ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

MÔN HỌC: CÁC THUẬT TOÁN THÔNG MINH NHÂN TẠO VÀ ỨNG DỤNG

Giảng viên: Nguyễn Ngọc Đức

Đề tài: Ứng dụng các phương pháp tìm kiếm cho việc thiết kế một tác tử (agent) trong một trò chơi đối kháng cụ thể.

Các thành viên tham gia:

Hồ Thiên Phúc – 1760147

Trương Văn Đắc – 1760274

Nguyễn Trung Nam – 18600177

Thành phố Hồ Chí Minh, tháng 1 năm 2022

Mục Lục

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin cảm ơn Ban giám hiệu nhà trường đã mở lớp để chúng em có cơ hội học tập, hoàn thành khóa học cuối cùng tại trường.

Tiếp theo, nhóm em xin cảm ơn đến Thầy **Nguyễn Ngọc Đức** – giảng viên bộ môn "Các thuật toán thông minh nhân tạo và ứng dụng", đã trang bị cho nhóm em những kiến thức, kỹ năng cần có để hoàn thành đồ án kết thúc môn lần này. Cảm ơn thầy đã tận tình giúp đỡ, hướng dẫn và phản hồi mail về những thắc mắc của nhóm trong quá trình làm bài.

Tuy nhiên trong quá trình hoàn thành đồ án, chắc chắn sẽ có những thiếu, sai xót, do đây là lần đầu nhóm em được tiếp xúc với bộ môn, cũng như đồ án này nên còn hạn chế. Mong thầy đóng góp ý kiến để nhóm hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn, kính chúc thầy sức khỏe và hạnh phúc!

I. Thông tin nhóm

- Nhóm 1
    - Hồ Thiên Phúc - 1760147
    - Trương Văn Đắc - 1760274
    - Nguyễn Trung Nam – 18600177

