

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO ĐỒ ÁN
MÔN HỌC NHẬP MÔN TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:

Trí tuệ nhân tạo trong trò chơi cờ Caro

Giáo viên hướng dẫn: TS. Nguyễn Nhật Quang

Nhóm sinh viên thực hiện:

<i>Họ và tên</i>	<i>MSSV</i>	<i>Lớp</i>
<i>Nguyễn Văn Thành</i>	<i>20194678</i>	<i>131403</i>
<i>Lê Huy Hoàng</i>	<i>20190053</i>	<i>131403</i>

Hà Nội, 2022

MỤC LỤC

NỘI DUNG	
1. Giới thiệu đề tài :	3
1.1. Lý do chọn đề tài:	3
1.2. Mô tả trò chơi:	4
2. Giải pháp:.....	6
2.1. Các thuật toán được sử dụng	6
A. Giải thuật tìm kiếm mini-max.....	7
2.2. Tổng quan giải thuật.....	7
2.3. Cách thức thực hiện	7
2.4. Áp dụng với trò chơi cờ Caro	8
2.5. Hàm lượng giá Heuristic	10
B. Phương pháp cắt tỉa anpha-beta.....	11
2.6. Tổng quan thuật toán.....	11
2.7. Cách thức thực hiện.....	11
2.8. Tối ưu và cải thiện thuật toán	13
3. Công nghệ triển khai	16
3.1. Các gói và thư viện hỗ trợ	16
4. Kết quả và đánh giá	17
4.1. Kết quả.....	17
4.2. Đánh giá.....	17
5. Các khó khăn, hạn chế và đề xuất cải tiến	18
5.1. Khó khăn.....	18
5.2. Hạn chế	18
5.3. Ý tưởng cải tiến	19
6. Kiến thức thu được	19
7. TÀI LIỆU THAM KHẢO	20

TRÒ CHƠI CỜ CARO

1. Giới thiệu đề tài

1.1. Lý do chọn đề tài

Ngày nay, trí tuệ nhân tạo (*artificial intelligence*) chắc chắn không còn quá xa lạ với mọi người, nhất là trong thời đại 4.0, thời đại bùng nổ của công nghệ. Một trong những ứng dụng đầu tiên của trí tuệ nhân tạo thông minh AI trên lĩnh vực game là trò chơi mang tên Nim được phát hành từ những năm 1951 – 1952. Tựa game này là một chương trình đánh Caro và cờ vua được Dietrich Prinz viết lên khi sử dụng cỗ máy Ferranti Mark 1. Tiếp sau đó là chương trình cờ Caro của Arthur Samuel. Kết quả vĩ đại nhất của các cỗ máy có tích hợp trí tuệ nhân tạo thông minh thời bấy giờ gọi tên Deep Blue của IBM, với thành tích đánh bại vua cờ Garry Kasparov trong năm 1997. Với mong muốn tìm hiểu các kiến thức về Trí tuệ nhân tạo đặc biệt là trong các trò chơi đối kháng, nhóm đã quyết định chọn tựa game Caro.

Tại sao là cờ caro mà không phải một trò chơi khác?

- Đây là một tựa game đối kháng có đồ họa khá đơn giản, từ đó thay vì phải tập trung quá nhiều vào đồ họa thì nhóm có thể tập trung vào phát triển và hiệu thuật toán.
- Trò chơi có mức độ phổ biến cao, có sự hứng thú trong quá trình phát triển và cả người thử nghiệm trò chơi.
- Phù hợp và có thể áp dụng các thuật toán đã học trên lớp, có nhiều phiên bản trước đó để so sánh.

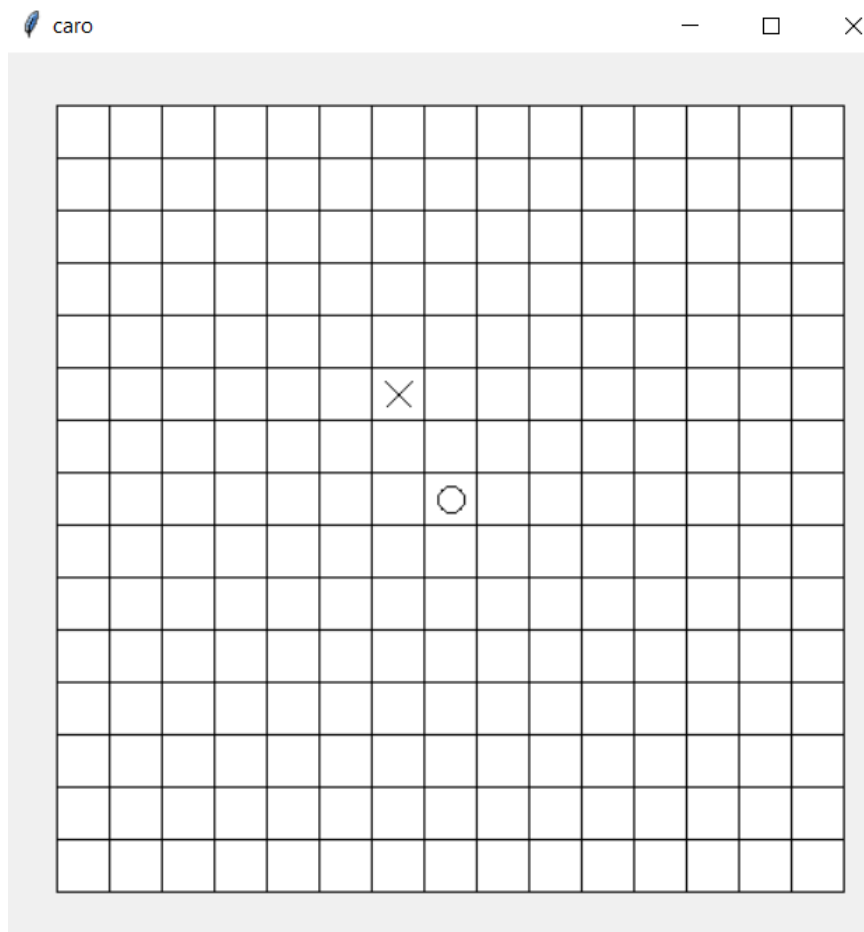
Từ những lý do đó, nhóm đã quyết định phát triển một chương trình chơi cờ Caro sử dụng trí tuệ nhân tạo với mong muốn chiến thắng con người.

