giải thuật Minimax kết hợp cắt tỉa Alpha-Beta với độ sâu cố định bằng 2 để tìm được nước đi tốt nhất.

- a) Ý tưởng giải quyết bài toán.
 - Bài toán được định nghĩa với 3 cách giải quyết như sau:
 - Kết thúc game với số điểm được “cược” trước, bên nào đạt tới số điểm đó trước bên đó thắng.
 - Kết thúc game với điều kiện ăn quân số “0” của đối thủ.
 - Kết thúc game với điều kiện hết thời gian suy nghĩ.
 - Ý tưởng được đưa ra để giải quyết các vấn đề của bài toán là:
 - Bước 1: Bài toán đưa về bài toán 2 người chơi để với thể hiện là: Biểu diễn bàn cờ đầy đủ quân cờ và có thể ăn quân cờ (tương tác người- người).
 - Bước 2: Bài toán được thêm với phần Game AI để áp dụng cơ sở lý thuyết ở chương 2.
 - b) Áp dụng phần lý thuyết trong mục Cơ sở lý thuyết để giải quyết bài toán.
 - Các phương thức của AI trong bài toán (lớp VanCo.java).
 - Hàm alphaBeta: Thực hiện tìm kiếm theo thuật toán Alpha-beta.
 - Hàm evaluate: Lượng giá thế cờ hiện tại.
 - Hàm moveAI : Hàm di chuyển nước đi AI của quân cờ.
 - Hàm moveEatChessAI : Hàm ăn quân cờ của đối phương trong nước đi AI.
 - Hàm tryMove : Hàm sinh nước đi thử của quân cờ.
 - Hàm removeTryMove : Hàm xóa nước đi thử.
- Sau đây chúng ta sẽ xem xét các phương thức AI chính của bài toán .
- Hàm Minimax kết hợp cắt tỉa Alpha-Beta.

```
AlphaBeta( Nuocdi, phe, depth, alpha, beta)

{
    Nếu depth = 0
        Tính Evaluate();
    bestValua = alpha;
    nếu phe = 1 thì bestValua = beta;
    Trong khi alpha < beta thì
    {
        phát sinh nước đi thử của 1 trạng thái.
        duyệt toàn bộ nước đi của trạng thái cờ đó.
        {
            Phát sinh nước đi thử;
            Đệ quy hàm AlphaBeta để cắt tỉa ;
            Hủy bỏ nước đi thử;
            Nếu value > bestValue
                gán nước đi tốt nhất là nước đi thử vừa đi;
        }
    }
}
```

Hình: Ví dụ



- Phương pháp lượng giá eval.

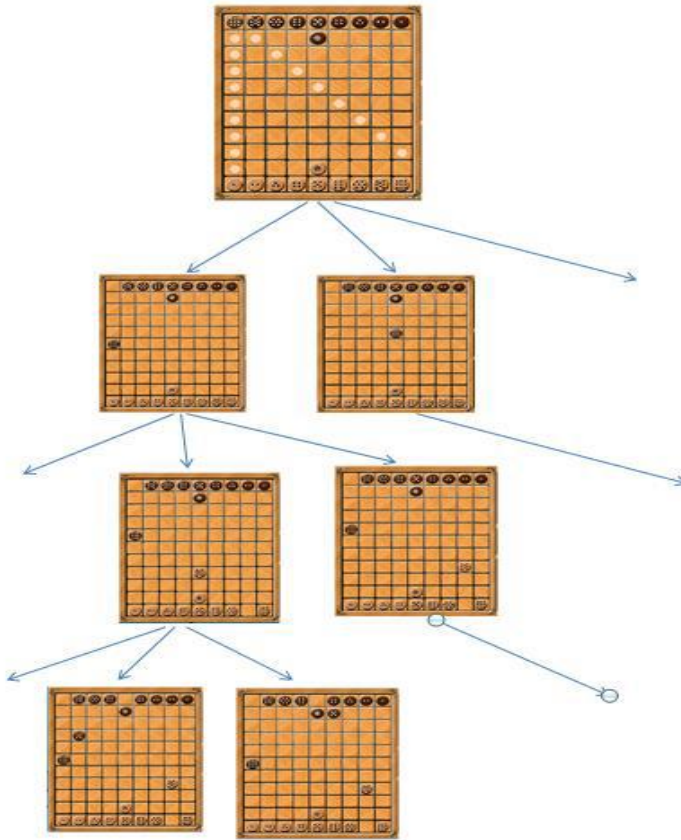
Hàm tính giá trị của thế cờ.

