|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  TRẦN XUÂN TRƯỜNG NGHIÊN CỨU VÀ PHÁT TRIỂN HỆ THỐNG CỜ VUA THÔNG MINH DỰA TRÊN AI 2025  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **TRẦN XUÂN TRƯỜNG**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGHIÊN CỨU VÀ PHÁT TRIỂN HỆ THỐNG CỜ VUA THÔNG MINH DỰA TRÊN AI**  **HƯNG YÊN - 2025** |

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGHIÊN CỨU VÀ PHÁT TRIỂN HỆ THỐNG CỜ VUA THÔNG MINH DỰA TRÊN AI**  **TRẦN XUÂN TRƯỜNG**  NGÀNH: KỸ THUẬT PHẦN MỀM  CHUYÊN NGÀNH: CÔNG NGHỆ WEB  **NGƯỜI HƯỚNG DẪN**  **NGUYỄN HOÀNG ĐIỆP**  **HƯNG YÊN - 2025** |

**NHẬN XÉT**

**Nhận xét của giảng viên hướng dẫn:**

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

**LỜI CAM ĐOAN**

Em xin cam đoan đồ án tốt nghiệp “Nghiên cứu và phát triển hệ thống cờ vua thông minh dựa trên ai” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai em xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

*Hưng Yên, ngày … tháng … năm…..*

Sinh viên

…………………………………..

MỤC LỤC

[MỤC LỤC 5](#_Toc197104680)

[DANH SÁCH CÁC THUẬT NGỮ 8](#_Toc197104681)

[DANH SÁCH BẢNG BIỂU 11](#_Toc197104682)

[DANH SÁCH HÌNH VẼ 12](#_Toc197104683)

[CHƯƠNG 1: MỞ ĐẦU 13](#_Toc197104684)

[1.1 Lý do chọn đồ án 13](#_Toc197104685)

[1.2 Mục tiêu của đồ án 16](#_Toc197104686)

[1.2.1 Mục tiêu tổng quát 16](#_Toc197104687)

[1.2.2 Mục tiêu cụ thể 16](#_Toc197104688)

[1.3 Giới hạn và phạm vi của đồ án 17](#_Toc197104689)

[1.3.1 Đối tượng nghiên cứu 17](#_Toc197104690)

[1.3.2 Phạm vi nghiên cứu 18](#_Toc197104691)

[1.4 Nội dung thực hiện 19](#_Toc197104692)

[1.5 Phương pháp tiếp cận 19](#_Toc197104693)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 23](#_Toc197104694)

[2.1 Giới thiệu về hệ thống AI 23](#_Toc197104695)

[2.2 Đánh giá vị trí cờ 23](#_Toc197104696)

[2.2.1 Đánh giá tĩnh 23](#_Toc197104697)

[2.2.2 Đánh giá động 24](#_Toc197104698)

[2.2.3 Tích hợp NNUE 25](#_Toc197104699)

[2.3 Thuật toán tìm kiếm 25](#_Toc197104700)

[2.3.1 Minimax cơ bản 25](#_Toc197104701)

[2.3.2 Tối ưu hoá tìm kiếm 26](#_Toc197104702)

[2.3.3 Kiến trúc hệ thống 27](#_Toc197104703)

[2.4 Xây dựng mô hình dựa trên các thuật toán nền tàng 28](#_Toc197104704)

[2.4.1 Các khái niệm ban đầu và bối cảnh lịch sử 28](#_Toc197104705)

[2.4.2 Thuật toán Minimax nền tảng 30](#_Toc197104706)

[2.4.3 Nâng cao hiệu quả với cắt tỉa Alpha-Beta 34](#_Toc197104707)

[2.4.4 Vai trò của hàm đánh giá 37](#_Toc197104708)

[2.4.5 Mở rộng tìm kiếm với tìm kiếm tĩnh lặng 41](#_Toc197104709)

[2.4.6 Tận dụng các tính toán trước với bảng chuyển vị 44](#_Toc197104710)

[2.4.7 Lợi thế chiến lược của sách khai cuộc 46](#_Toc197104711)

[2.4.8 Tối ưu hoá hơn nữa thông qua cắt tỉa nước đi rỗng 48](#_Toc197104712)

[2.4.9 Tìm kiếm sâu dần 50](#_Toc197104713)

[2.4.10 Các kỹ thuật sắp xếp nước đi 51](#_Toc197104714)

[2.5 Mô hình sử dụng NNUE để đánh giá thế cờ và Minimax với cắt tỉa Alpha-Beta để tìm nước đi tối ưu 52](#_Toc197104715)

[2.5.1 NNUE 52](#_Toc197104716)

[2.5.2 Thuật toán Minimax với cắt tỉa Alpha-Beta 53](#_Toc197104717)

[2.5.3 Hàm evaluate\_board 55](#_Toc197104718)

[2.5.4 Hàm preprocess\_fen 56](#_Toc197104719)

[2.5.5 Hàm find\_best\_move 56](#_Toc197104720)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 58](#_Toc197104721)

[3.1 Đặc tả yêu cầu phần mềm 58](#_Toc197104722)

[3.1.1 Các yêu cầu chức năng 58](#_Toc197104723)

[3.1.2 Biểu đồ lớp thực thể 62](#_Toc197104724)

[3.1.3 Các yêu cầu phi chức năng 64](#_Toc197104725)

[3.2 Thiết kế hệ thống 65](#_Toc197104726)

[3.2.1 Thiết kế kiến trúc 65](#_Toc197104727)

[3.2.2 Thiết kê cơ sở dữ liệu 66](#_Toc197104728)

[3.2.3 Thiết kế lớp đối tượng 69](#_Toc197104729)

[3.2.4 Thiết kế giao diện 72](#_Toc197104730)

[CHƯƠNG 4: TRIỂN KHAI WEBSITE 75](#_Toc197104731)

[4.1 Xây dựng Web với Gradio 75](#_Toc197104732)

[4.2 Xây dựng các chức năng 75](#_Toc197104733)

[4.2.1 Thiết kế giao diện người dùng cho ứng dụng AI trong trò chơi 75](#_Toc197104734)

[4.2.2 Xử lý đầu vào của người dùng và hiển thị trạng thái trò chơi 76](#_Toc197104735)

[4.3 Kiểm thử và triển khai ứng dụng 78](#_Toc197104736)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 80](#_Toc197104737)

[Kết quả đạt được 80](#_Toc197104738)

[Hạn chế của đề tài 80](#_Toc197104739)

[Hướng phát triển của đề tài 81](#_Toc197104740)

[TÀI LIỆU THAM KHẢO 82](#_Toc197104741)

[PHỤ LỤC 84](#_Toc197104742)

DANH SÁCH CÁC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
| UML | Unified Modeling Language | Ngôn ngữ mô hình hóa dùng để thiết kế hệ thống phần mềm. |
| AI | Artificial Intelligence | Trí tuệ nhân tạo – mô phỏng trí tuệ con người bằng máy tính. |
| CNN | Convolutional Neural Network | Mạng nơ-ron tích chập, dùng trong xử lý ảnh và mô hình bàn cờ. |
| NNUE | Efficiently Updatable Neural Network | Mạng nơ-ron có thể cập nhật hiệu quả, dùng để đánh giá thế cờ trong cờ vua. |
| MCTS | Monte Carlo Tree Search | Thuật toán tìm kiếm theo cây Monte Carlo dựa trên mô phỏng. |
| FEN | Forsyth-Edwards Notation | Định dạng mô tả trạng thái bàn cờ cờ vua. |
| PGN | Portable Game Notation | Định dạng lưu các nước đi trong ván cờ vua. |
| IDE | Integrated Development Environment | Môi trường phát triển tích hợp dùng cho lập trình. |
| API | Application Programming Interface | Giao diện lập trình ứng dụng – giúp giao tiếp giữa các phần mềm. |
| GUI | Graphical User Interface | Giao diện đồ họa người dùng. |
| DBMS | Database Management System | Hệ quản trị cơ sở dữ liệu. |
| CRUD | Create, Read, Update, Delete | Các thao tác cơ bản với dữ liệu. |
| HTML | HyperText Markup Language | Ngôn ngữ đánh dấu để tạo cấu trúc trang web. |
| CSS | Cascading Style Sheets | Ngôn ngữ định kiểu cho giao diện web. |
| JS | JavaScript | Ngôn ngữ lập trình phía client trong phát triển web. |
| AVX | Advanced Vector Extensions | Tập lệnh xử lý song song tăng tốc mạng NNUE. |
| SIMD | Single Instruction Multiple Data | Phép toán song song giúp tăng tốc xử lý mạng nơ-ron. |
| Alpha-Beta | Alpha-Beta Pruning | Kỹ thuật tối ưu hóa thuật toán tìm kiếm Minimax. |
| Minimax | Minimax Algorithm | Thuật toán chọn nước đi tối ưu trong trò chơi hai người. |
| Heuristic | Heuristic Search | Phương pháp tìm kiếm dựa vào đánh giá ước lượng. |
| Transposition | Transposition Table | Bảng lưu trữ các thế cờ đã được đánh giá để tránh tính lại. |
| Gradio | Gradio | Thư viện Python dùng để tạo giao diện web đơn giản cho mô hình AI. |
| Python | Python Programming Language | Ngôn ngữ lập trình cấp cao, dùng trong phát triển AI. |
| SQLite | SQLite Database | Cơ sở dữ liệu nhẹ, thường dùng trong ứng dụng nhỏ. |
| JSON | JavaScript Object Notation | Định dạng dữ liệu trao đổi giữa client-server. |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền tải dữ liệu trên web. |

DANH SÁCH BẢNG BIỂU

[Bảng 1.1 Các cột mốc lịch sử quan trọng trong cờ vua máy tính 18](#_Toc197100372)

[Bảng 1.2 So sánh các thuật toán AI phổ biến cho cờ vua 19](#_Toc197100373)

[Bảng 1.3 Các chỉ số đánh giá hệ thống cờ vua thông minh 20](#_Toc197100374)

[Bảng 2.1 Giá trị các quân cờ trong AI cờ vua(ví dụ) 38](#_Toc197100375)

[Bảng 2.2 Phạm vi điểm số của hàm đánh giá 39](#_Toc197100376)

[Bảng 2.3 Các kỹ thuật sắp xếp nước đi khác nhau 49](#_Toc197100377)

[Bảng 2.4 Mã giả minh hoạ hàm alphabeta 52](#_Toc197100378)

[Bảng 3.1 Bảng Users 65](#_Toc197100379)

[Bảng 3.2 Bảng Games 65](#_Toc197100380)

[Bảng 3.3 Bảng Moves 66](#_Toc197100381)

DANH SÁCH HÌNH VẼ

[Hình 2.1 Tổng quan kiến trúc và tham số mô hình NNUE 50](#_Toc197100364)

[Hình 3.1 Biểu đồ Ca sử dụng cho Phân hệ quản trị. 57](#_Toc197100365)

[Hình 3.2 Biểu đồ Ca sử dụng cho Phân hệ Người dùng 59](#_Toc197100366)

[Hình 3.3 Biểu đồ Lớp Thực thể 61](#_Toc197100367)

[Hình 3.4 Sơ đồ Kiến trúc Hệ thống 64](#_Toc197100368)

[Hình 3.5 Ca sử dụng: Bắt đầu Ván cờ Mới 69](#_Toc197100369)

[Hình 3.6 Ca sử dụng: Thực hiện Nước đi 69](#_Toc197100370)

[Hình 3.7 Biểu đồ lớp chi tiết 70](#_Toc197100371)

# MỞ ĐẦU

1.1 Lý do chọn đồ án

Trí tuệ nhân tạo (AI) đã và đang ngày càng khẳng định vai trò quan trọng của mình trong nhiều lĩnh vực của đời sống và khoa học, mang đến những thay đổi mang tính cách mạng trong cách công nghệ tương tác và hỗ trợ khả năng của con người. Trong lĩnh vực trò chơi, đặc biệt là các trò chơi mang tính chiến lược cao như cờ vua, AI đã chứng minh được tác động sâu sắc của mình, phát triển từ những hệ thống dựa trên luật lệ đơn giản đến các bộ máy phức tạp có khả năng vượt trội so với trình độ của con người. Điều này cho thấy tầm quan trọng của AI trong việc tạo ra các hệ thống thông minh có khả năng giải quyết các thách thức phức tạp.

Sự ứng dụng của AI trong lĩnh vực trò chơi không chỉ dừng lại ở việc tạo ra những đối thủ mạnh mẽ mà còn góp phần nâng cao trải nghiệm của người chơi thông qua việc tạo ra môi trường chơi game phản hồi nhanh nhạy, có khả năng thích ứng và đầy thử thách. Các hệ thống AI trong game có thể đảm nhiệm nhiều vai trò khác nhau, từ việc điều khiển các nhân vật không phải người chơi (NPC) một cách thông minh, tạo ra các màn chơi và nhiệm vụ đa dạng một cách tự động, đến việc cân bằng độ khó của trò chơi và kiểm thử lỗi một cách hiệu quả. Sự đa dạng trong ứng dụng và giá trị mà AI mang lại cho các hệ thống tương tác là không thể phủ nhận.

Việc phát triển một hệ thống cờ vua thông minh dựa trên AI là một nghiên cứu điển hình hấp dẫn trong việc ứng dụng các nguyên tắc của AI vào một bài toán có cấu trúc rõ ràng nhưng không kém phần phức tạp. Cờ vua, với luật chơi tường minh và không gian tìm kiếm rộng lớn, cung cấp một nền tảng lý tưởng để khám phá và phát triển các kỹ thuật AI. Sự tiến hóa của AI trong cờ vua phản ánh những bước tiến lớn hơn trong lĩnh vực AI nói chung, từ đó cho thấy đây là một lĩnh vực nghiên cứu quan trọng để hiểu rõ hơn về khả năng và giới hạn của các công nghệ AI hiện tại.

Những chương trình cờ vua ban đầu dựa trên các thuật toán cơ bản và bị giới hạn bởi sức mạnh tính toán. Sự phát triển đến các bộ máy hiện đại sử dụng học máy và mạng nơ-ron đã cho thấy một bước nhảy vọt đáng kể trong năng lực của AI. Nghiên cứu sự tiến hóa này cung cấp những hiểu biết sâu sắc về quỹ đạo phát triển của AI nói chung. Bên cạnh đó, thành công của AI trong cờ vua có những tác động rộng lớn đến các lĩnh vực giải quyết vấn đề phức tạp khác, minh chứng cho tiềm năng của AI trong việc đối mặt với những thách thức mà trước đây được cho là chỉ thuộc về trí tuệ con người. Các kỹ thuật được phát triển cho AI cờ vua, như các thuật toán tìm kiếm, hàm đánh giá và phương pháp học, có thể được điều chỉnh và áp dụng vào nhiều lĩnh vực khác đòi hỏi tư duy chiến lược và ra quyết định, chẳng hạn như logistics, tài chính và robot học.

Mặc dù đã có những thành tựu đáng chú ý trong lĩnh vực cờ vua máy tính, nhưng lĩnh vực này vẫn tiếp tục phát triển với các nghiên cứu không ngừng về các thuật toán, kiến trúc và phương pháp học mới. Điều này cho thấy sự cần thiết phải tiếp tục nghiên cứu và phát triển trong lĩnh vực này. Sự tích hợp các kỹ thuật AI tiên tiến như học máy và học sâu vào các bộ máy cờ vua đại diện cho một lĩnh vực nghiên cứu tiên phong với tiềm năng mang lại những cải tiến đáng kể về hiệu suất và sự hiểu biết về trò chơi. Việc khám phá những hạn chế của các bộ máy cờ vua hiện tại , chẳng hạn như sự thiếu hụt "cảm giác" về ván cờ hoặc những sai sót đôi khi xảy ra ở độ sâu tìm kiếm thấp, nhấn mạnh nhu cầu nghiên cứu nhằm giải quyết những thiếu sót này và mở rộng ranh giới của AI trong lĩnh vực này. Thách thức liên tục trong việc tạo ra các hệ thống AI không chỉ chơi giỏi mà còn thể hiện trực giác và sự sáng tạo giống như con người trong lối chơi vẫn là một động lực quan trọng cho các nghiên cứu đang diễn ra.

Sự phát triển nhanh chóng của AI, đặc biệt trong các lĩnh vực như mạng nơ-ron, tạo ra những cơ hội mới để phát triển các hệ thống cờ vua tinh vi và thông minh hơn, làm cho đây trở thành một lĩnh vực nghiên cứu cấp thiết và phù hợp. Thành công của AlphaZero trong việc sử dụng các kỹ thuật tự học đã cho thấy một sự thay đổi mô hình trong phát triển bộ máy cờ vua. Điều này cho thấy việc tiếp tục khám phá các phương pháp AI tiên tiến như vậy là rất quan trọng để tạo ra thế hệ hệ thống cờ vua thông minh tiếp theo. Hơn nữa, sự gia tăng khả năng tiếp cận các tài nguyên điện toán mạnh mẽ và các công cụ phát triển AI tạo điều kiện thuận lợi cho sinh viên và nhà nghiên cứu đóng góp một cách ý nghĩa vào lĩnh vực này, càng làm nổi bật tính cấp thiết và phù hợp của các dự án như vậy. Các nền tảng điện toán đám mây và các thư viện AI mã nguồn mở cung cấp cơ sở hạ tầng và công cụ cần thiết để phát triển và thử nghiệm các mô hình AI phức tạp, dân chủ hóa quyền truy cập vào nghiên cứu và phát triển AI tiên tiến.

Mặc dù các bộ máy cờ vua hiện đại thể hiện sức mạnh chơi vượt trội, một số người cho rằng chúng thiếu sự hiểu biết thực sự về trò chơi theo cách mà con người cảm nhận. Chúng chủ yếu dựa vào khả năng tính toán nước đi một cách cạn kiệt và có thể không phải lúc nào cũng ưu tiên các nước đi dựa trên những cân nhắc chiến lược sâu sắc theo cách mà một đại kiện tướng cờ vua thực hiện. Các bộ máy hiện tại đôi khi có thể bỏ lỡ những hiểu biết về vị trí tế nhị hoặc các kế hoạch chiến lược dài hạn mà người chơi cờ có thể nắm bắt được. Điều này cho thấy những hạn chế trong khả năng đánh giá các vị trí phức tạp đòi hỏi nhiều hơn chỉ là tính toán chiến thuật.

Có những lo ngại về khả năng người chơi cờ vua quá phụ thuộc vào các bộ máy trong quá trình chuẩn bị, điều này có thể dẫn đến sự suy giảm về khả năng sáng tạo và trực giác. Điều này gián tiếp chỉ ra những hạn chế trong "trí thông minh" của các bộ máy hiện tại trong việc thúc đẩy sự hiểu biết và phát triển của con người. Một số nghiên cứu cho thấy rằng một số loại vị trí nhất định, đặc biệt là những vị trí đòi hỏi sự hiểu biết sâu sắc về các "pháo đài" hoặc liên quan đến các nước đi duy nhất, vẫn có thể gây khó khăn cho ngay cả những bộ máy mạnh nhất, đặc biệt là trong điều kiện giới hạn thời gian. "Cảm giác" về ván cờ, khía cạnh tâm lý khi chơi với một đối thủ là người, là điều mà các hệ thống AI hiện tại không thể sao chép hoặc thích ứng được. Điều này làm nổi bật sự khác biệt cơ bản giữa trí tuệ nhân tạo và trí tuệ con người trong bối cảnh cờ vua.

Việc các bộ máy hiện tại dựa vào việc tìm kiếm và đánh giá hàm số rộng rãi, mặc dù hiệu quả, nhưng có thể không nắm bắt đầy đủ bản chất của sự hiểu biết cờ vua của con người, cho thấy một lĩnh vực tiềm năng để cải thiện thông qua việc kết hợp các phương pháp suy luận hoặc học tập giống con người hơn. Người chơi cờ vua thường dựa vào khả năng nhận dạng mẫu, trực giác và các khái niệm chiến lược để đánh giá các vị trí và chọn nước đi, đôi khi không cần tính toán hàng triệu biến thể. Việc khám phá các cách để truyền đạt những khả năng tương tự vào các hệ thống AI có thể dẫn đến lối chơi "thông minh" hơn. Lỗi và sự không chính xác, mặc dù ngày càng ít xảy ra ở các bộ máy hàng đầu, vẫn có thể xảy ra, đặc biệt là trong các vị trí phức tạp hoặc bất thường, chứng minh rằng ngay cả những hệ thống tiên tiến nhất cũng không phải là không thể sai lầm. Điều này đòi hỏi phải tiếp tục nghiên cứu và tinh chỉnh các hệ thống này.

1.2 Mục tiêu của đồ án

1.2.1 Mục tiêu tổng quát

Mục tiêu tổng quát của đồ án tốt nghiệp này là **nghiên cứu, thiết kế và phát triển một hệ thống cờ vua thông minh dựa trên các kỹ thuật của trí tuệ nhân tạo** có khả năng chơi cờ ở một trình độ nhất định. Mục tiêu này hoàn toàn phù hợp với tên đề tài đã được xác định.

1.2.2 Mục tiêu cụ thể

Nghiên cứu các thuật toán AI phổ biến được sử dụng trong cờ vua: Mục tiêu này bao gồm việc nghiên cứu kỹ lưỡng các thuật toán AI có liên quan như thuật toán Minimax, tỉa nhánh Alpha-Beta , tìm kiếm theo cây Monte Carlo (MCTS) và có thể là các kỹ thuật học sâu. Việc nghiên cứu này nhằm mục đích hiểu rõ nguyên tắc hoạt động, ưu điểm và hạn chế của từng thuật toán trong bối cảnh trò chơi cờ vua. Việc hiểu rõ lịch sử phát triển của các thuật toán được sử dụng trong các bộ máy cờ vua, từ Minimax đến mạng nơ-ron, sẽ cung cấp một nền tảng vững chắc cho việc lựa chọn và triển khai các kỹ thuật phù hợp cho đồ án. Sự thành công ban đầu của các thuật toán như Minimax và tỉa nhánh Alpha-Beta đã đặt nền móng cho các kỹ thuật tiên tiến hơn. Việc xem xét lịch sử này sẽ giúp đưa ra quyết định về các thuật toán cần khám phá và có khả năng tích hợp vào hệ thống cờ vua thông minh.

