TRƯỜNG ĐẠI HỌC PHENIKAA

**KHOA CƠ KHÍ – CƠ ĐIỆN TỬ**



**BÀI TẬP LỚN**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**(C++)**

**Mã học phần: CSE703030**

**Dự án: CHINESE CHESS**

Học kỳ 2 Năm học 2024-2025

**Nhóm số: 01**

**Mã lớp**: [CSE703030-1-2-24(N03)](https://canvas.phenikaa-uni.edu.vn/courses/14618)

**Giảng viên hướng dẫn:** TS. Đinh Thế Hưng

**HÀ NỘI, 2/2025**

TRƯỜNG ĐẠI HỌC PHENIKAA

**KHOA CƠ KHÍ – CƠ ĐIỆN TỬ**

**BÀI TẬP LỚN**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**(C++)**

**Mã học phần: CSE703030**

**Dự án: CHINESE CHESS**

Học kỳ 2 Năm học 2024-2025

|  |  |  |
| --- | --- | --- |
| **Các sinh viên thực hiện:** | | |
| STT | Họ và tên | Mã lớp |
| 1 | Hoàng Trung An | [CSE703030-1-2-24(N03)](https://canvas.phenikaa-uni.edu.vn/courses/14618) |
| 2 | Nguyễn Đức Bình | [CSE703030-1-2-24(N03)](https://canvas.phenikaa-uni.edu.vn/courses/14618) |
|  |  |  |

**Giảng viên hướng dẫn:**...............................

**HÀ NỘI, .............../2025**

**PHIẾU ĐÁNH GIÁ KẾT QUẢ BÀI TẬP LỚN**

**HỌC PHẦN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (C++)**

**Mã học phần: CSE703030**

Họ và tên sinh viên: Nguyễn Đức Bình

Hoàng Trung An

Mã số sinh viên: 22010960

22010740

Mã đề: Chinese chess (Cờ tướng)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **CĐR** | **Nội dung đánh giá** | **Điểm tối đa** | **Điểm** | **Ghi chú** |
| 1 | 1.3 | Vận hành các kiến thức trong C++ để có thể giải các vấn đề tính toán đơn giản. | 4,0 |  |  |
| 2 | 2.1 | Thực hiện được thuần thục việc thiết kế và lập trình hướng đối tượng đơn giản với ngôn ngữ lập trình C++ | 6,0 |  |  |
| **Tổng** | | | **10,0** |  |  |

*Hà Nội, ngày ……… tháng ……… năm 2025*

**Giảng viên đánh giá**

(Ký và ghi rõ họ tên)

Đinh Thế Hưng

**DỰ ÁN**

Dự án thiết kế game cờ tướng với hai chế độ chơi hai người trên cùng một thiết bị và chơi với máy. Dự án được xây dựng với hình thức offline,

*VD:*

Dự án:

Thiết kế game cờ tướng (Chinese chess)

Đầu vào:

* Chế độ chơi:
  + - * Chơi với máy
      * Hai người chơi;
* Level: Chỉ một chế độ
* Hình thức: Offline

Đầu ra:

* Ở chế độ chơi với máy: Lưu lại bàn cờ để có thể tiếp tục
* Ở chế độ 2 người chơi: Offline, có lưu lại bàn cờ để có thể tiếp tục tiếp.

|  |  |  |  |
| --- | --- | --- | --- |
| *Các thành viên trong nhóm số 01:* | | | |
| *STT* | *HỌ VÀ TÊN* | *MSSV* | *Lớp* |
| *1* | Nguyễn Đức Bình | 22010960 | [CSE703030-1-2-24(N03)](https://canvas.phenikaa-uni.edu.vn/courses/14618) |
| *2* | Hoàng Trung An | 22010740 | [CSE703030-1-2-24(N03)](https://canvas.phenikaa-uni.edu.vn/courses/14618) |

**Mục lục**

[**TÓM TẮT 2**](#_Toc187926024)

[**Lời nói đầu 3**](#_Toc187926025)

[**Chương 1: Giới thiệu 4**](#_Toc187926026)

[**1.1 Đặt vấn đề 4**](#_Toc187926027)

[**1.2 Các giải pháp đã có 4**](#_Toc187926028)

[**1.3 Giải pháp đề xuất 4**](#_Toc187926029)