1.2. Mô tả bài toán

Bài toán: Nhóm sẽ xây dựng một chương trình chơi cờ Caro để máy tính có thể thông minh hơn và chiến thắng được người chơi tuân thủ các luật cố định của trò chơi. Sử dụng các thuật toán đã biết kết hợp tri thức từ kinh nghiệm thực tế bổ sung để đánh bại chính con người.

Trò chơi cờ Caro:

a, Giao diện



Giao diện chương trình đánh cờ Caro

Giao diện bàn cờ Caro là một lưới ô vuông kích thước $n \times n$ (n thường là các số lẻ như 11,13,15,17...) . Với chương trình của nhóm, nhóm sử dụng bàn cờ 15×15 ($n=15$) do đây là bàn cờ tiêu chuẩn thi đấu Caro trên thế giới.

b, Cách chơi

Một số thuật ngữ :

Quân cờ: trong game sử dụng bộ dấu ký hiệu X/O .

Bàn cờ Caro

Sử dụng bàn cờ kích thước 15×15 (ô vuông)

Trên một bàn cờ chỉ cho phép 2 người tham gia.

Lượt chơi (tick X/O)

Một ván cờ có thể gồm nhiều lượt chơi.

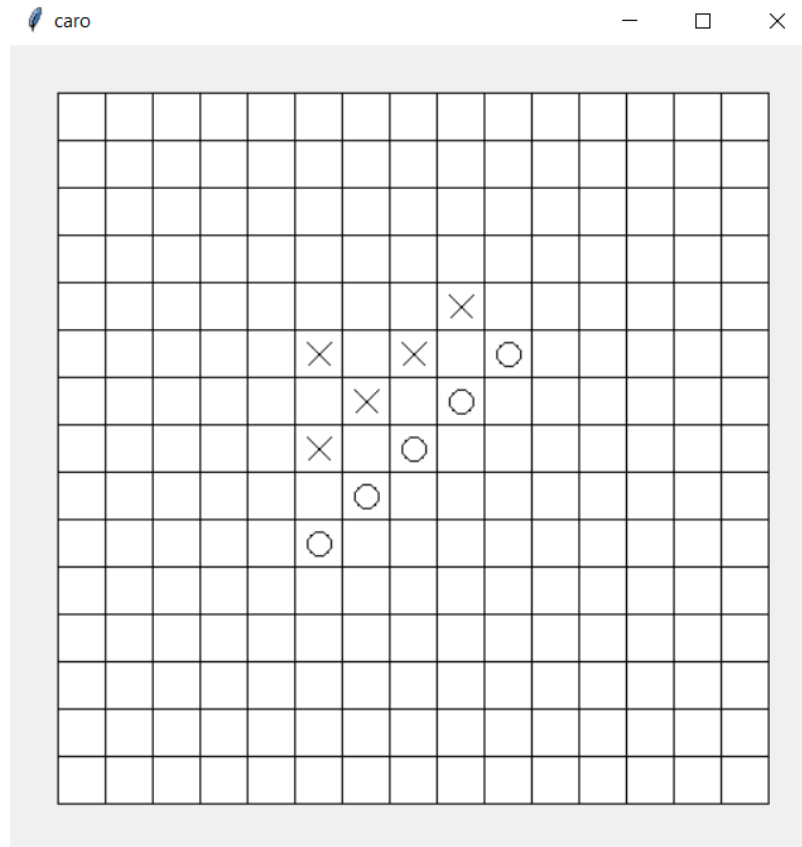
Mỗi lượt người chơi thực hiện tick các dấu X/O vào hình vuông trên bàn cờ.

Nước đi đầu:

Trong ván đầu tiên: chủ bàn là người đi đầu.

Các ván tiếp theo: người đi đầu là người thua cuộc của ván trước đó.

Luật cơ bản :



Quân O dành chiến thắng

- Một ván đấu được bắt đầu khi có đủ 2 người trên bàn.
- Người được quyền đi trước: đánh vào vị trí bất kỳ trên bàn cờ
- Khi tới lượt: người chơi phải tick vào một ô trên bàn cờ. Tick đủ 5 ô theo chiều dọc, chiều ngang hoặc đường chéo thì sẽ thắng.
- Trường hợp hòa xảy ra khi đã đánh hết các ô mà chưa có người chiến thắng.

2. Giải pháp:

2.1. Các thuật toán được sử dụng

A. Thuật toán Mini-max

2.2. Tổng quan về giải thuật

Giải thuật mini-max là gì?:

