**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----o0o----



**ĐỒ ÁN MÔN HỌC**

**MÔN: TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI : XÂY DỰNG TRÒ CHƠI Ô ĂN QUAN**

|  |  |
| --- | --- |
| **Giảng Viên** | **Thành viên nhóm** |
|  |  |
|  |  |
|  |  |
|  |  |

**TP. Hồ Chí Minh, tháng 12 năm 2024**

**BẢNG PHÂN CHIA CÔNG VIỆC**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sinh viên thực hiện** | **MSSV** | **Công việc** | **Mức độ đóng góp** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# LỜI CẢM ƠN

# LỜI CAM ĐOAN

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc185531012)

[LỜI CAM ĐOAN ii](#_Toc185531013)

[MỤC LỤC iii](#_Toc185531014)

[DANH MỤC HÌNH v](#_Toc185531015)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc185531016)

[1.1 Giới thiệu về trò chơi ô ăn quan: 1](#_Toc185531017)

[1.1.1 Nguồn gốc của trò chơi: 1](#_Toc185531018)

[1.1.2 Luật chơi của trò chơi ô ăn quan: 2](#_Toc185531019)

[1.2 Cơ sở lý thuyết 3](#_Toc185531020)

[1.2.1 Tìm kiếm có đối thủ 3](#_Toc185531021)

[1.2.2 Trò chơi đối kháng 5](#_Toc185531022)

[1.2.3 Giải thuật Minimax kết hợp cắt tỉa Alpha-Beta 8](#_Toc185531023)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ 16](#_Toc185531024)

[2.1 Phân tích bài toán: 16](#_Toc185531025)

[2.2 Cấu trúc của chương trình: 17](#_Toc185531026)

[2.2.1 Xây dựng hàm Minimax: 17](#_Toc185531027)

[2.2.2 Xây dựng hàm Minimax có cắt tỉa: 18](#_Toc185531028)

[2.3.4 Các hàm xử lý cơ bản trong chương trình: 18](#_Toc185531029)

[CHƯƠNG 3: CHƯƠNG TRÌNH ỨNG DỤNG 26](#_Toc185531030)

[3.1 Giới thiệu bàn chơi ô ăn quan: 26](#_Toc185531031)

[3.2 Cài đặt: 26](#_Toc185531032)

[3.2.1 Ngôn ngữ và công cụ sử dụng: 26](#_Toc185531033)

[3.2.2 Độ phức tạp của chương trình: 27](#_Toc185531034)

[3.3 Giới thiệu chương trình: 28](#_Toc185531035)

[CHƯƠNG 4: KẾT LUẬN 31](#_Toc185531036)

[4.1 Kết quả đạt được: 31](#_Toc185531037)

[4.2 Hạn chế: 31](#_Toc185531038)

[4.3 Định hướng phát triển thêm: 32](#_Toc185531039)

[TÀI LIỆU THAM KHẢO 33](#_Toc185531040)

# 

# DANH MỤC HÌNH

[Hình 1.1: Ô ăn quan dân gian 1](#_Toc185530992)

[Hình 1.2 Hình minh họa cây trò chơi 10](#_Toc185530993)

[Hình 1.3 Hình minh họa cây trò chơi sử dụng cắt tỉa Alpha-beta 14](#_Toc185530994)

[Hình 2.1 Hàm khởi tạo ô ăn quan 18](#_Toc185530995)

[Hình 2.2 Hàm xác định nước đi hợp lệ 19](#_Toc185530996)

[Hình 2.3 Hàm cập nhật nút 19](#_Toc185530997)

[Hình 2.4 Hàm kiểm tra điều kiện kết thúc 20](#_Toc185530998)

[Hình 2.5 Hàm cập nhật trạng thái 20](#_Toc185530999)

[Hình 2.6 Hàm di chuyển sang trái hoặc phải 21](#_Toc185531000)

[Hình 2.7 Hàm tính điểm 22](#_Toc185531001)

[Hình 2.8 Hàm minimax không có cắt tỉa Alpha-Beta 23](#_Toc185531002)

[Hình 2.9 Hàm minimax có cắt tỉa Alpha-Beta 24](#_Toc185531003)

[Hình 2.10 Hàm tham lam 25](#_Toc185531004)

[Hình 3.1: Bàn chơi ô ăn quan cho 2 người 26](#_Toc185531005)

[Hình 3.2: Trang chủ của ứng dụng 28](#_Toc185531006)

[Hình 3.3: Giao diện bàn chơi ô ăn quan cho 2 người 28](#_Toc185531007)

[Hình 3.4: Giao diện chọn hướng đi 29](#_Toc185531008)

[Hình 3.5: Giao diện khi di chuyển các quân trong ô ăn quan 29](#_Toc185531009)

[Hình 3.6: Giao diện khi rải thêm quân 30](#_Toc185531010)

[Hình 3.7 : Giao diện khi trò chơi kết thúc 30](#_Toc185531011)

# CHƯƠNG 1: GIỚI THIỆU

## Giới thiệu về trò chơi ô ăn quan:

### 1.1.1 Nguồn gốc của trò chơi:



Hình 1.1: Ô ăn quan dân gian

Ô ăn quan là một trò chơi dân gian truyền thống lâu đời của Việt Nam, gắn bó với tuổi thơ của biết bao thế hệ. Trò chơi này xuất phát từ sự sáng tạo đơn giản nhưng vô cùng tinh tế của người Việt xưa, nhằm mang đến niềm vui và sự gắn kết trong các cộng đồng làng quê. Không có tài liệu nào xác định chính xác thời điểm trò chơi này ra đời, nhưng các nhà nghiên cứu văn hóa dân gian cho rằng nó đã tồn tại từ hàng trăm năm trước, thậm chí có thể từ thời kỳ phong kiến. [1]

Ô ăn quan thường được biết đến với sự phổ biến ở miền Bắc và miền Trung Việt Nam, nhưng sau đó lan rộng ra khắp cả nước. Đây không chỉ là một trò chơi giải trí mà còn là một phương tiện giáo dục, giúp trẻ em rèn luyện tư duy, sự khéo léo và khả năng tính toán. Thời kỳ trước, khi các thiết bị công nghệ chưa phổ biến, ô ăn quan xuất hiện ở khắp các sân đình, sân trường, hay trên những khoảng đất trống trước nhà. [1]

Không chỉ Việt Nam, nhiều nước Đông Nam Á cũng có các trò chơi tương tự với các tên gọi khác nhau, như Congklak ở Indonesia hay Mancala ở châu Phi và Trung Đông. Tuy nhiên, ô ăn quan của Việt Nam mang đậm nét văn hóa riêng với luật chơi dễ hiểu và cách sắp xếp bàn chơi độc đáo, phản ánh sự sáng tạo dân dã của người Việt. [1]

### 1.1.2 Luật chơi của trò chơi ô ăn quan:

Ô ăn quan là một trò chơi dành cho hai người, sử dụng bàn chơi được vẽ hoặc kẻ trên mặt đất. Bàn chơi có hình chữ nhật với tổng cộng 10 ô vuông nhỏ và 2 ô lớn ở hai đầu, được gọi là “ô quan”. Các ô vuông nhỏ tượng trưng cho “dân” (hay quân), còn ô lớn là nơi để “quan”. Dụng cụ chơi rất đơn giản, thường là các viên sỏi, hạt nhãn, hay bất kỳ vật nhỏ nào dễ nhặt, được chia thành quân dân và quân quan.

**Luật chơi cơ bản:**

* **Chuẩn bị:**
  + Bàn chơi: Có 10 ô nhỏ xếp thành 2 hàng ngang (mỗi hàng 5 ô) và 2 ô lớn ở hai đầu.Tổng cộng có 12 ô.
  + Quân chơi: Gồm hai loại chính:
  + Quan: Mỗi ô quan chứa 10 viên
  + Dân: Mỗi ô nhỏ chứa số lượng quân bằng nhau, thường là 5 viên.
* **Cách chơi:**
  + Hai người chơi lần lượt di chuyển quân. Mỗi lượt, người chơi chọn một ô chứa dân ở phía mình, nhặt hết số quân trong ô đó, sau đó rải đều từng quân vào các ô tiếp theo theo chiều kim đồng hồ hoặc ngược chiều kiêm đồng hồ.
  + Nếu rải đến viên cuối cùng mà ô tiếp theo có quân, người chơi tiếp tục nhặt hết quân ở ô đó và rải tiếp. Nếu rơi vào ô trống, người chơi dừng lại.
  + Trường hợp ô trống đó liền kề với một ô có quân, người chơi được ăn toàn bộ quân trong ô kế tiếp.Quá trình này có thể lặp lại nhiều lần gọi là ăn liên hoàn.
  + Khi đến lượt đi của người chơi nếu ô quan vẫn còn mà người chơi đã hết dân ở các ô của mình thì sẽ tiến hành rải quân với mỗi ô 1 viên để tiếp tục.
* **Kết thúc ván chơi:**
  + Trò chơi kết thúc khi toàn bộ quan trên bàn chơi đã bị ăn hết.
  + Hai người đếm số quân mình thu thập được. Ai có nhiều quân hơn là người chiến thắng.Nếu số quân bằng nhau thì cả 2 cùng hòa.
* **Ý nghĩa của trò chơi**

Trò chơi ô ăn quan không chỉ mang tính chất giải trí mà còn ẩn chứa nhiều giá trị giáo dục và văn hóa sâu sắc. Trẻ em khi chơi có thể rèn luyện:

* Khả năng tư duy logic: Trò chơi đòi hỏi người chơi tính toán chiến lược, dự đoán bước đi của đối thủ và lựa chọn cách rải quân sao cho có lợi nhất.
* Sự khéo léo: Việc rải quân đều và chính xác cũng là một kỹ năng được rèn luyện qua từng lượt chơi.
* Tinh thần đồng đội và cạnh tranh lành mạnh: Trò chơi khuyến khích sự giao lưu, học hỏi lẫn nhau giữa các bạn nhỏ, đồng thời nuôi dưỡng tinh thần thi đấu công bằng.
* **Biến thể của trò chơi**

Qua thời gian, ô ăn quan đã có nhiều biến thể khác nhau tùy theo vùng miền, nhưng luật chơi cơ bản vẫn được giữ nguyên. Một số nơi sử dụng các vật liệu khác nhau để làm quân, hoặc thay đổi cách bố trí bàn chơi, tăng thêm số lượng ô để phù hợp với số người chơi nhiều hơn. Có những phiên bản nâng cấp của trò chơi trên các ứng dụng điện thoại, mang lại trải nghiệm hiện đại nhưng vẫn giữ nguyên tinh thần truyền thống.