II. Thuật toán

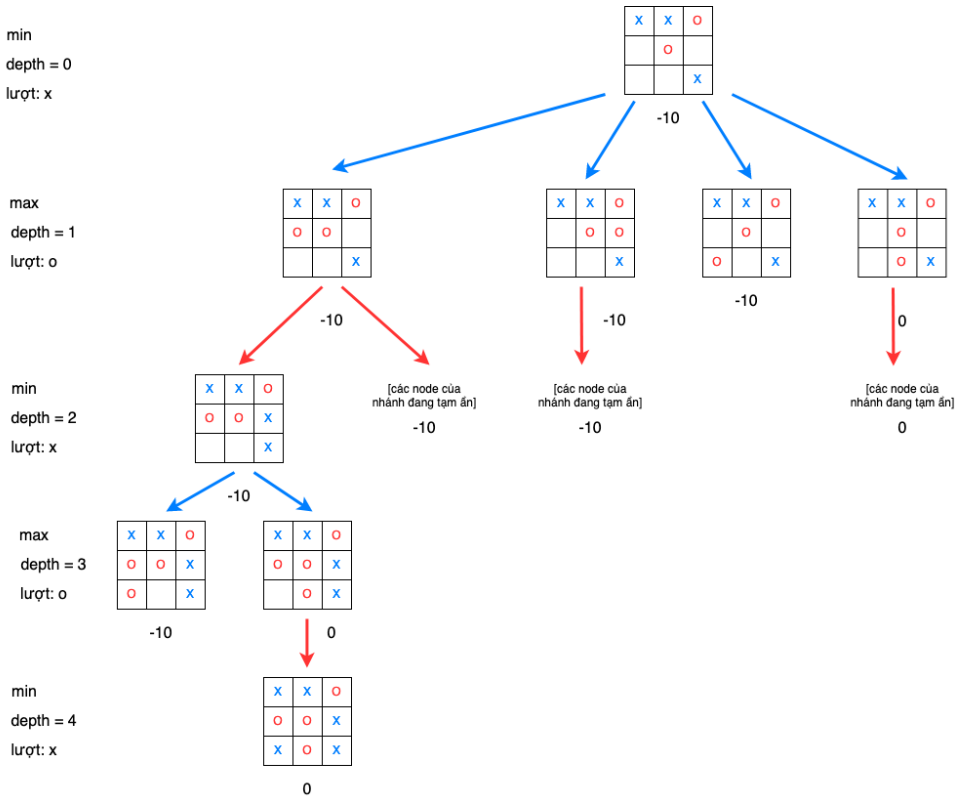
a. Thuật toán MiniMax

Thuật toán được sử dụng trong trò chơi này là thuật toán MiniMax. Minimax (còn gọi là minmax) là một phương pháp trong lý thuyết quyết định có mục đích là tối thiểu hóa (minimize) tổn thất vốn được dự tính có thể là "tối đa" (maximize). Có thể hiểu ngược lại là, nó nhằm tối đa hóa lợi ích vốn được dự tính là tối thiểu (maximin). Minimax 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.

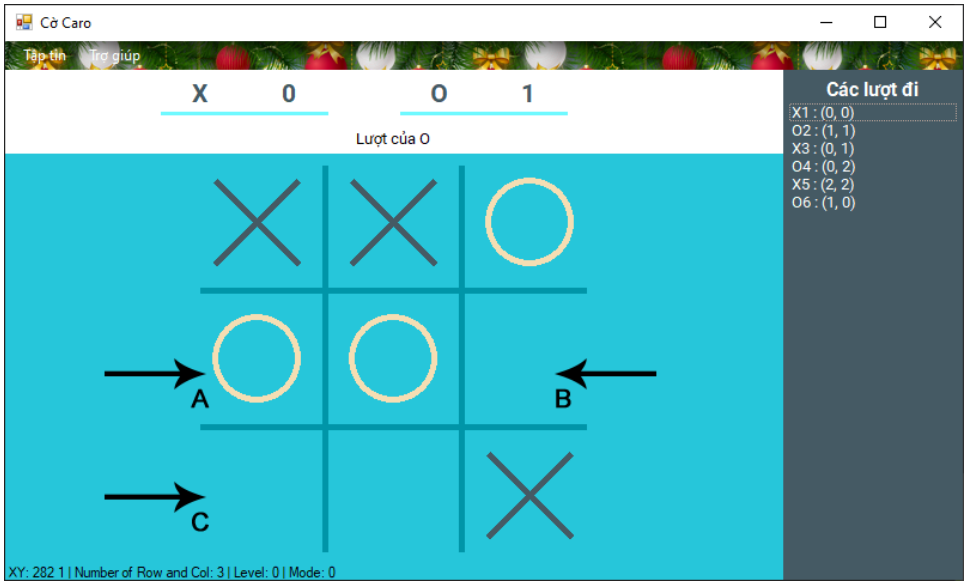
Tại sao cần dùng minimax trong trò chơi này? Hiện tại có rất nhiều thuật toán tìm kiếm để làm tác nhân thông minh trong game như A*, Heuristic... Mỗi thuật toán thì sẽ phù hợp với từng loại game cho nó. Những game đối kháng trong đối người chơi luân phiên đánh như cờ vua, cờ tường, caro... Người chơi 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 mình như thế nào? Cách xử lý đơn giản là giả sử đối thủ cũng có các nước đi tối ưu giống mình. Giải thuật Minimax áp dụng giả thuyết này để tìm kiếm không gian trạng thái của trò chơi. Trường hợp này thuật toán minimax sẽ đáp ứng những gì mình cần.

Giải thuật Minimax 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.