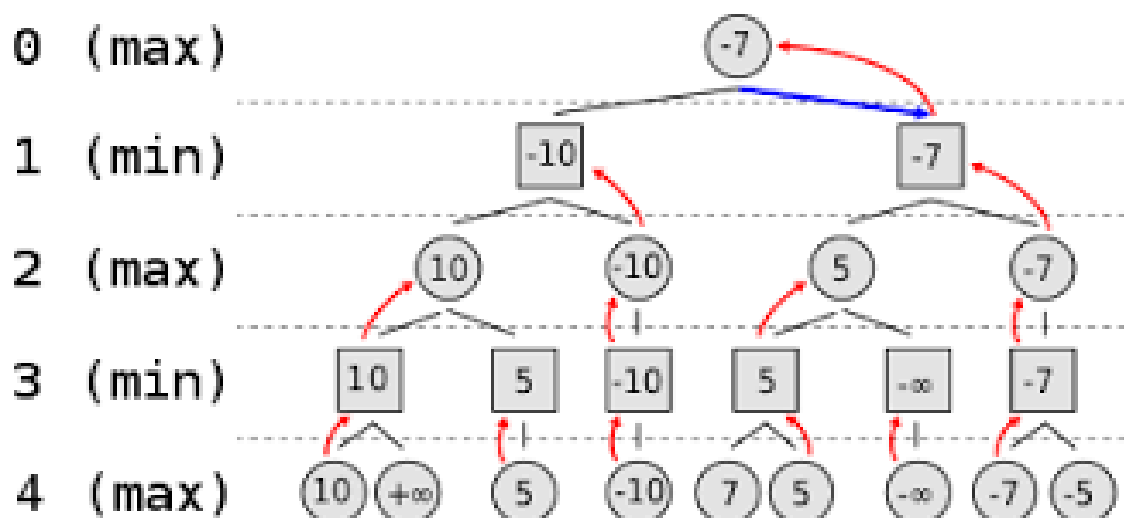
Minimax là giải thuật là một thuật toán đệ quy lựa chọn bước đi kế tiếp trong một trò chơi bằng cách định giá trị cho các Node trên cây trò chơi sau đó tìm Node có giá trị phù hợp để đi bước tiếp theo.

Mini-max có mục đích là tối thiểu hóa (minimize) tổn thất vốn được dự tính có thể là "tối đa" (maximize). Có thể hiểu ngược lại là, nó nhằm tối đa hóa lợi ích vốn được dự tính là tối thiểu (maximin).

Chiến lược của MAX bị ảnh hưởng (phụ thuộc) vào các nước đi của MIN và ngược lại. MAX cần chọn một chiến lược giúp cực đại hóa giá trị hàm mục tiêu –với giả sử là MIN đi các nước đi tối ưu. MIN cần chọn một chiến lược giúp cực tiểu hóa giá trị hàm mục tiêu.

Chiến lược này được xác định bằng việc xét giá trị MINIMAX đối với mỗi nút trong cây biểu diễn trò chơi. Chiến lược tối ưu đối với các trò chơi có không gian trạng thái xác định (deterministic states). MAX chọn nước đi ứng với giá trị MINIMAX cực đại (để đạt được giá trị cực đại của hàm mục tiêu). Ngược lại, MIN chọn nước đi ứng với giá trị MINIMAX cực tiểu.

2.3. Cách thức thực hiện



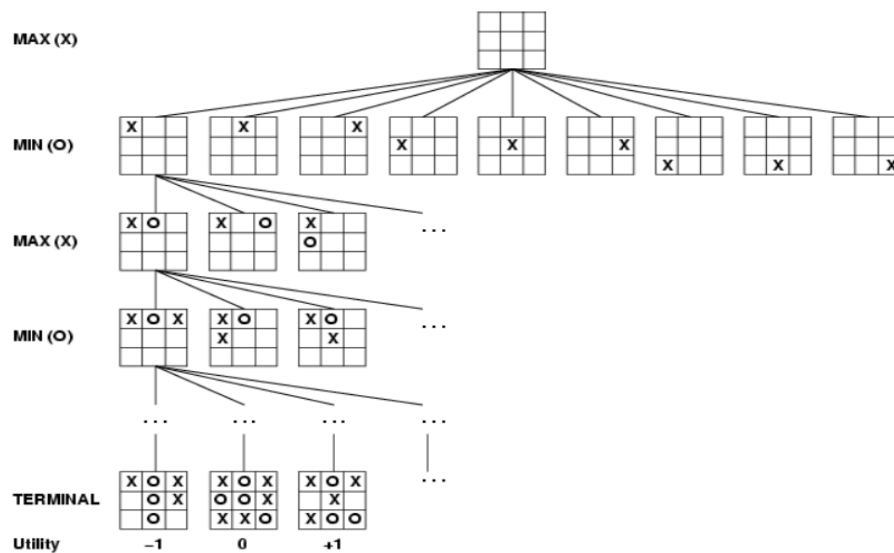
Cách thức chọn các giá trị hàm trên cây tuân theo giải thuật mini-max

Hai người chơi trong game được đại diện là MAX và MIN. MAX đại diện cho người chơi luôn muốn chiến thắng và cố gắng tối ưu hóa ưu thế của mình còn MIN đại diện cho người chơi cố gắng cho người MAX giành số điểm càng thấp càng tốt. Giải thuật Minimax thể hiện bằng cách định trị các Node trên cây trò chơi: Node thuộc lớp MAX thì gán cho nó giá trị lớn nhất của con Node đó. Node thuộc lớp MIN thì gán cho nó giá trị nhỏ nhất của con Node đó. Từ các giá trị này người chơi sẽ lựa chọn cho mình nước đi tiếp theo hợp lý nhất.

2.4. Áp dụng mini-max vào trò chơi cờ Caro

a, Tic-tac-toe (Caro 3×3)

Một phiên bản của giải thuật áp dụng cho các trò chơi như *tic-tac-toe* (phiên bản 3×3 của game Caro) khi mà mỗi người chơi có thể thắng, thua, hoặc hòa. Nếu người chơi A có thể thắng trong 1 nước đi, thì "nước đi tốt nhất" chính là nước đi để dẫn đến kết quả thắng đó. Nếu người B biết rằng có một nước đi mà dẫn đến tình huống người A có thể thắng ngay ở nước đi tiếp theo, trong khi nước đi khác thì sẽ dẫn đến tình huống mà người chơi A chỉ có thể, tốt nhất, là hòa thì nước đi tốt nhất của người B chính là nước đi sau.



Giải thuật mini-max vét cạn không gian tìm kiếm trong tic-tac-toe

Do bàn cờ có kích thước nhỏ, việc vét cạn tất cả các trường hợp là hoàn toàn có thể diễn ra ở bàn cờ 3×3 để tìm ra nước đi tối ưu.