[**1.4 (Nếu có) 4**](#_Toc187926030)

[**1.5 (nếu có) 4**](#_Toc187926031)

[**Chương 2: Thiết kế và triển khai 5**](#_Toc187926032)

[**2.1 Các yêu cầu chức năng 5**](#_Toc187926033)

[**2.2 Các yêu cầu phi chức năng 5**](#_Toc187926034)

[**2.3 Các ràng buộc 5**](#_Toc187926035)

[**2.3.1 Các ràng buộc về triển khai 5**](#_Toc187926036)

[**2.3.2 Các ràng buộc về kinh tế 5**](#_Toc187926037)

[**2.3.3 Các ràng buộc về đạo đức 6**](#_Toc187926038)

[**2.4 Mô hình hệ thống/Thiết kế giải pháp 6**](#_Toc187926039)

[**2.4.1 Các kịch bản của hệ thống (Use-cases) 6**](#_Toc187926040)

[**2.4.2 Mô hình Use-case 6**](#_Toc187926041)

[**2.4.3 Mô hình hướng đối tượng 6**](#_Toc187926042)

[**2.4.4 Các biểu đồ tuần tự 7**](#_Toc187926043)

[**2.4.5 Các màn hình giao diện với người dùng 7**](#_Toc187926044)

[**Chương 3: Các yếu tố khác của dự án 8**](#_Toc187926045)

[**3.1 Kế hoạch và phân chia nhiệm vụ 8**](#_Toc187926046)

[**3.2 Cách thức hoạt động để đảm bảo thực hiện đúng tiến trình 8**](#_Toc187926047)

[**3.3 Các vấn đề về đạo đúc và tính chuyên nghiệp trong khi thực hiện 8**](#_Toc187926048)

[**3.4 Các yếu tố tác động đến xã hội 8**](#_Toc187926049)

[**3.5 Chiến lược và kế hoạch học tập kiến thức mới 9**](#_Toc187926050)

[**KẾT LUẬN 10**](#_Toc187926051)

[**CODE 11**](#_Toc187926052)

[**Tài liệu tham khảo 12**](#_Toc187926053)

TÓM TẮT

Bài tập lớn này trình bày quá trình thiết kế và phát triển trò chơi cờ tướng sử dụng ngôn ngữ C++. Mục tiêu là tạo ra một trò chơi có tính tương tác cao, giúp người chơi rèn luyện tư duy chiến thuật qua các chế độ chơi đa dạng. Trò chơi có hai chế độ chính: chơi với máy tính và hai người chơi offline.

Trong chế độ chơi với máy tính, người chơi có thể chọn AI mô phỏng chiến lược phù hợp. Trò chơi còn cung cấp chế độ mách nước, gợi ý nước đi để nâng cao chiến thuật. Chế độ hai người chơi offline mang đến trải nghiệm cờ tướng truyền thống, giúp người chơi cạnh tranh và phát triển chiến thuật riêng.

Giao diện trò chơi được thiết kế trực quan, dễ sử dụng và ổn định, có thể lưu lại bàn cờ đang chơi để có thể tiếp tục bất kỳ lúc nào kể cả khi đã tắt ứng dụng. Mục tiêu của bài tập này là tạo ra một ứng dụng chơi cờ tướng hoàn chỉnh, giúp người chơi cải thiện kỹ năng tư duy chiến thuật và chơi cờ tướng.

**ABSTRACT**

This project presents the design and development process of a Chinese chess (Xiangqi) game using the C++ programming language. The goal is to create a highly interactive game that helps players develop strategic thinking through various gameplay modes. The game features two main modes: playing against the computer and two-player offline mode.

In the computer mode, players can select an AI that simulates different strategic approaches. The game also includes a hint system that suggests possible moves to enhance strategic thinking. The two-player offline mode offers a traditional Chinese chess experience, allowing players to compete and refine their own strategies.

The game's user interface is designed to be intuitive, easy to use, and stable. Additionally, it supports saving the current board state so that players can resume their game at any time, even after closing the application. The objective of this project is to develop a complete Chinese chess application that helps players improve their strategic thinking and gameplay skills.