## 1.2 Cơ sở lý thuyết

### 1.2.1 Tìm kiếm có đối thủ

Tìm kiếm có đối thủ là một phương pháp trong lĩnh vực trí tuệ nhân tạo (AI), được áp dụng trong các trò chơi đối kháng và những tình huống mà có hai hoặc nhiều bên tham gia với mục tiêu xung đột. Trái ngược với tìm kiếm thông thường trong các bài toán giải đố (như tìm đường đi ngắn nhất), tìm kiếm có đối thủ không chỉ đơn thuần là việc tối ưu hóa hành động của một bên, mà còn phải dự đoán và đối phó với hành động của đối thủ, từ đó xác định chiến lược tốt nhất cho mình. [1]

Trong các trò chơi đối kháng, chẳng hạn như cờ vua, cờ tướng, hay ô ăn quan, hai người chơi thay phiên nhau thực hiện nước đi. Mỗi nước đi có thể thay đổi trạng thái của trò chơi và dẫn đến những kết quả khác nhau. Người chơi phải dự đoán được phản ứng của đối thủ và chuẩn bị cho nhiều tình huống có thể xảy ra. Để đạt được điều này, tìm kiếm có đối thủ sử dụng một cấu trúc gọi là cây trò chơi (game tree). [1]

* **Cây trò chơi**

Cây trò chơi là một mô hình biểu diễn tất cả các trạng thái có thể xảy ra của trò chơi, trong đó:

* + Nút gốc : Là trạng thái hiện tại của trò chơi.
  + Nút con : Là các trạng thái có thể đạt được sau một hành động.
  + Lá của cây :Là các trạng thái kết thúc trò chơi hoặc khi không còn nước đi nào có thể thực hiện được.

Mỗi nút trong cây trò chơi được gán một giá trị để biểu thị mức độ thuận lợi hoặc bất lợi cho người chơi ở trạng thái đó. Việc duyệt cây trò chơi từ gốc đến các lá giúp người chơi có thể tính toán và xác định hành động tốt nhất trong lượt đi của mình.

* **Người chơi MAX và MIN**

Trong tìm kiếm có đối thủ, người chơi thường được phân thành hai loại:

* + MAX: Là người chơi cố gắng tối đa hóa giá trị của trạng thái. Mục tiêu của MAX là đạt được kết quả tốt nhất có thể cho mình.
  + MIN: Là người chơi cố gắng giảm thiểu giá trị của trạng thái. Mục tiêu của MIN là gây bất lợi tối đa cho MAX.

Hai người chơi sẽ luân phiên thực hiện các bước đi. Vì vậy, các nút trong cây trò chơi sẽ được phân loại theo lượt chơi:

* + Nếu nút thuộc lượt của MAX, MAX sẽ chọn nước đi dẫn đến trạng thái có giá trị cao nhất.
  + Nếu nút thuộc lượt của MIN, MIN sẽ chọn nước đi dẫn đến trạng thái có giá trị thấp nhất.
* **Chiến lược trong tìm kiếm có đối thủ**

Tìm kiếm có đối thủ không chỉ là việc chọn nước đi tốt nhất ở hiện tại mà còn phải dự đoán hành động của đối thủ trong tương lai. Điều này đòi hỏi người chơi phải có chiến lược dựa trên hai yếu tố chính:

* + Dự đoán phản ứng của đối thủ: Giả định rằng đối thủ cũng chơi tối ưu, người chơi phải xem xét tất cả các khả năng mà đối thủ có thể thực hiện.
  + Tối ưu hóa chiến lược của mình: Bằng cách dựa vào dự đoán phản ứng của đối thủ, người chơi chọn hành động có lợi nhất cho mình.

Ví dụ, trong trò chơi ô ăn quan, nếu người chơi A có thể thực hiện một nước đi để ăn được nhiều quân, họ cũng phải cân nhắc liệu nước đi đó có tạo cơ hội cho người chơi B thực hiện một nước đi tốt hơn hay không. Đây là một minh chứng điển hình cho việc áp dụng tìm kiếm có đối thủ.

* **Đánh giá trạng thái**

Trong tìm kiếm có đối thủ, việc đánh giá trạng thái của một nút trong cây trò chơi rất quan trọng. Để xác định giá trị của mỗi trạng thái:

* + Hàm đánh giá được sử dụng để tính toán mức độ tốt xấu của trạng thái đó.
  + Hàm đánh giá thường dựa trên các yếu tố như số lượng quân, vị trí quân trên bàn chơi, hoặc khả năng tạo lợi thế trong lượt chơi tiếp theo.

Trong các trò chơi đơn giản, giá trị trạng thái có thể là số lượng quân mà người chơi đã ăn được. Trong các trò chơi phức tạp hơn, hàm đánh giá có thể kết hợp nhiều yếu tố để phản ánh toàn diện lợi thế của mỗi trạng thái.

* **Ý nghĩa của tìm kiếm có đối thủ**

Tìm kiếm có đối thủ là cơ sở lý thuyết quan trọng trong việc xây dựng trí tuệ nhân tạo cho các trò chơi đối kháng. Nó giúp mô phỏng cách mà con người suy nghĩ và ra quyết định khi tham gia vào một cuộc cạnh tranh. Các ứng dụng của tìm kiếm có đối thủ không chỉ dừng lại ở trò chơi mà còn mở rộng ra các lĩnh vực như:

* + Thương thuyết trong kinh doanh.
  + Đưa ra chiến lược trong quân sự.
  + Phát triển hệ thống hỗ trợ ra quyết định trong môi trường có xung đột lợi ích.

Tóm lại, tìm kiếm có đối thủ cung cấp một cách tiếp cận mạnh mẽ để xử lý các bài toán mang tính cạnh tranh, trong đó hành động của một bên không chỉ phụ thuộc vào trạng thái hiện tại mà còn dựa vào sự dự đoán và thích nghi với hành động của đối phương. Trong trò chơi ô ăn quan, việc sử dụng tìm kiếm có đối thủ giúp người chơi xây dựng chiến lược thông minh và tối ưu hóa kết quả dựa trên các trạng thái khác nhau của trò chơi.[2]

### 1.2.2 Trò chơi đối kháng

* **Khái niệm trò chơi đối kháng**

Trò chơi đối kháng (adversarial games) là một loại hình trò chơi trong đó có hai hoặc nhiều người chơi tham gia, với mục tiêu mâu thuẫn hoặc xung đột lẫn nhau. Trong các trò chơi này, lợi ích của một người chơi thường tỷ lệ nghịch với lợi ích của người chơi khác. Nói cách khác, để một người chơi thắng lợi, đối thủ phải chịu thất bại.

Các trò chơi đối kháng rất phổ biến trong đời sống, từ những trò chơi dân gian như ô ăn quan, cờ vua, cờ tướng, cho đến các trò chơi hiện đại như poker hay các tựa game điện tử chiến thuật. Bản chất của các trò chơi này là tạo ra sự cạnh tranh trực tiếp giữa các người chơi, qua đó đòi hỏi họ phải sử dụng chiến lược, tư duy logic, và khả năng dự đoán để giành chiến thắng.

* **Phân loại trò chơi đối kháng**

Các trò chơi đối kháng thường được phân loại dựa trên một số tiêu chí như sau:

* + Dựa trên thông tin về trò chơi:
    - Trò chơi thông tin hoàn toàn: Mọi thông tin về trạng thái trò chơi đều được biết rõ ràng với tất cả người chơi. Ví dụ: cờ vua, cờ tướng, ô ăn quan.
    - Trò chơi thông tin không hoàn toàn: Một số thông tin về trạng thái trò chơi bị ẩn hoặc không được tiết lộ đầy đủ. Ví dụ: bài poker, bài tây.
  + Dựa trên yếu tố ngẫu nhiên:
    - Trò chơi không có yếu tố ngẫu nhiên: Kết quả của trò chơi hoàn toàn phụ thuộc vào kỹ năng và chiến lược của người chơi. Ví dụ: cờ vua, cờ caro.
    - Trò chơi có yếu tố ngẫu nhiên: Các yếu tố như xúc xắc hoặc bốc thăm có thể ảnh hưởng đến kết quả. Ví dụ: trò chơi cờ tỷ phú, game board như Ludo.
  + Dựa trên số lượng người chơi:
    - Trò chơi đối kháng hai người: Hai người chơi đối đầu trực tiếp. Đây là dạng phổ biến nhất, như cờ vua hay ô ăn quan.
    - Trò chơi đối kháng nhiều người: Có sự cạnh tranh giữa nhiều người chơi hoặc đội nhóm. Ví dụ: các trò chơi chiến thuật đội nhóm trong game điện tử.
* **Đặc điểm của trò chơi đối kháng**

Các trò chơi đối kháng có một số đặc điểm chính sau đây:

* Cạnh tranh trực tiếp:

Trong trò chơi đối kháng, các người chơi trực tiếp tranh đấu để đạt được mục tiêu của mình. Mỗi hành động của một người không chỉ ảnh hưởng đến kết quả của họ mà còn tác động đến chiến lược và kết quả của đối thủ.

* Yêu cầu chiến lược:

Người chơi cần xây dựng các chiến lược phù hợp để đối phó với các tình huống cụ thể. Điều này bao gồm việc dự đoán hành động của đối thủ và điều chỉnh chiến thuật của mình để tối ưu hóa lợi thế.

* Kết quả rõ ràng:

Thông thường, trò chơi đối kháng kết thúc với một kết quả rõ ràng: thắng, thua hoặc hòa. Kết quả này giúp xác định thành công của chiến lược mà mỗi người chơi áp dụng.

* Không gian trạng thái lớn:

Các trò chơi như cờ vua hoặc ô ăn quan có không gian trạng thái rất lớn, đòi hỏi người chơi phải tính toán nhiều bước đi trước để đạt được lợi thế.

* **Ví dụ về trò chơi đối kháng: Ô ăn quan**

Trò chơi đối kháng phổ biến trong văn hóa Việt Nam là ô ăn quan. Đây là một trò chơi thông tin hoàn toàn, không có yếu tố ngẫu nhiên. Mỗi người chơi cố gắng tối ưu hóa chiến lược của mình để ăn được nhiều quân và quan nhất, đồng thời ngăn đối thủ làm điều tương tự.

* **Đặc điểm đối kháng trong ô ăn quan thể hiện ở việc:**
  + Người chơi cần cân nhắc xem nước đi của mình sẽ tác động như thế nào đến các ô trên bàn chơi, tạo điều kiện thuận lợi hay bất lợi cho đối thủ.
  + Mỗi lượt đi của một người chơi không chỉ để ăn quân mà còn nhằm cản trở đối thủ tiếp cận các ô có lợi.
* **Mục tiêu trong trò chơi đối kháng**
  + Mục tiêu của trò chơi đối kháng là giành chiến thắng bằng cách đạt được lợi thế tối đa hoặc giảm thiểu lợi thế của đối thủ. Để làm được điều này, người chơi phải:
  + Tìm kiếm nước đi tối ưu: Chọn hành động đem lại lợi ích lớn nhất cho mình hoặc hạn chế tối đa cơ hội của đối thủ.
  + Dự đoán chiến lược của đối thủ: Phân tích các khả năng mà đối thủ có thể thực hiện và chuẩn bị các phản ứng phù hợp.
  + Cân bằng tấn công và phòng thủ: Một chiến lược tốt thường kết hợp giữa việc gia tăng lợi thế của mình và ngăn chặn đối thủ làm điều tương tự.
* **Ý nghĩa của trò chơi đối kháng trong nghiên cứu trí tuệ nhân tạo**

Trò chơi đối kháng đóng vai trò quan trọng trong việc nghiên cứu và phát triển trí tuệ nhân tạo. Bằng cách mô phỏng các trò chơi như cờ vua, ô ăn quan, hay trò chơi điện tử, các nhà nghiên cứu có thể:

* + Phát triển các thuật toán như Minimax hoặc Alpha-Beta để xây dựng chiến lược tối ưu.
  + Tạo ra các hệ thống AI có khả năng cạnh tranh với con người trong các trò chơi phức tạp.
  + Mô hình hóa các tình huống cạnh tranh trong đời thực, chẳng hạn như trong kinh doanh, chính trị, hoặc quân sự.