Thiết kế kiến trúc tổng thể của hệ thống cờ vua thông minh: Mục tiêu này tập trung vào việc xác định các thành phần khác nhau của hệ thống, chẳng hạn như bản thân engine cờ vua, mô-đun AI, giao diện người dùng (nếu có) và cách các thành phần này tương tác với nhau.

Xây dựng một engine cờ vua cơ bản có khả năng thực hiện các nước đi hợp lệ: Mục tiêu này liên quan đến việc triển khai các quy tắc cơ bản của cờ vua, bao gồm cách di chuyển của các quân cờ, các hành động bắt quân, chiếu, chiếu hết và các nước đi đặc biệt như nhập thành và bắt quân qua đường.

Áp dụng một hoặc nhiều thuật toán AI đã nghiên cứu để xây dựng khả năng suy nghĩ và đưa ra quyết định cho hệ thống: Đây là mục tiêu cốt lõi của đồ án, trong đó các thuật toán AI đã chọn sẽ được tích hợp vào engine cờ vua để cho phép nó đánh giá các vị trí trên bàn cờ và lựa chọn nước đi.

Đánh giá hiệu suất của hệ thống thông qua các trận đấu thử nghiệm với các đối thủ khác nhau (người chơi hoặc các hệ thống cờ vua khác): Mục tiêu này bao gồm việc thiết kế và tiến hành các thử nghiệm để đánh giá sức mạnh chơi và hiệu quả của hệ thống cờ vua thông minh đã phát triển. Điều này có thể bao gồm việc chơi với người chơi ở các cấp độ kỹ năng khác nhau hoặc so sánh hệ thống với các engine cờ vua hiện có. Quá trình đánh giá không chỉ nên xem xét sức mạnh chơi mà còn các khía cạnh khác của "trí thông minh" của hệ thống, chẳng hạn như độ sâu phân tích, chất lượng của các lựa chọn nước đi trong các loại vị trí khác nhau và khả năng học hỏi hoặc thích ứng của nó. Việc chỉ đo lường tỷ lệ thắng/thua có thể không nắm bắt đầy đủ trí thông minh của hệ thống. Việc phân tích hiệu suất của engine trong các kịch bản cụ thể, khả năng tránh các cạm bẫy đã biết và lý do đằng sau các nước đi của nó sẽ cung cấp một đánh giá toàn diện hơn.

1.3 Giới hạn và phạm vi của đồ án

1.3.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đồ án này là hệ thống cờ vua thông minh dựa trên trí tuệ nhân tạo. Điều này bao gồm bản thân ứng dụng phần mềm, bao gồm kiến trúc, việc triển khai engine cờ vua và các thuật toán AI được tích hợp.

* Các thuật toán AI liên quan đến trò chơi cờ vua: Cụ thể, các thuật toán như Minimax, tỉa nhánh Alpha-Beta, tìm kiếm theo cây Monte Carlo và có thể là các kiến trúc học sâu phù hợp với việc chơi game.
* Các hệ thống cờ vua thông minh hiện có: Điều này bao gồm việc nghiên cứu kiến trúc, thuật toán và hiệu suất của các engine cờ vua đã được thiết lập như Stockfish, Leela Chess Zero và các engine khác. Điều này sẽ cung cấp một tiêu chuẩn để so sánh và xác định các lĩnh vực tiềm năng để cải thiện hoặc các phương pháp tiếp cận mới.

1.3.2 Phạm vi nghiên cứu

Phạm vi không gian: Việc phát triển và thử nghiệm hệ thống có khả năng sẽ được thực hiện trong một môi trường lập trình cụ thể và trên các tài nguyên điện toán có sẵn. Ngôn ngữ sử dụng của dự án, python, pytorch, jupyter notebook, máy tính cá nhân của sinh viên thực hiện với core i5, sử dụng cpu, cấu hình trung.

Phạm vi thời gian: Đồ án sẽ được thực hiện trong khoảng thời gian 2-3 tháng được phân bổ cho dự án tốt nghiệp.

Ý nghĩa khoa học: Nghiên cứu này đóng góp vào lĩnh vực Trí tuệ nhân tạo bằng cách khám phá và ứng dụng các kỹ thuật AI vào một lĩnh vực bài toán phức tạp. Nó có khả năng mang lại những hiểu biết mới về thiết kế và triển khai các hệ thống chơi game thông minh. Dự án cũng có thể góp phần hiểu rõ hơn về điểm mạnh và điểm yếu của các thuật toán AI khác nhau trong bối cảnh các trò chơi chiến lược.

Ý nghĩa thực tiễn: Hệ thống cờ vua thông minh được phát triển có thể đóng vai trò là một công cụ hữu ích cho người chơi cờ vua để phân tích các ván cờ, cải thiện kỹ năng và khám phá các chiến lược khác nhau. Nó cũng có thể được sử dụng như một công cụ giáo dục để dạy các nguyên tắc cơ bản của cờ vua và các nguyên tắc của AI. Hơn nữa, các kỹ thuật và phương pháp luận được sử dụng trong dự án này có khả năng được áp dụng để phát triển các hệ thống thông minh cho các lĩnh vực phức tạp khác. Việc phát triển một hệ thống cờ vua thông minh hoạt động tốt cung cấp một minh chứng hữu hình về việc ứng dụng các nguyên tắc AI và có thể là một trải nghiệm học tập giá trị cho sinh viên. Quá trình thiết kế, triển khai và thử nghiệm một hệ thống như vậy đòi hỏi sự hiểu biết sâu sắc về cả cờ vua và AI, mang lại kinh nghiệm thực tế trong việc áp dụng kiến thức lý thuyết. Nghiên cứu về những hạn chế của các hệ thống hiện có và việc khám phá các phương pháp tiếp cận thay thế có khả năng dẫn đến các kỹ thuật hoặc cải tiến sáng tạo trong lĩnh vực cờ vua máy tính và AI nói chung. Bằng cách xác định những thiếu sót của các engine hiện tại , dự án này có thể khám phá các phương pháp mới hoặc sự kết hợp của các kỹ thuật hiện có để giải quyết những hạn chế này, có khả năng đóng góp kiến thức mới cho lĩnh vực này.

1.4 Nội dung thực hiện

Nghiên cứu tài liệu chuyên sâu về lịch sử phát triển của AI trong trò chơi nói chung và trong cờ vua nói riêng.

Nghiên cứu kỹ lưỡng các thuật toán AI đã chọn (ví dụ: Minimax, Alpha-Beta, MCTS, Học sâu) và ứng dụng của chúng trong các engine cờ vua.

Thiết kế kiến trúc tổng thể của hệ thống, bao gồm việc phân chia thành các mô-đun của engine cờ vua và thành phần AI.

Triển khai các chức năng cốt lõi của engine cờ vua (tạo nước đi, kiểm tra tính hợp lệ của nước đi, biểu diễn trạng thái ván cờ).

Tích hợp các thuật toán AI đã chọn vào engine cờ vua để cho phép đánh giá vị trí và lựa chọn nước đi.

Phát triển giao diện người dùng cơ bản (tùy thuộc vào phạm vi và thời gian của dự án).

Thiết kế và thực hiện các trường hợp thử nghiệm để đánh giá hiệu suất của hệ thống cờ vua thông minh so với nhiều đối thủ và tiêu chuẩn khác nhau.

Phân tích kết quả thử nghiệm và xác định các lĩnh vực cần cải thiện.

Viết tài liệu mô tả toàn bộ quá trình nghiên cứu và phát triển, bao gồm thiết kế, triển khai và đánh giá hệ thống.

1.5 Phương pháp tiếp cận

Nghiên cứu tài liệu: Tiến hành nghiên cứu toàn diện về các bài báo khoa học, tạp chí và tài liệu liên quan đến AI trong trò chơi, cờ vua máy tính và các thuật toán liên quan.

Phân tích và thiết kế hệ thống: Sử dụng các phương pháp phân tích yêu cầu của một hệ thống cờ vua thông minh và thiết kế kiến trúc và các thành phần của nó.

Lập trình hướng đối tượng: Áp dụng các nguyên tắc lập trình hướng đối tượng để thiết kế và triển khai engine cờ vua và các mô-đun AI(python có hỗ trợ tốt về việc triển khai này).

Phương pháp thử nghiệm và đánh giá: Xác định các chỉ số và quy trình để kiểm tra chức năng và hiệu suất của hệ thống đã phát triển, bao gồm kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hiệu suất thông qua các ván cờ mô phỏng và các trận đấu với người chơi hoặc các engine khác.

Phương pháp tiếp cận lặp: Có khả năng áp dụng một phương pháp phát triển lặp, trong đó một phiên bản cơ bản của hệ thống được xây dựng trước và sau đó được cải tiến dần với nhiều tính năng và khả năng AI tinh vi hơn.

Bảng 1.1 Các cột mốc lịch sử quan trọng trong cờ vua máy tính

|  |  |  |
| --- | --- | --- |
| Năm | Sự kiện chính | Ý nghĩa |
| 1950 | Alan Turing viết chương trình cờ vua đầu tiên (thực hiện thủ công) | Chứng minh rằng máy tính có thể chơi cờ. |
| 1958 | Một chương trình cờ vua đánh bại một người chơi (người mới học chơi) | Cho thấy kiến thức có thể được đưa vào chương trình cờ vua. |
| 1978 | Chess 4.7 bị đánh bại bởi David Levy | Thể hiện sự khác biệt giữa sức mạnh của chương trình và kỳ thủ hàng đầu. |
| 1988 | Deep Thought trở thành chương trình đầu tiên đánh bại một đại kiện tướng trong một giải đấu | Một bước tiến lớn trong sức mạnh chơi cờ của máy tính. |
| 1997 | Deep Blue đánh bại nhà vô địch thế giới Garry Kasparov | Lần đầu tiên một chương trình cờ vua đánh bại nhà vô địch thế giới đương kim trong một trận đấu. |
| 2017 | AlphaZero của DeepMind đánh bại Stockfish | Một cách tiếp cận mới dựa trên học sâu đã đạt được sức mạnh vượt trội. |

Bảng 1.2 So sánh các thuật toán AI phổ biến cho cờ vua

|  |  |  |
| --- | --- | --- |
| Thuật toán | Nguyên tắc cốt lõi | Ưu điểm |
| Minimax | Tìm cách tối đa hóa điểm số của người chơi hiện tại và tối thiểu hóa điểm số của đối thủ. | Đơn giản, dễ hiểu, đảm bảo tìm được nước đi tối ưu trong không gian tìm kiếm hữu hạn. |
| Tỉa nhánh Alpha-Beta | Một tối ưu hóa của thuật toán Minimax, loại bỏ các nhánh không cần thiết trong cây tìm kiếm. | Giảm đáng kể số lượng nút cần đánh giá so với Minimax, cho phép tìm kiếm sâu hơn. |
| Tìm kiếm theo cây Monte Carlo (MCTS) | Xây dựng cây tìm kiếm bằng cách mô phỏng ngẫu nhiên các ván cờ. | Phù hợp với các trò chơi có không gian tìm kiếm lớn hoặc hàm đánh giá khó xác định, không yêu cầu hàm đánh giá tĩnh rõ ràng. |
| Học sâu (Mạng nơ-ron) | Sử dụng mạng nơ-ron để học cách đánh giá các vị trí và chọn nước đi từ dữ liệu huấn luyện. | Có khả năng học các mẫu phức tạp và đưa ra các đánh giá chính xác, đạt được sức mạnh chơi rất cao. |

Bảng 1.3 Các chỉ số đánh giá hệ thống cờ vua thông minh

|  |  |  |
| --- | --- | --- |
| Chỉ số | Mô tả | Cách đo lường |
| Elo rating (ước tính) | Ước tính trình độ chơi cờ của hệ thống dựa trên kết quả các trận đấu. | Sử dụng các thuật toán xếp hạng như Elo sau các trận đấu với người chơi hoặc các engine khác. |
| Tỷ lệ Thắng/Thua/Hòa | Tỷ lệ phần trăm các trận thắng, thua và hòa của hệ thống trong các thử nghiệm. | Ghi lại kết quả của từng trận đấu. |
| Độ sâu tìm kiếm trung bình | Độ sâu trung bình mà hệ thống khám phá trong cây tìm kiếm trước khi đưa ra quyết định. | Theo dõi độ sâu tìm kiếm trong quá trình chơi. |
| Thời gian trung bình cho mỗi nước đi | Thời gian trung bình mà hệ thống cần để tính toán và đưa ra một nước đi. | Đo thời gian tính toán cho từng nước đi và tính trung bình. |
| Chỉ số | Mô tả | Cách đo lường |

# CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về hệ thống AI

Hệ thống trí tuệ nhân tạo (AI) được thiết kế để chơi cờ vua, kết hợp các kỹ thuật AI cổ điển và các phương pháp học máy hiện đại nhằm đạt được hiệu suất cao trong việc đánh giá vị trí và đưa ra quyết định nước đi. Hệ thống bao gồm ba thành phần chính: đánh giá vị trí cờ, thuật toán tìm kiếm nước đi, và kiến trúc hệ thống với các cấp độ từ cơ bản đến chuyên gia. Mục tiêu là tạo ra một hệ thống có khả năng chơi cờ vua ở mức độ cạnh tranh, từ trình độ nghiệp dư đến chuyên nghiệp.

Hệ thống này tận dụng các phương pháp truyền thống như thuật toán Minimax và đánh giá tĩnh, đồng thời tích hợp các kỹ thuật tiên tiến như mạng nơ-ron Efficiently Updatable Neural Network (NNUE) để nâng cao độ chính xác và hiệu quả. Sự kết hợp này phản ánh sự tiến hóa của AI trong các trò chơi chiến lược, từ các hệ thống dựa trên quy tắc đến các mô hình học máy phức tạp.

2.2 Đánh giá vị trí cờ

Đánh giá vị trí cờ là một yếu tố cốt lõi của hệ thống AI, cung cấp điểm số để xác định lợi thế tương đối của một vị trí trên bàn cờ. Quá trình đánh giá được chia thành các phương pháp tĩnh, động, và tích hợp mạng nơ-ron.

2.2.1 Đánh giá tĩnh

Giá trị vật liệu: Tính toán giá trị của các quân cờ (tốt, mã, tượng, xe, hậu) trên bàn cờ để xác định lợi thế vật chất. Ví dụ, một quân hậu thường được gán giá trị cao hơn nhiều so với một quân tốt.

Bảng điểm quân cờ (Piece Square Tables - PSQTs): Gán điểm số cho vị trí của từng quân cờ trên bàn cờ, thay đổi tùy theo giai đoạn đầu trận hoặc cuối trận.

Phân tích sâu: PSQTs được xây dựng dựa trên kinh nghiệm của các kỳ thủ hàng đầu hoặc phân tích thống kê từ các ván cờ thắng. Ví dụ, mã thường mạnh hơn ở trung tâm bàn cờ (ô d4, d5, e4, e5) vì có nhiều khả năng di chuyển, trong khi tượng thích các đường chéo dài để tối đa hóa tầm ảnh hưởng. Các bảng này giúp hệ thống ưu tiên các vị trí chiến lược cho từng loại quân.

Cấu trúc quân tốt: Phân tích các đặc điểm như tốt đôi, tốt cô lập, tốt lùi, tốt thông, chuỗi tốt, và lá chắn tốt.

Phân tích sâu: Tốt đôi và tốt cô lập thường là điểm yếu vì làm giảm sức mạnh phòng thủ và khả năng kiểm soát không gian. Ngược lại, tốt thông có thể trở thành lợi thế lớn trong cuối trận, đặc biệt nếu không bị chặn. Chuỗi tốt và lá chắn tốt giúp bảo vệ vua và duy trì cấu trúc phòng thủ vững chắc.

An toàn của vua: Đánh giá độ an toàn của vua dựa trên lá chắn tốt, các cột mở gần vua, và các mẫu tấn công.

Phân tích sâu: Vua thường an toàn hơn khi được che chắn bởi các quân tốt và không nằm trên các cột mở, nơi có thể bị tấn công bởi xe hoặc hậu. Hệ thống cũng xem xét các mẫu tấn công, như sự hiện diện của các quân tấn công gần vua hoặc các nước chiếu tiềm năng.

Độ di động: Xem xét số lượng nước đi hợp lệ, khả năng kiểm soát trung tâm, sự phối hợp giữa các quân cờ, và mức độ phát triển của các quân.

Phân tích sâu: Độ di động cao cho phép người chơi linh hoạt hơn, tạo ra nhiều mối đe dọa. Tuy nhiên, trong một số trường hợp, việc giảm độ di động của đối phương (ví dụ, bằng cách khóa các quân chủ lực) có thể là chiến lược hiệu quả.

2.2.2 Đánh giá động

Đánh giá dần dần (Tapered Evaluation): Sử dụng nội suy để kết hợp điểm số giữa giai đoạn giữa trận và cuối trận, đảm bảo đánh giá phù hợp với trạng thái hiện tại của ván cờ. Ví dụ, giá trị của quân mã có thể giảm trong cuối trận so với giữa trận, trong khi tốt thông trở nên quan trọng hơn.

Nhận diện mô hình: Xác định các mô hình chiến thuật (như đinh ba, chiếu hết), chiến lược (như kiểm soát trung tâm), và các kịch bản cuối trận cụ thể (như vua và tốt đấu vua) để nâng cao độ sâu chiến lược.

2.2.3 Tích hợp NNUE

Mạng nơ-ron Efficiently Updatable Neural Network (NNUE) được tích hợp để cung cấp đánh giá vị trí chính xác và hiệu quả hơn. NNUE sử dụng:

* Bộ biến đổi đặc trưng (Feature Transformer): Xử lý các đặc trưng như vị trí quân cờ, mẫu tấn công-phòng thủ, và cấu hình vật liệu, chuyển đổi thành không gian đặc trưng 256 chiều.
* Các lớp ẩn: Sử dụng hàm kích hoạt clipped ReLU và nhân ma trận thưa để xử lý sâu hơn các đặc trưng.
* Lớp đầu ra: Cung cấp một điểm số đánh giá duy nhất cho vị trí.

NNUE được tối ưu hóa với các kỹ thuật như SIMD (Single Instruction, Multiple Data) và quản lý bộ nhớ hiệu quả, đảm bảo tốc độ xử lý nhanh trong các ứng dụng thời gian thực.

Phân tích sâu: NNUE vượt trội so với các phương pháp đánh giá truyền thống vì nó có thể học các mẫu phức tạp từ dữ liệu, như sự tương tác giữa các quân cờ mà các quy tắc thủ công khó xác định. Ví dụ, NNUE có thể nhận ra rằng một quân mã ở vị trí cụ thể có giá trị cao hơn khi kết hợp với một cấu trúc tốt nhất định, điều mà PSQTs đơn giản không thể mô tả đầy đủ.

2.3 Thuật toán tìm kiếm

Thuật toán tìm kiếm là nền tảng để hệ thống AI xác định nước đi tốt nhất. Thuật toán chính được sử dụng là Minimax, được cải tiến với nhiều tối ưu hóa để tăng hiệu quả.

2.3.1 Minimax cơ bản

Thuật toán Minimax hoạt động bằng cách:

* Đánh giá tất cả các nước đi có thể trong một cây trò chơi, giả định cả hai người chơi đều chơi tối ưu.
* Người chơi tối đa hóa chọn nước đi có điểm số cao nhất, trong khi người chơi tối thiểu hóa chọn nước đi có điểm số thấp nhất.
* Khi đạt đến độ sâu tìm kiếm hoặc trạng thái kết thúc trò chơi, một hàm đánh giá được sử dụng để chấm điểm vị trí.

Độ phức tạp thời gian của Minimax cơ bản là O(b^d), trong đó b là số nhánh trung bình (số nước đi hợp lệ mỗi vị trí, khoảng 35 trong cờ vua) và d là độ sâu tìm kiếm.

Phân tích sâu: Minimax là một thuật toán lý tưởng trong lý thuyết trò chơi vì nó đảm bảo tìm ra nước đi tối ưu nếu có thể khám phá toàn bộ cây trò chơi. Tuy nhiên, trong cờ vua, số lượng vị trí có thể là rất lớn (khoảng 10^43), khiến việc tìm kiếm toàn bộ là không khả thi. Do đó, các tối ưu hóa là cần thiết để làm cho thuật toán thực tế.

2.3.2 Tối ưu hoá tìm kiếm

Cắt tỉa Alpha-Beta: Loại bỏ các nhánh không ảnh hưởng đến quyết định cuối cùng, giảm đáng kể số lượng nút cần đánh giá.

Phân tích sâu: Alpha-beta pruning hoạt động bằng cách duy trì hai giá trị, alpha (điểm số tốt nhất cho người tối đa hóa) và beta (điểm số tốt nhất cho người tối thiểu hóa). Nếu một nhánh cho thấy kết quả không thể tốt hơn giá trị hiện tại của alpha hoặc beta, nhánh đó sẽ bị cắt bỏ.

Bảng chuyển vị (Transposition Tables): Lưu trữ kết quả đánh giá của các vị trí đã tính toán để tránh tính toán lại.

Phân tích sâu: Vì cùng một vị trí có thể được đạt đến qua các chuỗi nước đi khác nhau (transpositions), bảng chuyển vị giúp tiết kiệm thời gian bằng cách tái sử dụng các đánh giá trước đó.

Tìm kiếm yên tĩnh (Quiescence Search): Mở rộng tìm kiếm cho các vị trí không ổn định (ví dụ, liên quan đến nước ăn) để đảm bảo đánh giá chính xác.

Tăng dần chiều sâu (Iterative Deepening): Tăng dần độ sâu tìm kiếm, cải thiện thứ tự nước đi và cho phép dừng sớm nếu cần.

Sắp xếp nước đi (Move Ordering): Ưu tiên các nước đi quan trọng (như nước ăn, nước kiểm, hoặc nước có lịch sử tốt) để tăng hiệu quả cắt tỉa.

Cắt tỉa nước đi muộn (Late Move Reduction): Giảm độ sâu tìm kiếm cho các nước đi ít tiềm năng.

Tìm kiếm biến thể chính (Principal Variation Search): Tập trung vào các đường đi hứa hẹn nhất để tối ưu hóa tìm kiếm.