Ta tạm quy định giá trị MAX lúc game thắng cho X = +10 và MIN lúc game thua cho X = -10 và lúc game hòa = 0. Lúc này ở lượt 1, MAX có thể đi được 1 trong 3 nước như hình. Vậy làm sao để chọn 1 trong 3 nước đó nước nào là tốt nhất để đi. Chúng ta dựa vào giá trị của từng nước để chọn nước tốt nhất, như ở đây 3 node đó thuộc lớp MAX nên chọn giá trị lớn nhất. Chúng ta bắt đầu tìm giá trị của từng node đó. Ở lớp MAX trong lượt 1, thì ta có node 1,2,3 được đánh số từ trái sáng phải như hình. Node 3 chúng ta đã là node lá (X win game) và có giá trị là +10. Còn 2 node 1,2 thì chưa biết giá trị của nó tại lượt 1 nên chúng ta dựa vào giá trị của các node con để định giá trị và bằng giá trị bé nhất của các node con ở lớp MIN tại lượt 2. Cứ tiếp tục tương tự như vậy đến lúc gặp node lá thì từ node lá đó ta suy ngược lại và ta tính được node 1 có giá trị là -10 và node 2 là 0. Vậy nước đi tốt nhất ở đây là như node 3 có giá trị lớn nhất là +10. 5.

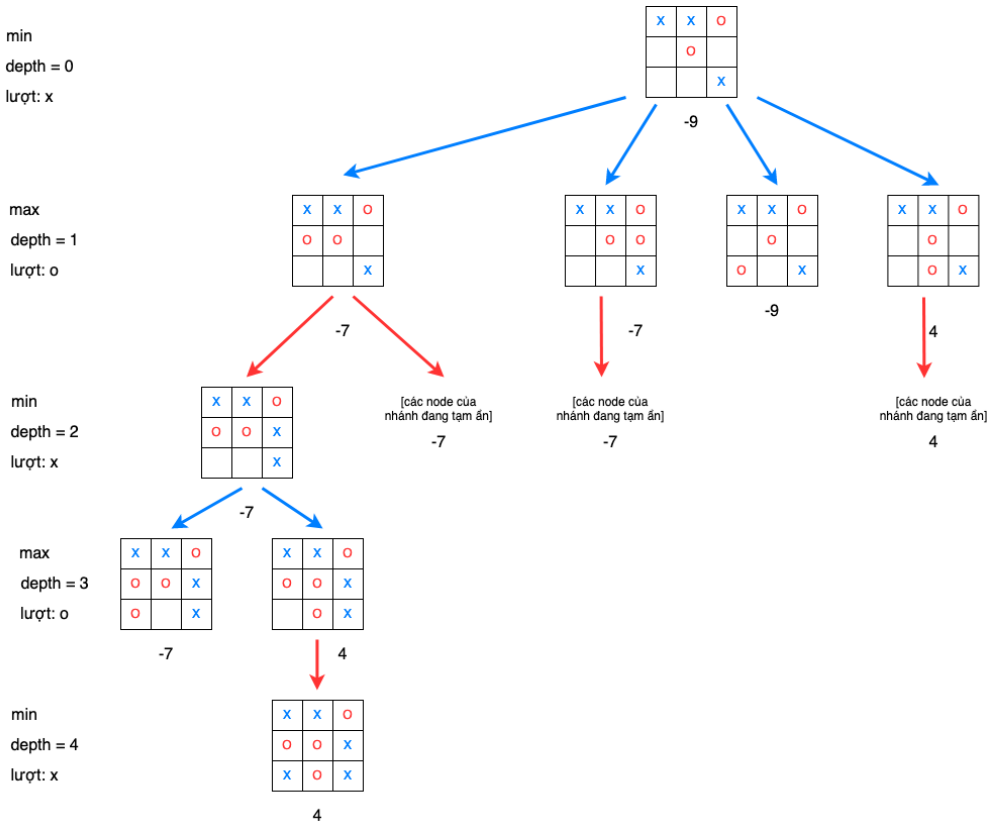


Hình 1: Các trường hợp trong trò chơi TicTacToe được mô tả dưới dạng cây (Ảnh minh họa)