Trò chơi đối kháng không chỉ là phương tiện giải trí mà còn là môi trường lý tưởng để thử nghiệm và học hỏi cách ra quyết định trong các tình huống có xung đột lợi ích. Nó dạy con người và máy móc cách cân nhắc, lập kế hoạch, và phản ứng một cách hiệu quả trong môi trường cạnh tranh. Trong trò chơi truyền thống như ô ăn quan, sự đối kháng không chỉ nằm ở mức độ tính toán mà còn ở khả năng dự đoán và thích nghi, giúp nâng cao trí tuệ và rèn luyện kỹ năng tư duy cho người chơi.

### 1.2.3 Giải thuật Minimax kết hợp cắt tỉa Alpha-Beta

* **Giải thuật Minimax**

Giải thuật Minimax là một phương pháp toán học và lập trình được thiết kế để tìm nước đi tối ưu trong các trò chơi đối kháng. Trong Minimax, hai người chơi được gọi là **MAX** và **MIN**:

* MAX cố gắng tối đa hóa điểm số hoặc lợi thế của mình.
* MIN cố gắng giảm thiểu lợi thế của MAX (hoặc tối đa hóa lợi thế cho mình nếu xét ngược lại).

Mục tiêu chính của Minimax là tối ưu hóa kết quả cho người chơi hiện tại (MAX hoặc MIN) trong khi giả định rằng đối thủ cũng đang chơi hoàn hảo, tức là luôn chọn nước đi tốt nhất có thể.

Giải thuật Minimax được áp dụng rộng rãi trong các trò chơi như cờ vua, cờ tướng, tic-tac-toe, ô ăn quan, và nhiều trò chơi đối kháng khác, nơi có sự luân phiên lượt đi giữa hai người chơi. Nó dựa trên việc xây dựng và duyệt một cây trò chơi (game tree), đại diện cho tất cả các trạng thái có thể xảy ra và hành động từ trạng thái hiện tại.

* **Nguyên tắc hoạt động của Minimax**

Minimax hoạt động dựa trên nguyên tắc duyệt toàn bộ cây trò chơi để tìm ra giá trị tốt nhất của trạng thái hiện tại. Mỗi nút trong cây đại diện cho một trạng thái của trò chơi, và các nhánh của cây đại diện cho các hành động có thể thực hiện từ trạng thái đó.

Quá trình được thực hiện như sau:

* + Xây dựng cây trò chơi:

Tất cả các trạng thái có thể xảy ra từ trạng thái hiện tại được biểu diễn trong cây trò chơi, với các nút gốc là trạng thái hiện tại và các nút lá là trạng thái kết thúc của trò chơi.

* + Đánh giá trạng thái ở nút lá:

Sử dụng hàm đánh giá (evaluation function) để gán giá trị cho mỗi trạng thái kết thúc, dựa trên lợi ích của trạng thái đó đối với người chơi MAX.

* + Quay ngược giá trị từ nút lá về nút gốc:

Tại các nút thuộc lượt của MAX, chọn giá trị lớn nhất từ các nút con (vì MAX muốn tối đa hóa điểm số).

Tại các nút thuộc lượt của MIN, chọn giá trị nhỏ nhất từ các nút con (vì MIN muốn tối thiểu hóa lợi thế của MAX).

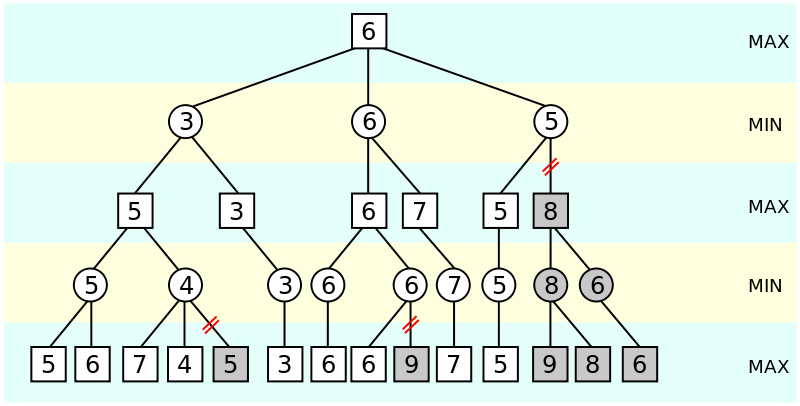
* + Quyết định nước đi:

Ở nút gốc (tương ứng với lượt chơi hiện tại), chọn hành động dẫn đến nút con có giá trị tốt nhất cho người chơi MAX.

* **Quy trình chi tiết của Minimax**

Ví dụ minh họa về cách hoạt động của giải thuật Minimax trong một trò chơi:

* + Người chơi MAX và MIN thay phiên nhau thực hiện nước đi.
  + Tại trạng thái hiện tại (nút gốc), cây trò chơi được xây dựng với tất cả các trạng thái kế tiếp có thể xảy ra.
  + Ở mỗi trạng thái lá (mức sâu nhất của cây), giá trị trạng thái được tính bằng một hàm đánh giá đơn giản, ví dụ: số điểm MAX đạt được trừ đi số điểm MIN đạt được.
  + Giá trị này được chuyển ngược lên các mức cao hơn trong cây:
    - Nếu nút thuộc lượt của MAX, chọn giá trị lớn nhất từ các nút con.
    - Nếu nút thuộc lượt của MIN, chọn giá trị nhỏ nhất từ các nút con.
* Giả sử có một cây trò chơi với các giá trị lá được đánh giá như sau:



Hình 1.2 Hình minh họa cây trò chơi

* Ở mức dưới cùng, giá trị trạng thái được gán cho từng nút lá.
* Ở mức MAX, chọn giá trị nhỏ nhất từ các nút con:
* Ở nhánh 1: MIN(5, 6) = 5
* Ở nhánh 2: MIN(7, 4, 5) = 4
* Ở nhánh 3: MIN(3) = 3
* Ở nhánh 4: MIN(6) = 6
* Ở nhánh 5: MIN(6, 9) = 6
* Ở nhánh 6: MIN(7) = 7
* Ở nhánh 7: MIN(5) = 5
* Ở nhánh 8: MIN(9, 8) = 8
* Ở nhánh 9: MIN(6) = 6
* Ở mức MIN, chọn giá trị lớn nhất từ các nút con:
* Ở nhánh 1: MAX(5, 4) = 5
* Ở nhánh 2: MAX(3) = 3
* Ở nhánh 3: MAX(6, 6) = 6
* Ở nhánh 4: MAX(7) = 7
* Ở nhánh 5: MAX(5, 5) = 5
* Ở nhánh 6: MAX(8, 6) = 8
* Ở mức MAX, chọn giá trị nhỏ nhất từ các nút con:
* Ở nhánh 1: MIN(5, 3) = 3
* Ở nhánh 2: MIN(6, 7) = 6
* Ở nhánh 3: MIN(5, 8) = 5
* Ở mức MIN, chọn giá trị lớn nhất từ các nút con:
* MAX(3, 6, 5) = 6

Do đó, MAX sẽ chọn nước đi dẫn đến trạng thái có giá trị 3.

* **Ưu điểm:**
  + Cung cấp giải pháp tối ưu:

Nếu cây trò chơi được duyệt đầy đủ, giải thuật Minimax luôn đưa ra nước đi tốt nhất có thể cho người chơi MAX.

* + Đơn giản và dễ hiểu:

Minimax dựa trên quy tắc logic đơn giản: MAX luôn chọn giá trị lớn nhất, MIN luôn chọn giá trị nhỏ nhất.

* + Khả năng dự đoán:

Minimax không chỉ phân tích trạng thái hiện tại mà còn dự đoán nhiều bước đi tiếp theo, giúp người chơi chuẩn bị trước cho các kịch bản có thể xảy ra.

* **Nhược điểm:**
  + Chi phí tính toán cao:

Số lượng nút trong cây trò chơi tăng theo cấp số nhân với độ sâu của cây. Với các trò chơi phức tạp như cờ vua hoặc ô ăn quan, việc duyệt toàn bộ cây là không khả thi.

* + Không hiệu quả với trò chơi lớn:

Trong các trò chơi có không gian trạng thái lớn, Minimax không thể áp dụng trực tiếp mà không có các cải tiến như cắt tỉa Alpha-Beta.

* + Phụ thuộc vào hàm đánh giá:

Kết quả của Minimax phụ thuộc nhiều vào độ chính xác của hàm đánh giá trạng thái, đặc biệt khi cây không được duyệt hết.

* **Ứng dụng của Minimax**

Minimax được sử dụng trong nhiều lĩnh vực, đặc biệt là trong các trò chơi đối kháng. Trong trò chơi ô ăn quan, giải thuật Minimax có thể giúp người chơi tính toán nước đi tối ưu bằng cách dự đoán các phản ứng của đối thủ và lựa chọn chiến lược phù hợp.

Ngoài trò chơi, Minimax còn được áp dụng trong:

* **Kinh tế học:** Dùng để phân tích các tình huống cạnh tranh giữa các doanh nghiệp.
* **Quân sự:** Hỗ trợ ra quyết định trong chiến lược đối phó.
* **Trí tuệ nhân tạo:** Phát triển các hệ thống có khả năng ra quyết định thông minh trong môi trường cạnh tranh.
* **Giải thuật cắt tỉa ALPHA-BETA**
* **Khái niệm cắt tỉa Alpha-Beta**

Cắt tỉa Alpha-Beta (Alpha-Beta Pruning) là một cải tiến của giải thuật Minimax, được sử dụng để giảm bớt số lượng trạng thái cần đánh giá trong cây trò chơi. Phương pháp này không làm thay đổi kết quả cuối cùng của Minimax nhưng giúp tăng hiệu quả tính toán, đặc biệt là trong các trò chơi có không gian trạng thái lớn như cờ vua, cờ tướng hay ô ăn quan.

Nguyên tắc của cắt tỉa Alpha-Beta là loại bỏ (cắt tỉa) các nhánh không cần thiết của cây trò chơi, tức là những nhánh mà kết quả của chúng không ảnh hưởng đến quyết định cuối cùng. Điều này giúp tiết kiệm tài nguyên tính toán và cho phép giải thuật xử lý các cây trò chơi sâu hơn trong cùng một khoảng thời gian.

* Ý tưởng chính của Alpha-Beta

Trong cây trò chơi, hai giá trị alpha và beta được sử dụng để lưu giữ thông tin về khoảng giá trị mà MAX và MIN đang quan tâm:

* + Alpha (α): Giá trị tốt nhất hiện tại mà MAX có thể đảm bảo tại bất kỳ nút nào thuộc lượt MAX.
  + Beta (β): Giá trị tốt nhất hiện tại mà MIN có thể đảm bảo tại bất kỳ nút nào thuộc lượt MIN.