Độ tốt của thế cờ sẽ tăng nếu nước đi phát sinh nước ăn quân và số quân tồn tại trên bàn cờ.

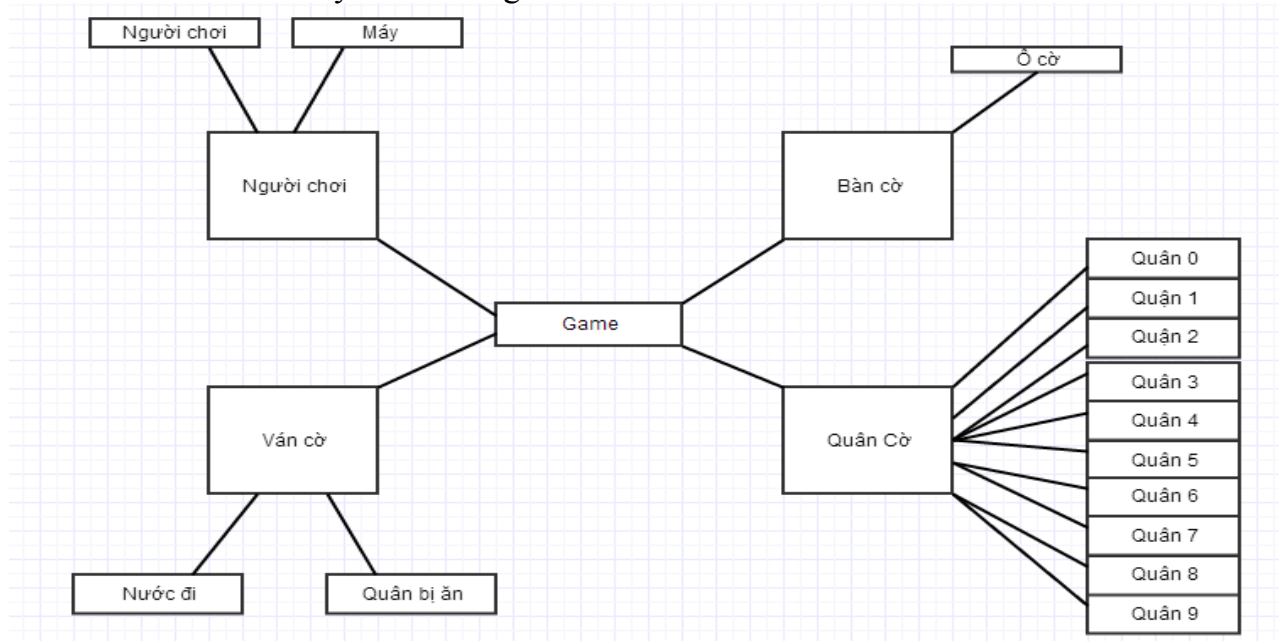
Nếu là quân 0 thì sẽ có value lớn nhất.

Các quân 1 đến 9 tăng dần các giá trị value theo quân (20, 30, 40,...,80,90,100).

- c) Cấu trúc dữ liệu và cách biểu diễn các trạng thái của bài toán.
- Biểu diễn quân trên bàn cờ và các không gian trạng thái.



- Cấu trúc cây của chương trình:

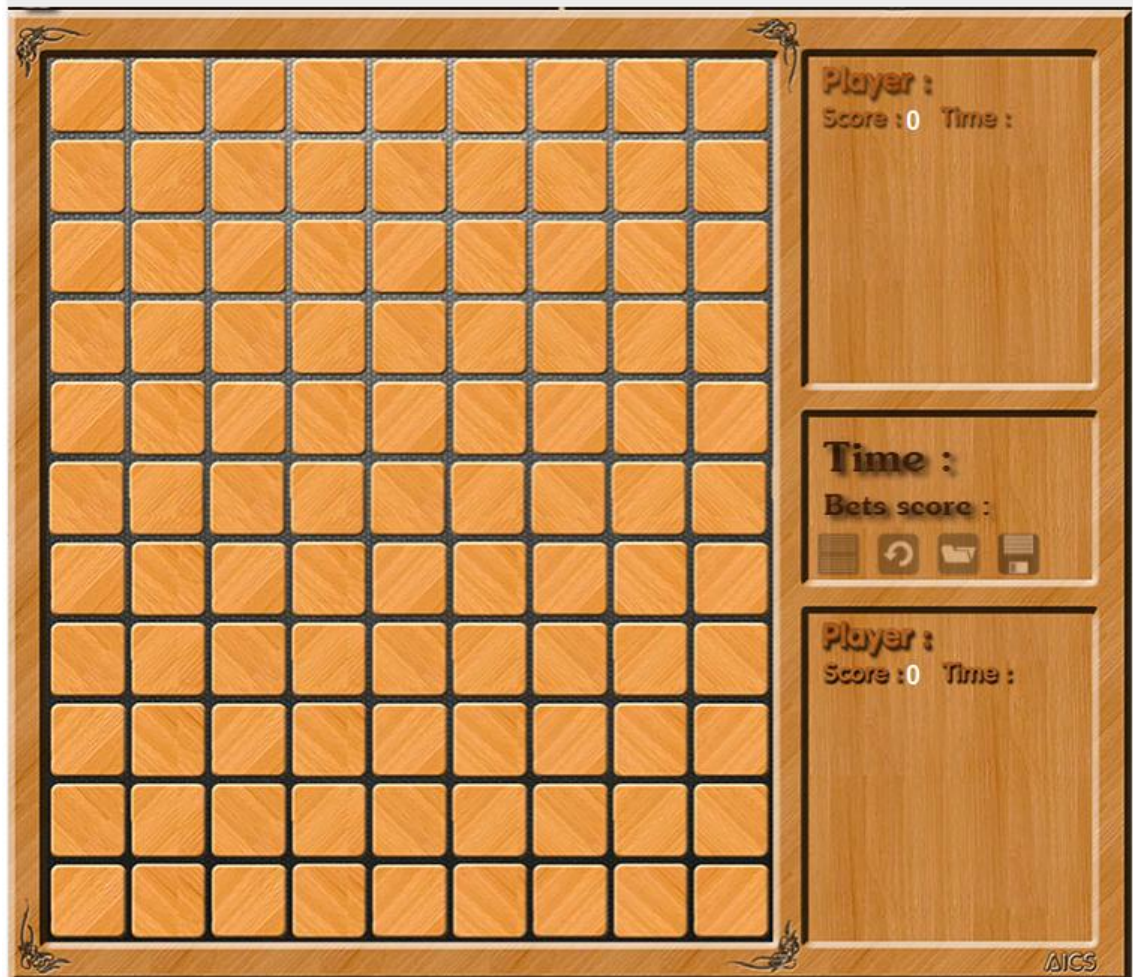


CHƯƠNG IV: CHƯƠNG TRÌNH XÂY DỰNG