2.3.3 Kiến trúc hệ thống

Hệ thống AI được thiết kế với các cấp độ kiến trúc khác nhau, từ cơ bản đến chuyên gia, mỗi cấp độ đại diện cho một bước tiến trong độ phức tạp và hiệu suất.

1. KnightVision (Cơ bản)

Mô tả: Sử dụng các phương pháp lập trình cờ vua cổ điển với tìm kiếm alpha-beta và đánh giá đơn giản dựa trên giá trị vật liệu và bảng điểm quân cờ.

Đặc điểm: Độ sâu tìm kiếm tối đa 4, xử lý đơn luồng, không yêu cầu học máy.

Mục đích: Phù hợp cho người mới học các khái niệm cơ bản về lập trình AI cờ vua.

1. CNN (Trung cấp)

Mô tả: Sử dụng mạng nơ-ron tích chập (Convolutional Neural Network - CNN) để nhận diện mô hình trên bàn cờ, xử lý bàn cờ 8x8 với 12 kênh đầu vào (đại diện cho các loại quân cờ).

Đặc điểm: Yêu cầu đào tạo trên hàng triệu vị trí, hỗ trợ tăng tốc GPU, tập trung vào nhận diện mô hình hơn là tìm kiếm sâu.

Mục đích: Giới thiệu các khái niệm học máy và xử lý dữ liệu lớn trong AI cờ vua.

1. NNUE (Nâng cao)

Mô tả: Kết hợp các phương pháp cổ điển với mạng nơ-ron NNUE, tập trung vào đánh giá hiệu quả và cập nhật tăng dần.

Đặc điểm: Sử dụng tối ưu hóa SIMD/AVX, kỹ thuật đặc trưng phức tạp, và kiến trúc mạng với 768 chiều đặc trưng đầu vào, 256 chiều ẩn, và 1 chiều đầu ra.

Mục đích: Cân bằng giữa tốc độ và độ chính xác, phù hợp cho các ứng dụng thời gian thực.

1. Stockfish (Chuyên gia)

Mô tả: Đại diện cho đỉnh cao của các động cơ cờ vua, tích hợp nhiều mạng đánh giá, thuật toán tìm kiếm nâng cao, và quản lý thời gian ở cấp độ giải đấu.

Đặc điểm: Hỗ trợ tìm kiếm đa luồng (lên đến 512 luồng), bảng băm lớn, và các chiến lược cắt tỉa phức tạp.

Mục đích: Đạt được sức mạnh chơi cờ ở mức độ thế giới, với Elo trên 3000.

2.4 Xây dựng mô hình dựa trên các thuật toán nền tàng

2.4.1 Các khái niệm ban đầu và bối cảnh lịch sử

Công trình nền tảng của Alan Turing vào năm 1946 đánh dấu sự khởi đầu của AI cờ vua. Turing, được công nhận là cha đẻ của trí tuệ nhân tạo, đã nhận thức sâu sắc tiềm năng của máy tính trong việc giải quyết các vấn đề phức tạp, và cờ vua không phải là ngoại lệ. Chương trình AI cờ vua đầu tiên, mặc dù chỉ là một khái niệm lý thuyết được phát triển trên giấy, đã được Turing xây dựng vào năm 1951. Sự quan tâm ban đầu của Turing đối với cờ vua như một vấn đề phức tạp có thể giải quyết bằng máy tính đã đặt nền móng cho nghiên cứu sâu hơn trong lĩnh vực này.

Cách tiếp cận ban đầu của Turing đối với AI cờ vua bao gồm một số thành phần cốt lõi. Đầu tiên, ông đã gán một giá trị số cho mỗi quân cờ (ví dụ: Tốt = 1, Tượng = 3,5, Xe = 5). Điều này cho phép chương trình định lượng lợi thế về quân số trong bất kỳ thế cờ nào. Thứ hai, chương trình sẽ xem xét mọi nước đi và phản nước đi có thể của đối thủ. Việc tìm kiếm toàn diện này nhằm mục đích dự đoán hậu quả của từng hành động. Thứ ba, mỗi thế cờ được tạo ra sẽ được đánh giá dựa trên tổng giá trị của các quân cờ mà mỗi người chơi nắm giữ. Nước đi nào mang lại giá trị lớn nhất cho người chơi sẽ được chọn làm nước đi tốt nhất. Turing cũng nhận ra rằng trong giai đoạn khai cuộc của một ván cờ, nhiều thế cờ có thể có giá trị quân số như nhau. Để giải quyết vấn đề này, ông đã kết hợp các tiêu chí vị trí vào hệ thống đánh giá. Các tiêu chí này có khả năng bao gồm các yếu tố ngoài số lượng quân cờ, chẳng hạn như sự an toàn của vua, quyền kiểm soát trung tâm và sự hoạt động của quân cờ, mặc dù các chi tiết cụ thể không được nêu rõ trong tài liệu tham khảo. Bằng cách gán giá trị cho các quân cờ, Turing đã tạo ra một thước đo định lượng để đánh giá các vị trí trên bàn cờ. Ý tưởng khám phá tất cả các nước đi có thể, ngay cả với những hạn chế, đã phản ánh một cách tiếp cận giải quyết vấn đề bằng vũ lực, một chiến lược ban đầu phổ biến trong phát triển AI. Việc ông nhận ra sự cần thiết của các tiêu chí vị trí ngoài giá trị vật chất đã thể hiện sự hiểu biết ban đầu về các sắc thái chiến lược của cờ vua.

Tiến sĩ Dietrich Prinz đã triển khai chương trình của Turing trên máy Ferranti Mark 1 vào năm 1951. Bất chấp những hạn chế về bộ nhớ của máy tính, chương trình có thể giải các bài toán "chiếu hết trong hai nước" trong vòng 15-20 phút bằng cách sử dụng phương pháp tìm kiếm vét cạn hàng nghìn nước đi có thể cho đến khi tìm thấy chuỗi chiếu hết. Thành công ban đầu này, mặc dù với những hạn chế về tính toán, đã chứng minh tính khả thi của việc sử dụng máy tính để chơi cờ vua, khơi dậy sự quan tâm và nghiên cứu sâu hơn trong lĩnh vực này. Khả năng giải quyết ngay cả những vấn đề cờ vua đơn giản như "chiếu hết trong hai nước" đã cho thấy tiềm năng của máy tính trong một lĩnh vực trước đây được coi là dành riêng cho trí tuệ con người. Thành tích này đã cung cấp sự xác nhận ban đầu cho việc áp dụng các phương pháp tính toán vào các trò chơi chiến lược. Thời gian giải (15-20 phút) cũng cho thấy những thách thức về tính toán vốn có trong cờ vua.

Thuật toán Minimax nổi lên như một lý thuyết nền tảng ban đầu quan trọng trong AI cờ vua. Thuật toán này là một khái niệm cốt lõi trong việc ra quyết định, nhằm mục đích khám phá quyết định nào sẽ tạo ra tổn thất nhỏ nhất có thể cùng với lợi ích tối thiểu lớn nhất. Nó đạt được điều này bằng cách đánh giá tất cả các chuỗi nước đi có thể, tạo ra cái mà các nhà khoa học máy tính gọi là "cây tìm kiếm". Thuật toán Minimax cung cấp một khuôn khổ có cấu trúc và logic để phát triển các chương trình chơi cờ vua, đặt nền tảng cho các thuật toán tìm kiếm phức tạp hơn. Bằng cách chính thức hóa khái niệm luân phiên giữa việc tối đa hóa kết quả của chính mình và tối thiểu hóa kết quả của đối thủ, thuật toán Minimax đã cung cấp một chiến lược tính toán rõ ràng để đưa ra các nước đi tối ưu trong một trò chơi có tổng bằng không gồm hai người chơi như cờ vua.

2.4.2 Thuật toán Minimax nền tảng

Thuật toán Minimax đóng vai trò là một thuật toán ra quyết định cơ bản được sử dụng trong trí tuệ nhân tạo, đặc biệt cho các trò chơi hai người chơi theo lượt như cờ ca-rô, cờ vua và cờ đam. Nó cho phép máy tính lên kế hoạch trước nhiều nước đi bằng cách xem xét tất cả các nước đi có thể của cả máy tính và đối thủ. Thuật toán này hoạt động dựa trên nguyên tắc là máy tính cố gắng tối đa hóa điểm số của chính mình trong khi giả định rằng đối thủ sẽ luôn cố gắng tối thiểu hóa điểm số của máy tính. Điều này dẫn đến tên gọi "Minimax", trong đó "Mini" đại diện cho mục tiêu của đối thủ là tối thiểu hóa kết quả của máy tính và "Max" đại diện cho mục tiêu của máy tính là tối đa hóa kết quả của chính mình.

Thuật toán Minimax hoạt động bằng cách khám phá tất cả các trạng thái trò chơi có thể trong tương lai từ trạng thái hiện tại. Nó xây dựng một cây trò chơi, trong đó mỗi nút đại diện cho một trạng thái trò chơi và mỗi nhánh đại diện cho một nước đi có thể. Cây trò chơi cung cấp một mô hình trực quan và khái niệm để hiểu vô số khả năng trong một trò chơi như cờ vua. Mỗi đường đi từ gốc (vị trí hiện tại) đến một lá (kết thúc trò chơi hoặc giới hạn độ sâu tìm kiếm) đại diện cho một chuỗi các nước đi tiềm năng. Hệ số phân nhánh của cây (số lượng nước đi hợp lệ ở mỗi bước) và độ sâu của cây (số lượng nước đi được xem xét trước) xác định độ phức tạp của việc tìm kiếm. Thuật toán sau đó đánh giá các nút cuối của cây này (đại diện cho các kịch bản kết thúc trò chơi như thắng, thua hoặc hòa) bằng cách sử dụng hệ thống tính điểm.

Thuật toán Minimax mô phỏng lượt đi của cả hai người chơi. Khi đến lượt máy tính (chế độ tối đa hóa), thuật toán sẽ khám phá tất cả các nước đi có thể và nhằm mục đích chọn nước đi dẫn đến điểm số cao nhất có thể. Nó giả định rằng đối thủ sẽ chơi tối ưu trong các lượt tiếp theo. Ngược lại, khi nó mô phỏng lượt đi của người chơi (chế độ tối thiểu hóa), thuật toán giả định người chơi sẽ thực hiện nước đi dẫn đến điểm số thấp nhất có thể cho máy tính (tức là kết quả tốt nhất cho người chơi). Quan điểm đối kháng này là trung tâm của thuật toán Minimax. Nó mô phỏng sự tương tác chiến lược giữa hai đối thủ thông minh, nơi mục tiêu của mỗi người chơi hoàn toàn trái ngược với mục tiêu của người kia. Giả định về lối chơi tối ưu cho phép AI đưa ra các quyết định mạnh mẽ trước các phản ứng tốt nhất có thể của đối thủ. Thuật toán đệ quy chuyển đổi giữa hai chế độ này cho đến khi đạt đến trạng thái cuối cùng của trò chơi. Tính chất đệ quy của Minimax cho phép nó khám phá cây trò chơi đến độ sâu mong muốn. Thuật toán bắt đầu bằng cách đánh giá các trạng thái cuối của trò chơi và sau đó truyền các giá trị này trở lại cây, từng cấp một, để xác định nước đi tối ưu ở trạng thái hiện tại. Đệ quy là một cách tự nhiên để triển khai thuật toán Minimax vì vấn đề tìm nước đi tốt nhất từ một trạng thái nhất định có thể được chia thành các bài toán con là tìm nước đi tốt nhất từ các trạng thái tiếp theo. Các trường hợp cơ bản cho đệ quy là các trạng thái cuối của trò chơi, nơi kết quả đã được biết.

Máy tính sử dụng hệ thống tính điểm để đánh giá kết quả của trò chơi. Một hệ thống tính điểm phổ biến trong các trò chơi như cờ ca-rô là: +10 điểm nếu máy tính thắng, -10 điểm nếu người chơi thắng, và 0 điểm nếu hòa. Việc gán điểm số cho các trạng thái cuối cung cấp cơ sở để thuật toán Minimax xác định giá trị tương đối của các chuỗi nước đi khác nhau. Các điểm số này đại diện cho kết quả cuối cùng của trò chơi từ quan điểm của người chơi tối đa hóa. Hệ thống tính điểm cho các trạng thái cuối phải phản ánh chính xác mức độ mong muốn của từng kết quả đối với người chơi tối đa hóa. Chiến thắng cho người chơi tối đa hóa phải có điểm số cao nhất, thua có điểm số thấp nhất và hòa có điểm số trung lập. Các điểm số này sau đó được sử dụng để hướng dẫn thuật toán trong việc chọn nước đi dẫn đến trạng thái cuối cùng thuận lợi nhất.

Thuật toán đệ quy chuyển đổi giữa các chế độ này cho đến khi đạt đến trạng thái cuối cùng của trò chơi. Sau khi tạo cây, máy tính cần "đánh giá các vị trí trên bàn cờ". Để quyết định phải làm gì, máy tính sẽ xem xét cây này và làm việc từ dưới lên. Các phép tính của nó được thiết lập sao cho nó tìm thấy các vị trí trên bàn cờ tốt nhất từ mỗi vị trí có thể mà quân đen sẽ đi (nó lấy giá trị tối đa). Một cấp độ cao hơn, nó giả định rằng quân đen sẽ chọn vị trí tồi tệ nhất có thể cho quân trắng (nó lấy giá trị tối thiểu). Cuối cùng, nó lấy giá trị tối đa của ba số trên cùng. Đó là nước đi mà máy tính sẽ thực hiện. Khi quân đen thực hiện nước đi, máy tính sẽ thực hiện toàn bộ quá trình này một lần nữa, tạo một cây mới và đánh giá tất cả các vị trí trên bàn cờ để tìm ra nước đi tiếp theo. Cách tiếp cận này được gọi là thuật toán minimax vì nó luân phiên giữa các giá trị tối đa và tối thiểu khi nó di chuyển lên cây.

Mục tiêu cuối cùng của thuật toán Minimax trong bối cảnh một trò chơi như cờ vua là xác định chuỗi các nước đi dẫn đến vị trí chiến thắng cho AI, giả sử đối thủ cũng chơi tối ưu. Bằng cách khám phá cây trò chơi và đánh giá các kết quả tiềm năng của các chuỗi nước đi khác nhau, thuật toán Minimax nhằm mục đích xác định đường đi tối đa hóa cơ hội đạt đến trạng thái cuối cùng đại diện cho chiến thắng của AI. Điều này không chỉ liên quan đến việc xem xét các hậu quả trực tiếp của một nước đi mà còn cả các tác động chiến lược lâu dài.

Trong các trò chơi phức tạp như cờ vua, việc khám phá toàn bộ cây trò chơi đến cuối cùng là điều bất khả thi về mặt tính toán. Do đó, đối với các triển khai thực tế của thuật toán Minimax, một hàm đánh giá là rất quan trọng để ước tính giá trị của các trạng thái trò chơi không phải là cuối cùng ở một độ sâu tìm kiếm nhất định. Hàm đánh giá này hoạt động như một heuristic, cung cấp ước tính về khả năng chiến thắng từ một vị trí trên bàn cờ nhất định. Điều này cho phép thuật toán Minimax đưa ra quyết định ngay cả khi nó không thể nhìn thấy toàn bộ trò chơi đến cuối cùng. Độ chính xác và hiệu quả của hàm đánh giá là những yếu tố quan trọng trong hiệu suất của AI cờ vua.

Độ phức tạp thời gian của thuật toán Minimax là O(b^m), trong đó b là hệ số phân nhánh (số lượng nước đi hợp lệ ở mỗi trạng thái) và m là độ sâu tối đa của cây trò chơi. Độ phức tạp không gian là O(bm). Độ phức tạp thời gian theo cấp số nhân này của thuật toán Minimax làm nổi bật những hạn chế về mặt tính toán của nó đối với các trò chơi có hệ số phân nhánh cao và không gian tìm kiếm sâu, chẳng hạn như cờ vua. Hệ số phân nhánh trong cờ vua (số lượng nước đi hợp lệ trung bình cho mỗi vị trí) là khoảng 35 và một ván cờ điển hình có thể kéo dài nhiều nước đi. Điều này dẫn đến một cây trò chơi khổng lồ, khiến việc tìm kiếm Minimax đầy đủ trở nên bất khả thi về mặt tính toán. Sự tăng trưởng theo cấp số nhân về số lượng nút cần đánh giá khi tăng độ sâu tìm kiếm nhấn mạnh sự cần thiết của các thuật toán tìm kiếm và heuristic hiệu quả.

Cờ vua hoàn toàn phù hợp với các tiêu chí của trò chơi thông tin hoàn hảo và tổng bằng không. Cả hai người chơi đều có tầm nhìn hoàn toàn về trạng thái bàn cờ, không có yếu tố ẩn giấu hoặc may rủi nào liên quan và mỗi nước đi đều dẫn đến một kết quả xác định. Hơn nữa, kết quả mang tính đối kháng nghiêm ngặt: chiến thắng cho một người chơi là thua cho người kia và một ván hòa không mang lại lợi ích cho ai. Các thuộc tính này làm cho Minimax trở thành một thuật toán phù hợp về mặt lý thuyết cho cờ vua. Khía cạnh thông tin hoàn hảo đảm bảo rằng AI có thể phân tích đầy đủ tất cả các trạng thái tương lai có thể mà không có sự không chắc chắn về thông tin ẩn. Tính chất tổng bằng không đơn giản hóa quá trình đánh giá bằng cách tập trung vào lợi thế tương đối của một người chơi so với người kia. Các đặc điểm này cho phép thuật toán Minimax đưa ra các quyết định tối ưu dựa trên một mô hình trò chơi hoàn chỉnh và xác định.

Thuật toán Minimax, mặc dù có cơ sở lý thuyết vững chắc để chơi tối ưu trong các trò chơi có thông tin hoàn hảo và tổng bằng không, nhưng phải đối mặt với những hạn chế thực tế đáng kể trong các trò chơi phức tạp như cờ vua do độ phức tạp về mặt tính toán của nó. Điều này đòi hỏi phải sử dụng các kỹ thuật tối ưu hóa và các hàm đánh giá heuristic để xây dựng AI cờ vua hiệu quả. Các nguyên tắc cốt lõi của Minimax cung cấp một khuôn khổ cơ bản để ra quyết định trong các trò chơi đối kháng. Tuy nhiên, quy mô rộng lớn của cây trò chơi cờ vua đòi hỏi các điều chỉnh và cải tiến thực tế để làm cho thuật toán có thể xử lý được về mặt tính toán. Các kỹ thuật như cắt tỉa Alpha-Beta và các hàm đánh giá được thiết kế tốt là rất quan trọng để thu hẹp khoảng cách giữa lý tưởng lý thuyết của Minimax và thực tế thực tế của việc xây dựng một chương trình chơi cờ vua mạnh mẽ.

2.4.3 Nâng cao hiệu quả với cắt tỉa Alpha-Beta

Cắt tỉa Alpha-Beta là một kỹ thuật tối ưu hóa cho thuật toán Minimax giúp giảm đáng kể thời gian tính toán. Nó đạt được điều này bằng cách cắt bỏ các nhánh trong cây trò chơi không cần tìm kiếm vì đã có sẵn một nước đi tốt hơn. Điều này cho phép tìm kiếm nhanh hơn và khả năng khám phá các cấp độ sâu hơn của cây trò chơi. Kỹ thuật này hoạt động bằng cách truyền hai tham số bổ sung, alpha và beta, vào hàm minimax. Alpha đại diện cho giá trị tốt nhất mà người chơi tối đa hóa hiện có thể đảm bảo ở cấp độ đó trở lên, trong khi beta đại diện cho giá trị tốt nhất mà người chơi tối thiểu hóa hiện có thể đảm bảo ở cấp độ đó trở xuống. Trong các lệnh gọi đệ quy của hàm minimax, các giá trị alpha và beta này được cập nhật. Đối với người chơi tối đa hóa, giá trị alpha được cập nhật với giá trị tối đa tìm thấy cho đến nay trong số các con của nó. Nếu tại bất kỳ thời điểm nào, giá trị beta (từ một tổ tiên tối thiểu hóa) nhỏ hơn hoặc bằng giá trị alpha hiện tại, điều đó có nghĩa là người chơi tối thiểu hóa sẽ không bao giờ chọn đường đi này và việc tìm kiếm các con còn lại của nút tối đa hóa hiện tại có thể dừng lại (cắt tỉa). Tương tự, đối với người chơi tối thiểu hóa, giá trị beta được cập nhật với giá trị tối thiểu tìm thấy cho đến nay trong số các con của nó. Nếu giá trị beta trở nên nhỏ hơn hoặc bằng giá trị alpha (từ một tổ tiên tối đa hóa), người chơi tối đa hóa sẽ không bao giờ chọn đường đi này và việc tìm kiếm các con còn lại của nút tối thiểu hóa hiện tại có thể bị chấm dứt.

Giá trị alpha và beta đóng vai trò là ngưỡng trong quá trình tìm kiếm. Alpha đại diện cho giới hạn dưới của điểm số mà người chơi tối đa hóa có thể đạt được và beta đại diện cho giới hạn trên của điểm số mà người chơi tối thiểu hóa có thể đạt được. Các giới hạn này liên tục được cập nhật khi quá trình tìm kiếm tiến triển. Bằng cách theo dõi các giới hạn này, thuật toán có thể xác định khi nào một nhánh cụ thể của cây tìm kiếm không còn phù hợp nữa. Nếu người chơi tối đa hóa tìm thấy một nước đi đảm bảo điểm số lớn hơn hoặc bằng beta, người chơi tối thiểu hóa sẽ không bao giờ chọn đường đi đó, vì vậy không cần khám phá thêm. Tương tự, nếu người chơi tối thiểu hóa tìm thấy một nước đi đảm bảo điểm số nhỏ hơn hoặc bằng alpha, người chơi tối đa hóa sẽ không bao giờ chọn đường đi đó. Điều kiện beta <= alpha (hoặc beta < alpha tùy thuộc vào cách triển khai cụ thể) là chìa khóa để cắt tỉa. Nó biểu thị rằng nhánh hiện tại của cây tìm kiếm không thể dẫn đến kết quả tốt hơn những gì người chơi tương ứng đã đạt được ở các cấp độ cao hơn của cây. Khi beta <= alpha, điều đó có nghĩa là người chơi tối thiểu hóa đã tìm thấy một nước đi đảm bảo điểm số tối đa là beta và người chơi tối đa hóa đã tìm thấy một nước đi đảm bảo điểm số tối thiểu là alpha. Nếu alpha lớn hơn hoặc bằng beta, không có khả năng nhánh đang được khám phá có thể dẫn đến kết quả tốt hơn cho bất kỳ người chơi nào, vì vậy nó có thể được cắt tỉa một cách an toàn, tiết kiệm nỗ lực tính toán.