Quá trình duyệt cây trò chơi với Alpha-Beta diễn ra như sau:

* + Khi duyệt đến một nút:

Nếu giá trị của nút đó vượt ra ngoài phạm vi mà đối thủ quan tâm (giá trị của alpha hoặc beta), các nhánh con của nút đó không cần phải được kiểm tra, vì chúng không ảnh hưởng đến quyết định của đối thủ.

Nếu giá trị nằm trong khoảng quan tâm, tiếp tục duyệt sâu hơn.

* + Cắt tỉa xảy ra khi:

Một nhánh được xác định là không thể dẫn đến một kết quả tốt hơn so với giá trị alpha hoặc beta hiện tại.

* + Ví dụ:

Tại một nút của MAX, nếu giá trị của nút con thấp hơn beta hiện tại, MIN chắc chắn sẽ không chọn nhánh này, và do đó không cần xét tiếp.

Tương tự, tại một nút của MIN, nếu giá trị của nút con lớn hơn alpha hiện tại, MAX chắc chắn sẽ không chọn nhánh này.

* **Quy trình cắt tỉa Alpha-Beta**

Alpha-Beta được triển khai trên cây trò chơi bằng cách sử dụng đệ quy, giống như Minimax, nhưng có thêm hai biến alpha và beta. Quy trình tổng quát:

* + Khởi tạo alpha là giá trị âm vô cùng (-∞), beta là giá trị dương vô cùng (+∞).
  + Khi duyệt qua các nút:
    - Tại lượt của MAX:

Cập nhật alpha nếu tìm được giá trị tốt hơn.

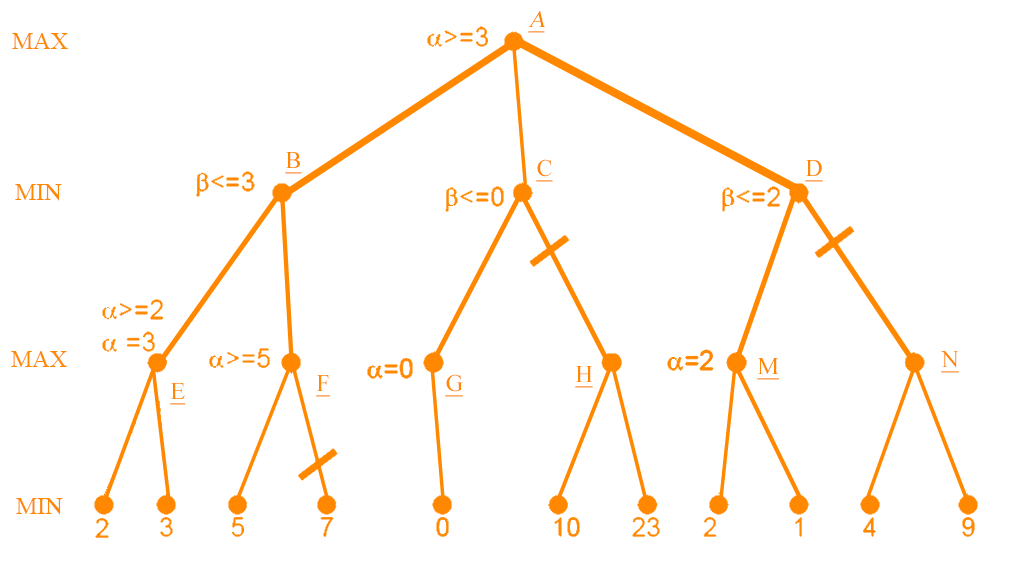
Ngừng duyệt nếu alpha >= beta (cắt tỉa).

* + - Tại lượt của MIN:

Cập nhật beta nếu tìm được giá trị nhỏ hơn.

Ngừng duyệt nếu beta <= alpha (cắt tỉa).

* + Kết thúc duyệt cây khi không còn nút nào cần xét, trả về giá trị tốt nhất đã tìm được.
* **Ví dụ minh họa**



Hình 1.3 Hình minh họa cây trò chơi sử dụng 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. Duyệt nguyên cây này sẽ khá dài dòng nhưng để bạn hiểu tôi sẽ viết ra các bước sau.

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).

Lưu ý là luôn luôn 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 con của B thì tại B – beta = 3, và tại nút A – alpha ≥ 3.

* **Ưu điểm**
  + Tăng hiệu quả:

Alpha-Beta loại bỏ các nhánh không cần thiết, giúp giảm số lượng nút cần duyệt. Trong trường hợp tốt nhất, Alpha-Beta có thể giảm độ phức tạp từ O(bd)(với b là số nhánh mỗi nút, d là độ sâu của cây) xuống còn O(bd/2). Điều này có nghĩa là nó có thể xử lý các cây sâu hơn gấp đôi so với Minimax thông thường.

* + Không thay đổi kết quả:

Dù có cắt tỉa, kết quả cuối cùng của Alpha-Beta vẫn giống với Minimax, đảm bảo tính chính xác.

* + Thích hợp cho các trò chơi lớn:

Các trò chơi như cờ vua, cờ tướng hay ô ăn quan có không gian trạng thái rất lớn, Alpha-Beta cho phép tối ưu hóa tính toán trong các bài toán này.

* **Nhược điểm**
  + Phụ thuộc vào thứ tự duyệt

Hiệu quả của Alpha-Beta phụ thuộc vào thứ tự duyệt các nút. Nếu các nút được duyệt theo thứ tự "tốt nhất trước", hiệu quả cắt tỉa sẽ tối ưu. Ngược lại, nếu thứ tự không thuận lợi, hiệu quả có thể giảm đáng kể.

* + Không phù hợp với trò chơi có thông tin không hoàn toàn:

Alpha-Beta chỉ hiệu quả với các trò chơi thông tin hoàn toàn (như cờ vua, ô ăn quan), nơi tất cả trạng thái có thể biết trước.

* + Đòi hỏi đánh giá trạng thái chính xác:

Hàm đánh giá phải được thiết kế tốt để cắt tỉa có hiệu quả.

* **Ứng dụng trong trò chơi ô ăn quan**

Trong trò chơi ô ăn quan, Alpha-Beta có thể được sử dụng để tính toán nước đi tối ưu bằng cách:

* Xây dựng cây trò chơi đại diện cho các trạng thái sau mỗi lượt đi.
* Sử dụng Alpha-Beta để cắt tỉa các nhánh không cần thiết, chỉ tập trung vào các nước đi có khả năng mang lại lợi thế lớn nhất.

Ví dụ, khi lựa chọn ô để di chuyển quân, Alpha-Beta sẽ giúp xác định các nước đi tốt nhất, đồng thời loại bỏ những lựa chọn dẫn đến kết quả bất lợi cho người chơi.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ

## 2.1 Phân tích bài toán:

Bài toán được đặt ra là xây dựng một chương trình chơi trò chơi ô ăn quan giữa hai người chơi, trong đó một người chơi có thể được thay thế bởi máy tính. Mục tiêu chính là phát triển một hệ thống trí tuệ nhân tạo để điều khiển máy tính trong vai trò người chơi đối kháng. Máy tính cần có khả năng đưa ra các nước đi tối ưu để tăng cơ hội chiến thắng.

Bài toán được phân tích dựa trên các yêu cầu sau:

* **Mô tả bài toán:**

Trò chơi ô ăn quan là một trò chơi dân gian truyền thống với quy tắc đơn giản nhưng mang tính chiến lược cao. Hai người chơi sẽ lần lượt đi, mỗi lượt đi phải chọn ô chứa quân và thực hiện rải quân theo luật định. Mục tiêu là ăn được nhiều quân nhất có thể.

* **Yêu cầu chức năng:**
  + Người chơi:

Có thể chơi với nhau hoặc chơi với máy.

Máy tính sẽ sử dụng thuật toán Minimax,

* Luật chơi:

Chương trình phải tuân thủ các quy tắc của trò chơi ô ăn quan.

Mỗi lượt đi phải kiểm tra trạng thái hợp lệ của bàn cờ và xác định nước đi tiếp theo.

* + Điểm số:

Điểm số của mỗi người chơi được tính dựa trên số quân mà họ ăn được.

* **Phân tích tính năng AI:**

Hỗ trợ người chơi: Máy tính cần phân tích trạng thái bàn cờ hiện tại để dự đoán các nước đi của đối thủ và lựa chọn chiến lược tốt nhất.

Cân bằng độ khó: Có thể áp dụng các chiến lược như tham lam hoặc sử dụng thuật toán Minimax để điều chỉnh mức độ thông minh của máy.

Hiệu quả tính toán: Để đảm bảo chương trình không bị chậm, cần tối ưu hóa bằng giải thuật cắt tỉa Alpha-Beta khi thực hiện tìm kiếm Minimax.

* **Các thách thức:**
* Không gian trạng thái của trò chơi rất lớn, đặc biệt khi số lượng quân trên bàn cờ cao. Điều này đòi hỏi các thuật toán phải được tối ưu hóa để đạt hiệu quả.
* Xây dựng hàm đánh giá trạng thái hợp lý để AI có thể đưa ra quyết định thông minh.
* Lập trình giao diện thân thiện để người chơi dễ dàng tương tác với chương trình.
* **Giải pháp:**

Cấu trúc chương trình: Chương trình sẽ được chia thành các hàm chính:

* Hàm xử lý trò chơi (bao gồm luật chơi, trạng thái bàn cờ).
* Hàm AI (các thuật toán Minimax).
* Hàm giao diện người dùng.

Thuật toán: Sử dụng Minimax để xử lý trò chơi.

Hàm đánh giá: Xây dựng hàm đánh giá trạng thái dựa trên các tiêu chí như: số quân mà người chơi ăn được,.

## 2.2 Cấu trúc của chương trình:

### 2.2.1 Xây dựng hàm Minimax:

Hàm Minimax được thiết kế để duyệt cây tìm kiếm và tìm ra nước đi tốt nhất dựa trên chiến lược tối ưu của cả hai bên:

Đầu vào:

* Trạng thái hiện tại của trò chơi.
* Độ sâu tối đa để dừng tìm kiếm.

Đầu ra:

* Giá trị tối ưu và nước đi tương ứng.
* Hoạt động chính:
* Nếu là MAX, chọn giá trị lớn nhất từ các nhánh con.
* Nếu là MIN, chọn giá trị nhỏ nhất từ các nhánh con.

Ưu điểm: Đảm bảo nước đi tối ưu nếu duyệt hết cây.

Nhược điểm: Tốn kém tài nguyên khi không gian tìm kiếm quá lớn.

### 2.2.2 Xây dựng hàm Minimax có cắt tỉa:

Hàm Minimax được thiết kế để duyệt cây tìm kiếm và tìm ra nước đi tốt nhất dựa trên chiến lược tối ưu của cả hai bên:

Đầu vào:

* Trạng thái hiện tại của trò chơi.
* Độ sâu tối đa để dừng tìm kiếm.