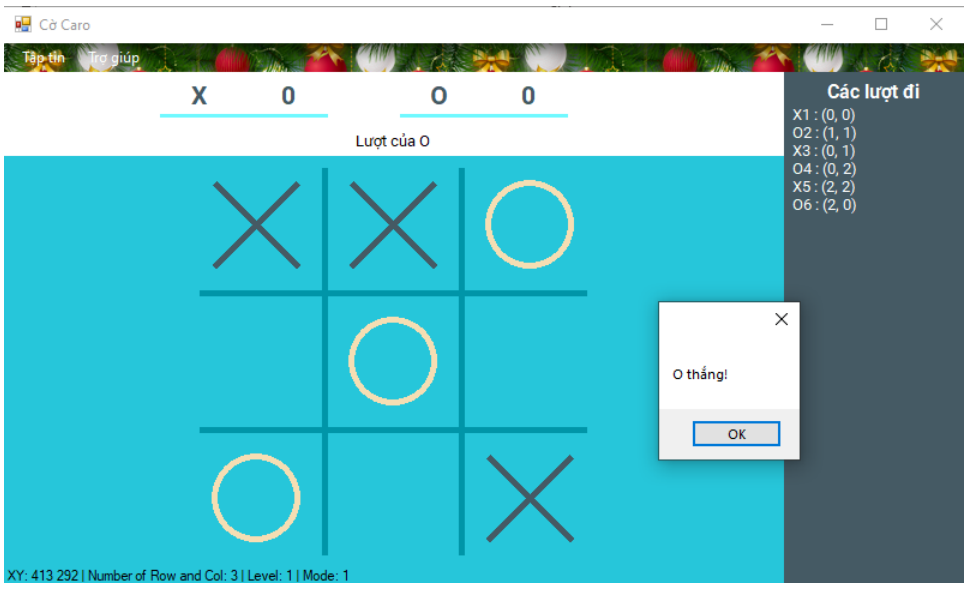


Hình 2: Trong trò chơi, O đại diện cho máy tính sẽ chọn đi tại A thay vì B hoặc C do thuật toán ban đầu chưa tối ưu

Trong hình trên ở lượt 1 sau khi duyệt hết các node có thể, ta thấy node thứ 2 và 3 đều cùng có +10 điểm. Rõ ràng thấy node thứ 3 có nước đi tối ưu hơn node thứ 2 nhưng vì cả hai đều có cùng điểm số nên mặc định node thứ 2 sẽ được chọn, từ đó ta thấy cách chọn nước đi của bên X chưa thật sự "quyết đoán". Vậy, chúng ta có cách nào để làm cho thuật toán trở nên "quyết đoán" hơn không? Ý tưởng giải quyết khá đơn giản là node lá nằm ở độ sâu lớn hơn thì điểm số của node đó sẽ ít hơn node lá nằm ở độ sâu nhỏ. Từ đó ta bổ sung vào thuật toán một giá trị depth (độ sâu) với điểm số được tính bằng cách $\boldsymbol{score_{final} = score - depth}$.



Hình 3: Sau khi cải tiến, thuật toán đã có thể đưa ra lựa chọn tối ưu hơn (Ảnh minh họa)