Các ví dụ minh họa rõ ràng sức mạnh của việc cắt tỉa Alpha-Beta trong việc giảm không gian tìm kiếm. Bằng cách tránh đánh giá toàn bộ các cây con được đảm bảo không ảnh hưởng đến quyết định cuối cùng, thuật toán có thể đạt được những cải thiện đáng kể về hiệu suất. Ví dụ, trong cờ vua, nếu một nước đi cho phép đối thủ chiếu hết, không cần phải xem xét bất kỳ nước đi nào khác của họ từ trạng thái đó. Một ví dụ khác, trong một cây trò chơi, nếu người chơi tối đa hóa tìm thấy một điểm số là 5 (alpha=5), và sau đó một nước đi khác cho phép người chơi tối thiểu hóa buộc một điểm số tối đa là 4 (beta=4), thì người chơi tối đa hóa có thể ngừng khám phá thêm dưới nước đi thứ hai vì 5 > 4. Những ví dụ này làm nổi bật logic cốt lõi của điều kiện cắt tỉa. Khi thuật toán phát hiện ra rằng việc khám phá một chuỗi các nước đi cụ thể không thể dẫn đến kết quả tốt hơn cho người chơi hiện tại so với những gì đã được thiết lập, nó sẽ cắt bỏ hiệu quả nhánh đó của cây tìm kiếm, ngăn chặn các tính toán không cần thiết và cho phép khám phá sâu hơn các đường đi hứa hẹn hơn.

Trong các kịch bản lý tưởng, nơi các nước đi được sắp xếp theo thứ tự tối ưu, việc cắt tỉa Alpha-Beta có khả năng giảm đáng kể số lượng nút được khám phá, có khả năng cho phép độ sâu tìm kiếm lớn gấp đôi so với những gì có thể với thuật toán Minimax đơn giản trong cùng một ràng buộc về thời gian. Tiềm năng cải thiện theo cấp số nhân này nhấn mạnh tầm quan trọng của các heuristic sắp xếp nước đi hiệu quả trong việc tối đa hóa lợi ích của việc cắt tỉa Alpha-Beta. Bằng cách ưu tiên khám phá các nước đi có nhiều khả năng dẫn đến việc cắt tỉa hơn, thuật toán có thể đạt được hiệu quả tối đa của nó.

Thứ tự xem xét các nước đi trong quá trình tìm kiếm Alpha-Beta ảnh hưởng đáng kể đến số lượng lần cắt tỉa đạt được. Việc khám phá các nước đi tốt nhất trước sẽ làm tăng khả năng cắt tỉa sớm, dẫn đến việc tìm kiếm hiệu quả hơn. Khi thuật toán gặp một nước đi tốt sớm trong quá trình khám phá một nút, nó có thể nhanh chóng thiết lập một giới hạn alpha hoặc beta chặt chẽ. Các nước đi tiếp theo tại nút đó tệ hơn giới hạn này sau đó có thể bị cắt tỉa mà không cần đánh giá thêm. Do đó, các heuristic ưu tiên việc tạo và đánh giá các nước đi hứa hẹn trước là rất quan trọng để tối đa hóa hiệu quả của việc cắt tỉa Alpha-Beta.

Cắt tỉa Alpha-Beta là một tối ưu hóa cơ bản cho thuật toán Minimax giúp giảm đáng kể không gian tìm kiếm trong AI cờ vua, cho phép phân tích sâu hơn và hiệu quả hơn các nước đi tiềm năng mà không ảnh hưởng đến tính tối ưu của kết quả. Hiệu quả của kỹ thuật này phụ thuộc nhiều vào thứ tự đánh giá các nước đi. Bằng cách cắt tỉa một cách thông minh các nhánh của cây trò chơi không thể ảnh hưởng đến quyết định cuối cùng, việc cắt tỉa Alpha-Beta giúp cho việc tìm kiếm sâu trong cờ vua trở nên khả thi về mặt tính toán. Tối ưu hóa này là điều cần thiết để xây dựng các công cụ cờ vua mạnh mẽ có thể phân tích trò chơi đến độ sâu đáng kể và đưa ra các quyết định sáng suốt dựa trên sự hiểu biết toàn diện về các trạng thái tương lai tiềm năng. Sự phụ thuộc vào thứ tự nước đi làm nổi bật tầm quan trọng của việc kết hợp kiến thức và heuristic cụ thể về cờ vua vào AI để hướng dẫn tìm kiếm đến các phần liên quan nhất của cây trò chơi.

2.4.4 Vai trò của hàm đánh giá

Hàm đánh giá là một thành phần quan trọng cho phép AI cờ vua đánh giá chất lượng của một thế cờ nhất định. Vì việc tìm kiếm đầy đủ đến cuối ván cờ thường là điều không thể, hàm này hoạt động như một heuristic, cung cấp ước tính về giá trị của thế cờ và hướng dẫn tìm kiếm đến các nước đi hứa hẹn. Thuật toán Minimax dựa vào việc có một giá trị cho mỗi trạng thái cuối cùng. Trong thực tế, đối với các trò chơi phức tạp như cờ vua, việc tìm kiếm bị giới hạn ở một độ sâu nhất định. Ở độ sâu này, hàm đánh giá sẽ can thiệp để cung cấp giá trị ước tính cho các nút không phải là cuối cùng. Ước tính này cho phép thuật toán Minimax (thường kết hợp với cắt tỉa Alpha-Beta) hoạt động hiệu quả bằng cách so sánh các ưu điểm tương đối của các nước đi khác nhau mà không cần đạt đến kết thúc thực tế của trò chơi. Độ chính xác và hiệu quả của hàm đánh giá là những yếu tố quan trọng trong hiệu suất của AI cờ vua.

Hàm đánh giá cung cấp một cái nhìn tức thời về chất lượng của bàn cờ mà không cần xem xét các nước đi trong tương lai. Bản chất "tĩnh" này có nghĩa là nó dựa vào các đặc điểm hiện diện trong thế cờ hiện tại để dự đoán thành công trong tương lai. Các khía cạnh động hơn như các mối đe dọa tức thời hoặc các chuỗi chiến thuật thường được xử lý bởi thuật toán tìm kiếm (chẳng hạn như Minimax và Tìm kiếm yên tĩnh). Trong khi thuật toán tìm kiếm khám phá hậu quả của các nước đi qua nhiều lớp, hàm đánh giá cung cấp một đánh giá nhanh chóng về bàn cờ tại bất kỳ thời điểm nào trong quá trình tìm kiếm đó. Sự phân tách trách nhiệm này cho phép một cách tiếp cận hiệu quả hơn, trong đó hàm đánh giá tập trung vào việc nhận dạng các mẫu và đặc điểm trong trạng thái hiện tại, và thuật toán tìm kiếm khám phá các trạng thái tương lai tiềm năng phát sinh từ các nước đi khác nhau.

Đầu ra của hàm đánh giá thường là một điểm số số học biểu thị lợi thế cho một bên. Điểm số này cung cấp một thước đo định lượng về trạng thái bàn cờ, cho phép AI so sánh các vị trí khác nhau và chọn nước đi dẫn đến điểm số cao nhất từ quan điểm của nó (người chơi tối đa hóa). Dấu và độ lớn của điểm số cho biết hướng và mức độ của lợi thế. Thang điểm và đơn vị của điểm số này (ví dụ: centipawn) cung cấp một cách tiêu chuẩn để diễn giải đánh giá của công cụ về vị trí. Centipawn cung cấp một thang đo chi tiết để đo lường lợi thế về vị trí, cho phép các công cụ cờ vua phân biệt giữa những khác biệt rất nhỏ trong trạng thái bàn cờ. Độ chính xác này rất quan trọng để đưa ra các quyết định tối ưu, đặc biệt trong các ván đấu cấp cao, nơi ngay cả những lợi thế nhỏ cũng có thể mang tính quyết định. Bằng cách sử dụng một đơn vị phân số của một quân tốt, các công cụ cờ vua có thể biểu thị bằng số ngay cả những lợi thế nhỏ về vị trí. Mức độ chi tiết này cho phép AI đưa ra các đánh giá sắc thái hơn và chọn các nước đi cải thiện dần vị trí của nó, ngay cả khi chúng không dẫn đến lợi thế vật chất ngay lập tức.

Hàm đánh giá xem xét nhiều thành phần và yếu tố chính để đánh giá sức mạnh của một thế cờ. **Cân bằng vật chất** thường là yếu tố được cân nhắc nhiều nhất trong các hàm đánh giá, phản ánh lợi thế cơ bản của việc có nhiều quân hơn hoặc quân có giá trị cao hơn. Các giá trị cụ thể được gán cho mỗi quân cờ đại diện cho sức mạnh tương đối và tiềm năng đóng góp vào chiến thắng của ván cờ. Thông thường, các giá trị này là: Tốt = 1, Mã/Tượng = 3, Xe = 5, Hậu = 9 và Vua có giá trị rất lớn (ví dụ: vô cùng). Trong cờ vua, lợi thế về vật chất thường chuyển trực tiếp thành khả năng chiến thắng cao hơn. Do đó, việc đánh giá chính xác sự cân bằng vật chất là mục tiêu chính của hàm đánh giá. Các giá trị điểm được gán dựa trên sức mạnh tương đối được chấp nhận chung của các quân cờ, mặc dù các giá trị này có thể được điều chỉnh nhẹ tùy thuộc vào công cụ cụ thể và giai đoạn của trò chơi. **Tính cơ động**, số lượng nước đi hợp lệ có sẵn cho mỗi quân cờ, là một yếu tố vị trí quan trọng khác. Các quân cờ có nhiều nước đi hơn có thể gây ảnh hưởng lớn hơn trên bàn cờ, kiểm soát nhiều ô hơn và tạo ra nhiều mối đe dọa hơn. Mức độ cơ động cao hơn thường cho thấy một vị trí tích cực và thuận lợi hơn. Các quân cờ bị hạn chế di chuyển có tác động hạn chế đến trò chơi. Ngược lại, các quân cờ có nhiều nước đi hợp lệ có thể tham gia vào các cuộc tấn công và phòng thủ trên một khu vực rộng lớn hơn của bàn cờ. Do đó, các hàm đánh giá thường thưởng cho các vị trí mà quân cờ của người chơi có tính cơ động cao và phạt các vị trí mà quân cờ của đối thủ di động trong khi quân cờ của họ bị hạn chế. **Sự an toàn của vua** là tối quan trọng trong cờ vua, vì mất vua dẫn đến thua ván cờ. Do đó, các hàm đánh giá cân nhắc kỹ lưỡng sự an toàn của cả hai vua, thưởng cho những vua được bảo vệ tốt và phạt những vua bị lộ. Một vị vua dễ bị tấn công là một điểm yếu đáng kể mà đối thủ sẽ cố gắng khai thác. Các yếu tố như sự hiện diện của các quân tốt che chắn vua, sự vắng mặt của các cột mở có thể cho phép xe hoặc hậu của đối phương tấn công và sự gần gũi của các quân cờ đối phương đều góp phần đánh giá sự an toàn của vua. Hàm đánh giá cần phản ánh chính xác mức độ đe dọa đối với vua. **Cấu trúc tốt** ảnh hưởng đáng kể đến bối cảnh chiến lược của ván cờ. Các cấu trúc tốt mạnh mẽ có thể hỗ trợ các quân cờ và kiểm soát các ô quan trọng, trong khi các cấu trúc tốt yếu có thể tạo ra các lỗ hổng và mục tiêu cho đối thủ. Các hàm đánh giá đánh giá các khía cạnh khác nhau của cấu trúc tốt để xác định sức mạnh hoặc điểm yếu tổng thể của nó. Cấu trúc tốt thường được coi là một yếu tố chiến lược dài hạn hơn của trò chơi. Các yếu tố như các quân tốt kết nối, các đảo quân tốt, các quân tốt thông và các điểm yếu của quân tốt (ví dụ: quân tốt bị cô lập, quân tốt kép, quân tốt lạc hậu) đều góp phần vào việc đánh giá tổng thể cấu trúc tốt và tác động của nó đến vị trí. **Kiểm soát không gian**, đặc biệt là các ô trung tâm của bàn cờ, mang lại cho người chơi nhiều lựa chọn hơn để di chuyển quân và hạn chế sự di chuyển của đối thủ. Lợi thế không gian này thường có thể chuyển thành lợi thế chiến lược. Các quân cờ nằm ở hoặc kiểm soát trung tâm bàn cờ thường có nhiều nước đi tiềm năng hơn và có thể gây ảnh hưởng đến một khu vực rộng lớn hơn. Việc kiểm soát các ô quan trọng cũng có thể ngăn đối thủ phát triển quân cờ hiệu quả hoặc thực hiện các cuộc tấn công. Các hàm đánh giá thường thưởng cho các vị trí mà người chơi kiểm soát nhiều ô quan trọng hơn. **Sự phối hợp quân cờ** cho phép người chơi hỗ trợ lẫn nhau, tạo ra các cuộc tấn công mạnh mẽ và phòng thủ hiệu quả. Một hàm đánh giá cần nhận ra và thưởng cho các vị trí mà quân cờ của người chơi đang phối hợp hài hòa. Phối hợp bao gồm các yếu tố như các quân cờ bảo vệ lẫn nhau, các quân cờ phối hợp để kiểm soát các ô quan trọng hoặc tấn công các quân cờ chủ chốt của đối phương và sự hài hòa và hiệp lực tổng thể của lực lượng người chơi. Một đội quân phối hợp tốt thường mạnh hơn một tập hợp các quân cờ mạnh riêng lẻ. **Tempo**, hoặc quyền chủ động, có thể là một lợi thế đáng kể trong cờ vua, vì nó cho phép người chơi điều khiển nhịp độ của trò chơi và buộc đối thủ phải phản ứng. Các hàm đánh giá có thể kết hợp một cách tinh tế giá trị của việc có nước đi tiếp theo. Khái niệm tempo liên quan đến hiệu quả phát triển và khả năng tạo ra các mối đe dọa. Một người chơi đi trước trong việc phát triển hoặc có thể tạo ra các mối đe dọa mà đối thủ phải phản ứng sẽ có quyền chủ động. Các hàm đánh giá có thể gán một phần thưởng nhỏ cho bên có lượt đi, phản ánh lợi thế này.

Các công cụ cờ vua khác nhau có thể sử dụng các hàm đánh giá khác nhau và gán các trọng số khác nhau cho các yếu tố này. Việc triển khai và điều chỉnh cụ thể hàm đánh giá là những yếu tố phân biệt chính giữa các công cụ cờ vua khác nhau. Các trọng số được gán cho các yếu tố khác nhau phản ánh sự hiểu biết chiến lược của nhà phát triển công cụ và có thể ảnh hưởng đáng kể đến phong cách chơi và sức mạnh của công cụ. Không có một cách "đúng" duy nhất để đánh giá một thế cờ. Các công cụ khác nhau có thể ưu tiên các khía cạnh khác nhau của trò chơi. Ví dụ, một công cụ có thể coi trọng vật chất hơn một chút so với một công cụ khác, trong khi một công cụ khác có thể chú trọng hơn đến sự an toàn của vua hoặc sự hoạt động của quân cờ. Quá trình xác định trọng số tối ưu cho các yếu tố này là một lĩnh vực nghiên cứu và phát triển phức tạp và liên tục trong AI cờ vua.

Bảng 2.1 Giá trị các quân cờ trong AI cờ vua(ví dụ)

|  |  |
| --- | --- |
| Quân cờ | Giá trị |
| Tốt | 1 |
| Mã | 3 |
| Tượng | 3 (hoặc 3.5 1 ) |
| Xe | 5 |
| Hậu | 9 |
| Vua | Rất lớn (ví dụ: 200 hoặc vô cùng) |

Bảng 2.2 Phạm vi điểm số của hàm đánh giá

|  |  |
| --- | --- |
| Phạm vi | Đánh giá |
| -0.26 đến 0.26 | Bình đẳng |
| 0.27 đến 0.7 | Lợi thế nhỏ cho Trắng |
| 0.7 đến 1.5 | Lợi thế rõ ràng cho Trắng |
| trên 1.5 | Lợi thế quyết định cho Trắng |
| -0.27 đến -0.7 | Lợi thế nhỏ cho Đen |
| -0.7 đến -1.5 | Lợi thế rõ ràng cho Đen |
| trên -1.5 | Lợi thế quyết định cho Đen |
| cộng bất kỳ số nào | Chiếu hết |

2.4.5 Mở rộng tìm kiếm với tìm kiếm tĩnh lặng

Tìm kiếm yên tĩnh là một thuật toán cần thiết để giải quyết hiệu ứng đường chân trời trong các thuật toán tìm kiếm giới hạn độ sâu trong cờ vua. Hiệu ứng đường chân trời gây ra một vấn đề đáng kể cho các thuật toán tìm kiếm giới hạn độ sâu trong cờ vua. Bằng cách chỉ nhìn trước một số lượng nước đi cố định, AI có thể bỏ lỡ các diễn biến chiến thuật quan trọng hoặc các hậu quả lâu dài nằm ngay ngoài độ sâu tìm kiếm của nó, dẫn đến các quyết định có khả năng sai sót. Hãy tưởng tượng một kịch bản mà một người chơi hy sinh một quân cờ ở đường chân trời tìm kiếm, khiến AI tin rằng nó đã đạt được lợi thế. Tuy nhiên, một nước đi sau đường chân trời, đối thủ có một đòn phản công tàn khốc mà AI đã không thấy trước. Hiệu ứng đường chân trời gây ra sự thiển cận này, nơi đánh giá của AI dựa trên một bức tranh không đầy đủ về trạng thái trò chơi trong tương lai.

Trong các thế cờ có các cuộc trao đổi chiến thuật đang diễn ra hoặc các mối đe dọa tức thời, việc đánh giá tĩnh ở độ sâu tìm kiếm cố định có thể gây hiểu lầm nghiêm trọng. Giá trị của thế cờ có thể thay đổi đáng kể trong nước đi tiếp theo và một công cụ không nhìn xa hơn đường chân trời của nó có thể thực hiện một nước đi có vẻ thuận lợi nhưng thực tế lại là một sai lầm. Hãy tưởng tượng một chuỗi các lần bắt quân. Nếu quá trình tìm kiếm chính dừng lại ở giữa chuỗi này, hàm đánh giá có khả năng đưa ra đánh giá không chính xác về sự cân bằng vật chất cuối cùng. Tìm kiếm yên tĩnh nhằm mục đích mở rộng quá trình tìm kiếm cụ thể trong các thế cờ "nhiễu" hoặc "không ổn định" này cho đến khi đạt được trạng thái yên tĩnh hơn, nơi có thể thực hiện đánh giá tĩnh đáng tin cậy hơn.

Tìm kiếm yên tĩnh chọn lọc làm sâu sắc thêm quá trình tìm kiếm trong các phần của cây trò chơi nơi đánh giá có khả năng dao động đáng kể trong tương lai gần. Việc mở rộng tập trung này giúp ổn định đánh giá và cung cấp đánh giá chính xác hơn về giá trị thực của thế cờ. Thay vì tìm kiếm tất cả các nước đi có thể đến độ sâu lớn hơn, điều này sẽ rất tốn kém về mặt tính toán, tìm kiếm yên tĩnh nhắm mục tiêu vào các loại nước đi cụ thể, chủ yếu là các lần bắt quân, trong các thế cờ được coi là "không yên tĩnh". Mục tiêu là đạt đến trạng thái "yên tĩnh" nơi không có những thay đổi chiến thuật lớn nào sắp xảy ra, cho phép hàm đánh giá tĩnh cung cấp một điểm số đáng tin cậy hơn.

Tìm kiếm yên tĩnh thường tập trung vào các nước đi có thể thay đổi đáng kể đánh giá, chẳng hạn như các lần bắt quân. Việc bắt quân thường là loại nước đi quan trọng nhất về mặt thay đổi cân bằng vật chất và đánh giá tổng thể của một thế cờ. Bằng cách tập trung tìm kiếm mở rộng vào các lần bắt quân, tìm kiếm yên tĩnh nhằm mục đích giải quyết các chuỗi chiến thuật tức thời và đạt đến trạng thái ổn định hơn để đánh giá. Một lần bắt quân có thể dẫn đến việc trực tiếp giành được vật chất, đây là một yếu tố chính trong hàm đánh giá. Do đó, điều quan trọng là phải khám phá những hậu quả tức thời của các lần bắt quân ngoài độ sâu tìm kiếm chính để tránh đánh giá sai giá trị của một thế cờ. Tìm kiếm yên tĩnh ưu tiên các loại nước đi này để đảm bảo đánh giá chính xác hơn.

Khái niệm "đứng im" cung cấp một đường cơ sở cho việc tìm kiếm yên tĩnh. Nếu đánh giá tĩnh hiện tại đã đủ tốt và không có lần bắt quân nào có thể cải thiện đáng kể nó, thì có thể kết thúc tìm kiếm sớm, tiết kiệm tài nguyên tính toán. Ý tưởng đằng sau việc đứng im là người chơi hiện tại luôn có tùy chọn thực hiện một nước đi không phải là bắt quân. Nếu đánh giá tĩnh của thế cờ hiện tại đã thuận lợi và không có lần bắt quân nào ngay lập tức có thể cải thiện nó, thì việc tìm kiếm không cần khám phá thêm các lần bắt quân từ vị trí này. Điểm số đứng im đóng vai trò là giới hạn dưới vì người chơi luôn có thể chọn thực hiện một nước đi không phải là bắt quân và đạt được ít nhất điểm số đó.