Đầu ra:

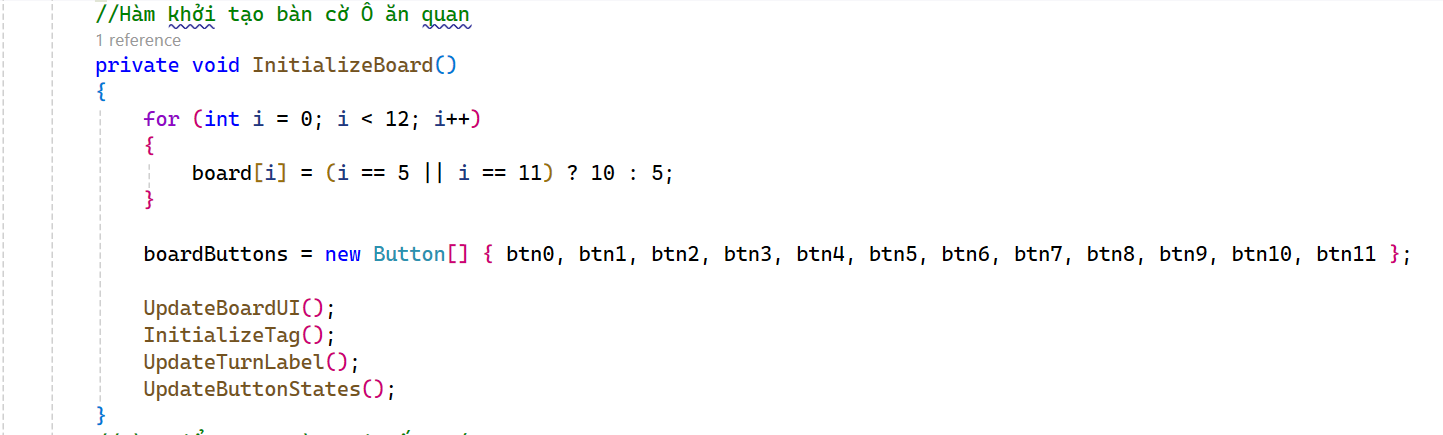
* Giá trị tối ưu và nước đi tương ứng.
* Hoạt động chính:
* Nếu là MAX, chọn giá trị lớn nhất từ các nhánh con.
* Nếu là MIN, chọn giá trị nhỏ nhất từ các nhánh con.

Ưu điểm:

* Giảm số lượng trạng thái cần đánh giá.
* Duy trì kết quả chính xác như Minimax.

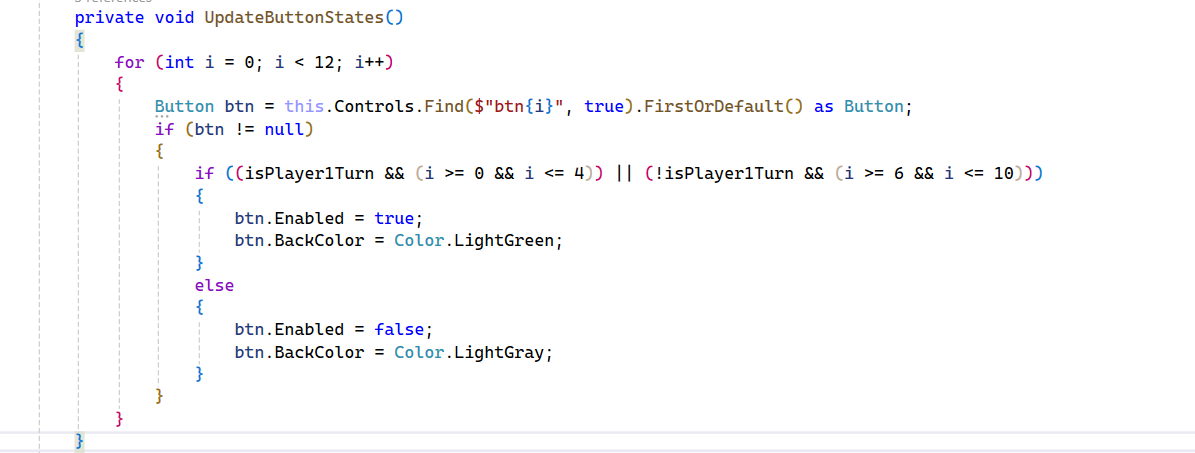
### 2.3.4 Các hàm xử lý cơ bản trong chương trình:

* Hàm khởi tạo trạng thái:
  + Tạo trạng thái trò chơi ban đầu.

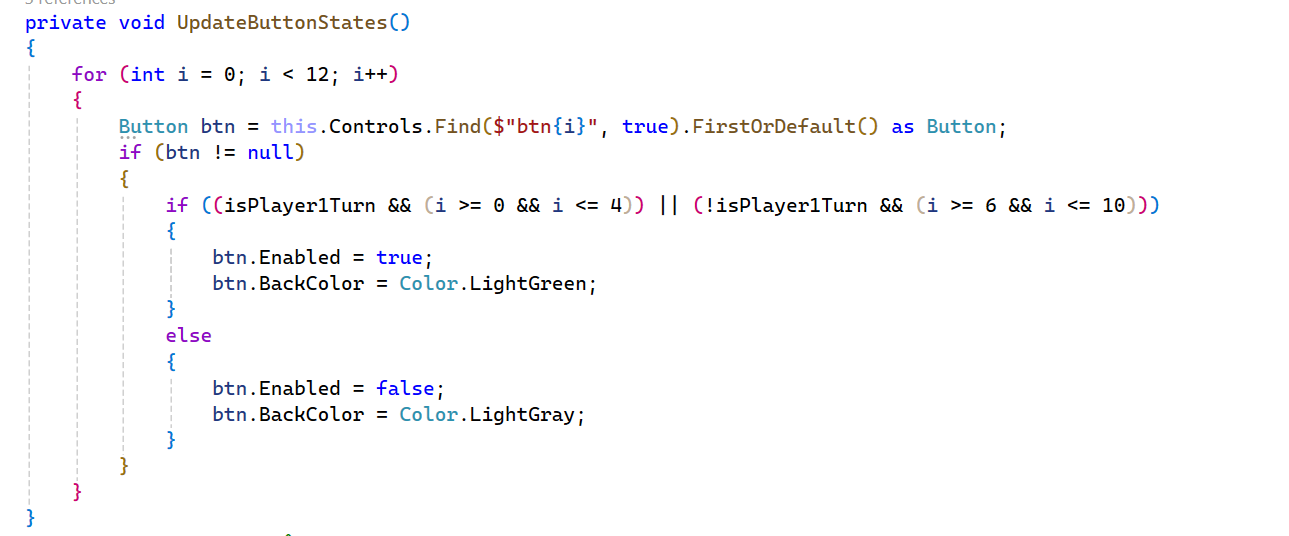


Hình 2.1 Hàm khởi tạo ô ăn quan

* Hàm xác định nước đi hợp lệ:
  + Xác định các ô trống hoặc các nước đi có thể thực hiện

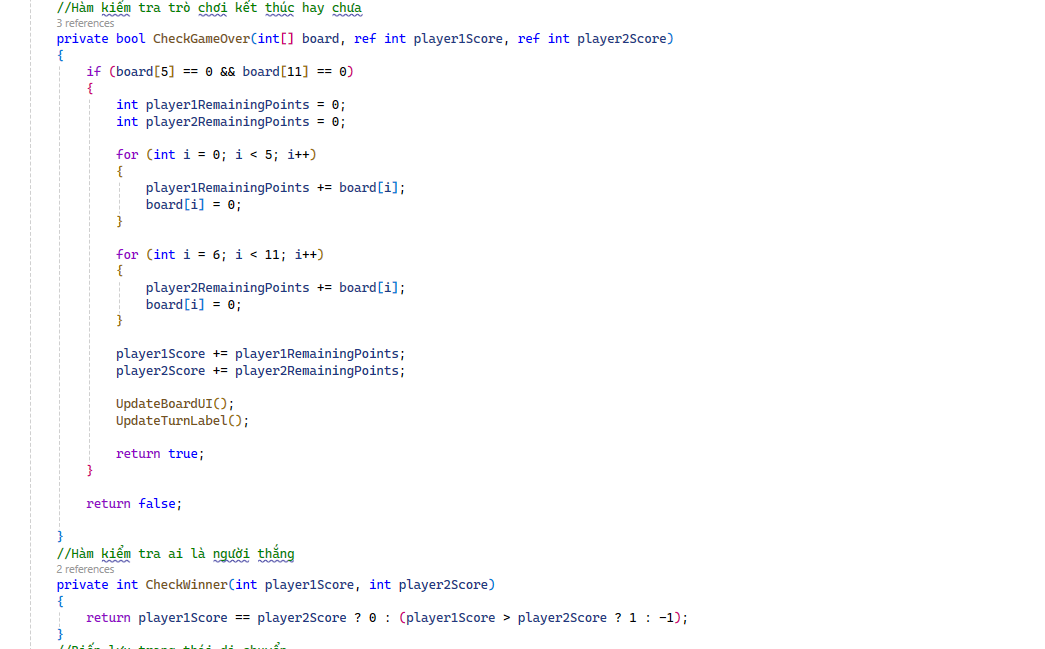


Hình 2.2 Hàm xác định nước đi hợp lệ

* Hàm cập nhật nút:
  + Hàm cập nhật trạng thái nút thuộc về người chơi 

Hình 2.3 Hàm cập nhật nút

* Hàm kiểm tra điều kiện kết thúc:
  + Xác định liệu trò chơi đã kết thúc (thắng, thua, hòa).



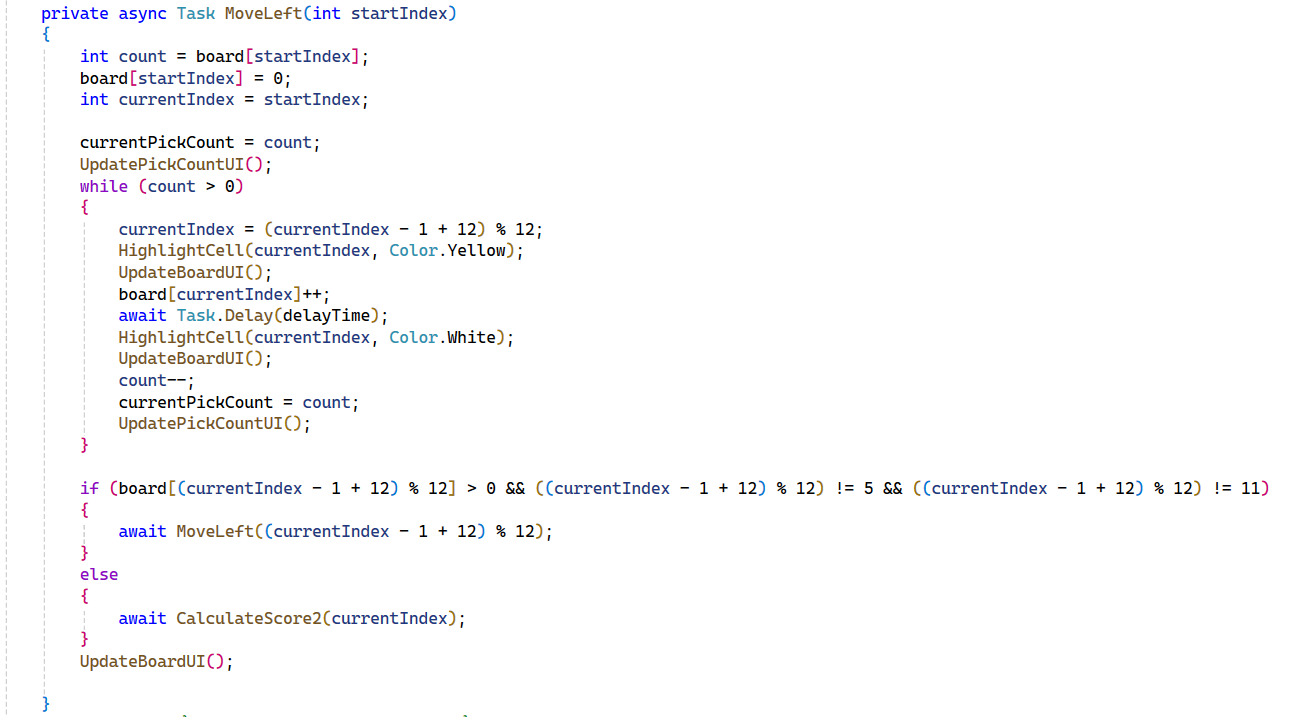
Hình 2.4 Hàm kiểm tra điều kiện kết thúc

