**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**ĐỀ TÀI**

**THIẾT KẾ TRÒ CHƠI ĐÁNH CARO GIỮA NGƯỜI VÀ MÁY**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: TS. Nguyễn Đình Hiển  Sinh viên thực hiện: | | |
| Võ Đoàn Hoàng Long | **-** | Phạm Hoàng Hải |
| Nguyễn Thanh Hòa | **-** | Nguyễn Lê Vinh |
| Nguyễn Ngọc Vũ Triều | **-** | Lê Trần Minh Hiền |
| Đoàn Phước Nhật | **-** | Nguyễn Thị Ngọc Hiền |
| Đặng Đình Duy | **-** | Lê Âu Hải |
| Lớp: Công nghệ thông tin  Khóa: 60 |  |  |

TP. Hồ Chí Minh, năm 2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**ĐỀ TÀI**

**THIẾT KẾ TRÒ CHƠI ĐÁNH CARO GIỮA NGƯỜI VÀ MÁY**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: TS. Nguyễn Đình Hiển  Sinh viên thực hiện: | | |
| Võ Đoàn Hoàng Long | **-** | Phạm Hoàng Hải |
| Nguyễn Thanh Hòa | **-** | Nguyễn Lê Vinh |
| Nguyễn Ngọc Vũ Triều | **-** | Lê Trần Minh Hiền |
| Đoàn Phước Nhật | **-** | Nguyễn Thị Ngọc Hiền |
| Đặng Đình Duy | **-** | Lê Âu Hải |
| Lớp: Công nghệ thông tin  Khóa: 60 |  |  |

TP. Hồ Chí Minh, năm 2022

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI** | **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM** |
| **PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** | **Độc lập – Tự do – Hạnh phúc** |

**NHIỆM VỤ THIẾT KẾ ĐỒ ÁN CUỐI KỲ**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

1. **Tên đề tài**

Thiết kế trò chơi đánh caro giữa người và máy.

1. **Mục đích, yêu cầu**
   1. **Mục đích:**

Xây dựng “Thiết kế trò chơi đánh caro giữa người và máy” với mục đích chính là ứng dụng được các thuật toán vào trò chơi. Trò chơi caro được xây dựng trên nền tảng WinForms.

* 1. **Yêu cầu:**
* **Yêu cầu cơ bản**
* Sử dụng ngôn ngữ lập trình C#.
* Sử dụng công cụ Visual Studio 2019.
* **Yêu cầu về tổ chức code**
  + Tổ chức code ngay ngắn dễ đọc, lược bỏ các thành phần rườm rà.

1. **Các kết quả chính dự kiến sẽ đạt được**

* Hoàn thiện trò chơi với 2 chế độ: chơi với người, chơi với máy.

1. **Giáo viên và cán bộ hướng dẫn**

Họ tên giảng viên: TS. Nguyễn Đình Hiển.

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải.

MỤC LỤC

[NHIỆM VỤ THIẾT KẾ ĐỒ ÁN CUỐI KỲ 1](#_Toc103540625)

[MỤC LỤC 2](#_Toc103540626)

[DANH MỤC HÌNH ẢNH 3](#_Toc103540627)

[LỜI MỞ ĐẦU 4](#_Toc103540628)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 5](#_Toc103540629)

[1. Công cụ Visual Studio 2019 5](#_Toc103540630)

[CHƯƠNG 2. CHIẾN LƯỢC MINIMAX 6](#_Toc103540631)

[2. 1 Giới thiệu 6](#_Toc103540632)

[2.1. 1 Trò chơi có tổng bằng không (Zero-sum-game) 6](#_Toc103540633)

[2.1. 2 Cây trò chơi. 6](#_Toc103540634)

[2. 2 Giải thuật Minimax 7](#_Toc103540635)

[2.2. 1 Ý tưởng 8](#_Toc103540636)

[2.2. 2 Áp dụng giải thuật Minimax đến độ sâu lớp cố định 12](#_Toc103540637)

[2.2. 3 Thủ tục Minimax 15](#_Toc103540638)

[2.2. 4 Đánh giá 17](#_Toc103540639)

[CHƯƠNG 3. THIẾT KẾ GIAO DIỆN 18](#_Toc103540640)

[CHƯƠNG 4. KẾT LUẬN 20](#_Toc103540641)

[TÀI LIỆU THAM KHẢO 21](#_Toc103540642)