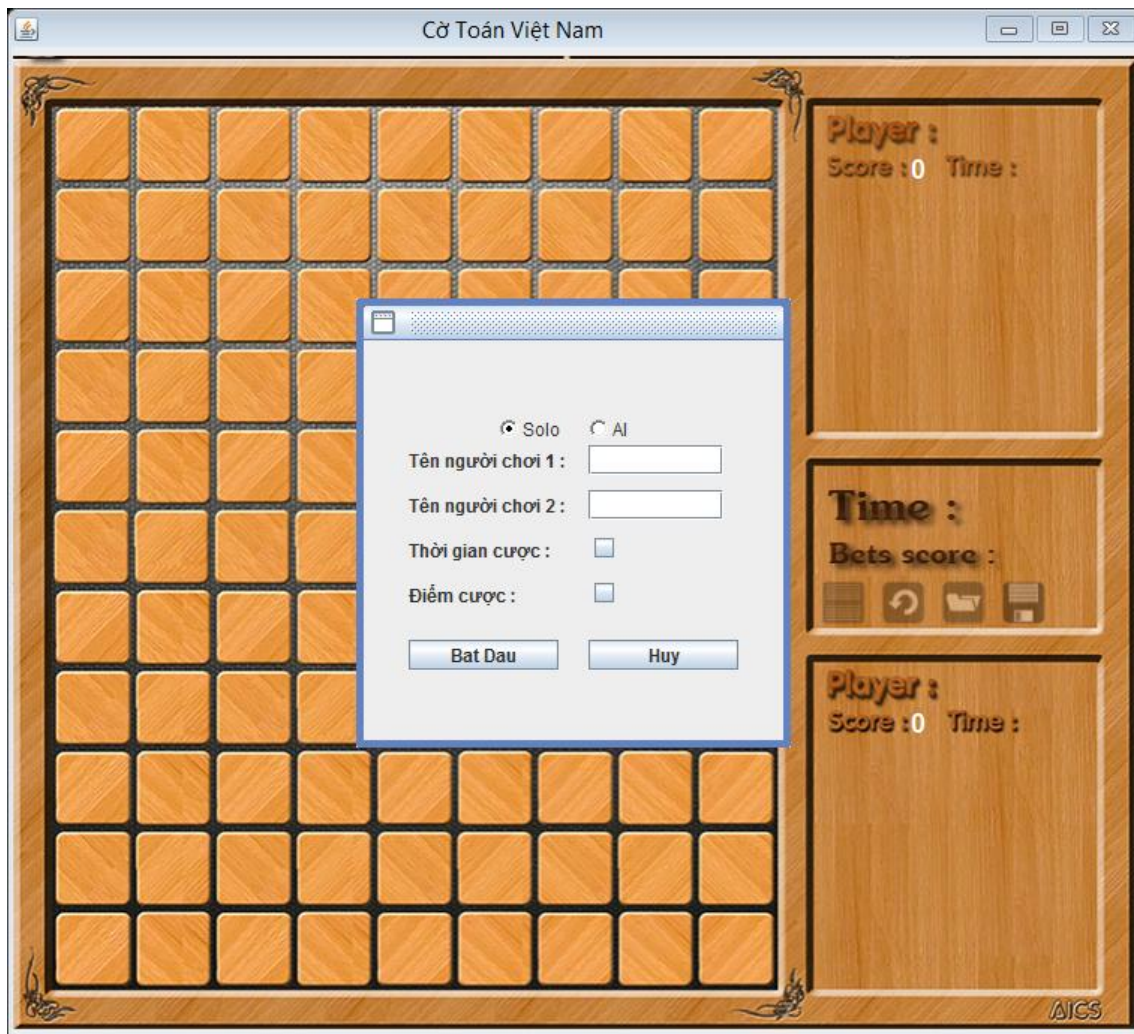
IV.1 Giới thiệu chương trình.

Giao diện của game cờ toán.