Hàm lượng giá cũng rất đơn giản bằng việc gán giá trị -1 cho kết quả thua ở độ sâu cuối cùng (bất lợi cho người đánh), 0 cho kết quả hòa và 1 cho kết quả thắng (thuận lợi cho người đánh)

b, Caro (gomoku)

Đối với trò chơi Caro kích thước $n \times n$ có sự tương đồng với trò chơi tic-tac-toe, điểm khác biệt ở chỗ bàn cờ lớn hơn và số nước liên tiếp để thắng là 5 quân cờ.

Bàn cờ Caro được tăng kích thước lên $n \times n$, đồng nghĩa với việc không gian tìm kiếm tăng lên rất nhiều lần qua mỗi độ sâu.

Điều gì xảy ra khi chúng ta xét toàn bộ không gian tìm kiếm như trò chơi tic-tac-toe ?

Ta sẽ so sánh số nước cờ trong không gian tìm kiếm cần so sánh để tìm ra nước đi đầu tiên giữa bàn cờ 3×3 và bàn cờ 15×15 nếu vét cạn không gian !!!

Với bàn cờ 3×3

- *Với độ sâu dept=1 ta có 9 trường hợp các nước đi*
- *Với độ sâu dept=2 ta có $9 \times 8 = 72$ trường hợp các nước đi*
- *Với độ sâu dept=2 ta có $9 \times 8 \times 7 = 504$ trường hợp các nước đi*
-
- ...
- *Với độ sâu dept=9 ta có $9! = 362880$ trường hợp các nước đi*

Với bàn cờ 15×15

- *Với độ sâu dept=1 ta có $15 \times 15 = 225$ trường hợp các nước đi*
- *Với độ sâu dept=2 ta có $225 \times 224 = 50400$ trường hợp các nước đi*
- *Với độ sâu dept=3 ta có $225 \times 224 \times 223 = 11239200$ trường hợp các nước đi*
-
- ...
- *Với độ sâu dept=225 ta có $225!$ trường hợp các nước đi – một con số rất rất lớn !!!*

Càng về các nước đi cuối không gian sẽ nhỏ đi nhưng vẫn là rất rất lớn nếu ta xét tới độ sâu tối đa.

Mặc dù việc vét cạn không gian tìm kiếm có thể tìm được nước đi tốt nhất tuy nhiên với việc không gian tìm kiếm quá rộng. Nhóm đã tiến hành giải thuật mini-max với độ sâu nhất định ($dept=1,2,3,4,\dots$) với mong muốn giảm bớt không gian tìm kiếm nước đi phù hợp với trò chơi.

2.5. Hàm lượng giá Heuristic

Với việc xét không gian tìm kiếm tới một độ sâu nhất định, ta cần một hàm lượng giá đánh giá đúng và chính xác về mức độ thuật lợi của nước đi tại độ sâu đó.

Thông qua việc tham khảo kinh nghiệm chơi cờ, nhóm đã xây được hàm lượng giá như sau :

Với mỗi một *nước cờ* trong không gian hàm lượng giá sẽ có giá trị bằng tổng giá trị các *thế cờ thuận lợi* có trong bàn cờ tại thời điểm đó.

Ta sẽ có các *thế cờ thuận lợi* như sau (ký hiệu 1 đại diện cho ký hiệu của quân cờ đối tượng đang đánh, ký hiệu R đại diện cho 1 ô trống chưa ai đánh) :

- *Thế chắc thắng* có dạng $(1,1,1,1,1) + 100000$ điểm
- *Thế rất thuật lợi* có dạng $(1,1,1,1,R)$, $(1,1,1,R,1)$, $(1,1,R,1,1)$, $(1,R,1,1,1)$, $(R,1,1,1,1) + 10000$ điểm (4 trên 5 quân cùng ký hiệu, 1 ô trống)
- *Thế thuật lợi* có dạng $(1,1,1,R,R)$, $(1,1,R,R,1)$, $(1,R,R,1,1)$, $(R,R,1,1,1)$, $(R,1,R,1,1), \dots + 500$ điểm (3 trên 5 quân cùng ký hiệu, 2 ô trống)
- *Thế trung bình* có dạng $(1,1,R,R,R)$, $(1,R,R,R,1)$, $(R,R,R,1,1)$, $(R,R,1,R,1)$, $(R,1,R,1,R), \dots + 10$ điểm (2 trên 5 quân cùng ký hiệu, 3 ô trống)
- *Thế thuận lợi kém* có dạng $(1,R,R,R,R)$, $(R,1,R,R,R)$, $(R,R,1,R,R)$, $(R,R,R,R,1)$, $+ 1$ điểm (1 trên 5 quân cùng ký hiệu, 4 ô trống)

Điểm của hàm heuristic bằng tổng điểm các thế cờ thuận lợi tại bàn cờ tại thời điểm đó. Giả sử trên bàn cờ có 2 thế rất thuận lợi, 1 thế thuận lợi, 1 thế trung bình thì tổng điểm sẽ là $10000+500+10=10510$ điểm

B. Phương pháp cắt tỉa Anpha-Beta

2.6. Tổng quan thuật toán

Vấn đề : Giải thuật tìm kiếm MINIMAX vấp phải vấn đề bùng nổ (mức hàm mũ) các khả năng nước đi cần phải xét → không phù hợp với nhiều bài toán trò chơi thực tế khi đề cao thời gian chạy, xử lý (*runtime*).

Chúng ta có thể cắt tỉa (bỏ đi – không xét đến) một số nhánh tìm kiếm trong cây biểu diễn trò chơi.