DANH MỤC HÌNH ẢNH

[Hình 3. 1 Giao diện chơi giữa Player 1 – Player 2 20](#_Toc103538819)

[Hình 3. 2 Giao diện chuyển sang chế độ Play with computer 20](#_Toc103538820)

[Hình 3. 3 Bắt sự kiện chặn 2 đầu vẫn thắng. 21](#_Toc103538821)

LỜI MỞ ĐẦU

Để hoàn thành đề tài cuối kì này trước hết em xin gửi đến quý thầy, cô **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh** lời cảm ơn chân thành vì đã truyền đạt cho em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt em xin gửi đến thầy Nguyễn Đình Hiển cảm ơn sâu sắc nhất vì thầy đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình thực hiện đề tài. Tuy đề tài không được lớn nhưng nếu không được sự hướng dẫn chỉ bảo tận tình của thầy thì đề tài bài tập lớn này khó có thể hoàn thành được.

Vì thời gian làm đề tài đồ án cuối kì có hạn cũng như hiểu biết của nhóm còn hạn chế, chúng em cũng đã nỗ lực hết sức để hoàn thành bài báo cáo đồ án cuối kì một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi.  Chúng em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc quý thầy cô trong **Bộ môn Công nghệ thông tin** đang công tác tại Bộ phận một cửa – Phòng Tổ chức hành chính luôn mạnh khoẻ, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Chúng em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày 17 tháng 05 năm 2022

**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

1. Công cụ Visual Studio 2019

**Visual studio** là một trong những công cụ hỗ trợ lập trình website rất nổi tiếng nhất hiện nay của Mcrosoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ Microsoft. Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Visual Studio 2019 như một cú lột xác hoàn hảo với sự cải thiện toàn diện cả về giao diện thao tác và hiệu suất. Giao diện Start cũng trở nên đơn giản và dễ dàng hơn cho các ‘coder’ để thao tác nhanh hơn (nhất là khi muốn clone một repos từ Git hoặc mở một project/ folder có sẵn), màn hình chọn template code cũng được cải thiện, vùng soạn thảo code được mở rộng không gian hơn, trải nghiệm tìm kiếm mới, trình debug thông minh hơn

**CHƯƠNG 2. CHIẾN LƯỢC MINIMAX**

* 1. **Giới thiệu**

Thuật toán Minimax là thuật toán tìm kiếm chuyên dùng để trả về chuỗi nước đi tối ưu cho một người chơi trong trò chơi có tổng bằng không. 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).

Một trò chơi áp dụng chiến lược như tic-tac-toe, khi mà mỗi người chơi có thể thắng, thua, hoặc hòa. Nếu người chơi A có thể thắng trong một 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.

Ta sẽ nắm rõ, thế nào là một nước đi "tốt nhất". Giải thuật Minimax giúp tìm ra nước đi tốt nhất, bằng cách đi ngược từ cuối trò chơi trở về đầu. Tại mỗi bước, nó sẽ ước định rằng người A đang cố gắng tối đa hóa cơ hội thắng của A khi đến phiên anh ta, còn ở nước đi kế tiếp thì người chơi B cố gắng để tổi thiểu hóa cơ hội thắng của A.

1. **Trò chơi có tổng bằng không (Zero-sum-game)**

Trò chơi có tổng bằng không 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 tuý, cuối cùng dẫn tới tổng (+1- 1) = 0.

Cờ vua là một trò chơi có tổng bằng không 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. Thể thao là những ví dụ điển hình nhất của trò chơi có tổng bằng không. 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 không ấy .

Việc đầu tư kinh doanh chứng khoán cũng chính là một trò chơi có tổng bằng không, bởi vì ở đó, số tiền thua lỗ của nhà đầu tư này sẽ là tiền lãi của nhà đầu tư khác. Nhà đầu tư có thể mất trắng hoặc thắng lớn.