Nhiều chương trình cờ vua xử lý đặc biệt các nước chiếu và các nước tránh chiếu trong tìm kiếm yên tĩnh. Một thế cờ mà bên đi quân đang bị chiếu được coi là không yên tĩnh, vì có một mối đe dọa ngay lập tức cần được giải quyết. Trong những trường hợp như vậy, tất cả các nước đi hợp lệ để thoát khỏi chiếu thường được tìm kiếm. Việc đứng im không được phép khi đang bị chiếu vì không có đảm bảo về một nước đi có thể phù hợp với alpha và tất cả các nước đi đều đang được xem xét, không chỉ các lần bắt quân.

Để tránh tính toán quá mức, điều quan trọng là phải giới hạn độ sâu và phạm vi của tìm kiếm yên tĩnh. Mặc dù các tìm kiếm yên tĩnh thường ngắn, chúng có thể tiêu tốn một phần đáng kể các nút tìm kiếm (50%-90%). Do đó, các kỹ thuật cắt tỉa như cắt tỉa delta có thể được sử dụng để hạn chế không gian tìm kiếm trong quá trình tìm kiếm yên tĩnh.

Tìm kiếm yên tĩnh là một phần mở rộng quan trọng đối với thuật toán tìm kiếm chính, đảm bảo sự ổn định về chiến thuật ở đường chân trời tìm kiếm, ngăn công cụ đưa ra các nước đi không hợp lý dựa trên các đánh giá không đầy đủ. Bằng cách tập trung vào các thế cờ thay đổi động và đặc biệt là vào các lần bắt quân, nó cung cấp một đánh giá chính xác hơn về trạng thái trò chơi.

2.4.6 Tận dụng các tính toán trước với bảng chuyển vị

Bảng chuyển vị là một thành phần cơ bản của các công cụ cờ vua hiệu quả, hoạt động như một dạng ghi nhớ để lưu trữ các vị trí đã thấy trước đó và các đánh giá liên quan từ quá trình tìm kiếm cây trò chơi. Kỹ thuật này đặc biệt hữu ích trong các trò chơi có thông tin hoàn hảo như cờ vua. Trong quá trình tìm kiếm vét cạn, các chương trình cờ vua thường xuyên gặp lại các vị trí trò chơi giống nhau thông qua các chuỗi nước đi khác nhau, một hiện tượng được gọi là chuyển vị.

Bảng chuyển vị hoạt động như một bộ nhớ cache lớn lưu trữ thông tin về các vị trí đã được tìm kiếm trước đó. Khi thuật toán chơi trò chơi gặp một vị trí đã được phân tích trước đó, thông tin được lưu trữ trong Bảng chuyển vị sẽ được truy xuất thay vì phân tích lại vị trí từ đầu. Việc "ghi nhớ" các tính toán trước đây này giúp tiết kiệm đáng kể thời gian và tài nguyên. Ngay cả khi thông tin được lưu trữ (chẳng hạn như độ sâu tìm kiếm hoặc điểm đánh giá chính xác) không hoàn toàn phù hợp với tình huống hiện tại, nước đi tốt nhất được tìm thấy trong quá trình tìm kiếm trước đó vẫn có thể có giá trị để cải thiện thứ tự xem xét các nước đi, có khả năng dẫn đến việc cắt tỉa cây tìm kiếm nhanh hơn. Điều này đặc biệt có lợi trong tìm kiếm sâu dần, nơi kết quả từ các tìm kiếm nông hơn có thể cung cấp thông tin cho các tìm kiếm sâu hơn.

Mỗi mục trong bảng chuyển vị thường chứa các thông tin sau: Khóa băm (Zobrist hoặc BCH), một chữ ký gần như duy nhất của thế cờ được sử dụng để nhanh chóng xác định vị trí khi thăm dò bảng. Khóa này rất quan trọng để xác minh rằng thông tin được lưu trữ tương ứng với vị trí hiện tại, đặc biệt trong trường hợp xảy ra xung đột. Nước đi tốt nhất hoặc nước đi bác bỏ, nước đi được tìm thấy là tốt nhất trong vị trí đã phân tích trước đó hoặc một nước đi dẫn đến sự bác bỏ một dòng chơi. Nước đi này được ưu tiên trong quá trình tìm kiếm hiện tại do khả năng nó sẽ lại có liên quan. Độ sâu (bản nháp), độ sâu mà vị trí đã được tìm kiếm trong phân tích trước đó. Thông tin này giúp xác định xem kết quả được lưu trữ có liên quan đến độ sâu tìm kiếm hiện tại hay không. Điểm số, điểm đánh giá của vị trí thu được từ quá trình tìm kiếm trước đó. Đây có thể là điểm số chính xác hoặc giới hạn (trên hoặc dưới) của giá trị thực của vị trí. Loại nút, cho biết loại nút tìm kiếm đã tạo ra điểm số được lưu trữ. Các loại phổ biến bao gồm nút PV (nút biến chính), nút tất cả và nút cắt. Tuổi, một giá trị được sử dụng để xác định thời điểm ghi đè các mục cũ hơn trong bảng để nhường chỗ cho các vị trí được tìm kiếm gần đây hơn. Điều này giúp đảm bảo rằng bảng chứa thông tin liên quan đến trạng thái hiện tại của trò chơi.

Các hàm băm được sử dụng để chuyển đổi các thế cờ phức tạp thành một biểu diễn số nhỏ gọn (khóa băm). Điều này cho phép lưu trữ và truy xuất thông tin hiệu quả trong Bảng chuyển vị. Các phương pháp băm phổ biến được sử dụng trong các chương trình cờ vua bao gồm băm Zobrist và băm BCH. Do giới hạn bộ nhớ, toàn bộ khóa băm 64 bit thường không được sử dụng làm địa chỉ trực tiếp trong bảng băm. Thay vào đó, một phần của khóa băm được sử dụng để tính toán một chỉ số (địa chỉ) trong bảng, thường bằng cách lấy khóa băm modulo kích thước của bảng (thường là lũy thừa của hai). Vì số lượng các vị trí cờ vua có thể lớn hơn rất nhiều so với kích thước của Bảng chuyển vị, nên không thể tránh khỏi việc đôi khi các vị trí khác nhau sẽ ánh xạ đến cùng một chỉ số trong bảng (xung đột chỉ số hoặc lỗi loại 2) hoặc thậm chí cùng một khóa băm (xung đột khóa hoặc lỗi loại 1). Để xử lý xung đột chỉ số, toàn bộ hoặc một phần đáng kể của khóa băm được lưu trữ trong mục bảng cùng với các thông tin khác. Khi một vị trí được tra cứu, khóa băm được lưu trữ sẽ được so sánh với khóa băm của vị trí hiện tại để đảm bảo khớp. Xung đột khóa hiếm hơn nhưng có thể dẫn đến kết quả không chính xác nếu không được xử lý cẩn thận. Các kỹ thuật như lưu trữ các nước đi giả hợp lệ có thể giúp giảm thiểu tác động của xung đột khóa.

Khi Bảng chuyển vị đầy và cần lưu trữ một vị trí mới, một chiến lược thay thế sẽ được sử dụng để quyết định mục hiện có nào sẽ bị ghi đè. Có nhiều chiến lược khác nhau, xem xét các yếu tố như: Luôn thay thế, chiến lược đơn giản nhất, nơi mọi mục mới ngay lập tức ghi đè mục hiện có tại chỉ số đã tính toán. Ưu tiên độ sâu, ưu tiên giữ các mục đã được tìm kiếm đến độ sâu lớn hơn, vì chúng đại diện cho nỗ lực tính toán đáng kể hơn. Hệ thống hai cấp, sử dụng hai bảng riêng biệt (hoặc các khe cắm cho mỗi chỉ số), một ưu tiên độ sâu và một ưu tiên luôn thay thế. Lão hóa, giảm dần "tuổi" của các mục theo thời gian và các mục mới hơn thay thế các mục cũ hơn, ngay cả khi các mục cũ hơn có độ sâu lớn hơn.

Thông tin được lưu trữ trong Bảng chuyển vị (độ sâu, điểm số và giới hạn) có thể được sử dụng để thực hiện việc cắt tỉa trong quá trình tìm kiếm alpha-beta, giảm thêm không gian tìm kiếm. Nếu Bảng chuyển vị chứa một mục cho vị trí hiện tại với độ sâu đủ và điểm số đáp ứng các điều kiện alpha-beta (ví dụ: giới hạn dưới lớn hơn hoặc bằng beta hoặc giới hạn trên nhỏ hơn hoặc bằng alpha), nhánh hiện tại của tìm kiếm có thể bị cắt tỉa. Nước đi tốt nhất được lưu trữ trong Bảng chuyển vị thường được thử đầu tiên trong quá trình sắp xếp thứ tự các nước đi. "Nước đi băm" này có khả năng là một nước đi tốt và thường có thể dẫn đến việc cắt tỉa sớm, cải thiện đáng kể hiệu quả tìm kiếm. Trong các công cụ cờ vua hiện đại sử dụng tìm kiếm song song trên CPU đa lõi, một bảng chuyển vị toàn cục được chia sẻ bởi nhiều luồng là rất quan trọng để đạt hiệu quả. Việc triển khai bảng chia sẻ này một cách an toàn cho luồng (đôi khi sử dụng các kỹ thuật không khóa) là điều quan trọng để tránh các nút thắt cổ chai về hiệu suất. Ngoài Bảng chuyển vị chính, các chương trình cờ vua thường sử dụng các bảng băm khác để lưu trữ các loại thông tin khác nhau, chẳng hạn như điểm đánh giá, số lượng vật chất và đánh giá cấu trúc tốt.

Bảng chuyển vị là một thành phần cơ bản của AI chơi game mạnh mẽ, đặc biệt trong các trò chơi có thông tin hoàn hảo như cờ vua. Bằng cách lưu trữ và sử dụng lại hiệu quả kết quả của các tính toán trước đó, chúng cho phép các chương trình tìm kiếm cây trò chơi hiệu quả hơn nhiều và đạt được trình độ chơi cao hơn đáng kể. Hiệu quả của bảng chuyển vị phụ thuộc vào chất lượng của hàm băm và hiệu quả của chiến lược thay thế.

2.4.7 Lợi thế chiến lược của sách khai cuộc

Các chương trình cờ vua thường sử dụng Sách Khai cuộc, là các cơ sở dữ liệu về các nước đi đã được tính toán trước cho giai đoạn đầu của một ván cờ. Các cơ sở dữ liệu này được tham khảo miễn là các nước đi của đối thủ được nhận dạng trong sách. Thông thường, các dòng khai cuộc phổ biến hơn được lưu trữ ở độ sâu lớn hơn so với các dòng ít phổ biến hơn. Khi một nước đi do đối thủ thực hiện không được tìm thấy trong sách khai cuộc, chương trình cờ vua sẽ chuyển sang các thuật toán tìm kiếm thông thường của nó để xác định nước đi tiếp theo.

Các mục đích chính của việc sử dụng sách khai cuộc trong AI cờ vua là: Tiết kiệm thời gian: Thay vì các tìm kiếm tốn kém về mặt tính toán ngay từ đầu ván cờ, sách khai cuộc cho phép AI thực hiện các nước đi khai cuộc gần như ngay lập tức. Cung cấp chất lượng chơi cao hơn: Ngay cả các tìm kiếm sâu có thể bỏ lỡ những lợi thế chiến lược dài hạn tinh tế. Sách khai cuộc, dựa trên lý thuyết cờ vua đã được thiết lập, có thể hướng dẫn AI thông qua các khai cuộc mang lại triển vọng tổng thể tốt hơn. Giới thiệu sự đa dạng: Văn học cờ vua chứa vô số các dòng khai cuộc khác nhau. Bằng cách chọn ngẫu nhiên các nước đi từ sách khai cuộc, AI có thể giới thiệu nhiều sự đa dạng hơn vào lối chơi của mình so với bản chất thường mang tính quyết định của các thuật toán tìm kiếm.

Sách khai cuộc thường được lưu trữ ở hai loại chính: Văn bản: Chúng có thể đọc và chỉnh sửa được, giúp người dùng dễ hiểu và sửa đổi bằng các trình soạn thảo văn bản tiêu chuẩn. Tuy nhiên, chúng có thể tốn dung lượng và chậm hơn khi tìm kiếm, đặc biệt đối với các bộ sưu tập lớn các vị trí khai cuộc. Các định dạng văn bản bao gồm EPD, PGN và các định dạng đặc biệt như định dạng do Kathe Spracklen đề xuất. Nhị phân: Các định dạng này có lợi thế hơn cho các công cụ cờ vua do hiệu quả về không gian và khả năng truy cập và tìm kiếm nhanh chóng. Chúng có thể là các chuyển đổi đơn giản từ sách dựa trên văn bản hoặc, phổ biến hơn, sử dụng các giá trị băm từ các chuyển vị để lưu trữ các vị trí cùng với thông tin bổ sung như tần suất xuất hiện, thống kê thắng/thua/hòa và xếp hạng Elo của người chơi liên quan đến các vị trí đó. Định dạng PolyGlot là một ví dụ phổ biến về định dạng sách khai cuộc nhị phân. Cần có phần mềm đặc biệt để xây dựng và chỉnh sửa sách khai cuộc nhị phân.

Sách khai cuộc có thể được tạo ra thông qua một số phương pháp: Thủ công: Một nhà thiết kế có sở thích cụ thể đối với một số dòng khai cuộc nhất định có thể tạo sách thủ công. Điều này thường được thực hiện khi chuẩn bị một chương trình cờ vua cho một ván đấu quan trọng với một người chơi mạnh. Từ bộ sưu tập ván cờ: Một cơ sở dữ liệu có thể được tạo tự động bằng cách chọn một số lượng lớn các ván cờ (ví dụ: các ván cờ do các đại kiện tướng chơi hoặc các ván cờ hòa) và lưu trữ tất cả các vị trí đạt được đến một độ sâu nhất định. Mặc dù phương pháp này có thể gây ra lỗi, nhưng các chương trình thường có một ngưỡng cho số lần tối thiểu một vị trí phải xuất hiện để được đưa vào sách. Bằng cách tính toán: Bản thân các công cụ cờ vua có thể được sử dụng để đánh giá các vị trí trong một cây sách khai cuộc tiềm năng, cho phép chúng xếp hạng các nước đi và quyết định có nên tiếp tục khám phá một dòng cụ thể hay không.

Các sách khai cuộc nhị phân hiệu quả thường sử dụng các giá trị băm (từ các chuyển vị) để lưu trữ vị trí cùng với thông tin bổ sung như tần suất xuất hiện, thống kê thắng/thua/hòa và xếp hạng Elo của người chơi liên quan đến các vị trí đó. Điều này cho phép truy cập và tìm kiếm nhanh chóng trong cơ sở dữ liệu khai cuộc.

Khi một chương trình cờ vua phải quyết định một nước đi, nó sẽ tra cứu (bằng cách sử dụng tìm kiếm nhị phân) bằng cách sử dụng giá trị băm của vị trí hiện tại để thu thập tất cả thông tin đi kèm với giá trị băm đó và sau đó đưa ra quyết định dựa trên đó. Việc sử dụng sách khai cuộc cho phép AI chơi các nước đi đầu tiên của một ván cờ ở trình độ rất cao, dựa trên nhiều thế kỷ phân tích của con người và máy tính. Điều này không chỉ tiết kiệm thời gian tìm kiếm quý giá mà còn đảm bảo rằng công cụ đạt được một vị trí trung cuộc vững chắc về mặt chiến lược. Tuy nhiên, sự phụ thuộc vào các dòng đã được tính toán trước có nghĩa là các công cụ có thể dễ bị tổn thương trước các nước đi khai cuộc mới lạ hoặc ít phổ biến không có trong sách của chúng.

2.4.8 Tối ưu hoá hơn nữa thông qua cắt tỉa nước đi rỗng

Cắt tỉa nước đi rỗng là một kỹ thuật khác được sử dụng trong các chương trình cờ vua để giảm không gian tìm kiếm. Ý tưởng cốt lõi là thử một nước đi "rỗng" hoặc "bỏ qua" và xem liệu việc tìm kiếm giảm độ sâu vẫn gây ra việc cắt tỉa beta hay không. Kỹ thuật này dựa trên quan sát rằng trong hầu hết các vị trí cờ vua, việc có một nước đi hợp lệ tốt hơn là không làm gì cả.

Nguyên tắc cơ bản của cắt tỉa nước đi rỗng là: nếu sau khi đối thủ vừa đi một nước và AI vẫn ở vị trí tốt hơn, thì việc cho phép đối thủ đi thêm một nước nữa (AI bỏ qua lượt của mình) mà AI vẫn tốt hơn, thì toàn bộ nhánh phát sinh từ nước đi thêm mà đối thủ vừa thực hiện có thể bị cắt tỉa. Lý do là: nếu AI vẫn tốt hơn sau khi bỏ qua một lượt đi, thì chắc chắn AI sẽ tốt hơn nếu AI KHÔNG bỏ qua lượt đi của mình. Do đó, nếu việc tìm kiếm giảm độ sâu sau một nước đi rỗng vẫn dẫn đến điểm số đủ cao để gây ra việc cắt tỉa beta, chương trình có thể tự tin rằng nước đi hợp lệ tốt nhất cũng sẽ dẫn đến việc cắt tỉa. Điều này cho phép chương trình cắt tỉa cây tìm kiếm mà không cần khám phá đầy đủ tất cả các nước đi hợp lệ có thể.

Cắt tỉa nước đi rỗng hoạt động bằng cách bỏ qua lượt đi của bên đi quân và sau đó thực hiện tìm kiếm alpha-beta trên vị trí kết quả với độ sâu nông hơn so với khi không sử dụng heuristic nước đi rỗng. Nếu tìm kiếm nông này tạo ra việc cắt tỉa, chương trình sẽ giả định rằng việc tìm kiếm độ sâu đầy đủ khi không bỏ qua lượt đi cũng sẽ tạo ra việc cắt tỉa. Điều này cho phép chương trình tìm thấy việc cắt tỉa nhanh hơn vì tìm kiếm nông nhanh hơn tìm kiếm sâu hơn. Giá trị giảm độ sâu, thường được ký hiệu là 'R', được sử dụng trong tìm kiếm nước đi rỗng. Các giá trị phổ biến cho các triển khai cơ bản là 3 hoặc 4.

Tuy nhiên, heuristic nước đi rỗng dựa trên giả định rằng việc có một nước đi luôn tốt hơn là không có nước đi. Giả định này không đúng trong các vị trí zugzwang, nơi bất kỳ nước đi nào cũng sẽ làm xấu đi vị trí của người chơi. Do đó, việc sử dụng cắt tỉa nước đi rỗng cần được giới hạn trong các tình huống nhất định để tránh các sai lầm chiến thuật. Các hạn chế thường bao gồm không sử dụng heuristic nước đi rỗng nếu bên đi quân đang bị chiếu, chỉ còn vua và tốt, hoặc có một số lượng quân cờ nhỏ còn lại. Một cách tiếp cận khác để giải quyết vấn đề zugzwang là cắt tỉa nước đi rỗng đã được xác minh, trong đó việc tìm kiếm tiếp tục với độ sâu giảm thay vì cắt tỉa ngay lập tức.

Mặc dù có những hạn chế này, cắt tỉa nước đi rỗng là một heuristic mạnh mẽ có thể giảm đáng kể không gian tìm kiếm trong các chương trình cờ vua và được coi là một trong những kỹ thuật cắt tỉa quan trọng nhất. Ngay cả khi loại bỏ tất cả các tối ưu hóa khác khỏi Stockfish, việc giữ lại chỉ riêng cắt tỉa nước đi rỗng vẫn tạo ra một công cụ rất mạnh mẽ.

2.4.9 Tìm kiếm sâu dần

Tìm kiếm sâu dần (Iterative Deepening - ID) là một chiến lược tìm kiếm trong AI chơi cờ, trong đó chương trình bắt đầu bằng cách thực hiện một tìm kiếm rất nông của cây trò chơi, thường là đến độ sâu một lớp (một nước đi cho mỗi người chơi). Sau khi hoàn thành tìm kiếm nông này, chương trình tăng độ sâu tìm kiếm thêm một và thực hiện một tìm kiếm khác của toàn bộ cây trò chơi đến độ sâu mới này. Quá trình tăng dần độ sâu tìm kiếm và tìm kiếm lại từ vị trí gốc được lặp lại cho đến khi hết thời gian được phân bổ cho việc tìm kiếm.

Cải thiện việc sắp xếp nước đi:Một trong những lợi ích đáng kể nhất của ID là kết quả từ các tìm kiếm nông hơn có thể được sử dụng để cải thiện hiệu quả của các tìm kiếm sâu hơn. Ví dụ, nước đi tốt nhất được tìm thấy trong một tìm kiếm một lớp có khả năng là một ứng cử viên tốt trong một tìm kiếm hai lớp. Bằng cách tìm kiếm nước đi "tốt nhất" này trước trong lần lặp tiếp theo, AI có thể đạt được việc cắt tỉa Alpha-Beta tốt hơn, dẫn đến việc khám phá không gian tìm kiếm nhanh hơn và hiệu quả hơn. ID cũng kết nối với các kỹ thuật sắp xếp nước đi khác như nước đi Hash (được điền bởi các tìm kiếm nông hơn) và History Heuristic (tích lũy thông tin qua các lần lặp).

Quản lý thời gian hiệu quả: ID cho phép engine luôn có một "nước đi tốt nhất cho đến nay" để trả về nếu thời gian được phân bổ hết trước khi tìm kiếm đến độ sâu cuối cùng hoàn tất. Điều này đặc biệt quan trọng đối với các trò chơi có kiểm soát thời gian.

Tăng hiệu quả tìm kiếm: Đáng ngạc nhiên, tìm kiếm sâu dần đôi khi có thể nhanh hơn tìm kiếm trực tiếp đến một độ sâu cố định. Điều này chủ yếu là do việc sắp xếp nước đi được cải thiện dẫn đến việc cắt tỉa tốt hơn. Chi phí tìm kiếm lại ở các độ sâu nông hơn thường được bù đắp bởi lợi ích của việc cắt tỉa tốt hơn ở các độ sâu lớn hơn.