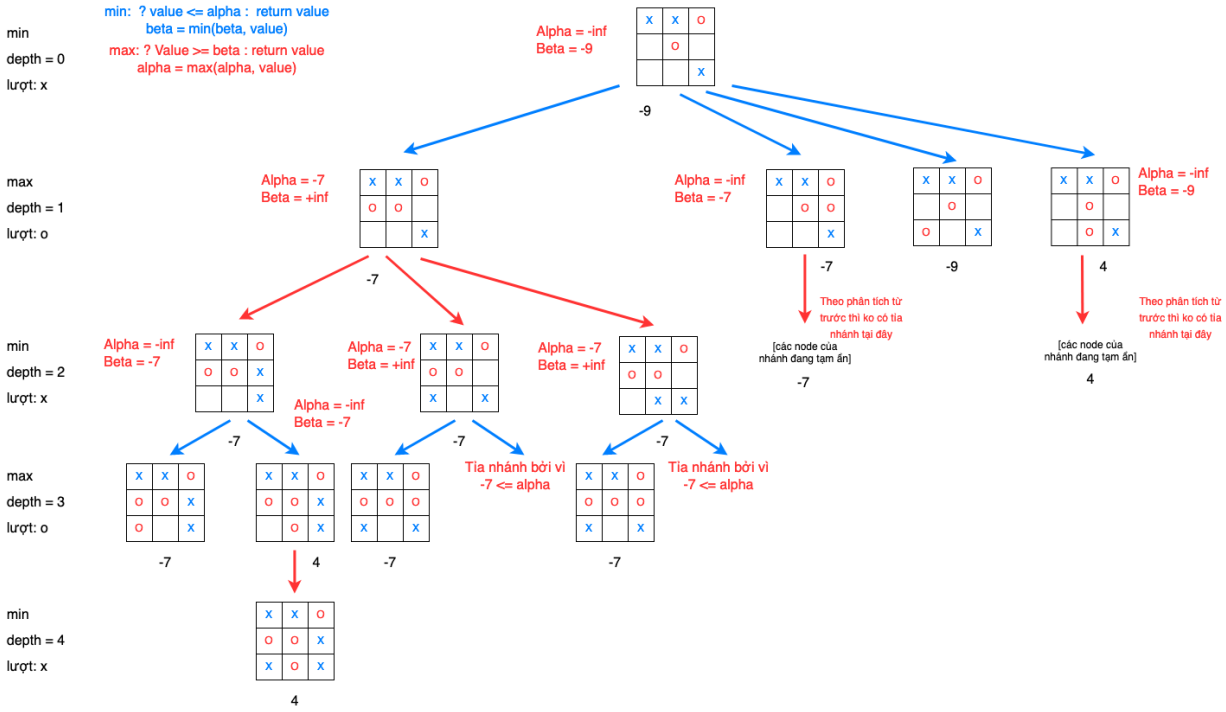


Hình 4: Lần này O quyết đoán hơn trong việc kết thúc trận đấu

b. Mở rộng

Trong trò chơi TicTacToe (3x3) thuật toán MiniMax tỏ ra hiệu quả trong việc chọn các nước đi phù hợp. Với n là tổng số ô trong bàn cờ, ta có n! các node hay cụ thể là 9! = 362880 node trong bàn cờ 3x3 nên nhược điểm của nó chính là tốn rất nhiều thời gian tính toán do phải duyệt qua tất cả các trường hợp có thể xảy ra. Việc mở rộng bàn cờ lên 4x4 hoặc 5x5 với thuật toán MiniMax tỏ ra không khả thi bởi vì sẽ phải duyệt qua 16! node (đối với bàn cờ 4x4) và 25! node (đối với bàn cờ 5x5). Vì thế, để có thể cài đặt trên bàn cờ 4x4 hoặc 5x5 thì phải sử dụng thêm Alpha-beta Pruning để giảm bớt các node cần phải duyệt.

Chiến lược cắt tỉa Alpha-beta, 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.