- Đầu tiên là giao diện trước khi bắt đầu game







Hình 4.1: Giao diện khi bắt đầu

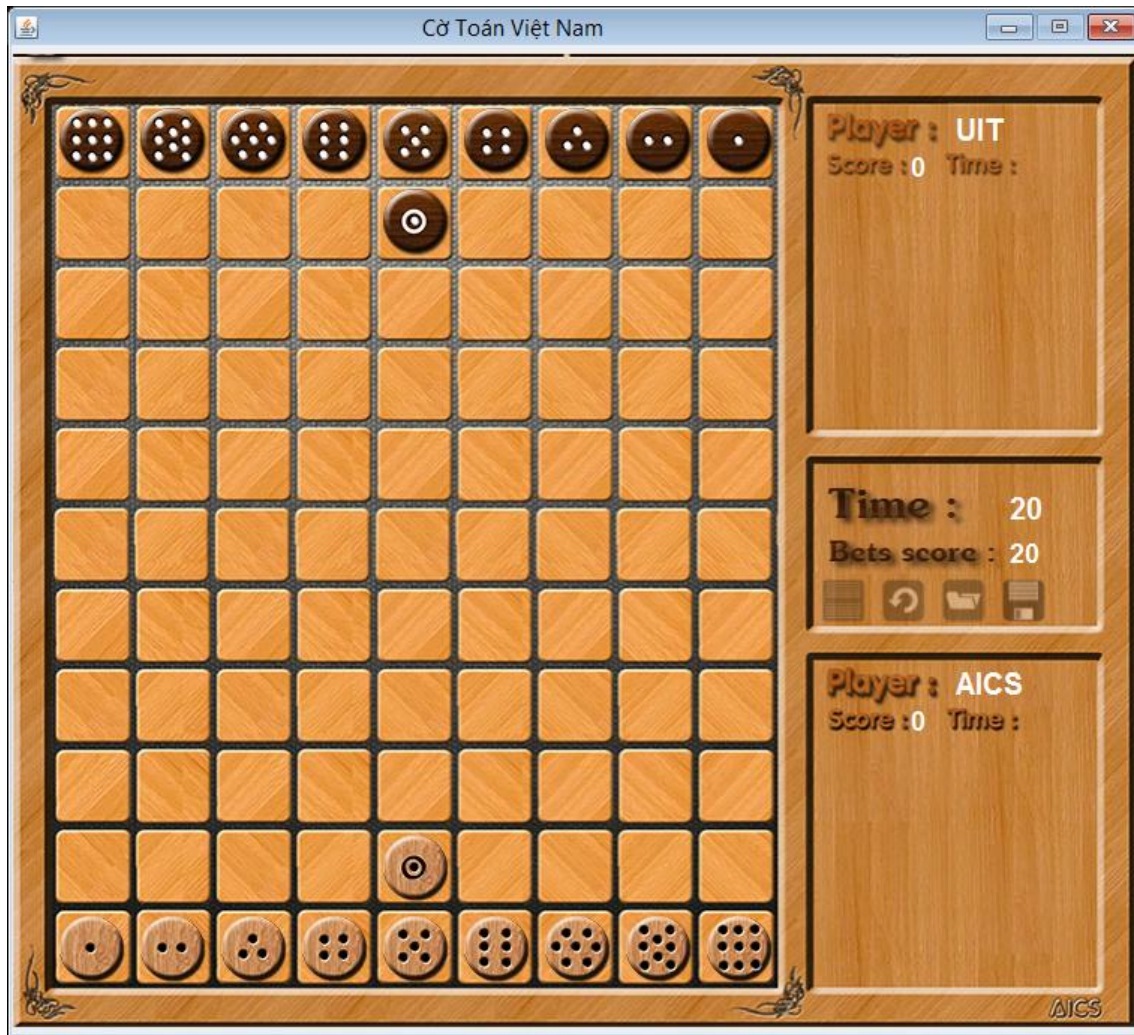


Hình 4.2: Chọn chế độ và đặt tên người chơi.

Giao diện gồm các nút chức năng: New Game, Undo , Open, Save

Trong đó:

-  ván mới có chức năng tạo ra ván cờ mới, trong đó bao gồm chọn chế độ chơi người hay máy, đặt tên người chơi và có chế độ cược điểm , đặt thời gian suy nghĩ cùng với 2 chức năng bắt đầu và thoát (hình 4.2).
-  Mở ván có chức năng mở lại những ván cờ đã được lưu trước đó.
-  Lưu ván cờ đang chơi.
-  Trở về nước đi trước đó.



Hình 4.3: Giao diện chính khi vào game.

IV.2. Ngôn ngữ sử dụng

Java trên nền java jdk 1.7

Công cụ: Eclipse Kepler.



IV.3. Các hàm chính trong chương trình.

1. Class BanCo.java
 - BanCo(): Khởi tạo bàn cờ của game.
 - addChess(): Thêm các hình ảnh của quân cờ vào bàn cờ.
 - mouseClicked(MouseEvent e): Các sự kiện khi click chuột vào các control trên bàn cờ (New Game, Undo, Open, Save).
2. Class OCo.java.
 - mouseClicked(MouseEvent e): Sự kiện khi click chuột vào ảnh jCanMove để di chuyển quân cờ.
3. Class Game.java.
 - Game(): Khởi tạo game.
4. Class Main.java.
 - Hàm main để chạy chương trình.
5. Class NewGame.java.
 - Khởi tạo form con để bắt đầu game.
6. Class VanCo.java.
 - addChess() : Thêm quân cờ vào bàn cờ.
 - putChess(QuanCo q, int i, int j) : đặt quân cờ q vào vị trí j,i trên bàn cờ.
 - eatChess(QuanCo q) : Hàm ăn quân cờ q.
 - saveGameLog() : Lưu tập các nước đi của quân cờ trong ván cờ.
 - undo() : Trở về nước đi trước của quân cờ.
 - save() : Lưu ván cờ đang chơi.
 - open() : Mở ván cờ đã lưu để chơi tiếp.
7. Class NgnoiChoi.java.
 - Khởi tạo người chơi của game.
8. Class QuanCo.java
 - mouseClicked(MouseEvent e) : Sự kiện click chuột vào quân cờ để di chuyển quân cờ.
 - createChess(...): Hàm khởi tạo quân cờ.
 - getChess(int i, int j) : Lấy quân cờ ở vị trí i,j trong bàn cờ.
 - drawChess() : Hàm vẽ quân cờ lên bàn cờ.
 - abstract int checkCheck(int i, int j) : Kiểm tra vị trí i,j quân cờ có được di chuyển đến không. Các lớp quân cờ con kế thừa (quân 0, quân 1..., quân 9).
 - move(int i, int j, int n) : xác định các vị trí mà quân cờ có thể di chuyển tới.
 - valueChess(int I, int j, int gtri) : Xác định toàn bộ các nước có thể đi được của quân cờ, kể cả những nước ăn quân đối phương.
 - moveAIQ() : Hàm sinh mọi nước đi của 1 quân cờ tại một thế cờ.
9. Util.java
 - loadImageCanMove() : load ảnh CanMove lên từ data.
 - loadImageChess() : load ảnh Quân cờ lên từ data
 - lockChess() : Khóa toàn bộ quân cờ trên bàn.
10. AI.java
 - saveGameLogAI(): Lưu các nước đi AI của quân cờ.