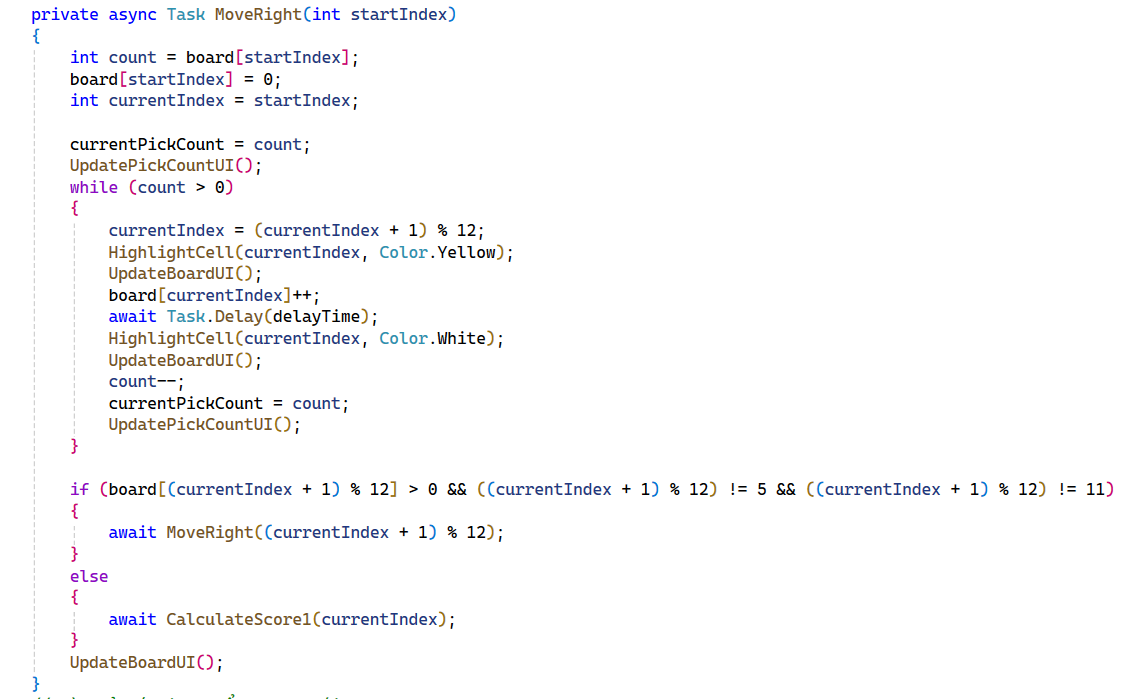
* Hàm cập nhật trạng thái:
  + Áp dụng nước đi và cập nhật trạng thái trò chơi.



Hình 2.5 Hàm cập nhật trạng thái

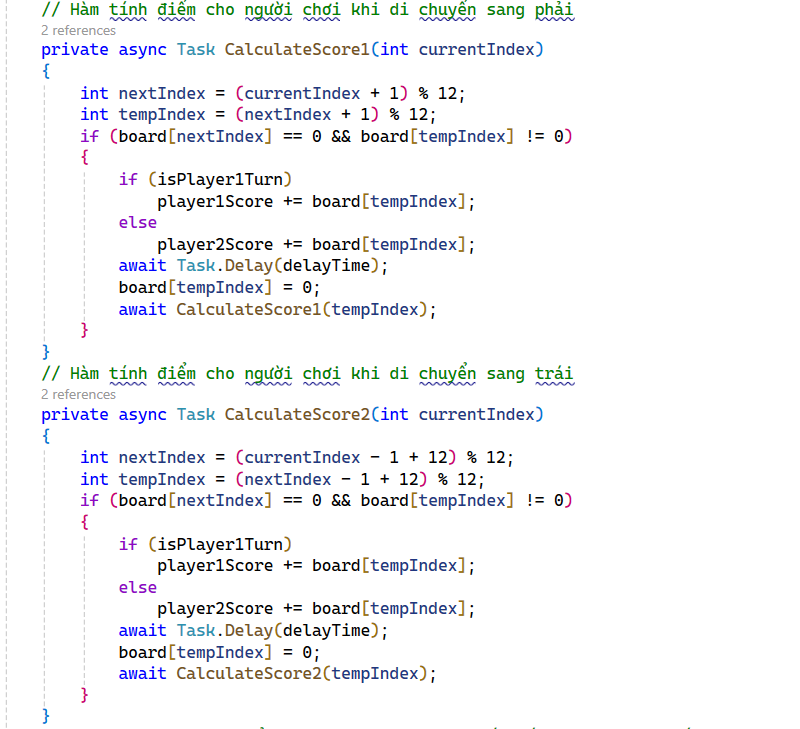
* Hàm di chuyển sang trái hoặc phải:





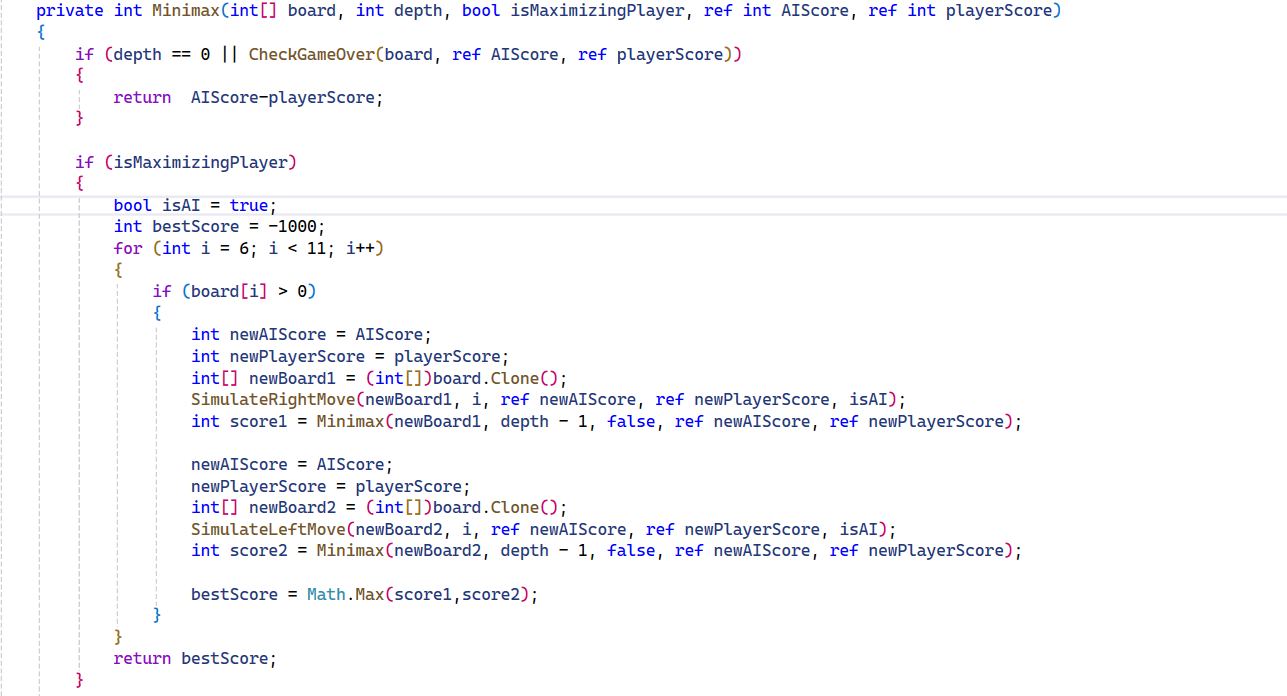
Hình 2.6 Hàm di chuyển sang trái hoặc phải

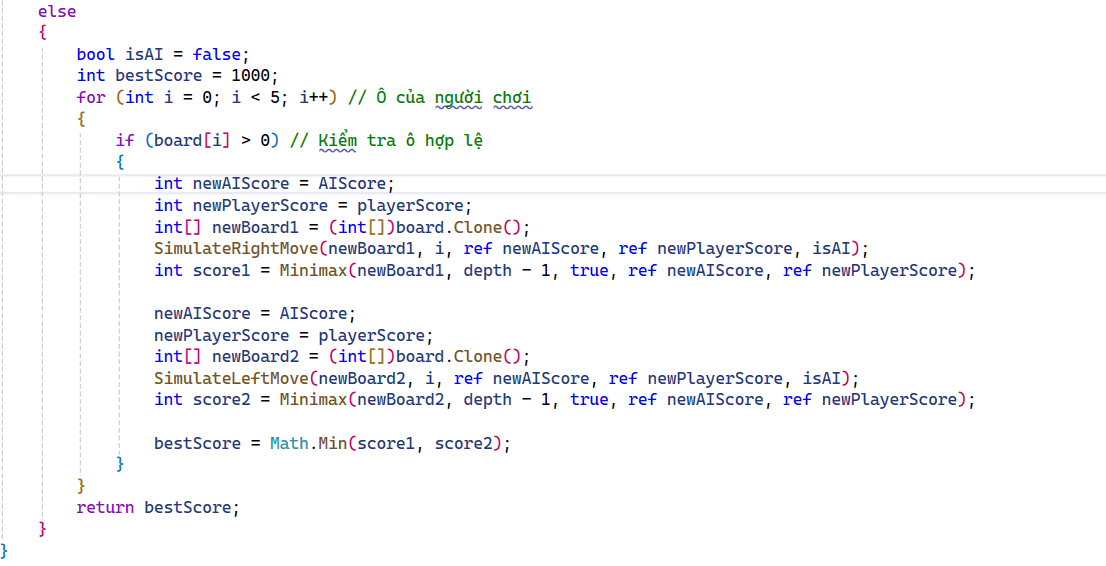
* Hàm tính điểm khi di chuyển sang trái hoặc phải:



Hình 2.7 Hàm tính điểm

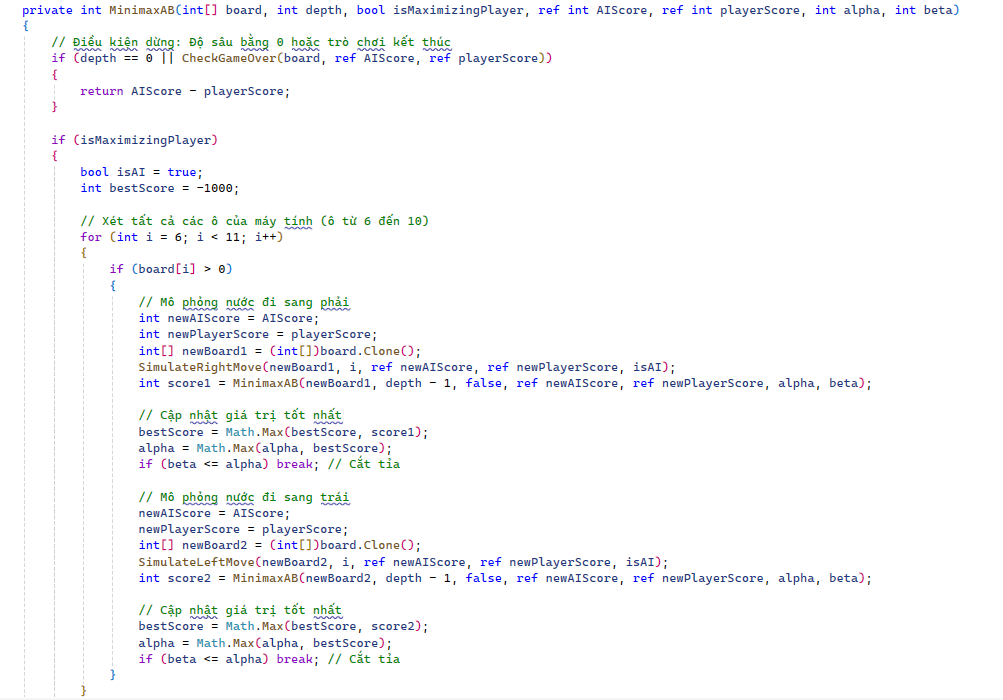
* Hàm minimax:
  + Hàm minimax không có cắt tỉa Alpha-beta.

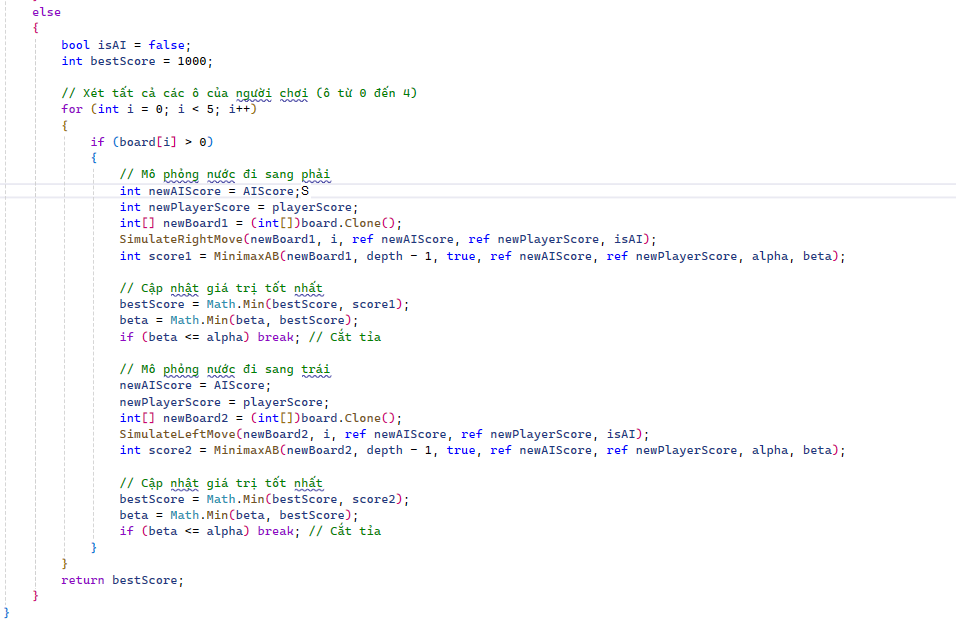




Hình 2.8 Hàm minimax không có cắt tỉa Alpha-Beta

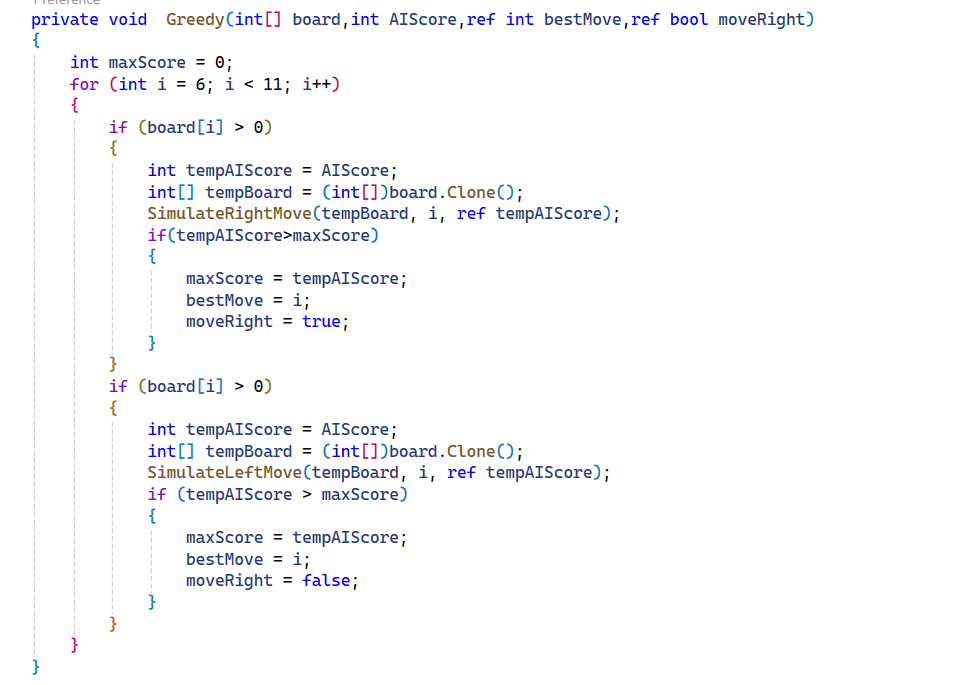
* + Hàm minimax có cắt tỉa Alpha-beta.





Hình 2.9 Hàm minimax có cắt tỉa Alpha-Beta

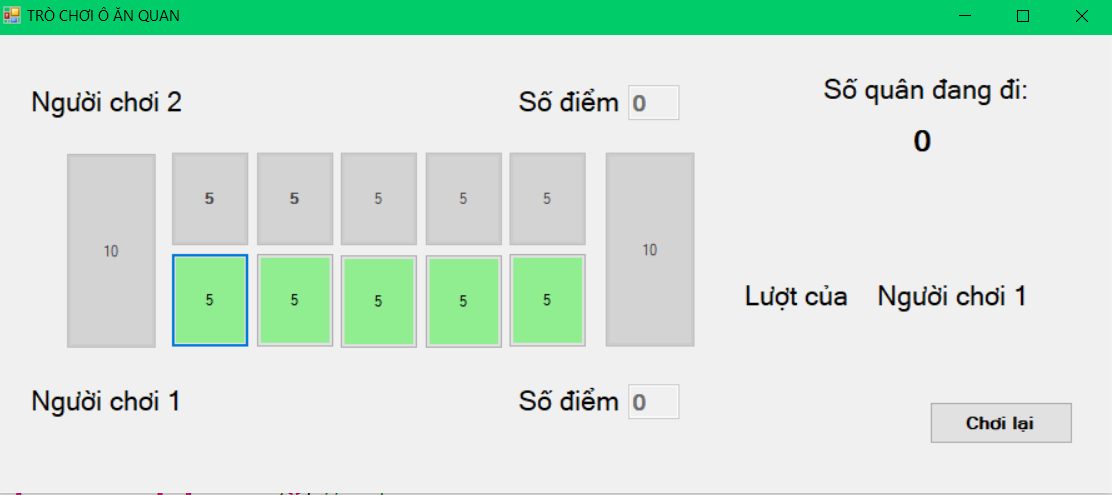
* Hàm tham lam:



Hình 2.10 Hàm tham lam

# CHƯƠNG 3: CHƯƠNG TRÌNH ỨNG DỤNG

## 3.1 Giới thiệu bàn chơi ô ăn quan:



Hình 3.1: Bàn chơi ô ăn quan cho 2 người

**Bàn chơi:** bàn chơi được kẻ thành một hình chữ nhật rồi chia hình chữ nhật đó thành mười ô vuông, mỗi bên có năm ô đối xứng nhau. Ở hai cạnh ngắn hơn của hình chữ nhật, kẻ hai ô hình bán nguyệt hoặc hình vòng cung hướng ra phía ngoài. Các ô hình vuông gọi là ô dân còn hai ô hình bán nguyệt hoặc vòng cung gọi là ô quan.

**Quân chơi:** gồm hai loại quan và dân. Số lượng quan là 10, dân có số lượng là 5 và tùy theo luật chơi mà số lượng quân và dân có thể khác nhau.

**Bố trí quân chơi:** quan được đặt trong hai ô hình bán nguyệt hoặc cánh cung, mỗi ô một quân, dân được bố trí vào các ô vuông với số quân đều nhau, mỗi ô 5 dân.

## 3.2 Cài đặt:

### 3.2.1 Ngôn ngữ và công cụ sử dụng:

Trong quá trình thực hiện đề tài, nhóm đã lựa chọn sử dụng các ngôn ngữ lập trình và công cụ hỗ trợ phù hợp để đảm bảo tính hiệu quả và đáp ứng đầy đủ yêu cầu của đề tài. Cụ thể như sau:

* **Ngôn ngữ lập trình:**

Nhóm sử dụng ngôn ngữ lập trình C# để xây dựng ứng dụng. Đây cũng là một ngôn ngữ được tối ưu hóa để làm việc với Windows Forms, cho phép xây dựng giao diện đồ họa dễ sử dụng và trực quan.

* **Công cụ phát triển:**

Nhóm lựa chọn Visual Studio làm môi trường phát triển Windows Forms (WinForm):

Sự kết hợp giữa ngôn ngữ C#, công cụ Visual Studio và Windows Forms đã giúp nhóm hoàn thành ứng dụng với giao diện thân thiện, chức năng đầy đủ, và hiệu suất ổn định, đáp ứng tốt các yêu cầu đặt ra trong đề tài.