Hình 5: Hình minh họa cắt tỉa Alpha-beta

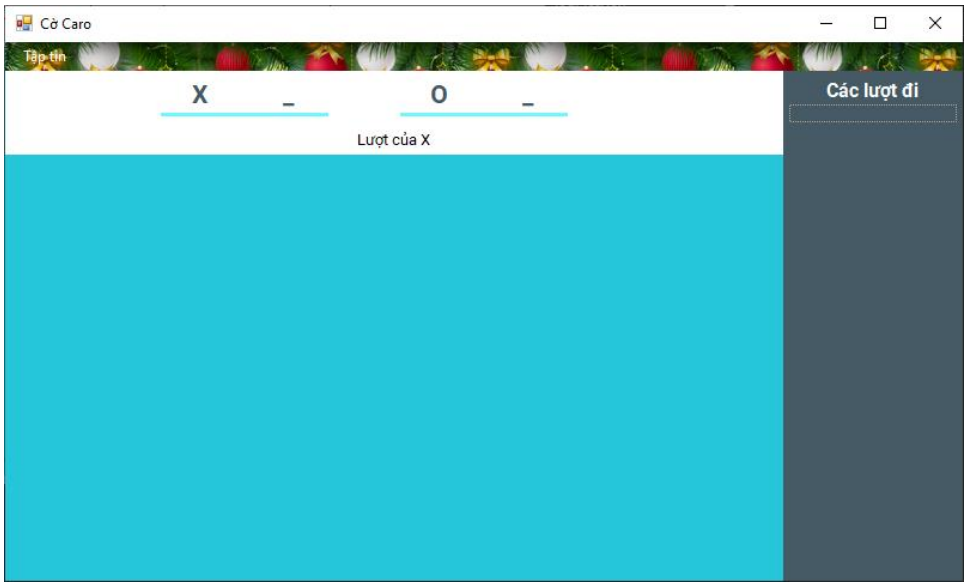
Ở đây chúng ta cũng xét từ trái qua phải bắt đầu từ nút gốc và nút con bên trái sẽ được ưu tiên duyệt trước. Xét duyệt từ trên gốc xuống sâu (vì ban đầu chưa hề tồn tại giá trị alpha hay beta của các nút). Nút đầu tiên ta duyệt là E sẽ gặp giá trị 2 (alpha ≥ 2), khi đó ở trên chưa có giá trị beta để ta có thể so sánh nên sẽ bắt đầu

duyệt con tiếp theo của nút E đó và ở đây ta sẽ chọn cho alpha = 3 (Max). Ta duyệt từ trái sang phải và phải lần lượt từng nhánh một, sau đó sang nhánh tiếp theo cùng gốc. Vậy nên tiếp theo chúng ta sẽ đưa giá trị alpha này lên nút B (Min) và nút B – beta ≤ 3, sau đó nút F sẽ được duyệt, và ta phải tìm alpha của F. Khi duyệt con đầu tiên mang giá trị 5 vậy alpha của F – alpha ≥ 5. Tại B – beta ≤ 3 và tại F – alpha ≥ 5. Như vậy chúng ta không cần xem xét các nút con còn lại của F vì cái ta cần ở đây chỉ là khoảng ≤ 3 nên ta cắt toàn bộ các con còn lại. Sau khi duyệt toàn bộ các con của B thì tại B – beta = 3, và tại nút A – alpha ≥ 3.