- `moveAI(QuanCo q, int i, int j)` : Hàm di chuyển quân cờ trong nước đi AI.
- `moveChessAI()` : Hàm sinh nước đi AI của quân cờ.
- `alphaBeta(QuanCo q, ...)` : Hàm Minimax kết hợp cắt tỉa Alpha –Beta để tìm ra nước đi tốt cho nước đi AI tiếp theo.

IV.4. Kết quả chương trình.

Xây dựng được một game Cờ Toán tương đối hoàn chỉnh.

Thuật toán xử lý AI giải quyết tương đối nhanh và cho kết quả chính xác.

CHƯƠNG V: TỔNG KẾT ĐÁNH GIÁ

V.1. Tổng kết.

Chương trình đã được hoàn thiện với đầy đủ các chức năng cơ bản và giao diện hoàn chỉnh của một game cờ đối kháng. Quá trình thực hiện nhóm có tham khảo một số đề tài ứng dụng game đối kháng thuộc loại bài toán tổng bằng 0 như: Cờ vua, cờ tướng, cờ caro...

Tuy vậy quá trình làm cũng gặp phải một số khó khăn :

- Là loại game còn khá mới, lạ lẫm nên việc tìm hiểu cách chơi và mẹo chơi mất nhiều thời gian.
- Là loại game có lượng bùng nổ lớn (mũ 78) lớn hơn rất nhiều so với các game tham khảo như cờ vua (mũ 16), cờ tướng (mũ 32)... nên việc quản lý bùng nổ và tối ưu cách đánh gây nhiều khó khăn.
- Chương trình không có open source nên nhóm phải đi hoàn toàn từ đầu.

Tuy gặp nhiều khó khăn trong việc tìm hiểu và cài đặt chương trình. Tuy nhiên các thành viên nhóm đều nỗ lực hết mình để hoàn thành công việc được giao.

Game Cờ Toán không được phổ biến và kinh nghiệm chơi chưa có nhiều vì thế còn nhiều sai sót, bất cập trong quá trình lập trình và còn nhiều hạn chế. Nhóm sẽ khắc phục chúng để chương trình tối ưu hơn.

Mọi sự góp ý xin gửi qua địa chỉ: <mailto:lamhx393@gmail.com>

V.2. Ưu điểm.

- Chương trình chạy tốt , có thể đánh ở chế độ 2 người chơi và đánh với máy.
- Thời gian tính toán nhanh.
- Tối ưu khá nhiều trong code và lập trình theo hướng đối tượng.

V.3 Hạn chế.

- Độ thông minh chưa cao.

- Độ sâu tính toán chưa lớn.
- Thuật toán phát sinh nhiều khi bị treo game.

V.4. Hướng phát triển.

Việc tìm hiểu và phát triển cờ trên máy tính sẽ là cơ sở để phát triển cho các ứng dụng thông minh. Đề án của nhóm sẽ được phát triển thêm các chức năng mới để giúp chương trình thực sự được hoàn thiện.

- Thể hiện vị trí vừa di chuyển.
- Xây dựng trên các thiết bị di động.
- Nâng cấp hàm tính giá trị để chia level cho game.
- Xây dựng đồ họa chuyên nghiệp.

TÀI LIỆU THAM KHẢO

- [1] Hướng dẫn chơi cờ ở diễn đàn: <http://cotoan.vnvista.com/>.
- [2] Nguyễn Thị Lệ, giải thuật tìm kiếm Minmax và ứng dụng trò chơi có tổng bằng không
- [3] Các tài liệu khác về giải thuật trên www.congdongviet.com/.
- [4] Các tài liệu về lập trình Java trên <http://congdongjava.com/>
- [5] Tham khảo các API trong Java trên <http://docs.oracle.com/javase/7/docs/api/>