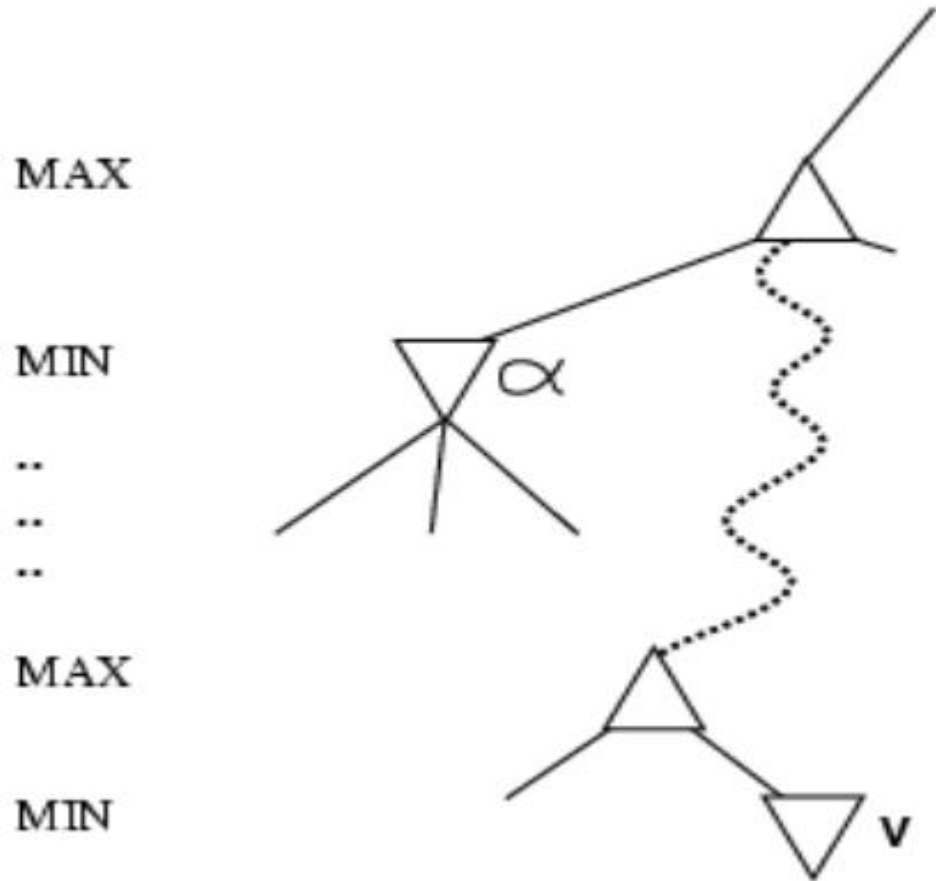
Phương pháp cắt tỉa Anpha-Beta (Alpha-beta pruning)

Ý tưởng: Nếu một nhánh tìm kiếm nào đó không thể cải thiện giá trị hàm tiện ích mà MAX (hoặc MIN) đã có, thì MAX (hoặc MIN) không cần xét đến nhánh tìm kiếm đó nữa (ta “cắt tỉa” nhánh tìm kiếm đó). Việc cắt tỉa các nhánh tìm kiếm “tồi” sẽ không ảnh hưởng đến kết quả cuối cùng (chất lượng của nước đi tìm được)

2.7. Cách thức thực hiện

Nút Max có một giá trị alpha (lớn hơn hoặc bằng alpha – luôn tăng), nút min có một giá trị beta (nhỏ hơn hoặc bằng beta – luôn giảm). Khi chưa có alpha và beta xác định thì thực hiện tìm kiếm sâu (depth-first) để xác định được alpha, beta, và truyền ngược lên các nút cha.

Giả sử ta có 1 cây như hình dưới đây :



α là giá trị của nước đi tốt nhất đối với MAX (giá trị tối đa) tính đến hiện tại đối với nhánh tìm kiếm.

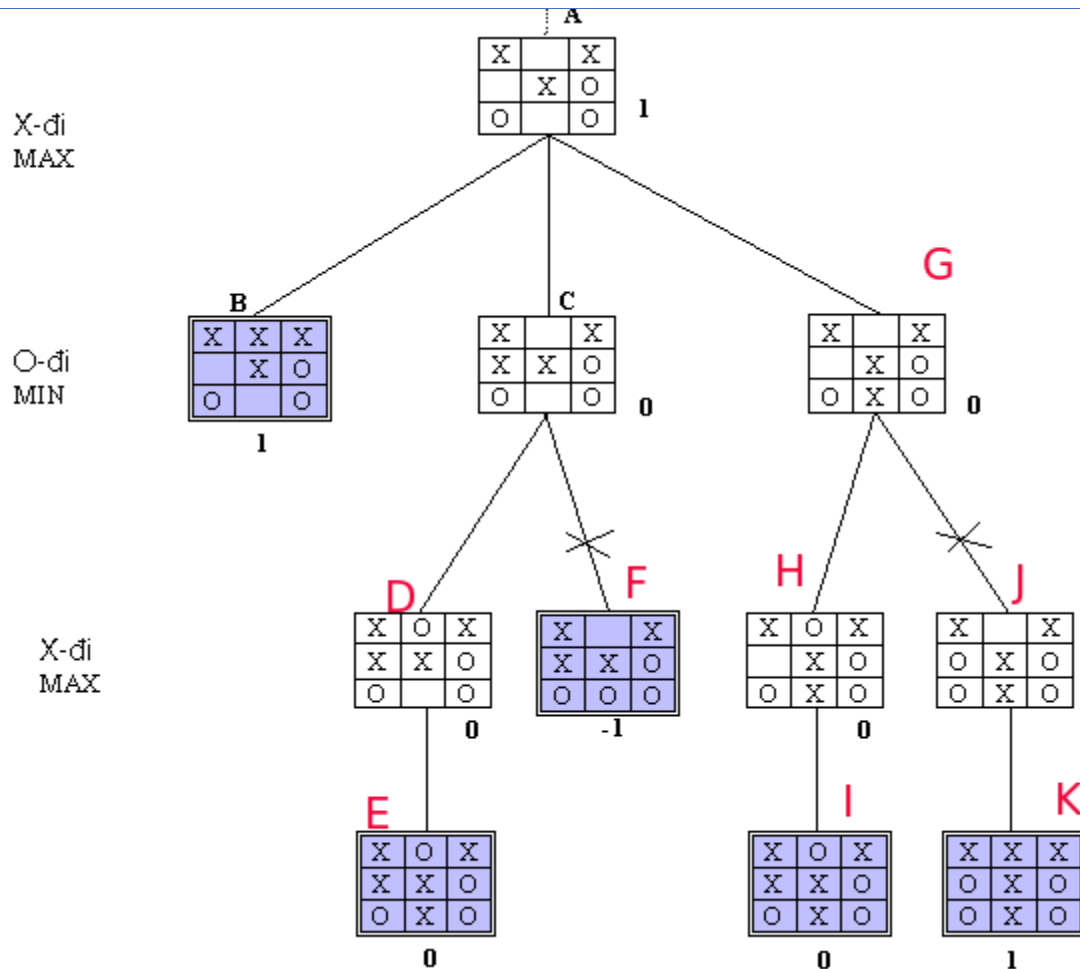
Nếu v là giá trị tồi hơn α , MAX sẽ bỏ qua nước đi ứng với v