Hiệp lực với Bảng chuyển vị: ID giúp điền Bảng chuyển vị với kết quả từ các độ sâu tìm kiếm khác nhau, có thể được sử dụng lại trong các lần lặp tiếp theo, cải thiện hơn nữa hiệu quả.

Cung cấp chiến lược dự phòng: Trong trường hợp hết thời gian đột ngột, engine có một nước đi hợp lý từ lần lặp cuối cùng đã hoàn thành.

2.4.10 Các kỹ thuật sắp xếp nước đi

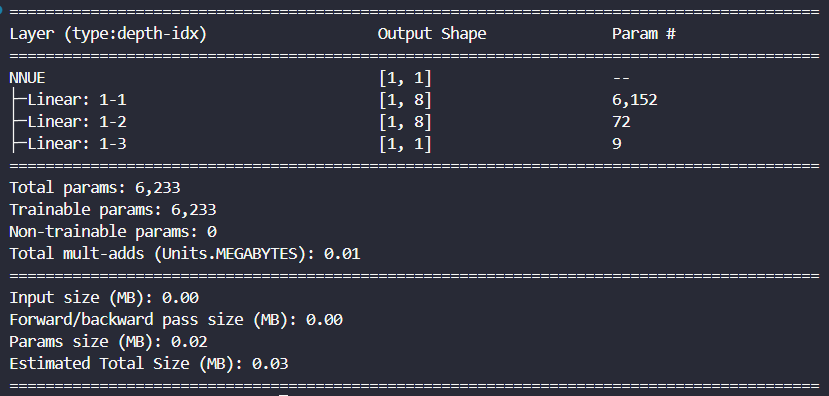
Bảng 2.3 Các kỹ thuật sắp xếp nước đi khác nhau

|  |  |  |
| --- | --- | --- |
| **Kỹ thuật** | **Mục đích chính** | **Cách cải thiện việc cắt tỉa Alpha-Beta** |
| Nước đi Hash | Ưu tiên nước đi tốt nhất từ các tìm kiếm trước đó | Tăng khả năng tìm thấy nước đi tối ưu sớm, dẫn đến việc cắt tỉa sớm hơn và hiệu quả hơn |
| Heuristic bắt quân (MVV-LVA) | Ưu tiên các nước đi bắt quân có lợi về mặt vật chất | Tìm thấy các nước đi có lợi sớm, thắt chặt các giới hạn alpha và beta để cắt tỉa hiệu quả hơn |
| Killer Heuristic | Ưu tiên các nước đi yên tĩnh gây ra ngưỡng beta trong các nút anh chị em | Tăng khả năng tìm thấy ngưỡng beta sớm ở các vị trí tương tự, dẫn đến việc cắt tỉa nhiều hơn |
| History Heuristic | Ưu tiên các nước đi không bắt quân đã dẫn đến ngưỡng beta trong quá khứ | Hướng dẫn tìm kiếm đến các nước đi có khả năng cao gây ra ngưỡng cắt tỉa dựa trên thành công lịch sử |
| Principal Variation Move Ordering | Ưu tiên các nước đi trong dòng biến chính từ các lần lặp trước | Tập trung tìm kiếm vào con đường hứa hẹn nhất, tăng tốc độ hội tụ đến nước đi tối ưu và cải thiện việc cắt tỉa |

2.5 Mô hình sử dụng NNUE để đánh giá thế cờ và Minimax với cắt tỉa Alpha-Beta để tìm nước đi tối ưu

2.5.1 NNUE

NNUE là một kiến trúc mạng nơ-ron được thiết kế đặc biệt để đánh giá các thế cờ một cách hiệu quả. Nó được biết đến với khả năng đưa ra các đánh giá chính xác một cách nhanh chóng, điều này rất quan trọng trong các engine cờ vua thời gian thực.



Hình 2.1 Tổng quan kiến trúc và tham số mô hình NNUE

Kiến trúc mạng: Mạng này bao gồm ba lớp fully connected (tuyến tính):

* fc1: Lớp đầu tiên nhận đầu vào có kích thước 768 và chuyển thành đầu ra có kích thước 8.
* fc2: Lớp thứ hai nhận đầu vào có kích thước 8 và chuyển thành đầu ra có kích thước 8.
* fc3: Lớp thứ ba nhận đầu vào có kích thước 8 và chuyển thành đầu ra có kích thước 1. Lớp này thường đại diện cho giá trị đánh giá của thế cờ (ví dụ: lợi thế cho Trắng hay Đen).

Hàm kích hoạt: Mô hình sử dụng một hàm kích hoạt tùy chỉnh có tên clipped\_relu. Hàm này tương tự như hàm ReLU (Rectified Linear Unit) nhưng giới hạn đầu ra trong khoảng từ 0 đến 1. Điều này có thể giúp ổn định quá trình huấn luyện và đảm bảo rằng các giá trị trung gian nằm trong một phạm vi nhất định.

Hàm forward: Hàm forward định nghĩa cách dữ liệu đi qua mạng. Trong trường hợp này:

1. Đầu vào x được truyền qua lớp fc1.
2. Kết quả được áp dụng hàm clipped\_relu.
3. Kết quả tiếp tục được truyền qua lớp fc2.
4. Kết quả lại được áp dụng hàm clipped\_relu.
5. Cuối cùng, kết quả được truyền qua lớp fc3 để tạo ra đầu ra cuối cùng, là giá trị đánh giá của thế cờ.

Mục đích trong AI cờ vua: NNUE được sử dụng như một hàm đánh giá tinh vi. Thay vì dựa vào các quy tắc và heuristic được lập trình thủ công để đánh giá một thế cờ (ví dụ: đếm quân, kiểm soát trung tâm), NNUE học cách đánh giá các thế cờ từ dữ liệu huấn luyện (thường là hàng triệu ván cờ tự chơi hoặc ván cờ của người chơi mạnh). Điều này cho phép AI nắm bắt các mẫu và sắc thái phức tạp trong cờ vua mà các phương pháp truyền thống có thể bỏ lỡ.

2.5.2 Thuật toán Minimax với cắt tỉa Alpha-Beta

Thuật toán Minimax là một thuật toán tìm kiếm được sử dụng trong lý thuyết trò chơi để tìm ra nước đi tối ưu cho một người chơi, giả định rằng đối thủ cũng sẽ chơi tối ưu. Cắt tỉa Alpha-Beta là một kỹ thuật tối ưu hóa giúp giảm số lượng nút mà thuật toán Minimax cần khám phá trong cây trò chơi, làm cho nó hiệu quả hơn về mặt tính toán.

Bảng 2.4 Mã giả minh hoạ hàm alphabeta

|  |
| --- |
| def alphabeta(*self*, *board*, *depth*, *alpha*, *beta*, *maximizing\_player*):      # Trường hợp cơ bản      if *depth* == 0 or *board*.is\_game\_over():          return *self*.evaluate\_board(*board*), None      if *maximizing\_player*:          max\_eval = float('-inf')          best\_move = None          for move in *board*.legal\_moves:  *board*.push(move)              eval\_score, \_ = *self*.alphabeta(*board*, *depth*-1, *alpha*, *beta*, False)  *board*.pop()              if eval\_score > max\_eval:                  max\_eval = eval\_score                  best\_move = move  *alpha* = max(*alpha*, eval\_score)              if *beta* <= *alpha*:                  break  # Cắt tỉa Beta          return max\_eval, best\_move      else:          min\_eval = float('inf')          best\_move = None          for move in *board*.legal\_moves:  *board*.push(move)              eval\_score, \_ = *self*.alphabeta(*board*, *depth*-1, *alpha*, *beta*, True)  *board*.pop()              if eval\_score < min\_eval:                  min\_eval = eval\_score                  best\_move = move  *beta* = min(*beta*, eval\_score)              if *beta* <= *alpha*:                  break  # Cắt tỉa Alpha          return min\_eval, best\_move |

2.5.3 Hàm evaluate\_board

Kiểm tra trạng thái kết thúc: Đầu tiên, nó kiểm tra xem ván cờ đã kết thúc chưa (chiếu hết, hòa). Nếu có, nó trả về một giá trị rất lớn (dương cho thắng, âm cho thua) hoặc 0 cho hòa.

Sử dụng NNUE: Nếu ván cờ chưa kết thúc, hàm sẽ gọi preprocess\_fen(board.fen()) để chuyển đổi trạng thái bàn cờ hiện tại (ở định dạng FEN) thành một tensor đầu vào mà mô hình NNUE có thể hiểu được. Sau đó, nó truyền tensor này qua mô hình NNUE (self.nnue\_3(inp)) để nhận được điểm số đánh giá.

Giá trị trả về: Hàm trả về điểm số đánh giá do mô hình NNUE cung cấp. Điểm số này biểu thị lợi thế của một bên so với bên kia.

2.5.4 Hàm preprocess\_fen

Hàm này chịu trách nhiệm chuyển đổi một chuỗi FEN (Forsyth-Edwards Notation), một ký hiệu tiêu chuẩn để mô tả một vị trí cụ thể trên bàn cờ, thành một tensor PyTorch có thể được sử dụng làm đầu vào cho mô hình NNUE.

1. Tạo đối tượng bàn cờ: Nó tạo một đối tượng chess.Board từ chuỗi FEN đầu vào.
2. Tạo tensor đặc trưng: Nó khởi tạo một tensor PyTorch có kích thước (64, 6, 2) với tất cả các giá trị là 0.

* 64 đại diện cho 64 ô trên bàn cờ.
* 6 đại diện cho 6 loại quân (Tốt, Mã, Tượng, Xe, Hậu, Vua).
* 2 đại diện cho 2 màu (Trắng và Đen).

1. Điền tensor đặc trưng: Nó lặp qua tất cả các ô trên bàn cờ. Nếu một quân cờ tồn tại trên ô đó:

* Nó xác định màu của quân cờ (color).
* Nó xác định loại quân cờ (piece\_type) và trừ đi 1 để có chỉ số từ 0 đến 5.
* Nó đặt giá trị của tensor tại vị trí tương ứng (ô, loại quân, màu) thành 1. Điều này tạo ra một biểu diễn one-hot của các quân cờ trên bàn cờ.

1. Thay đổi hình dạng tensor: Cuối cùng, nó thay đổi hình dạng của tensor từ (64, 6, 2) thành (768) bằng cách làm phẳng nó. Hình dạng này phù hợp với lớp đầu vào của mô hình NNUE (64 \* 6 \* 2 = 768).
2. Giá trị trả về: Hàm trả về tensor đặc trưng đã được tiền xử lý.

2.5.5 Hàm find\_best\_move

Hàm này là điểm vào chính để tìm nước đi tốt nhất cho trạng thái bàn cờ hiện tại.

1. Kiểm tra ván cờ kết thúc: Nó kiểm tra xem ván cờ đã kết thúc chưa. Nếu có, nó trả về None.
2. Khởi tạo: Nó khởi tạo các biến để theo dõi nước đi tốt nhất (best\_move), giá trị tốt nhất (best\_value), và các giá trị alpha và beta cho thuật toán Alpha-Beta.
3. Lặp qua các nước đi hợp lệ: Nó lặp qua tất cả các nước đi hợp lệ trên bàn cờ.
4. Gọi Alpha-Beta: Đối với mỗi nước đi, nó thực hiện nước đi đó, gọi hàm self.alphabeta để đánh giá thế cờ sau nước đi đó (với độ sâu tìm kiếm là self.depth - 1 và đổi lượt người chơi), và sau đó hoàn tác nước đi.
5. Cập nhật nước đi tốt nhất:

* Nếu đến lượt Trắng (người chơi tối đa hóa), nó sẽ cập nhật best\_move và best\_value nếu giá trị trả về từ alphabeta lớn hơn best\_value hiện tại. Nó cũng cập nhật alpha.
* Nếu đến lượt Đen (người chơi tối thiểu hóa), nó sẽ cập nhật best\_move và best\_value nếu giá trị trả về từ alphabeta nhỏ hơn best\_value hiện tại. Nó cũng cập nhật beta.

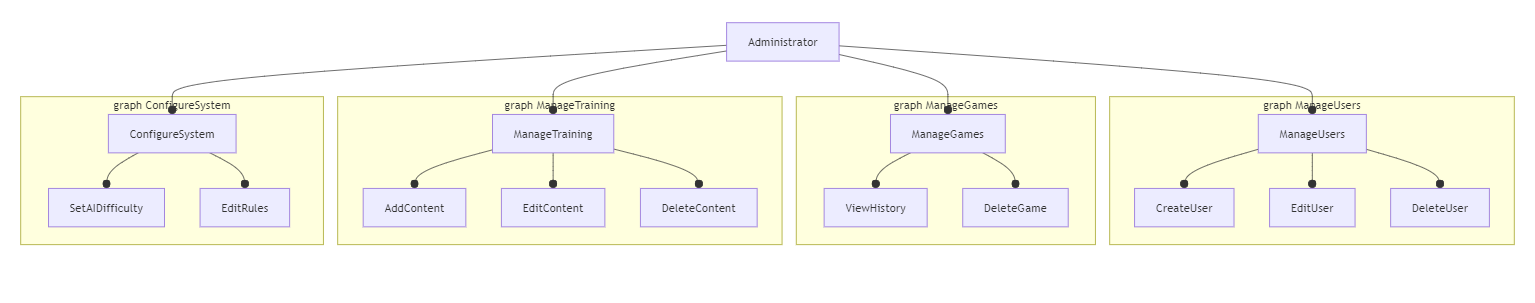
1. Giá trị trả về: Hàm trả về nước đi tốt nhất được tìm thấy.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

* 1. Đặc tả yêu cầu phần mềm
     1. Các yêu cầu chức năng
        1. Chức năng của phân hệ quản trị

Trong bối cảnh phát triển một hệ thống cờ vua thông minh toàn diện, việc bao gồm một phân hệ quản trị nội dung sẽ mang lại nhiều lợi ích. Nó sẽ cho phép quản lý hiệu quả người dùng, dữ liệu trò chơi và các tài nguyên đào tạo, nâng cao đáng kể trải nghiệm người dùng và khả năng mở rộng của hệ thống. Dựa trên các xu hướng trong phát triển trò chơi hiện đại và các hệ thống học tập trực tuyến, một phân hệ quản trị nội dung có thể bao gồm các chức năng sau:

* Quản lý Tài khoản Người dùng: Chức năng này sẽ cho phép người dùng tạo, cập nhật và xóa tài khoản của họ. Nó cũng sẽ bao gồm các tính năng như xác thực người dùng, quản lý mật khẩu và có thể là lưu trữ các tùy chọn và tiến trình của người dùng.
* Quản lý Cơ sở Dữ liệu Ván cờ: Hệ thống sẽ cần lưu trữ thông tin về các ván cờ đã chơi, bao gồm các nước đi, kết quả và có thể là phân tích ván cờ. Phân hệ này sẽ cho phép người dùng tìm kiếm, xem và xóa các ván cờ trong lịch sử của họ.
* Quản lý Nội dung Đào tạo: Nếu hệ thống có các tính năng đào tạo, phân hệ quản trị nội dung sẽ cho phép quản trị viên thêm, sửa đổi và xóa nội dung đào tạo như các bài tập chiến thuật, các bài giảng về khai cuộc, trung cuộc và tàn cuộc.
* Quản lý Cấu hình Hệ thống: Phân hệ này cung cấp giao diện cho quản trị viên để cấu hình các tham số hệ thống, như mức độ khó của AI, các biến thể cờ vua được hỗ trợ hoặc các quy tắc trò chơi khác.



Hình 3.1 Biểu đồ Ca sử dụng cho Phân hệ quản trị.

Luồng Sự kiện cho một số Ca sử dụng:

Ca sử dụng: Tạo Người dùng

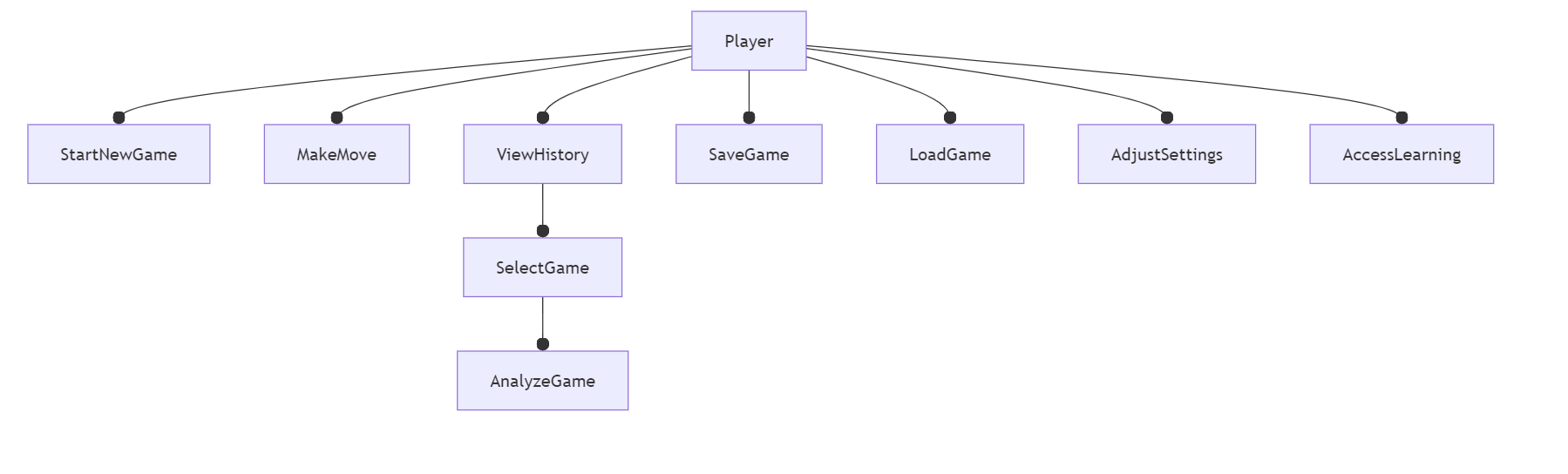
1. Quản trị viên chọn chức năng "Tạo Người dùng".
2. Hệ thống hiển thị biểu mẫu yêu cầu thông tin người dùng (tên người dùng, mật khẩu, email, v.v.).
3. Quản trị viên nhập thông tin và gửi biểu mẫu.
4. Hệ thống xác thực thông tin.
5. Nếu hợp lệ, hệ thống tạo tài khoản người dùng mới và hiển thị thông báo thành công.
6. Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu quản trị viên sửa thông tin.

Ca sử dụng: Xóa Ván cờ

1. Quản trị viên chọn chức năng "Quản lý Ván cờ".
2. Hệ thống hiển thị danh sách các ván cờ (có thể có bộ lọc hoặc chức năng tìm kiếm).
3. Quản trị viên chọn một ván cờ để xóa.
4. Hệ thống yêu cầu xác nhận xóa.
5. Nếu quản trị viên xác nhận, hệ thống sẽ xóa ván cờ khỏi cơ sở dữ liệu và hiển thị thông báo thành công.
   * + 1. Chức năng của phân hệ người dung

Phân hệ người dùng là giao diện chính để người chơi tương tác với hệ thống cờ vua thông minh. Nghiên cứu cho thấy các hệ thống cờ vua AI hiện tại cung cấp nhiều chức năng khác nhau cho người dùng. Chức năng cốt lõi là khả năng chơi cờ với AI ở các mức độ khó khác nhau. Ngoài ra, nhiều hệ thống cho phép người dùng xem lại lịch sử các ván cờ đã chơi của họ để phân tích và học hỏi. Một số hệ thống còn cung cấp các công cụ phân tích ván cờ, chẳng hạn như đánh giá nước đi hoặc gợi ý các nước đi tốt hơn. Dựa trên các chức năng phổ biến này và các khả năng tiềm năng của một hệ thống cờ vua thông minh dựa trên AI, phân hệ người dùng nên bao gồm các chức năng sau:

* Chơi cờ với AI: Người dùng có thể bắt đầu một ván cờ mới với AI. Hệ thống sẽ cho phép người dùng chọn mức độ khó của AI, có thể từ người mới bắt đầu đến nâng cao. Hệ thống cũng sẽ quản lý lượt đi, thực hiện các nước đi của người dùng và AI, và xác định kết quả của ván cờ.
* Xem lại Lịch sử Ván cờ: Người dùng có thể truy cập danh sách các ván cờ đã chơi của họ. Họ có thể chọn một ván cờ từ lịch sử để xem lại các nước đi đã thực hiện.
* Phân tích Ván cờ: Người dùng có thể chọn một ván cờ (từ lịch sử hoặc một ván cờ đang diễn ra) và sử dụng các công cụ để phân tích nó. Điều này có thể bao gồm việc xem xét đánh giá của AI về các vị trí khác nhau, nhận gợi ý về các nước đi tiềm năng hoặc khám phá các biến thể khác nhau.
* Lưu và Tải Ván cờ: Người dùng có thể có tùy chọn lưu một ván cờ đang diễn ra và tải nó sau này để tiếp tục hoặc phân tích.
* Cài đặt: Người dùng có thể điều chỉnh các cài đặt liên quan đến giao diện (ví dụ: chủ đề bàn cờ, kiểu quân cờ) hoặc hành vi của AI (ví dụ: thời gian suy nghĩ cho mỗi nước đi).
* Truy cập Tài liệu Học tập: Nếu hệ thống tích hợp các tài liệu học tập, người dùng có thể truy cập chúng thông qua phân hệ người dùng.



Hình 3.2 Biểu đồ Ca sử dụng cho Phân hệ Người dùng

Luồng Sự kiện cho một số Ca sử dụng:

Ca sử dụng: Bắt đầu Ván cờ Mới

1. Người chơi chọn tùy chọn "Ván cờ Mới".
2. Hệ thống hiển thị các tùy chọn cho ván cờ mới (ví dụ: mức độ khó của AI, màu quân).
3. Người chơi chọn các tùy chọn mong muốn.
4. Hệ thống khởi tạo một ván cờ mới với AI theo các tùy chọn đã chọn và hiển thị bàn cờ ở vị trí bắt đầu.