LỜI NÓI ĐẦU

Bài tập lớn này là một cơ hội tuyệt vời để chúng em có thể áp dụng những kiến thức đã học vào thực tế và phát triển một sản phẩm hoàn chỉnh. Qua quá trình nghiên cứu và làm việc, chúng em nhận thấy bài tập này không chỉ giúp tôi nâng cao kỹ năng lập trình mà còn rèn luyện khả năng tư duy logic, phân tích vấn đề và giải quyết các tình huống thực tế trong lập trình. Việc thiết kế và phát triển một trò chơi cờ tướng bằng ngôn ngữ C++ đã giúp tôi hiểu rõ hơn về cách xây dựng hệ thống, từ việc lập trình cho đến thiết kế giao diện người dùng.

Sơ đồ hệ thống của bài tập lớn đã được thiết kế khoa học, đảm bảo tính tương tác cao và hiệu quả trong việc xử lý các thao tác của người chơi. Bài tập này không chỉ có ý nghĩa trong việc rèn luyện kỹ năng mà còn giúp tôi có cái nhìn tổng quan về quá trình phát triển phần mềm từ ý tưởng đến sản phẩm hoàn chỉnh.

Cuối cùng, nhóm chúng em xin gửi lời cảm ơn chân thành đến giảng viên và các bạn trong nhóm đã cùng nỗ lực, cố gắng trong suốt quá trình thực hiện bài tập. Hy vọng kết quả cuối cùng là bài tập này sẽ mang lại những giá trị thực tiễn và góp phần nâng cao kiến thức cho mọi người.

Chương 1: Giới thiệu

* 1. Đặt vấn đề

Trong thời đại công nghệ hiện nay, với sự phát triển mạnh mẽ của các trò chơi trực tuyến, cờ tướng vẫn giữ được sức hấp dẫn bền vững như một trò chơi chiến thuật truyền thống. Tuy nhiên, nhiều trò chơi cờ tướng hiện đại ngày nay lại tập trung quá nhiều vào các yếu tố như đồ họa 3D, chế độ chơi trực tuyến, và các tính năng phức tạp khác.

Một trong những lỗ hổng trong thị trường game hiện nay là sự thiếu vắng của những trò chơi cờ tướng tối giản, chỉ tập trung vào trải nghiệm chơi cờ giữa hai người ở cùng một máy, giúp người chơi có thể tương tác trực tiếp và tận hưởng không gian thư giãn, truyền thống. Chính vì vậy, ý tưởng phát triển một game cờ tướng đơn giản nhưng chân thực đã được hình thành, nơi hai người có thể ngồi cạnh nhau và cùng trải qua trận đấu một cách trực tiếp mà không cần sự can thiệp của các yếu tố ngoại lai.

Trò chơi cờ tướng này sẽ tập trung vào sự tương tác trực tiếp giữa hai người chơi, tạo ra một không gian giao lưu và trao đổi ý tưởng chiến thuật, giúp người chơi có thể trải nghiệm cảm giác thi đấu thực sự. Đây là một lựa chọn lý tưởng cho những ai yêu thích cờ tướng và mong muốn tránh sự phức tạp của các tính năng online, nhưng vẫn muốn tận hưởng một trận đấu thú vị, đầy thử thách và đậm chất truyền thống.

* 1. Các giải pháp đã có

Khi nghiên cứu các giải pháp đã có trong thị trường game cờ tướng, nhóm đã tìm thấy một số nền tảng phổ biến hiện nay:

• **Cờ Tướng Online (ChessVn):**

* Cờ Tướng Online là một nền tảng cờ tướng trực tuyến phổ biến, cho phép người chơi thi đấu với máy hoặc đối kháng với người chơi khác trên toàn cầu.
* Tính năng chính bao gồm chơi cờ với máy, tham gia các giải đấu trực tuyến, và theo dõi các trận đấu của người chơi khác.
* Tuy nhiên, giao diện của game khá phức tạp và các tính năng như chat trực tuyến, bảng xếp hạng, hoặc các sự kiện online đôi khi làm mất đi sự tập trung của người chơi, đặc biệt là những người chỉ muốn chơi với bạn bè trên cùng một máy.

• **Cờ Tướng 2022:**

* Cờ Tướng 2022 là một game cờ tướng được nhiều người yêu thích, cung cấp chế độ chơi với máy hoặc đấu với bạn bè trực tuyến.
* Mặc dù có giao diện đẹp và dễ sử dụng, game vẫn thiên về việc kết nối và đấu với những người chơi khác trực tuyến. Việc chơi trên cùng một máy, dù có thể, nhưng chưa thực sự thuận tiện và dễ dàng, người chơi phải thực hiện một số thao tác phụ để chia sẻ màn hình hoặc thay đổi chế độ chơi.