--->Cắt tỉa nhánh ứng với v

β được định nghĩa tương tự đối với MIN.

Việc cắt tỉa sẽ làm giảm các nhánh tìm kiếm không cần thiết.

Một ví dụ cắt tỉa trong bàn cờ 3×3



Áp dụng phương pháp cắt tỉa Anpha-Beta cho tic-tac-toe

2.8. Tối ưu và cải thiện thuật toán

a, Phiên bản sử dụng thuần giải thuật mini-max

Đây là phiên bản đầu tiên mà nhóm xây dựng khi sử dụng thuần giải thuật mini-max. Khi xét hết tất cả các trường hợp của cây trò chơi. Mặc dù có thể tìm được nước đi tối ưu cục bộ tuy nhiên do chưa áp dụng các phương pháp làm giảm thời gian tính toán nên thời gian tính toán rất chậm.

b, Phiên bản sử dụng giải thuật mini-max kết hợp phương pháp cắt tỉa anpha-beta

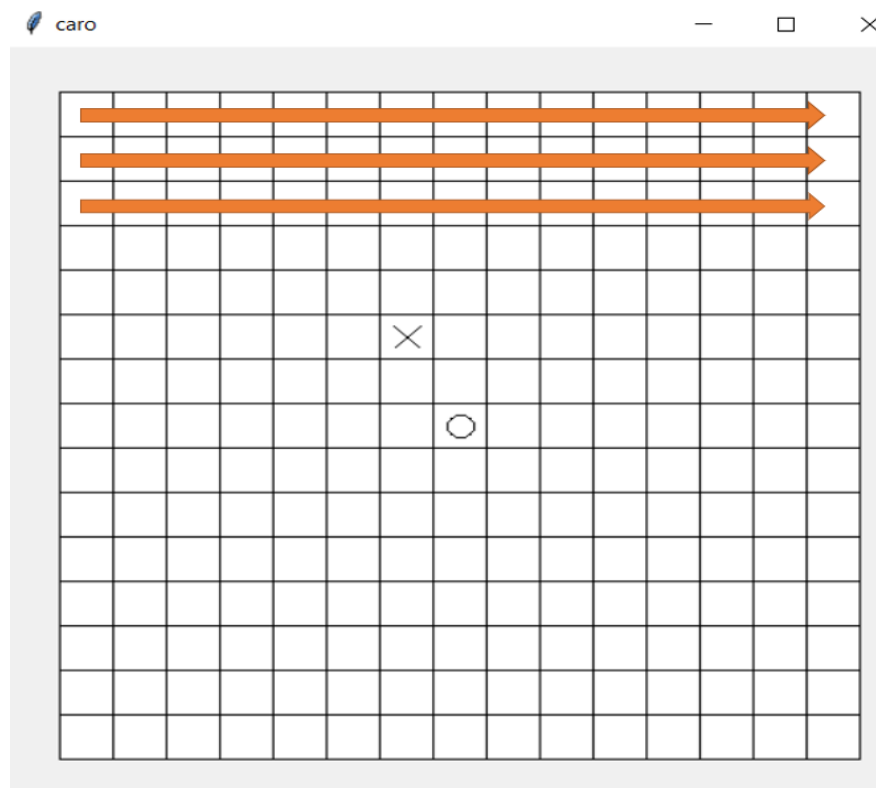
Với áp dụng việc cắt tỉa trong quá trình tìm nước đi trong cây trò chơi. Thời gian tính toán đã giảm rõ rệt do đã giảm bớt các trường hợp phải xét.

c, Phiên bản sử dụng giải thuật mini-max kết hợp phương pháp cắt tỉa anpha-beta, duyệt các nước lân cận

Thay vì duyệt hết các ô chưa đánh trong cây trò chơi, thì chỉ duyệt các ô lân cận với các ô đã đánh và mới đánh(các ô mà nhóm cho rằng quan trọng hơn các ô ở xa) điều này làm giảm thời gian tính toán khi thu hẹp được không gian tìm kiếm