1. **Cây trò chơi.**

Các trạng thái bàn cờ khác nhau (hay còn gọi là một thế cờ, tình huống cờ) trong quá trình chơi có thể biểu diễn thành một cây tìm kiếm (được gọi là cây trò chơi - hình 1.3) và ta sẽ tiến hành tìm kiếm trên cây để tìm được nước đi tốt nhất. Cây trò chơi có các nút là các tình huống khác nhau của bàn cờ, các nhánh nối giữa các nút sẽ cho biết từ một tình huống bàn cờ chuyển sang tình huống khác thông qua chỉ một nước đi đơn nào đó. Tuy nhiên, các nước đi này diễn ra theo cặp do hai đấu thủ lần lượt tiến hành.Độ sâu của cây trò chơi là số tầng của cây. Thuật ngữ “nước đi” được thống nhất là một lần đi của một đấu thủ hoặc một lần đi phản ứng lại của đối thủ bên kia. Chú ý điều này khác với thói quen dùng trong thực tế một nước đi bao gồm lần đi của ta và một lần đi của đối thủ. Nói cách khác, nước đi ở đây thực chất chỉ là "nửa nước" theo cách hiểu của làng cờ.

Trạng thái bàn cờ gốc (ply=0)

Lượt “ta” đi

Trạng thái bàn cờ mới (ply=1)

Lượt đối phương đi

Trạng thái bàn cờ mới (ply=2)

**Hình 2. 1** Cây trò chơi

Tóm lại: Cây trò chơi dùng các nút để thể hiện trạng thái của trò chơi. Cây này là một dạng của cây ngữ nghĩa, có các nhánh ứng với việc chuyển cấu hình sau một nước đi. Có thể xem hai nhánh xuất phát từ một nút là hai quyết định của hai đối thủ.

Gọi *p* là số mức của cây thì độ sâu của cây là *d= p-1*. Mỗi lựa chọn hay bước chuyển là một nước đi.

* 1. **Giải thuật Minimax**

Xét một trò chơi đối kháng trong đó hai người thay phiên nhau đi nước của mình như cờ vua, cờ tướng, cờ carô,... Trò chơi có một trạng thái bắt đầu và mỗi nước đi sẽ biến đổi trạng thái hiện hành thành một trạng thái mới. Trò chơi sẽ kết thúc theo một quy định nào đó, theo đó thì cuộc chơi sẽ dẫn đến một trạng thái phản ánh có một người thắng cuộc hoặc một trạng thái mà cả hai đấu thủ không thể phát triển được nước đi của mình, ta gọi nó là trạng thái hòa cờ. Ta tìm cách phân tích xem từ một trạng thái nào đó sẽ dẫn đến đấu thủ nào sẽ thắng với điều kiện cả hai đấu thủ đều có trình độ như nhau.

Vấn đề chơi cờ có thể xem như vấn đề tìm kiếm, trong không gian trạng thái. Mỗi trạng thái là một tình thế.

Trạng thái ban đầu là sự sắp xếp các quân cờ của hai bên lúc bắt đầu cuộc chơi.

Các toán tử là các nước đi hợp lệ.

Các trạng thái kết thúc là các tình thế mà cuộc chơi dừng, thường được xác định bởi một số điều kiện dừng.

Một hàm kết cục ứng với mỗi trạng thái kết thúc với một giá trị nào đó. Ta có thể xác định hàm kết cuộc là hàm nhận giá trị 1 tại các trạng thái kết thúc là thắng, -1 tại các trạng thái kết thúc là thua và 0 tại trạng thái kết thúc là hòa.

1. **Ý tưởng**

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. Cây trò chơi vì thế còn có tên là cây MIN-MAX. Trên cây trò chơi 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 1***: Xét trò chơi carô có 9 ô (Tic tac toe). Hai người MAX và MIN thay phiên nhau đi X hoặc O (MAX đi X, MIN đi O). Người nào đi được 3 ô thẳng hàng (ngang, dọc, xiên) thì thắng cuộc. Nếu đã hết ô đi mà chưa phân thắng bại thì hai đấu thủ hòa nhau. Một phần của trò chơi này được biểu diễn bởi cây sau:

Diagram

Description automatically generated

**Hình 2. 2** Một phần cây trò chơi trong trò chơi tic-tac-toe.

Trong cây trò chơi trên, các nút lá được tô nền và viền khung đôi để dễ phân biệt với các nút khác. Ta có thể 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:

1 nếu tại đó người đi X đã thắng

-1 nếu tại đó người đi X đã thua

0 nếu hai đấu thủ đã hòa nhau

Như vậy từ một trạng thái bất kỳ, đến lượt mình, người đi X sẽ chọn cho mình một nước đi sao cho dẫn đến trạng thái có giá trị lớn nhất (trong trường hợp này là 1). Ta nói X chọn nước đi MAX, nút mà từ đó X chọn nước đi của mình được gọi là nút MAX. Người đi O đến lượt mình sẽ chọn một nước đi sao cho dẫn đến trạng thái có giá trị nhỏ nhất (trong trường hợp này là -1, khi đó X sẽ thua và do đó O sẽ thắng). Ta nói O chọn nước đi MIN, nút mà từ đó O chọn nước đi của mình được gọi là nút MIN. Áp dụng chiến lược Minimax cho một nhánh trong cây trò chơi của trò chơi Tic-tac-toe ta có giá trị (phía trên mỗi nút) của các nút được thể hiện trong hình 2.1.