III. Ứng dụng trò chơi TicTacToe

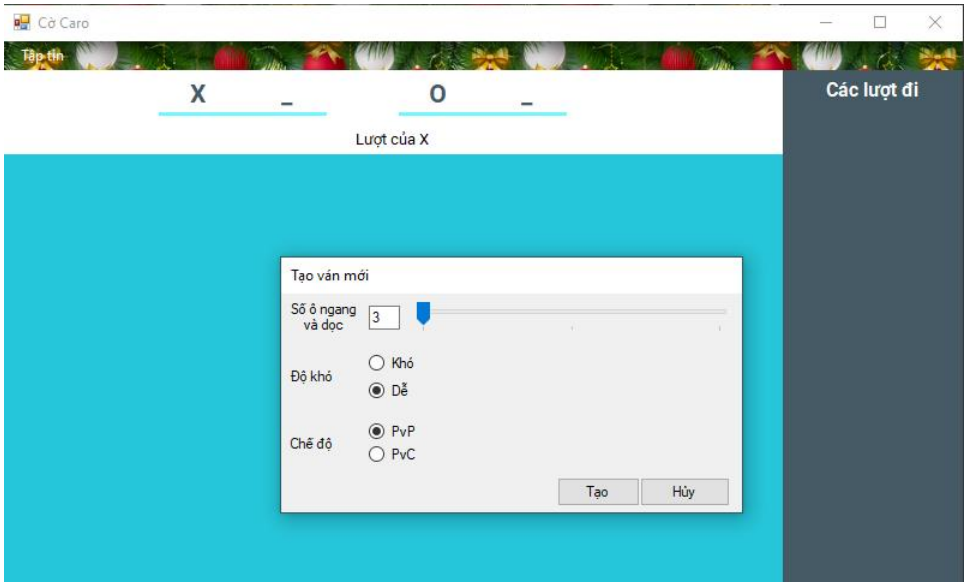
a. Giới thiệu ứng dụng

Ứng dụng mà nhóm xây dựng là trò chơi TicTacToe (Ca rô) có sử dụng thuật toán thông minh MiniMax. TicTacToe là một trò chơi phổ biến dùng viết trên bàn cờ giấy có chín ô, 3x3 (trong bài làm này nhóm mở rộng thêm 4x4 và 5x5). Hai người chơi, người dùng ký hiệu O, người kia dùng ký hiệu X, lần lượt điền ký hiệu của mình vào các ô. Người thắng là người thể tạo được đầu tiên một dãy ký hiệu của mình, ngang dọc hay chéo đều được. Ứng dụng được lập trình bằng ngôn ngữ C# và thư viện đồ họa Windows Forms.



Hình 6: Giao diện khi vừa mở trò chơi

b. Tạo ván chơi mới



Hình 7: Màn hình tạo ván mới

i. Chế độ chơi PvP – đấu người với người

Ở giao diện màn hình chính, chọn Tập tin → Tạo ván mới. Màn hình Tạo ván mới sẽ hiện ra, sau đó chọn Số ô ngang và dọc – mặc định là 3, chọn Độ khó – mặc định là Dễ, chọn chế độ PvP để chơi với người, cuối cùng nhấn Tạo để bắt đầu trò chơi.

ii. Chế độ chơi PvC – đấu người với máy

Ở giao diện màn hình chính, chọn Tập tin → Tạo ván mới. Màn hình Tạo ván mới sẽ hiện ra, sau đó chọn Số ô ngang và dọc – mặc định là 3, chọn Độ khó – mặc định là Dễ, chọn chế độ PvC để chơi với máy, cuối cùng nhấn Tạo để bắt đầu trò chơi

c. Nội dung code

i. Lớp và các phương thức

Để lưu trữ trạng thái của bàn cờ và canh chỉnh khoảng cách của bàn cờ trong màn hình, nhóm đã tạo lớp **Board** để quản lý chúng. Lớp **Board** có những thuộc tính và phương thức như sau:

- Thuộc tính
  - Width: chiều rộng của bàn cờ
  - Height: chiều cao của bàn cờ
  - MarginX: lề trên và dưới
  - MarginY: lề trái và phải
  - CrossSpace: chiều cao của 1 ô trong bàn cờ
  - BarSpace: chiều rộng của 1 ô trong bàn cờ
  - NumOfCol: số lượng cột
  - NumOfRow: số lượng dòng
  - Content: trạng thái của bàn cờ
  - Turn: lượt đi của X hoặc O
- Phương thức
  - IsInside(x, y): kiểm tra vị trí X, Y của con trỏ có nằm trong bàn cờ hay không
  - ChangeTurn(): thay đổi lượt đi
  - ConvertPositionXYtoIJ(posX, posY): chuyển đổi vị trí con trỏ thành vị trí trên bàn cờ
  - ConvertPositionIJtoXY(posI, posJ): chuyền đổi vị trí trên bàn cờ thành vị trí con trỏ
  - ChangeSize(horizontal, vertical): thay đổi chiều rộng và cao của bàn cờ
  - ResetContent(): xóa toàn bộ lượt đi trên bàn cờ
  - TickAt(posX, posY, turn): ghi lại lượt đi của turn lên bàn cờ tại vị trí posX và posY
  - CheckWin(): kiểm tra ai là người chiến thắng ván cờ

ii. Cài đặt thuật toán MiniMax

Để cài đặt thuật toán MiniMax lên trò chơi, nhóm đã tạo lớp **MiniMax**.