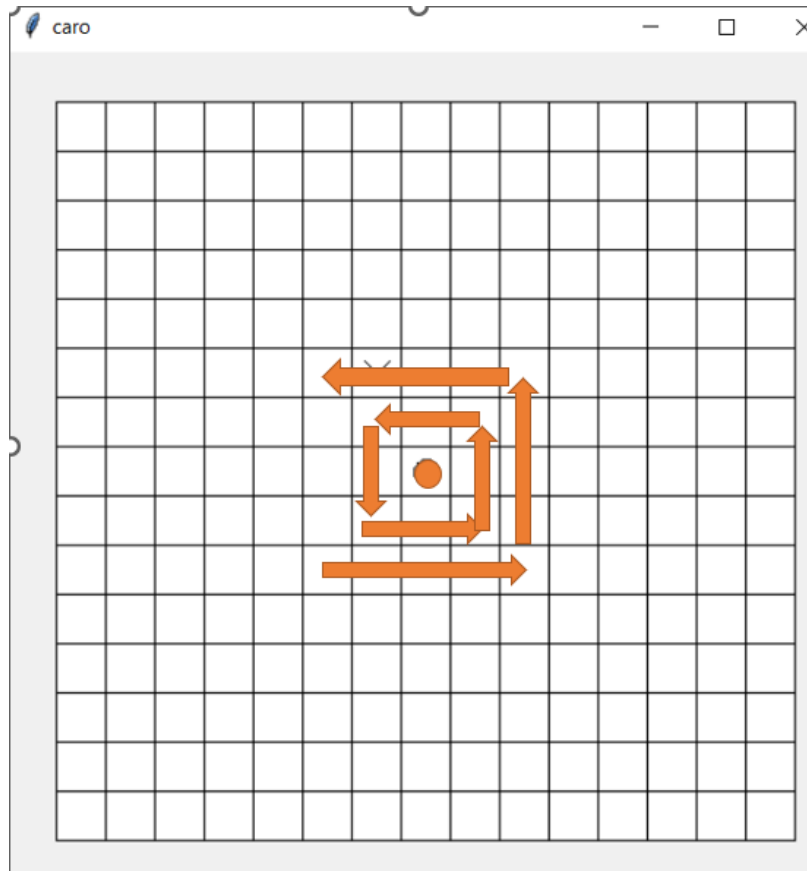
d, Thay đổi thứ tự duyệt trong cây trò chơi

Việc cắt tỉa anpha-beta phụ thuộc nhiều vào việc ta sẽ gặp nốt tối ưu sớm hay muộn. Nếu gặp sớm ta sẽ cắt tỉa được nhiều nhánh. Ngược lại nếu nốt tối ưu được duyệt cuối, các nhánh cắt tỉa sẽ bị giảm bớt dẫn tới kém hiệu quả.



Duyệt các ô trên cây trò chơi thông thường

Thông thường cách duyệt của mọi người sẽ là từ trái qua phải, từ trên xuống dưới. Tuy nhiên với kinh nghiệm chơi cờ, nhóm nhận thấy người chơi thường có xu hướng đánh vào các ô trung tâm để lấy lợi thế về không gian trò chơi. Vì vậy các ô có trọng số cao thường rơi vào lân cận các ô trung tâm. Vì vậy giải pháp nhóm đưa ra là đổi cách duyệt tuần tự thành *duyet hình xoắn ốc* với hi vọng tìm được các ô tối ưu sớm hơn.



Duyệt các ô theo xoắn ốc

e, Cập nhật hàm lượng giá Heuristic

Việc cập nhật giá Heuristic được cập nhật đệ quy theo bước trước thay vì việc xét duyệt toàn bộ bàn cờ để tìm ra số *thế cờ thuận lợi*.

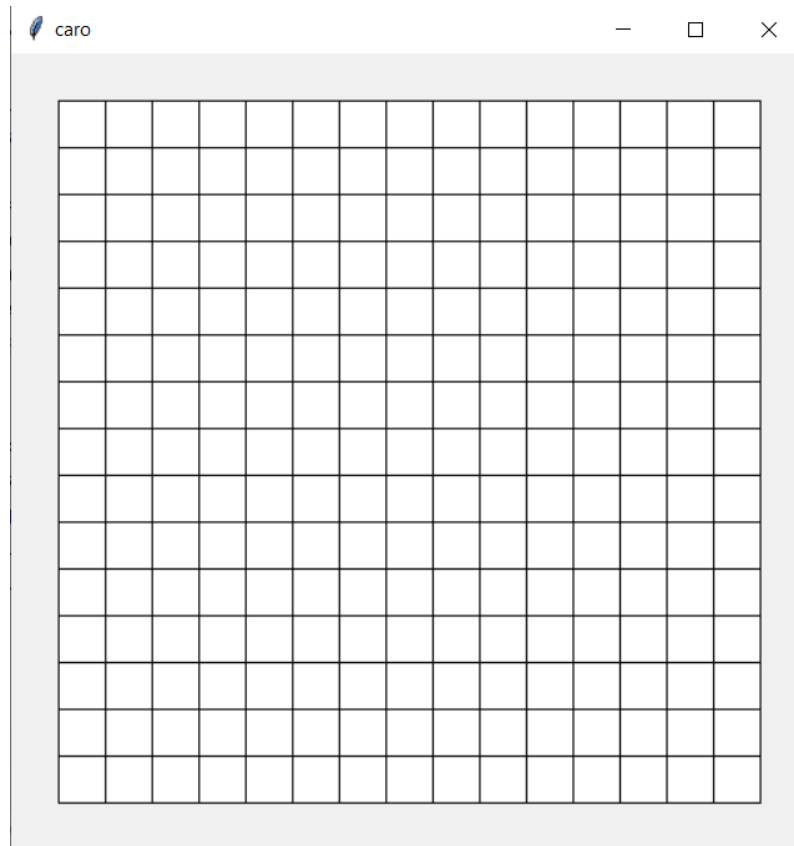
Giả sử ở bước đánh trước hàm lượng giá là x điểm. Khi người chơi đánh 1 nước hàm lượng giá chỉ xét trong phạm vi xung quanh ô đó (xét vùng 9×9 với trung tâm là ô đang đánh). Việc cập nhật như sau:

- Trong vùng 9×9 trung tâm là ô vừa đánh, ta cộng thêm điểm thế cờ lợi thế ứng quân cờ vừa đánh. (Điểm cập nhật ở phần 2.5)

- Đồng thời trừ bớt số điểm quân vừa đánh ảnh hưởng tới các nước lợi thế của đối thủ ở các nước trước đó trong vùng đang xét. (Điểm cập nhật ở phần 2.5)

3. Công nghệ triển khai

3.1. Các gói và thư viện hỗ trợ



Giao diện đồ họa trong chương trình

- Nhóm đã sử dụng ngôn ngữ Python để lập trình trò chơi.
- Về phần GUI, giao diện nhóm sử dụng **Tkinter** - là thư viện GUI tiêu chuẩn cho Python. Tkinter trong Python cung cấp một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI.
- Về phần thuật toán, nhóm không sử dụng thêm các thư viện nào khác.