*Ví dụ 2:* Xét trò chơi Nim. Để chơi Nim, một số token (vật biểu hiện như đồng xu, lá bài, mảnh gỗ…) được đặt trên bàn giữa hai đối thủ. Ở mỗi nước đi, người chơi phải chia đống token thành hai đống nhỏ có số lượng khác nhau. Người chơi nào đến lượt mà không chia được là thua cuộc. Ứng với một token vừa phải, không gian trạng thái này có thể triển khai đến cùng. Hình sau biểu diễn không gian trạng thái của trò chơi có 7 token:

Diagram

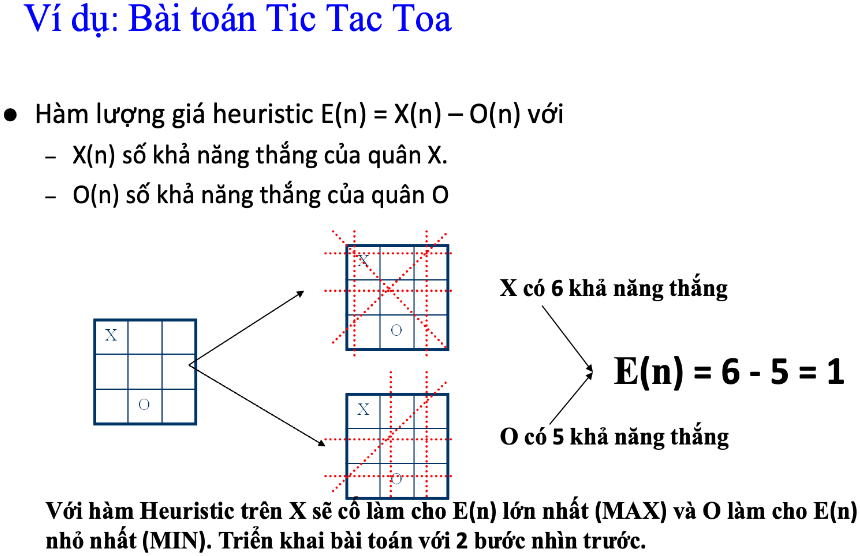
Description automatically generated

**Hình 2. 3** Không gian trạng thái của trò chơi Nim.

Khi chơi các trò chơi khó có thể triển khai hết không gian trạng thái, khó khăn chủ yếu là phải tính toán phản ứng của đối thủ. Một cách xử lý đơn giản nhất là giả sử đối thủ của chúng ta cũng sử dụng kiến thức về không gian trạng thái giống như chúng ta và áp dụng kiến thức đó kiên định để thắng cuộc. Mặc dù giả thiết này có những hạn chế của nó nhưng nó cũng cho chúng ta một cơ sở hợp lý để dự đoán hành vi của đối thủ. Giải thuật Minimax sẽ tìm kiếm không gian của trò chơi này theo giả thiết đó. Áp dụng chiến lược Minimax, chúng ta đánh dấu luân phiên từng mức trong không gian tìm kiếm phù hợp với đối thủ có nước đi ở mức đó. Trong ví dụ trên, MIN được quyền đi trước, từng nút lá được gán giá trị 1 hay 0 tùy theo kết quả đó là thắng cuộc đối với MAX hay MIN. Kết quả của việc áp dụng Minimax vào đồ thị không gian trạng thái đối với trò chơi Nim được thể hiện như hình trên. Vì tất cả các nước đi đầu tiên có thể xảy ra cho MIN sẽ dẫn đến các nút có giá trị 1 nên đối thủ MAX luôn có thể bắt trò chơi giành thắng lợi cho mình bất kể nước đi đầu tiên của MIN là như thế nào (đường đi thắng lợi của MAX được cho theo mũi tên đậm).