Ca sử dụng: Thực hiện Nước đi

1. Người chơi chọn một quân cờ trên bàn cờ.
2. Hệ thống làm nổi bật các ô hợp lệ mà quân cờ có thể di chuyển đến.
3. Người chơi chọn ô đích.
4. Hệ thống kiểm tra xem nước đi có hợp lệ không theo luật cờ vua.
5. Nếu hợp lệ, hệ thống sẽ di chuyển quân cờ đến ô đích và cập nhật trạng thái ván cờ.
6. Hệ thống sau đó chuyển lượt cho AI.

Ca sử dụng: Xem Lịch sử

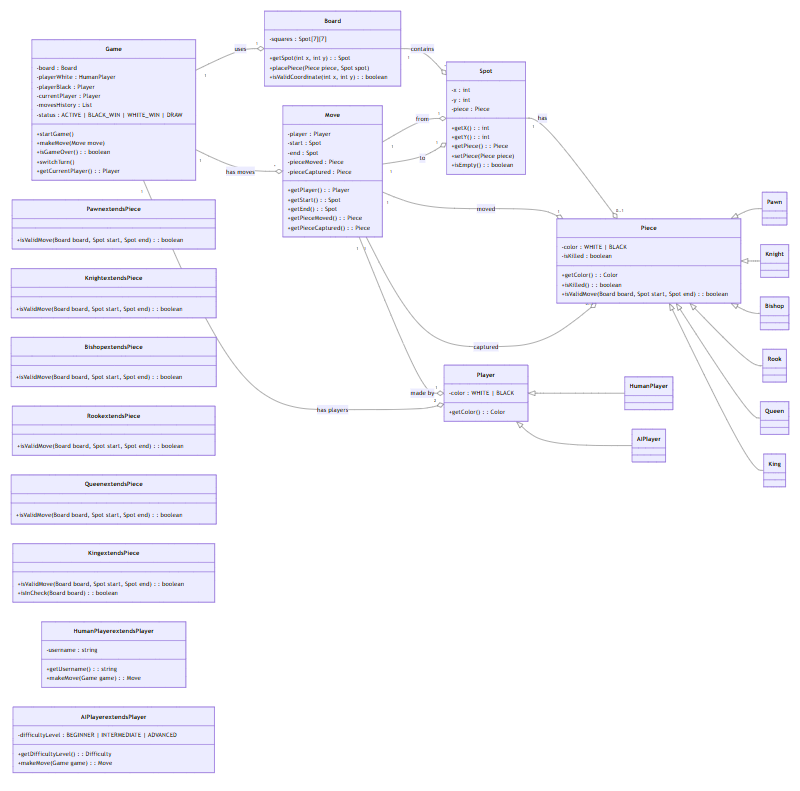
1. Người chơi chọn tùy chọn "Lịch sử".
2. Hệ thống hiển thị danh sách các ván cờ đã chơi của người chơi, có thể sắp xếp theo ngày, đối thủ hoặc kết quả.
3. Người chơi có thể chọn một ván cờ từ danh sách để xem lại (ca sử dụng: Chọn Ván cờ).

Ca sử dụng: Chọn Ván cờ

1. (Tiếp theo từ ca sử dụng "Xem Lịch sử") Người chơi chọn một ván cờ từ danh sách.
2. Hệ thống hiển thị bàn cờ ở vị trí bắt đầu của ván cờ đã chọn.
3. Người chơi có thể sử dụng các nút điều khiển (ví dụ: "Nước tiếp theo", "Nước trước") để xem lại các nước đi của ván cờ.
4. Người chơi có thể có tùy chọn để phân tích ván cờ đã chọn (ca sử dụng: Phân tích Ván cờ).
   * 1. Biểu đồ lớp thực thể

Dựa trên các yêu cầu chức năng đã xác định, các lớp thực thể chính trong hệ thống cờ vua thông minh này bao gồm:

* Board: Đại diện cho bàn cờ 8x8. Nó sẽ chứa một mảng hoặc cấu trúc dữ liệu khác để theo dõi vị trí của các quân cờ trên mỗi ô.
* Piece: Một lớp trừu tượng đại diện cho một quân cờ. Các lớp cụ thể sẽ kế thừa từ lớp này cho từng loại quân cờ (Pawn, Knight, Bishop, Rook, Queen, King). Lớp này sẽ chứa các thuộc tính chung như màu sắc và trạng thái (còn sống hay đã bị bắt).
* Pawn, Knight, Bishop, Rook, Queen, King: Các lớp cụ thể đại diện cho từng loại quân cờ, kế thừa từ lớp Piece. Mỗi lớp sẽ triển khai các quy tắc di chuyển cụ thể cho loại quân cờ đó.
* Player: Một lớp trừu tượng đại diện cho một người chơi. Các lớp cụ thể sẽ kế thừa từ lớp này cho người chơi là con người (HumanPlayer) và người chơi AI (AIPlayer).
* HumanPlayer: Một lớp cụ thể đại diện cho một người chơi là con người, có thể chứa thông tin như tên người dùng.
* AIPlayer: Một lớp cụ thể đại diện cho người chơi AI. Nó có thể chứa các thuộc tính liên quan đến mức độ khó hoặc thuật toán AI được sử dụng.
* Move: Đại diện cho một nước đi trong ván cờ. Nó sẽ chứa thông tin về quân cờ nào đã di chuyển, từ ô nào đến ô nào, và liệu có quân cờ nào bị bắt hay không.
* Game: Đại diện cho một ván cờ. Nó sẽ chứa một thể hiện của Board, hai thể hiện của Player (một HumanPlayer và một AIPlayer), người chơi hiện tại đang đến lượt, và có thể là lịch sử các nước đi đã thực hiện.
* Spot: Đại diện cho một ô trên bàn cờ. Nó có thể chứa một tham chiếu đến quân cờ đang ở trên ô đó.



Hình 3.3 Biểu đồ Lớp Thực thể

* + 1. Các yêu cầu phi chức năng

Hiệu suất:

* Yêu cầu: Thời gian phản hồi cho mỗi nước đi của AI phải nằm trong giới hạn chấp nhận được để duy trì sự tương tác của người chơi. Yêu cầu này có thể khác nhau tùy thuộc vào mức độ khó của AI (ví dụ: < 1 giây cho người mới bắt đầu, < 5 giây cho trung cấp, < 30 giây cho nâng cao).
* Mô tả: Người chơi mong đợi AI phản hồi một cách hợp lý trong thời gian ngắn. Thời gian phản hồi quá dài có thể dẫn đến trải nghiệm người dùng kém.

Độ tin cậy:

* Yêu cầu: Hệ thống phải hoạt động ổn định mà không gặp sự cố hoặc lỗi trong quá trình chơi game. Trạng thái của ván cờ phải được duy trì nhất quán.
* Mô tả: Độ tin cậy đảm bảo rằng người chơi có thể tin tưởng vào hệ thống để chơi game mà không bị gián đoạn hoặc mất dữ liệu.

Khả năng sử dụng:

* Yêu cầu: Giao diện người dùng phải trực quan và dễ điều hướng. Bàn cờ và quá trình chọn nước đi phải rõ ràng và đơn giản.
* Mô tả: Khả năng sử dụng tốt sẽ làm cho hệ thống dễ tiếp cận với nhiều đối tượng người chơi, từ người mới bắt đầu đến người chơi có kinh nghiệm.

Bảo mật:

* Yêu cầu: Nếu hệ thống lưu trữ thông tin người dùng (ví dụ: tài khoản, lịch sử trò chơi), dữ liệu này phải được bảo vệ khỏi truy cập trái phép. Các cơ chế xác thực an toàn phải được triển khai.
* Mô tả: Bảo mật là rất quan trọng để bảo vệ quyền riêng tư và dữ liệu của người dùng.

Khả năng mở rộng:

* Yêu cầu: Hệ thống nên có khả năng xử lý một số lượng người dùng đồng thời hợp lý mà không làm giảm đáng kể hiệu suất (nếu là ứng dụng mạng). Động cơ AI nên đủ hiệu quả để chạy trên phần cứng tiêu dùng tiêu chuẩn.
* Mô tả: Khả năng mở rộng đảm bảo rằng hệ thống có thể đáp ứng sự tăng trưởng về số lượng người dùng và duy trì hiệu suất chấp nhận được trong các điều kiện tải khác nhau.

Khả năng bảo trì:

* Yêu cầu: Mã nguồn của hệ thống phải được cấu trúc và ghi chú rõ ràng để dễ dàng bảo trì, sửa lỗi và nâng cấp trong tương lai.
* Mô tả: Khả năng bảo trì tốt giúp giảm chi phí và công sức cần thiết để cập nhật và hỗ trợ hệ thống theo thời gian.
  1. Thiết kế hệ thống
     1. Thiết kế kiến trúc

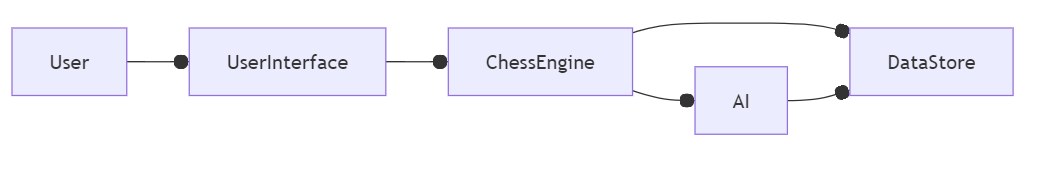
Dựa trên các yêu cầu đã xác định, một kiến trúc ba lớp (three-layer architecture) hoặc kiến trúc nhiều lớp (layered architecture) sẽ phù hợp cho hệ thống cờ vua thông minh này. Kiến trúc này giúp tách biệt các mối quan tâm khác nhau của hệ thống thành các lớp riêng biệt, làm cho hệ thống dễ hiểu, phát triển và bảo trì hơn. Các lớp chính trong kiến trúc này sẽ là:

Lớp Giao diện Người dùng (Presentation Layer): Lớp này chịu trách nhiệm tương tác với người dùng. Nó bao gồm các thành phần giao diện như bàn cờ hiển thị, khu vực hiển thị thông tin (ví dụ: lượt đi, kết quả), và các nút điều khiển (ví dụ: bắt đầu ván cờ mới, thực hiện nước đi, xem lịch sử). Lớp này sẽ nhận đầu vào của người dùng (ví dụ: chọn nước đi) và hiển thị thông tin từ lớp logic nghiệp vụ.

Lớp Logic Nghiệp vụ / Engine Cờ vua (Business Logic Layer / Chess Engine): Lớp này chứa logic cốt lõi của trò chơi cờ vua. Nó bao gồm các quy tắc của cờ vua, xác thực nước đi, quản lý trạng thái ván cờ (ví dụ: vị trí các quân cờ, lượt đi), và xác định kết quả của ván cờ (ví dụ: chiếu hết, hòa). Lớp này sẽ nhận các yêu cầu từ lớp giao diện người dùng, xử lý chúng và cập nhật trạng thái ván cờ.

Lớp AI (AI Layer): Lớp này chịu trách nhiệm triển khai các thuật toán AI để xác định nước đi tốt nhất cho máy tính. Nó sẽ nhận trạng thái hiện tại của ván cờ từ lớp logic nghiệp vụ và trả về nước đi được AI lựa chọn. Các thuật toán AI có thể bao gồm Minimax với cắt tỉa Alpha-Beta hoặc các phương pháp tiên tiến hơn như Tìm kiếm Cây Monte Carlo hoặc mạng nơ-ron.

Lớp Lưu trữ Dữ liệu (Data Access Layer): Lớp này chịu trách nhiệm quản lý việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Nó sẽ cung cấp các phương thức để lưu trữ thông tin người dùng, lịch sử các ván cờ đã chơi, và có thể là các cấu hình hệ thống. Lớp logic nghiệp vụ và lớp AI (cho mục đích học máy trong các hệ thống phức tạp hơn) có thể tương tác với lớp này.



Hình 3.4 Sơ đồ Kiến trúc Hệ thống

Trong kiến trúc này, mỗi lớp có một trách nhiệm riêng biệt và chỉ tương tác với các lớp liền kề. Ví dụ, lớp giao diện người dùng không trực tiếp tương tác với lớp AI hoặc lớp lưu trữ dữ liệu. Thay vào đó, nó gửi yêu cầu đến lớp logic nghiệp vụ, lớp này sau đó điều phối các hoạt động cần thiết với các lớp khác. Điều này giúp tăng tính mô-đun, khả năng bảo trì và khả năng kiểm thử của hệ thống.

* + 1. Thiết kê cơ sở dữ liệu

**(i) Mô hình Cơ sở Dữ liệu Quan hệ:**

Dựa trên biểu đồ lớp thực thể và giả định rằng hệ thống cần lưu trữ thông tin về người dùng và lịch sử các ván cờ, mô hình cơ sở dữ liệu quan hệ có thể bao gồm các bảng sau:

* **Users:** Lưu trữ thông tin về người dùng đã đăng ký.
* **Games:** Lưu trữ thông tin về các ván cờ đã chơi.
* **Moves:** Lưu trữ chi tiết về từng nước đi trong mỗi ván cờ.

**(ii) Mô tả Cấu trúc Bảng:**

Bảng 3.1 Bảng Users

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| UserID | (INT, PRIMARY KEY, AUTO\_INCREMENT) | Mã định danh duy nhất cho mỗi người dùng. |
| Username | (VARCHAR(255), UNIQUE, NOT NULL) | Tên người dùng. |
| PasswordHash | (VARCHAR(255), NOT NULL) | Hash của mật khẩu người dùng. |
| Email | (VARCHAR(255), UNIQUE) | Địa chỉ email của người dùng. |
| RegistrationDate | (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP) | Ngày đăng ký của người dùng. |

Bảng 3.2 Bảng Games

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| GameID | (INT, PRIMARY KEY, AUTO\_INCREMENT) | Mã định danh duy nhất cho mỗi ván cờ. |
| Player1ID | (INT, FOREIGN KEY REFERENCES Users(UserID)) | Mã của người chơi thứ nhất (có thể là người dùng). |
| Player2ID | (INT, FOREIGN KEY REFERENCES Users(UserID), NULLABLE) | Mã của người chơi thứ hai (có thể là người dùng hoặc NULL nếu là AI). |
| AILevel | (VARCHAR(50)) | Mức độ khó của AI trong ván cờ (nếu có). |
| StartTime | (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP) | Thời điểm bắt đầu ván cờ. |
| EndTime | (TIMESTAMP) | Thời điểm kết thúc ván cờ. |
| Result | (VARCHAR(50)) | Kết quả của ván cờ (ví dụ: White wins, Black wins, Draw). |
| FEN | (TEXT) | Chuỗi FEN đại diện cho trạng thái cuối cùng của ván cờ. |
| PGN | (TEXT) | Chuỗi PGN chứa toàn bộ các nước đi của ván cờ (tùy chọn). |

Bảng 3.3 Bảng Moves

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| MoveID | (INT, PRIMARY KEY, AUTO\_INCREMENT) | Mã định danh duy nhất cho mỗi nước đi. |
| GameID | (INT, FOREIGN KEY REFERENCES Games(GameID), NOT NULL) | Mã của ván cờ mà nước đi thuộc về. |
| MoveNumber | (INT, NOT NULL) | Số thứ tự của nước đi trong ván cờ. |
| PlayerID | (INT, FOREIGN KEY REFERENCES Users(UserID), NULLABLE) | Mã của người chơi thực hiện nước đi (NULL nếu là AI). |
| FromSquare | (VARCHAR(2), NOT NULL) | Ô xuất phát của nước đi (ví dụ: "e2"). |
| ToSquare | (VARCHAR(2), NOT NULL) | Ô đích của nước đi (ví dụ: "e4"). |
| Piece | (VARCHAR(50), NOT NULL) | Loại quân cờ đã di chuyển (ví dụ: "Pawn", "Knight"). |
| CapturedPiece | (VARCHAR(50)) | Loại quân cờ bị bắt (nếu có). |
| MoveTime | (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP) | Thời điểm thực hiện nước đi. |

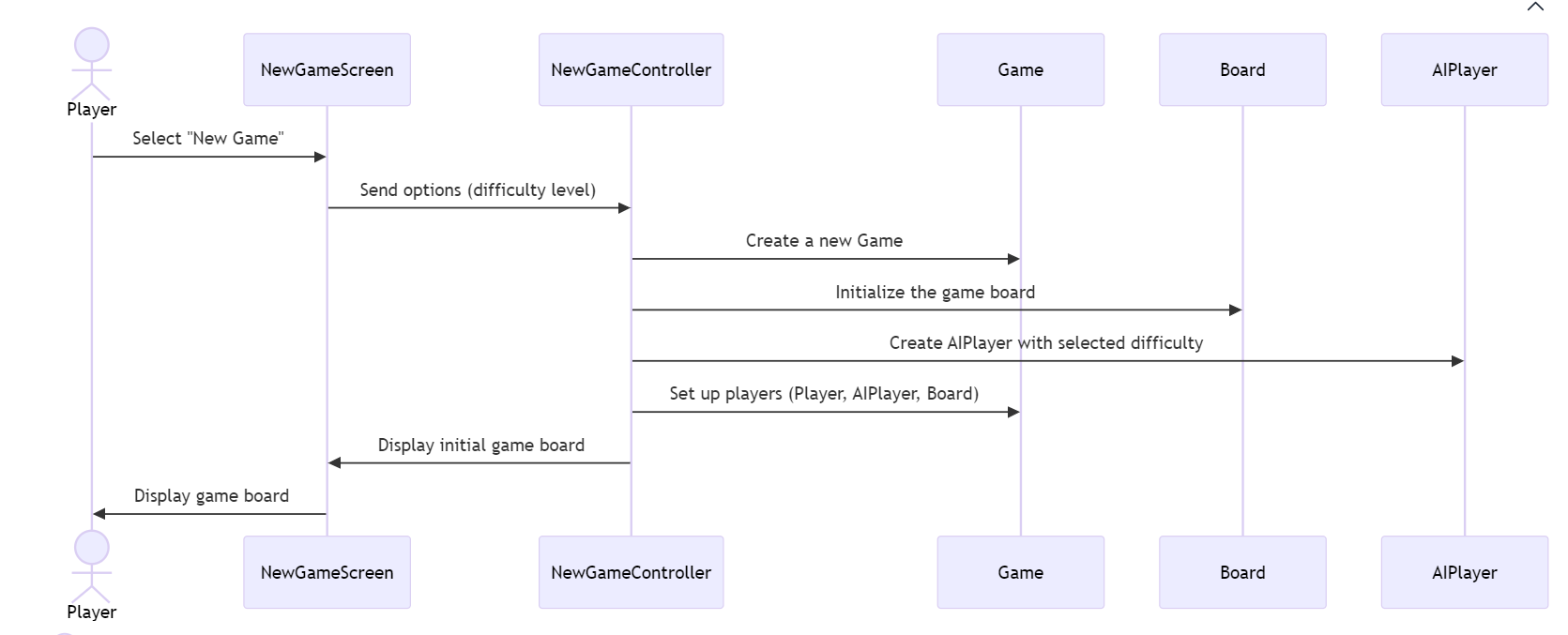
* + 1. Thiết kế lớp đối tượng
       1. Biểu đồ lớp VOPC của các ca sử dụng

Ca sử dụng: Bắt đầu Ván cờ Mới

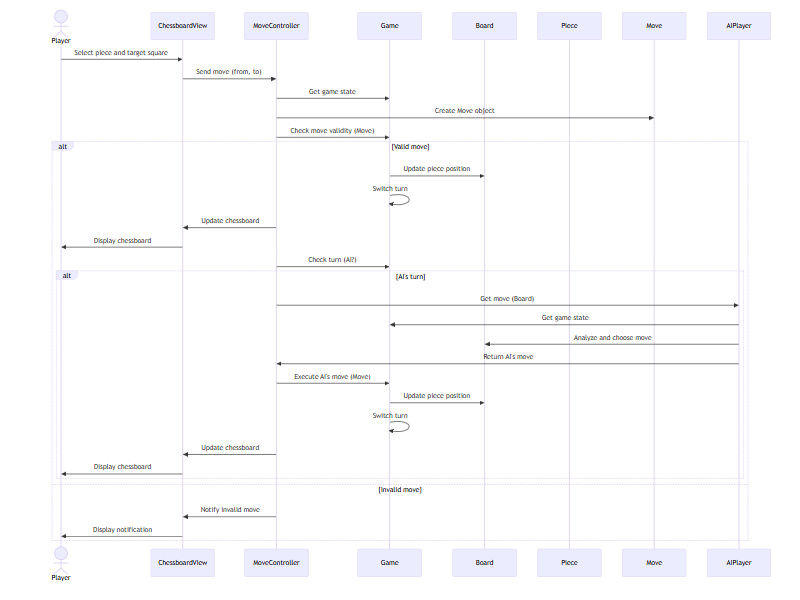
* **View (V):** NewGameScreen - Hiển thị các tùy chọn cho ván cờ mới (mức độ khó, màu quân).
* **Object/Entity (O):** Game - Thể hiện một ván cờ mới. AIPlayer - Thể hiện người chơi AI với mức độ khó đã chọn. Board - Thể hiện bàn cờ ở trạng thái ban đầu.
* **Process (P):** InitializeGameProcess - Chứa logic để tạo một thể hiện Game mới, khởi tạo Board và tạo AIPlayer với mức độ khó đã chọn.
* **Control (C):** NewGameController - Xử lý tương tác của người dùng trên NewGameScreen, thu thập tùy chọn và kích hoạt InitializeGameProcess.