4. Kết quả và đánh giá

4.1. Kết quả

- ✓ Máy đã đi các nước đi đúng luật đề ra.
- ✓ Các phần đồ họa game đã hoàn thiện ở mức độ cơ bản và thân thiện với người dùng.
- ✓ Chương trình đã có thể đánh cờ ngang ngửa với các thành viên trong lớp (đều đã biết chơi cờ Caro từ mức khá).
- ✓ Đã áp dụng được tri thức của con người (hàm heuristic) để tính toán các nước đi phù hợp.

4.2. Đánh giá

- *Giải thuật mini-max kết hợp cắt tỉa anpha-beta có sự cải tiến rõ rệt so với giải thuật mini-max thuần túy, do đã lược bỏ các nhánh không cần xét đến trong cây trò chơi.*
- *Việc duyệt thuật toán duyệt các phần tử lân cận với các nước đi đã đánh so với việc duyệt toàn bộ các ô làm tăng đáng kể thời gian, tuy nhiên hiệu quả sẽ không bằng thuật toán duyệt toàn bộ.*
- *Độ sâu có vai trò quan trọng, then chốt đến chất lượng của nước đi, càng tăng độ sâu các nước đi càng chính xác tuy nhiên đánh đổi lại về mặt thời gian sẽ tăng lên.*

5. Khó khăn và giải pháp cải tiến

5.1. Khó khăn

- Do đề tài được phân theo nhóm nên đã có những sự khó khăn trong quá trình làm việc nhóm như bất đồng quan điểm hay xung đột về thời gian họp nhóm,....
- Phong cách lập trình khác nhau khiến việc triển khai coding gặp nhiều khó khăn. Ví dụ như cách đặt tên biến, tên file, không comment code,...

5.2. Hạn chế

- Chưa xây dựng được như một phần mềm hoàn chỉnh thân thiện với người sử dụng.
- Chưa có thời gian tính giờ cho mỗi lượt đi.
- Chưa có các thông báo thắng cuộc/ thua cuộc trong game.

5.3. Giải pháp cải tiến

a, Về thời gian chạy

Về mặt ngôn ngữ lập trình, nhóm dự định trong tương lai sẽ sử dụng các ngôn ngữ có tốc độ cải thiện so với python(ví dụ java, C++,...) .

b, Về giải pháp mới

Đề xuất phương pháp mới : Reinforcement Learning cho game cờ Caro.

Trong khi trong các phương pháp lý thuyết trò chơi nói chung, ví dụ thuật toán min-max, thuật toán luôn giả định chúng ta có một đối thủ hoàn hảo, công việc phải thực hiện là tối đa hóa phần thưởng của mình và giảm thiểu phần thưởng của đối thủ (tối đa hóa điểm của mình và tối thiểu hóa điểm của đối thủ), trong học củng cố, chúng ta không cần giả định đối thủ của chúng ta là 1 thiên tài xuất chúng, nhưng chúng ta vẫn thu được mô hình với kết quả rất tốt. Bằng cách coi đối thủ là một phần của môi trường mà chúng ta có thể tương tác, sau một số lần lặp lại nhất định, đối thủ có thể lập kế hoạch trước mà không cần chúng ta phải làm gì cả. Ưu điểm của phương pháp này là

giảm số lượng không gian tìm kiếm và giảm số phép toán suy luận phải thực hiện, nhưng nó có thể đạt được kỹ năng hiện đại chỉ bằng cách thử và học.

Nhóm đã đọc qua ý tưởng và có thể thực hiện trong thời gian sắp tới.

(nguồn <https://www.phamduytung.com/blog/2020-12-26---tic-tac-toe/>)

6. Kiến thức thu được

Qua quá trình làm việc nhóm hoàn thành đề án môn học nhóm đã học hỏi và tiếp thu được nhiều bài học quý báu.

- ✓ Cách làm việc nhóm, bàn luận và đưa ra cách ý tưởng để triển khai trong chương trình.
- ✓ Các kiến thức, nội dung trong bài giảng được cụ thể hóa bằng chương trình cụ thể giúp bản thân hiểu rõ hơn về bài học (cụ thể là giải thuật mini-max và phương pháp cắt tỉa Anpha-Beta)
- ✓ Cách giải quyết vấn đề, tối ưu chương trình, khai thác các kiến thức kinh nghiệm vào giải thuật.
- ✓ Các giai đoạn, quy trình phát triển một chương trình được lên kế hoạch và triển khai một cách rõ ràng.

TÀI LIỆU THAM KHẢO

Slide bài giảng “Nhập môn Trí tuệ nhân tạo” – TS. Nguyễn Nhật Quang

<https://vi.wikipedia.org/wiki/Minimax>

<https://viblo.asia/p/thuat-toan-minimax-ai-trong-game-APqzeaVVzVe>

<https://www.stdio.vn/giai-thuat-lap-trinh/giai-thuat-tim-kiem-minimax-s1EVnH>

<https://www.stdio.vn/giai-thuat-lap-trinh/giai-thuat-cat-tia-alpha-beta-Wu7F1>

<https://thegioibongda.com.vn/ky-thuat-cat-tia-alpha-beta-giai-bai-toan-tro-choi-doi-khang/>

<https://viettuts.vn/python-tkinter>