1. **Áp dụng 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 (ply), 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 .

Chart, line chart

Description automatically generated

**Hình 2. 4** Minimax đối với không gian trạng thái giả

Diagram

Description automatically generated

**Hình 2. 5** Minimax hai lớp được áp dụng vào nước đi mở đầu trò chơi Caro.

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 + ∞, còn với trạng thái bắt buộc thắng cuộc cho MIN thì được đánh giá là - ∞.

1. **Thủ tục Minimax**

Giả sử chúng ta có một bộ phân tích thế cờ có thể áp dụng tất cả các luật, các phương pháp đánh cờ khác nhau vào từng thế cờ và chuyển đổi chúng thành một con số đại diện (cho điểm thế cờ). Mặt khác, ta giả sử con số đó là dương khi áp dụng cho thế cờ của một đấu thủ (được gọi là người chơi cực đại – người chơi MAX), và là âm khi áp dụng cho đấu thủ bên kia (được gọi là người chơi cực tiểu – người chơi MIN). Quá trình tính toán cho điểm thế cờ được gọi là lượng giá tĩnh (static evaluation). Hàm thực hiện việc tính toán được gọi là một bộ lượng giá tĩnh và giá trị nhận được gọi là điểm lượng giá tĩnh.

Cả hai đấu thủ đều cố gắng đi như thế nào đó để đạt được điểm tuyệt đối lớn nhất. Người chơi cực đại (MAX) sẽ tìm những nước đi dẫn đến điểm của mình trở nên lớn hơn (hay cao nhất có thể được) hay điểm của đối thủ bớt âm hơn (nhỏ hơn về giá trị tuyệt đối). Còn đấu thủ của anh ta, người chơi cực tiểu (MIN), lại ra sức phản kháng lại, để dẫn tới điểm âm của anh ta âm hơn hay điểm dương của đối thủ nhỏ đi .

Từ ý tưởng ta có thể suy ra các bước của thuật toán Minimax như sau:

- Nếu như đạt đến giới hạn tìm kiếm (đến tầng dưới cùng của cây tìm kiếm tức nút lá), tính giá trị tĩnh của thế cờ hiện tại ứng với người chơi ở đó.

Ghi nhớ kết quả.  
- Nếu như mức đang xét là của người chơi cực tiểu (nút MIN), áp dụng thủ tục Minimax này cho các con của nó. Ghi nhớ kết quả nhỏ nhất.

- Nếu như mức đang xét là của người chơi cực đại (nút MAX), áp dụng thủ tục Minimax này cho các con của nó. Ghi nhớ kết quả lớn nhất.

Từ ý tưởng phân tích trên ta có thể xây dựng thủ tục Minimax như sau:

Hàm Minimax nhận vào một thế cờ pos và trả về giá trị của thế cờ đó.

Nếu thế cờ pos tương ứng với nút lá trong cây trò chơi thì trả về giá trị đã được gán cho nút lá. Ngược lại ta cho pos một giá trị tạm value là -∞ hoặc ∞ tùy thuộc pos là nút MAX hay MIN và xét các thế cờ con của pos. Sau khi một con của pos có giá trị V thì đặt lại value= max(value, V) nếu n là nút MAX và value= min(value,V) nếu n là nút MIN. Khi tất cả các con của n đã được xét thì giá trị tạm value của pos trở thành giá trị của nó. (INFINITY thể hiện cho giá trị vô cùng) .

Ta có giả mã cho giải thuật Minimax như sau:

function Minimax(pos): integer;

value, best : integer;

begin

if **pos là nút lá** then return **eval(pos)**

else

begin

*{Khởi tạo giá trị tạm cho best }*

if **pos là nút MAX** then

best:= -INFINITY

else best := INFINITY;

*{hàm genPos sinh ra mọi nước đi từ thế cờ pos}*

**genPos(pos);**

*{Xét tất cả các con của pos, mỗi lần xác định được giá trị của một nút con, ta phải đặt lại giá trị tạm value. Khi đã xét hết tất cả các con thì value là giá trị của n}*

while (**còn lấy được một nước đi m**) do

begin

pos := Tính thế cờ mới nhờ đi m;  
 value = Minimax(pos);

if **pos là nút MAX** then

if (value > best) then best := value;

if **pos là nút MIN** then

if (value < best) then best := value;

end;

Minimax := best;

end;

end;

Xem xét đoạn chương trình trên ta thấy:

- Có hai hàm mới là hàm eval(pos) và hàm genPos(pos). Hàm eval(pos) thực hiện việc tính giá trị (lượng giá) của thế cờ pos. Hàm genPos(pos) sinh ra tất cả các nước đi có thể từ thế cờ pos hiện tại. Việc xây dựng hai hàm này sẽ phụ thuộc vào từng trò chơi cụ thể.