### 3.2.2 Độ phức tạp của chương trình:

* **Thuật toán Minimax có cắt tỉa alpha-beta**
* Không cắt tỉa: Thuật toán Minimax duyệt toàn bộ cây trạng thái trò chơi. Độ phức tạp thời gian là O(b^d), trong đó:
  + b: Số lượng nước đi trung bình tại mỗi trạng thái (độ rộng của cây).
  + d: Chiều sâu của cây tìm kiếm (số lượt đi mà thuật toán dự đoán).
* Có cắt tỉa alpha-beta:
  + Kỹ thuật cắt tỉa alpha-beta giúp loại bỏ những nhánh cây không cần thiết. Khi áp dụng hiệu quả, độ phức tạp giảm xuống còn O(b^{d/2}) trong trường hợp tốt nhất, vì số lượng nút cần kiểm tra giảm đáng kể.
* **Thuật toán tham lam**
* Thuật toán tham lam chỉ xét trạng thái hiện tại mà không mở rộng cây trạng thái. Do đó, độ phức tạp thời gian chủ yếu phụ thuộc vào số lượng nước đi khả dĩ tại trạng thái hiện tại.
* Độ phức tạp: O(b) trong đó:
  + b: Số lượng nước đi trung bình (giống Minimax).
* **Kết luận**
* **Minimax có cắt tỉa**: Hiệu quả hơn khi cần chiến lược dài hạn và quyết định chính xác. Tuy nhiên, chi phí tính toán vẫn cao nếu chiều sâu ddd lớn.
* **Tham lam:** Phù hợp với các bài toán hoặc trạng thái cần phản hồi nhanh, nhưng không thích hợp cho chiến lược phức tạp trong trò chơi như Ô ăn quan.

## 3.3 Giới thiệu chương trình:

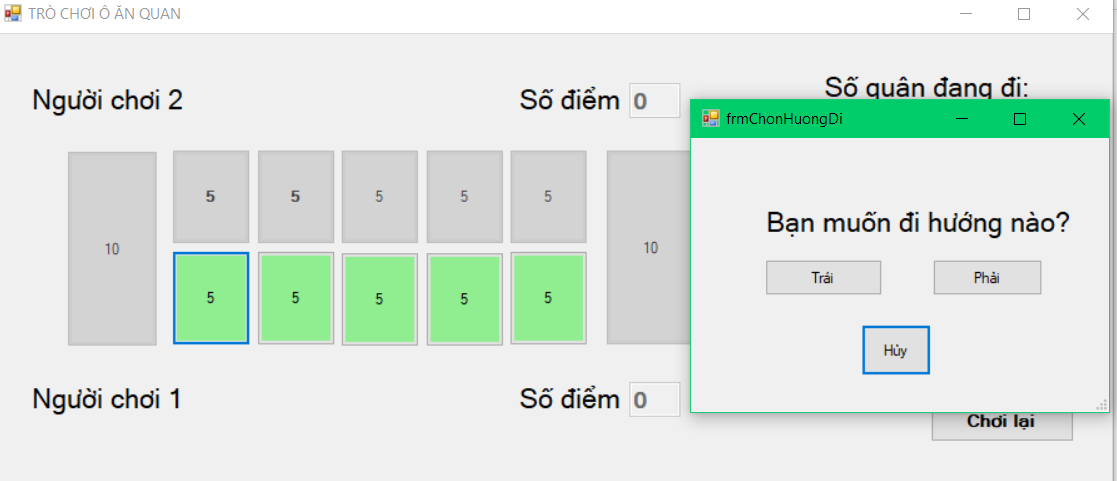
**

Hình 3.2: Trang chủ của ứng dụng

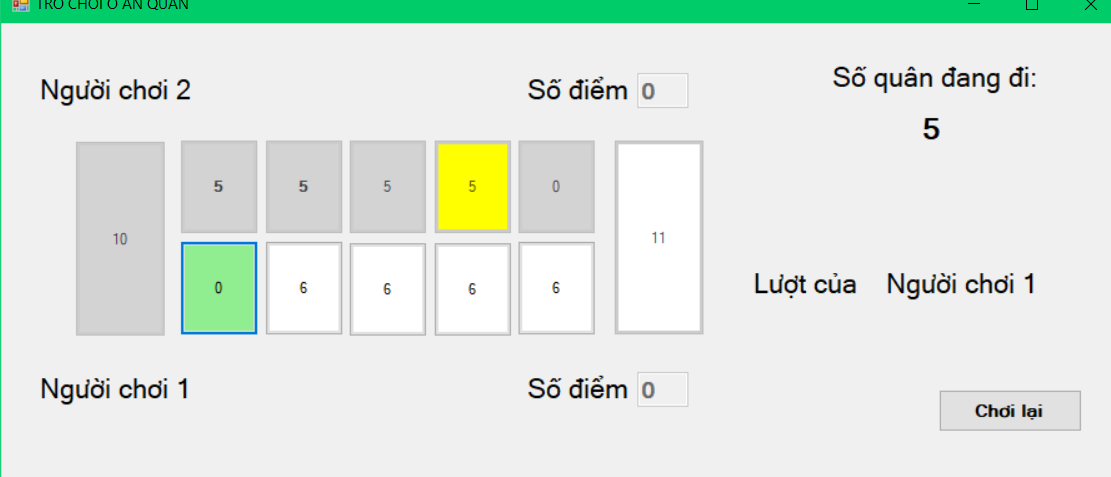
*A screenshot of a computer

Description automatically generated*

Hình 3.3: Giao diện bàn chơi ô ăn quan cho 2 người

**

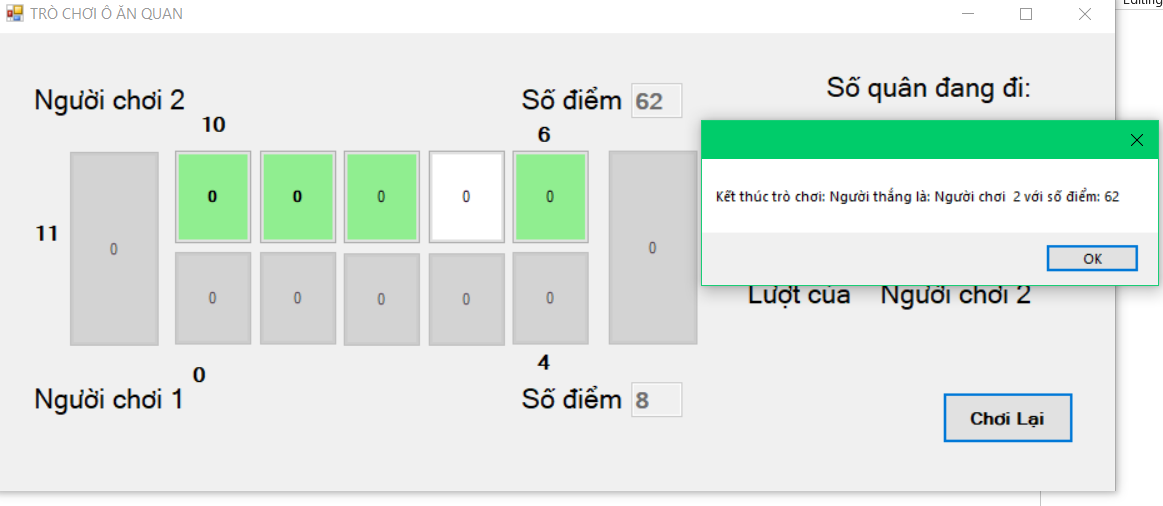
Hình 3.4: Giao diện chọn hướng đi

**

Hình 3.5: Giao diện khi di chuyển các quân trong ô ăn quan

**

Hình 3.6: Giao diện khi rải thêm quân

**

Hình 3.7 : Giao diện khi trò chơi kết thúc

# CHƯƠNG 4: KẾT LUẬN

## 4.1 Kết quả đạt được:

Sau quá trình nghiên cứu và phát triển, nhóm đã đạt được những kết quả cụ thể như sau:

* **Hiểu và triển khai trò chơi Ô ăn quan trên nền tảng Desktop:**

Nhóm đã nghiên cứu, tìm hiểu quy tắc và logic của trò chơi truyền thống Ô ăn quan, từ đó áp dụng để phát triển thành công phiên bản kỹ thuật số hoạt động trên máy tính.

Áp dụng được thuật toán minimax,cắt tỉa Alpha-Beta và thuật toán tham lam

* **Xây dựng chương trình hoàn chỉnh:**

Một chương trình cụ thể đã được nhóm xây dựng với đầy đủ các thành phần cần thiết để mô phỏng trò chơi Ô ăn quan, bao gồm cả giao diện người dùng và xử lý logic trò chơi.

* **Đảm bảo chạy đúng luật chơi:**

Chương trình được thiết kế để tuân thủ chặt chẽ các quy tắc của trò chơi Ô ăn quan, đảm bảo tính chính xác và công bằng khi người chơi tương tác.

* **Hoàn thành các chức năng cơ bản:**

Nhóm đã hoàn thành việc phát triển các tính năng cơ bản của game, bao gồm thao tác chơi, ghi nhận điểm số, và xử lý lượt đi giữa các người chơi.

* **Giao diện thân thiện và dễ sử dụng:**

Giao diện của trò chơi được thiết kế trực quan, giúp người chơi dễ dàng tương tác và trải nghiệm một cách thoải mái.

Những kết quả trên thể hiện sự nỗ lực của nhóm trong việc chuyển hóa một trò chơi dân gian thành một ứng dụng hiện đại, mang đến trải nghiệm mới mẻ nhưng vẫn giữ được tinh thần truyền thống.

## 4.2 Hạn chế:

Mặc dù đã đạt được nhiều kết quả tích cực, chương trình vẫn tồn tại một số hạn chế cần khắc phục trong tương lai:

* Giao diện chưa thực sự hấp dẫn
* Thiếu chức năng kết nối giữa hai máy
* Chưa tích hợp yếu tố thời gian

## 4.3 Định hướng phát triển thêm:

Để nâng cao chất lượng và mở rộng tính năng của trò chơi, nhóm đề ra các định hướng phát triển trong tương lai như sau:

* Cải thiện giao diện người dùng
* Tích hợp chế độ chơi trực tuyến
* Bổ sung tính năng đếm thời gian
* Xây dựng phiên bản đa nền tảng
* Thêm tùy chỉnh và nâng cấp trò chơi

# TÀI LIỆU THAM KHẢO

1. **Vũ Trọng Quang**, *“Giải thuật cắt tỉa Alpha-beta”*, IO STREAM. [[Xem tại đây].](https://www.iostream.co/article/giai-thuat-cat-tia-alpha-beta-Wu7F1)
2. **ICHI.PRO**, *“Hiểu thuật toán Minimax”*, ICHI.PRO. [[Xem tại đây].](https://ichi.pro/vi/hieu-thuat-toan-minimax-125780313305798)
3. **WEKIPEDIA**, *“Minimax”*, WEKIPEDIA. [[Xem tại đây].](https://vi.wikipedia.org/wiki/Minimax)
4. **WEKIPEDIA**, *“Ô ăn quan”*, WEKIPEDIA. [[Xem tại đây].](https://vi.wikipedia.org/wiki/%C3%94_%C4%83n_quan)
5. **Bro**, *“Thuật toán Minimax (AI trong Game)”*, VIBLO. [[Xem tại đây].](https://viblo.asia/p/thuat-toan-minimax-ai-trong-game-APqzeaVVzVe)