- Thuộc tính
    - NumOfCol: số cột của bàn cờ
    - NumOfRow: số dòng của bàn cờ
    - ScoreTable: bảng điểm ứng với mỗi trường hợp thắng thua
    - Level: độ khó mà người dùng chọn
- Phương thức
    - CheckWin(content): kiểm tra ai là người chiến thắng ván cờ
    - Score(state): tính điểm số dựa vào ScoreTable
    - Action(state): tính toán các nước đi hợp lệ có thể đi
    - MaxValue(state, step): tính điểm số lớn nhất trong các nước đi
    - MinValue(state, step): tính điểm số nhỏ nhất trong các nước đi
    - MinimaxDecision(board, turn): lựa chọn nước đi tối ưu trong thuật toán MiniMax

iii. Cài đặt thuật toán Cắt tỉa Alpha-beta

Tương tự thuật toán MiniMax, nhóm cũng tạo lớp Cắt tỉa **Alpha-beta** dựa trên lớp **MiniMax**

- Thuộc tính
    - NumOfCol: số cột của bàn cờ
    - NumOfRow: số dòng của bàn cờ
    - ScoreTable: bảng điểm ứng với mỗi trường hợp thắng thua
    - Level: độ khó mà người dùng chọn
- Phương thức
    - CheckWin(content): kiểm tra ai là người chiến thắng ván cờ
    - Score(state): tính điểm số dựa vào ScoreTable
    - Action(state): tính toán các nước đi hợp lệ có thể đi
    - MaxValue(state, step): tính điểm số lớn nhất trong các nước đi
    - MinValue(state, step): tính điểm số nhỏ nhất trong các nước đi
    - AlphaBetaDecision(board, turn): lựa chọn nước đi tối ưu trong thuật toán MiniMax

IV. Đánh giá

a. Ưu điểm

i. Thuật toán MiniMax

- Thuật toán có thể tìm ra lựa chọn tối ưu nhất trong trò chơi
- Dễ đọc hiểu
- Dễ dàng cài đặt trên ứng dụng

ii. Thuật toán Cắt tỉa Alpha-beta

- Thuật toán có thể tìm ra một sự lựa chọn tương đối tốt với số lượng lần duyệt qua các node là ít hơn nhiều so với MiniMax

b. Nhược điểm

i. Thuật toán MiniMax

- Thuật toán dù tìm ra lời giải tốt nhưng tốn rất nhiều thời gian do duyệt qua số lượng node rất lớn (n! node)

ii. Thuật toán Cắt tỉa Alpha-beta

- Thuật toán có thể đưa ra quyết định không chính xác do đã bỏ qua một lượng lớn các node trạng thái

c. Cách giải quyết

- Đối với bàn cờ 3x3, nhóm sử dụng thuật toán MiniMax để có lời giải tối ưu nhất và thời gian chạy cũng không quá lớn
- Đối với bàn cờ 4x4 và lớn hơn, nhóm sử dụng thuật toán Cắt tỉa Alpha-beta để giảm thời gian duyệt node, nhưng đánh đổi là lời giải chưa tối ưu

d. Các vấn đề còn tồn đọng

- Ở bàn cờ 4x4 dù máy tính có thể đưa ra lời giải nhưng lời giải đó chưa thực sự tối ưu
- Ở bàn cờ 5x5 máy tính không thể đưa ra lời giải do số lần duyệt node quá lớn!

V. Tài liệu tham khảo

- Stuart, J. R., & Peter, N. (2003). Artificial Intelligence: A Modern Approach (Third Edition)
- https://www.stdio.vn/giai-thuat-lap-trinh/giai-thuat-cat-tia-alpha-beta-Wu7F1 (truy cập ngày 20/12/2021)
- https://viblo.asia/p/thuat-toan-minimax-ai-trong-game-APqzeaVVzVe (truy cập ngày 20/12/2021)