Hàm đánh giá Eval ứng với mỗi trạng thái (thế cờ) pos của trò chơi với một giá trị số Eval(pos). Giá trị này là sự đánh giá độ lợi thế của trạng thái pos. Trạng thái pos càng thuận lợi cho MAX thì Eval(pos) là số dương càng lớn, pos càng thuật lợi cho MIN thì eval(pos) là số âm càng nhỏ, Eval(pos) 0 đối với trạng thái không lợi thế cho ai cả. Chất lượng của chương trình chơi cờ phụ thuộc rất nhiều vào hàm đánh giá. Nếu hàm đánh giá cho ta sự đánh giá không chính xác về các trạng thái, nó có thể hướng ta đi tới trạng thái được xem là tốt, nhưng thực tế lại rất lợi cho ta. Thiết kế một hàm đánh tốt là một việc khó. Đòi hỏi ta phải quan tâm đến nhiều nhân tố. Ở đây có sự mâu thuẫn giữa độ chính xác của hàm đánh giá và thời gian tính của nó. Hàm đánh giá chính xác sẽ đòi hỏi rất nhiều thời gian tính toán, mà người chơi lại .

1. **Đánh giá**

Về lí thuyết, chiến lược Minimax cho phép ta tìm được nước đi tối ưu. Song nó không thực tế, chúng ta sẽ không có đủ thời gian để tính được nước đi tối ưu. Bởi vì thuật toán Minimax đòi hỏi ta phải xem xét toàn bộ các đỉnh của cây trò chơi. Trong các trò chơi hay, cây trò chơi là cực kì lớn. Chẳng hạn, đối với cờ vua, chỉ tính đến độ sâu 40, thì cây trò chơi đã có khoảng 10120 đỉnh! Nếu cây có độ cao m, và tại mỗi đỉnh có b nước đi độ phức tạp về thời gian của thuật toán Minimax là O(bm).

Bản chất của thuật toán là tìm kiếm theo chiều sâu nên độ phức tạp không gian là O(bm).

Có thể tiết kiệm được nhiều thời gian bằng việc dùng các thuật toán thông minh hơn như thuật toán Alpha-beta, thuật toán này không thăm tất cả các nút lá mà vẫn cho kết quả đúng với thuật toán Minimax.

CHƯƠNG 3. THIẾT KẾ GIAO DIỆN

Chart, scatter chart

Description automatically generated  
Hình 3. 1 Giao diện chơi giữa Player 1 – Player 2

Chart

Description automatically generated  
Hình 3. 2 Giao diện chuyển sang chế độ Play with computer

A picture containing chart

Description automatically generated  
Hình 3. 3 Bắt sự kiện chặn 2 đầu vẫn thắng.

CHƯƠNG 4. KẾT LUẬN

* 1. **Kết quả đạt được**

Trong suốt khoảng thời gian nghiên cứu và thực hiện đề tài, nhóm đã cố gắng làm hết khả năng của mình và vì thời gian làm có hạn nên kết quả đạt được vẫn còn nhiều hạn chế nhưng nhóm đã học hỏi được khá nhiều kiến thức mới mẻ cũng như chuyên sâu về thuật toán, lập trình... Và những việc chúng em đã đạt được như sau:

* Hoàn thiện trò chơi caro với các chức năng:
* Có 2 chế độ chơi: người và người, người và máy.
* Giải quyết được các sự kiện người dùng đánh trùng vào ô đã đánh trước đó.
* Giải quyết được sự kiện: chặn 2 đầu vẫn thắng khi hoàn thiện 5 ô (đường thẳng, đường chéo, đường ngang).

TÀI LIỆU THAM KHẢO

**[1]. TS. Vương Quân Hoàng**, *Lý thuyết trò chơi, 2011.*

**[2]. Biagio Ricceri & Stephen Simons**, *Minimax Theory and Applications, 2011.*

**Source code**: https://github.com/phuocnhatfabulous/Caro.git