**Ca sử dụng: Thực hiện Nước đi**

* **View (V):** ChessboardView - Hiển thị trạng thái hiện tại của bàn cờ và cho phép người dùng chọn quân cờ và ô di chuyển.
* **Object/Entity (O):** Game - Thể hiện trạng thái hiện tại của ván cờ. Board - Thể hiện bàn cờ hiện tại. Piece - Quân cờ được người dùng chọn. Move - Nước đi mà người dùng muốn thực hiện.
* **Process (P):** ValidateMoveProcess - Chứa logic để kiểm tra xem nước đi của người dùng có hợp lệ theo luật cờ vua không. UpdateGameStateProcess - Chứa logic để cập nhật trạng thái Board và Game sau một nước đi hợp lệ, và kích hoạt lượt đi của AI nếu cần. AIMoveProcess - Chứa logic để AI chọn và thực hiện nước đi của nó.
* **Control (C):** MoveController - Xử lý tương tác của người dùng trên ChessboardView, gọi ValidateMoveProcess, UpdateGameStateProcess và AIMoveProcess.
  + - 1. Biểu đồ tuần tự

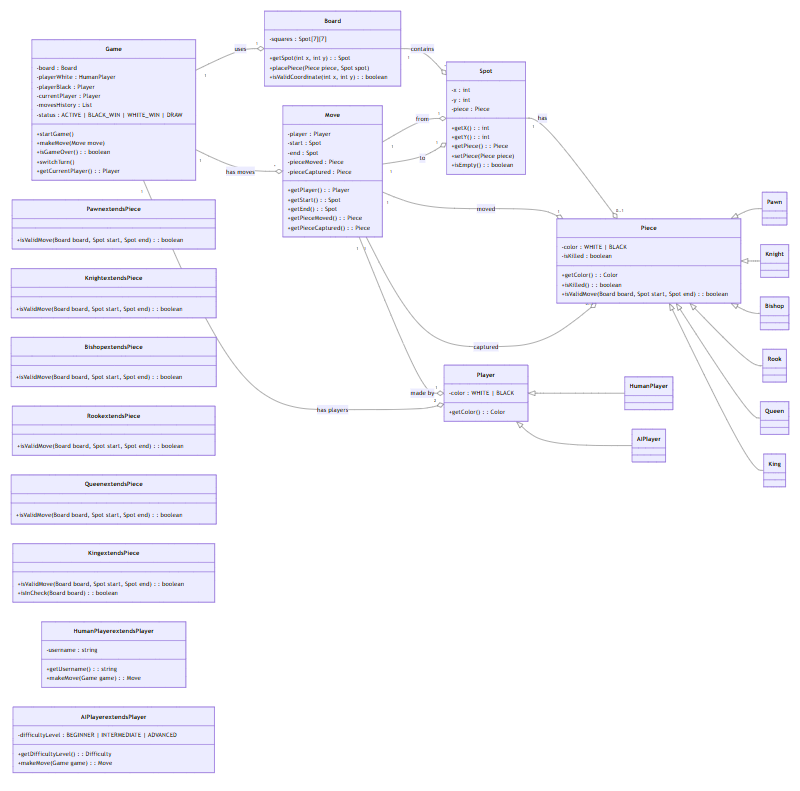


Hình 3.5 Ca sử dụng: Bắt đầu Ván cờ Mới



Hình 3.6 Ca sử dụng: Thực hiện Nước đi

* + - 1. Biểu đồ lớp chi tiết



Hình 3.7 Biểu đồ lớp chi tiết

* + 1. Thiết kế giao diện

Giao diện người dùng của hệ thống cờ vua thông minh sẽ được thiết kế để trực quan và dễ sử dụng cho người chơi ở mọi trình độ. Các thành phần giao diện chính sẽ bao gồm:

* **Bàn cờ:** Bàn cờ 8x8 sẽ là thành phần trung tâm của giao diện. Nó sẽ hiển thị vị trí hiện tại của tất cả các quân cờ. Người dùng sẽ tương tác với bàn cờ để chọn quân cờ và thực hiện nước đi. Bàn cờ nên có khả năng làm nổi bật các ô hợp lệ mà một quân cờ có thể di chuyển đến sau khi người dùng chọn quân cờ đó.
* **Khu vực Hiển thị Thông tin:** Khu vực này sẽ hiển thị các thông tin quan trọng về ván cờ hiện tại, chẳng hạn như:
  + Người chơi nào đang đến lượt.
  + Kết quả của ván cờ (ví dụ: chiếu hết, hòa).
  + Lịch sử các nước đi đã được thực hiện trong ván cờ, có thể hiển thị theo dạng danh sách hoặc ký hiệu đại số.
  + Thông tin về các quân cờ đã bị bắt.
  + (Tùy chọn) Đánh giá vị trí hiện tại của ván cờ theo quan điểm của AI.
* **Nút Điều khiển:** Các nút điều khiển sẽ cho phép người dùng thực hiện các hành động khác nhau, bao gồm:
  + **Ván cờ Mới:** Bắt đầu một ván cờ mới với AI. Nút này có thể mở ra một hộp thoại cho phép người dùng chọn mức độ khó của AI và màu quân mà họ muốn chơi.
  + **Thực hiện Nước đi:** (Có thể không cần nút riêng biệt, nước đi được thực hiện khi người dùng chọn ô đích).
  + **Xem Lịch sử:** Mở một màn hình hoặc khu vực hiển thị lịch sử các ván cờ đã chơi của người dùng.
  + **Phân tích Ván cờ:** (Chỉ khả dụng khi đang xem lại một ván cờ) Kích hoạt chức năng phân tích ván cờ của AI cho vị trí hiện tại.
  + **Lưu Ván cờ:** Lưu ván cờ hiện tại để chơi tiếp sau này.
  + **Tải Ván cờ:** Tải một ván cờ đã lưu trước đó.
  + **Cài đặt:** Mở một màn hình cho phép người dùng điều chỉnh các cài đặt giao diện và AI.
  + **Undo/Redo:** Cho phép người dùng hoàn tác hoặc làm lại nước đi cuối cùng (có thể có giới hạn về số lần).

**Luồng Điều hướng giữa các Màn hình chính:**

1. **Màn hình Bắt đầu:** Hiển thị khi ứng dụng khởi động. Có thể bao gồm các tùy chọn như "Chơi với AI", "Xem Lịch sử", "Cài đặt".
2. **Màn hình Ván cờ:** Hiển thị bàn cờ và các thành phần liên quan trong khi chơi game.
3. **Màn hình Lịch sử Ván cờ:** Hiển thị danh sách các ván cờ đã chơi. Người dùng có thể chọn một ván cờ để xem lại.
4. **Màn hình Xem lại Ván cờ:** Hiển thị bàn cờ ở trạng thái của một ván cờ đã chọn từ lịch sử, cho phép người dùng xem lại các nước đi.
5. **Màn hình Cài đặt:** Cho phép người dùng tùy chỉnh các tùy chọn giao diện và AI.
6. **Màn hình Kết thúc Ván cờ:** Hiển thị khi ván cờ kết thúc, thông báo kết quả và có thể cung cấp các tùy chọn như chơi lại hoặc phân tích ván cờ.

# TRIỂN KHAI WEBSITE

* 1. Xây dựng Web với Gradio

Gradio cung cấp một tập hợp phong phú các thành phần dựng sẵn có thể được sử dụng làm khối xây dựng để tạo giao diện web tương tác cho các ứng dụng AI. Các thành phần này trừu tượng hóa sự phức tạp của phát triển web, cho phép các nhà phát triển tập trung vào logic AI. Đối với đầu vào, Gradio cung cấp các thành phần như Textbox để nhập văn bản , Slider để nhập số trong một phạm vi , Checkbox và CheckboxGroup cho các tùy chọn boolean và đa lựa chọn , Dropdown để chọn từ danh sách các tùy chọn và các thành phần để xử lý phương tiện như Image, Audio và File.

Đối với đầu ra, Gradio cung cấp các thành phần như Textbox để hiển thị văn bản , Image để hiển thị hình ảnh , Label để hiển thị nhãn phân loại , Gallery để hiển thị nhiều hình ảnh , Plot để hiển thị các loại biểu đồ khác nhau , Audio và Video để hiển thị phương tiện và HTML để hiển thị nội dung HTML tùy ý. Gradio cũng cung cấp các thành phần bố cục như Row và Column để sắp xếp các phần tử giao diện theo chiều ngang và chiều dọc, cũng như Tabs để tổ chức nội dung thành các phần khác nhau. Thành phần gr.State cho phép duy trì trạng thái giữa các tương tác của người dùng. Bộ thành phần phong phú của Gradio cho phép tạo ra các giao diện web trực quan và tương tác cho AI trong trò chơi, cho phép người dùng dễ dàng tương tác với trò chơi và đối thủ AI.

Đối với một trò chơi cờ vua, chúng ta có thể cần một thành phần Image để hiển thị bàn cờ, một Textbox để người dùng nhập nước đi và một thành phần Textbox hoặc Markdown khác để hiển thị trạng thái trò chơi và các nước đi của AI. Các thành phần bố cục như Row và Column có thể giúp tổ chức các phần tử này trên trang.

* 1. Xây dựng các chức năng
     1. Thiết kế giao diện người dùng cho ứng dụng AI trong trò chơi

Một yếu tố quan trọng của giao diện trò chơi cờ vua sẽ là một biểu diễn trực quan của bàn cờ. Điều này có thể đạt được bằng cách sử dụng thành phần gr.Image, với bàn cờ thực tế được hiển thị dưới dạng hình ảnh SVG bằng hàm chess.svg.board() của thư viện python-chess. Điều này cung cấp một biểu diễn trực quan rõ ràng và có khả năng mở rộng về trạng thái của trò chơi.

Đối với đầu vào của người dùng, một gr.Textbox có nhãn "Nước đi của bạn (ví dụ: e2e4)" có thể cho phép người chơi nhập các nước đi của họ theo ký hiệu đại số. Ngoài ra, người ta có thể khám phá việc sử dụng một phương pháp trực quan hơn với các ô có thể nhấp trên bàn cờ, mặc dù điều này có thể đòi hỏi việc triển khai tùy chỉnh nhiều hơn. Một thành phần gr.Textbox hoặc gr.Markdown riêng biệt có thể được sử dụng để hiển thị trạng thái trò chơi, chẳng hạn như đến lượt ai đi, trạng thái chiếu và bất kỳ thông báo nào về nước đi của AI. Để kiểm soát sức mạnh của AI, một thành phần gr.Slider hoặc gr.Dropdown có thể cho phép người dùng chọn độ sâu tìm kiếm cho thuật toán Minimax, do đó điều chỉnh độ khó của đối thủ.

Thiết kế giao diện người dùng nên ưu tiên sự rõ ràng và dễ sử dụng, giúp người chơi dễ dàng hiểu trạng thái trò chơi và tương tác với đối thủ AI. Một bố cục rõ ràng với màn hình hiển thị bàn cờ nổi bật, phương pháp nhập nước đi đơn giản và phản hồi rõ ràng về trạng thái trò chơi góp phần tạo nên trải nghiệm người dùng tích cực. Việc cung cấp các tùy chọn để điều chỉnh độ khó của AI có thể phục vụ cho các trình độ kỹ năng khác nhau.

* + 1. Xử lý đầu vào của người dùng và hiển thị trạng thái trò chơi

Khi người dùng nhập một nước đi vào gr.Textbox và gửi nó (ví dụ: bằng cách nhấn Enter hoặc nhấp vào nút), một trình lắng nghe sự kiện Gradio (sử dụng phương thức .change() hoặc .click()) có thể kích hoạt một hàm Python để xử lý đầu vào này. Hàm này sẽ nhận nước đi của người dùng dưới dạng chuỗi, xác thực nó bằng thư viện python-chess (ví dụ: board.parse\_san() và board.is\_legal() từ ) và nếu hợp lệ, áp dụng nước đi đó vào đối tượng chess.Board nội bộ (board.push() từ ). Sau nước đi của người dùng (hoặc nước đi của AI, như sẽ thảo luận ở điểm tiếp theo), trạng thái đã cập nhật của đối tượng chess.Board có thể được sử dụng để tạo ra một biểu diễn SVG mới của bàn cờ bằng cách sử dụng chess.svg.board(board=current\_board). Chuỗi SVG này sau đó có thể được truyền vào tham số value của thành phần gr.Image để cập nhật hiển thị trực quan trong giao diện web.

Các thông báo trạng thái trò chơi, chẳng hạn như "Đến lượt Trắng đi" hoặc "Chiếu hết!", có thể được tạo động dựa trên trạng thái của đối tượng chess.Board (ví dụ: sử dụng board.turn từ và board.is\_checkmate() từ ) và được cập nhật trong thành phần gr.Markdown. Bản chất phản ứng của Gradio, với các trình lắng nghe sự kiện và khả năng cập nhật động các giá trị thành phần, cho phép tạo ra trải nghiệm trò chơi cờ vua thời gian thực và tương tác trong trình duyệt web. Khi người dùng thực hiện một nước đi, giao diện người dùng sẽ ngay lập tức cập nhật để phản ánh thay đổi này. Tương tự, khi AI thực hiện một nước đi, bàn cờ sẽ cập nhật lại. Các cơ chế xử lý sự kiện và cập nhật thành phần của Gradio cho phép tương tác liền mạch này.

* + 1. **Tích hợp logic trò chơi và thuật toán AI**

Lượt đi của AI có thể được kích hoạt sau khi một nước đi hợp lệ của người dùng được thực hiện và áp dụng cho đối tượng chess.Board. Một trình lắng nghe sự kiện (có lẽ trên nút được sử dụng để gửi nước đi của người dùng) sau đó có thể gọi một hàm Python triển khai thuật toán Minimax. Hàm này sẽ nhận đối tượng chess.Board hiện tại làm đầu vào, xác định nước đi tốt nhất của AI và trả về nước đi này (có khả năng ở định dạng SAN). Nước đi của AI, được trả về bởi hàm Minimax, sau đó có thể được áp dụng cho đối tượng chess.Board bằng cách sử dụng board.push\_san(). Sau nước đi của AI, trạng thái bàn cờ lại được cập nhật trong thành phần gr.Image, và trạng thái trò chơi (ví dụ: đến lượt ai đi) được cập nhật trong thành phần gr.Markdown. Độ sâu tìm kiếm cho thuật toán Minimax, được điều khiển bởi thành phần gr.Slider hoặc gr.Dropdown, có thể được truyền dưới dạng đối số cho hàm AI, cho phép người dùng ảnh hưởng đến sức mạnh chơi của AI.

Khả năng kết nối các sự kiện UI với các hàm Python của Gradio cho phép tích hợp rõ ràng và mô-đun hóa logic trò chơi (được xử lý bởi thư viện cờ vua Python) và việc ra quyết định của AI (được triển khai bằng Minimax) trong một ứng dụng web. Người dùng tương tác với trò chơi thông qua giao diện Gradio. Nước đi của họ kích hoạt logic trò chơi. Sau khi nước đi của người dùng được xử lý, hàm AI được gọi, hàm này sử dụng logic trò chơi để xác định phản hồi tốt nhất của nó. Phản hồi này sau đó được áp dụng bằng cách sử dụng logic trò chơi và UI được cập nhật để phản ánh nước đi của AI, tạo ra một trò chơi tương tác theo lượt.

* 1. Kiểm thử và triển khai ứng dụng

Kiểm thử đơn vị: Các thành phần riêng lẻ của logic trò chơi, chẳng hạn như việc tạo và xác thực nước đi trong thư viện python-chess, có thể được kiểm thử bằng mô-đun unittest tích hợp của Python hoặc một framework giàu tính năng hơn như pytest. Ví dụ: người ta có thể viết các bài kiểm thử để đảm bảo rằng board.legal\_moves xác định chính xác tất cả các nước đi hợp lệ từ nhiều vị trí khác nhau. Tương tự, hàm đánh giá được sử dụng trong thuật toán Minimax có thể được kiểm thử với các vị trí cờ vua đã biết để xác minh độ chính xác của nó trong việc gán điểm số.

Kiểm thử tích hợp: Kiểm thử tích hợp sẽ tập trung vào việc xác minh rằng thư viện python-chess và thuật toán Minimax hoạt động cùng nhau như mong đợi. Ví dụ: người ta có thể kiểm thử các tình huống trong đó AI xác định và thực hiện chính xác một nước đi để tránh bị chiếu hết hoặc để bắt một quân cờ của đối phương.

Đánh giá sức mạnh AI: Sức mạnh chơi tổng thể của AI có thể được đánh giá bằng cách cho nó chơi với chính nó hoặc với các công cụ cờ vua khác có xếp hạng Elo đã biết. Các công cụ và phương pháp tồn tại để ước tính xếp hạng Elo của một công cụ cờ vua dựa trên hiệu suất của nó trong một loạt các trò chơi.

Kiểm thử đầu cuối (E2E): Kiểm thử đầu cuối mô phỏng toàn bộ quy trình làm việc của người dùng, từ tương tác với các thành phần UI trong trình duyệt đến xác minh kết quả cuối cùng. Các công cụ như Playwright (được đề cập trong ) hoặc Selenium có thể được sử dụng để tự động hóa các tương tác của trình duyệt và kiểm thử ứng dụng từ quan điểm của người dùng. Ví dụ: một bài kiểm thử E2E có thể bao gồm việc người dùng thực hiện một loạt các nước đi và xác minh rằng bàn cờ cập nhật chính xác và AI đưa ra phản hồi hợp lệ.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

Trong đồ án này, chúng tôi đã xây dựng thành công một mô hình AI sử dụng thuật toán Minimax với kỹ thuật cắt tỉa Alpha-Beta, kết hợp với mạng nơ-ron sâu NNUE (Neural Network-based Universal Evaluation) để đánh giá các vị trí trong trò chơi. Mô hình này đã được áp dụng cho trò chơi cờ vua và đạt được những kết quả đáng chú ý.

* Hiệu suất của mô hình: Mô hình AI có khả năng chơi ở mức độ cao, có thể cạnh tranh và đánh bại người chơi con người có kinh nghiệm ở mức độ trung bình. Trong các trận đấu thử nghiệm, AI đã thể hiện khả năng đưa ra các nước đi chiến lược và tối ưu, đặc biệt trong các tình huống phức tạp.
* Tốc độ và hiệu quả: Việc sử dụng kỹ thuật cắt tỉa Alpha-Beta đã giúp giảm đáng kể số lượng nút cần đánh giá trong cây trò chơi, từ đó tăng tốc độ tính toán. Kết hợp với NNUE, quá trình đánh giá vị trí được thực hiện nhanh chóng và chính xác hơn so với các phương pháp truyền thống. Điều này cho phép AI suy nghĩ sâu hơn trong khoảng thời gian hợp lý.
* Kết quả thực nghiệm: Chúng tôi đã tiến hành các thí nghiệm so sánh với các phiên bản AI khác, bao gồm cả những phiên bản sử dụng thuật toán Minimax thuần túy mà không có NNUE. Kết quả cho thấy mô hình của chúng tôi có tỷ lệ thắng cao hơn và thời gian phản hồi nhanh hơn, chứng minh sự hiệu quả của việc tích hợp NNUE vào thuật toán Minimax.

Hạn chế của đề tài

Độ sâu của cây trò chơi: Trong các trò chơi có không gian trạng thái lớn như cờ vua, việc tính toán đến một độ sâu nhất định vẫn gặp khó khăn do yêu cầu về thời gian và tài nguyên tính toán. Mặc dù Alpha-Beta pruning đã giúp giảm thiểu vấn đề này, nhưng với các tình huống phức tạp, việc mở rộng độ sâu vẫn là một thách thức.

Chất lượng của NNUE: Hiệu suất của NNUE phụ thuộc nhiều vào chất lượng và độ đa dạng của dữ liệu huấn luyện. Trong đồ án này, chúng tôi đã sử dụng một tập dữ liệu hạn chế, điều này có thể ảnh hưởng đến khả năng đánh giá chính xác các vị trí phức tạp hoặc hiếm gặp.

Tính tổng quát: Mô hình hiện tại được thiết kế và tối ưu cho cờ vua, do đó có thể không dễ dàng chuyển sang các trò chơi khác mà không cần điều chỉnh hoặc huấn luyện lại.

Hướng phát triển của đề tài

Cải thiện NNUE: Tiếp tục thu thập và mở rộng tập dữ liệu huấn luyện để bao gồm nhiều tình huống và biến thể của trò chơi. Ngoài ra, thử nghiệm với các kiến trúc mạng nơ-ron khác nhau để tìm ra cấu trúc tối ưu cho việc đánh giá vị trí.

Tối ưu hóa thuật toán: Nghiên cứu và triển khai các kỹ thuật pruning tiên tiến hơn hoặc các heuristic đặc biệt cho trò chơi để giảm thiểu thời gian tính toán mà không làm giảm chất lượng của các nước đi.

Mở rộng sang các trò chơi khác: Áp dụng mô hình này cho các trò chơi chiến lược khác để đánh giá tính tổng quát và khả năng thích ứng của phương pháp. Điều này có thể bao gồm việc điều chỉnh NNUE và các tham số của thuật toán Minimax cho phù hợp với từng trò chơi.

Tích hợp học máy khác: Kết hợp với các kỹ thuật học máy như reinforcement learning để cho phép AI học hỏi từ kinh nghiệm chơi game, từ đó cải thiện khả năng ra quyết định qua thời gian.

TÀI LIỆU THAM KHẢO

x

|  |  |
| --- | --- |
| [1] | Russell, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson Education. |
| [2] | Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. |
| [3] | Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Lai, M., ... & Lillicrap, T. (2018). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv preprint arXiv:1712.01815. |
| [4] | Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Lillicrap, T. (2018). DeepMind's AlphaZero: Shedding new light on the grand games of chess, shogi and Go. Science, 362(6419), 1140-1144. |
| [5] | Stockfish. (n.d.). Stockfish Chess Engine. Truy cập từ https://stockfishchess.org/ |
| [6] | LCZero. (n.d.). Leela Chess Zero. Truy cập từ https://lczero.org/ |
| [7] | PyChess. (n.d.). PyChess. Truy cập từ https://www.pychess.org/ |
| [8] | Levy, D., & Newborn, M. (1991). A Survey of Computer Chess. Trong Computer Chess Compendium (tr. 1-24). Springer. |
| [9] | Kasparov, G. (2017). The Evolution of Computer Chess. Truy cập từ https://www.gary-kasparov.com/the-evolution-of-computer-chess |
| [10] | Wikipedia. (n.d.). Computer Chess. Truy cập từ https://en.wikipedia.org/wiki/Computer\_chess |
| [11] | ChessBase. (n.d.). ChessBase. Truy cập từ https://www.chessbase.com/ |
| [12] | Buckland, M. (2005). Programming Game AI by Example. Wordware Publishing. |
| [13] | Millington, I. (2012). Artificial Intelligence for Games (2nd ed.). CRC Press. |
| [14] | Boden, M. A. (2016). AI: A Very Short Introduction. Oxford University Press. |
|  |  |

PHỤ LỤC