Các game cờ tướng hiện nay đa số tích hợp nhiều tính năng như chơi trực tuyến, thi đấu trong các giải đấu, hoặc có các chế độ giúp người chơi cải thiện chiến thuật. Tuy nhiên, điểm yếu chung của các game này là họ không tập trung vào trải nghiệm đơn giản giữa hai người chơi trên cùng một máy. Các tính năng phụ như chơi trực tuyến hay giảng dạy chiến thuật làm mất đi sự nhẹ nhàng, dễ tiếp cận, và không gian thư giãn cho những người chỉ muốn chơi cờ tướng theo cách truyền thống, cùng nhau trên một màn hình.

* 1. Giải pháp đề xuất

Nhận thức được những hạn chế của các giải pháp hiện tại và hiểu rõ nhu cầu của người chơi cờ tướng, giải pháp được đề xuất là phát triển một game cờ tướng tối giản, nhưng hoàn toàn tập trung vào việc mang lại trải nghiệm trực tiếp giữa hai người chơi trên cùng một máy.

Trò chơi sẽ được thiết kế với giao diện đơn giản, dễ sử dụng và thân thiện, giúp người chơi dễ dàng làm quen mà không gặp phải sự phức tạp từ các tính năng không cần thiết. Chức năng chính của game là tạo ra không gian giao lưu trực tiếp giữa hai người chơi, cho phép họ ngồi cạnh nhau và chơi trên cùng một màn hình. Tất cả các thao tác di chuyển quân cờ và các tương tác khác sẽ được thực hiện trực tiếp trên màn hình chia sẻ.

Mục tiêu của trò chơi là tối ưu hóa trải nghiệm chơi cờ tướng, tập trung vào việc mang lại sự thú vị và sự tương tác chân thật giữa hai người chơi. Trò chơi này sẽ tạo ra một không gian giải trí thư giãn, lý tưởng cho những người muốn chia sẻ đam mê cờ tướng truyền thống với bạn bè, gia đình một cách gần gũi và dễ dàng, không bị phân tâm bởi các tính năng phức tạp hay yêu cầu kết nối trực tuyến.

Chương 2: Thiết kế và triển khai

* 1. Các yêu cầu chức năng
* Giao diện đơn giản: Giao diện chính sẽ hiển thị bốn tuỳ chọn chính: “Start new game ” , “Continue”, “Human-Machine” và "Exit", tối ưu hóa trải nghiệm sử dụng.
* Tiện dụng cho mọi người: Có nhắc nước đi của mọi quân cờ theo luật cờ vua khi chọn vào để di chuyển quân cờ đó.
  1. Các yêu cầu phi chức năng
* Hỗ trợ ngôn ngữ tiếng Anh.
* Giao diện tông màu trắng.
* Hình ảnh quân cờ trực quan.
  1. Các ràng buộc
     1. Các ràng buộc về triển khai
* Hiệu suất mượt mà để đảm bảo trải nghiệm người dùng.
* Đáp ứng luật chơi đúng luật cờ tướng.
  + 1. Các ràng buộc về kinh tế
* Sản phẩm phi lợi nhuận: miễn phí và không quảng cáo
* Không vi phạm bản quyền hình ảnh, mã nguồn …
  + 1. Các ràng buộc về đạo đức
* Ngôn ngữ và hình ảnh phù hợp
* Tôn trọng người chơi và công bằng của game
  1. Mô hình hệ thống/Thiết kế giải pháp
     1. Các kịch bản của hệ thống (Use-cases)

a. **Màn hình chính của game:**

• **Lựa chọn "Start new game"**: Hệ thống sẽ tạo một trận đấu mới giữa hai người chơi.

• **Lựa chọn "Exit"**: Game sẽ tắt và đóng ứng dụng.

• **Lựa chọn "Continue"**: Sẽ mở hộp thoại Explorer để chọn file TXT mở bàn cờ đã được lưu.

• **Lựa chọn "Human -Machine"**: Hệ thống sẽ tạo một trận đấy mới giữa người và máy.

b. **Trong trận đấu:** • **Di chuyển quân cờ:**

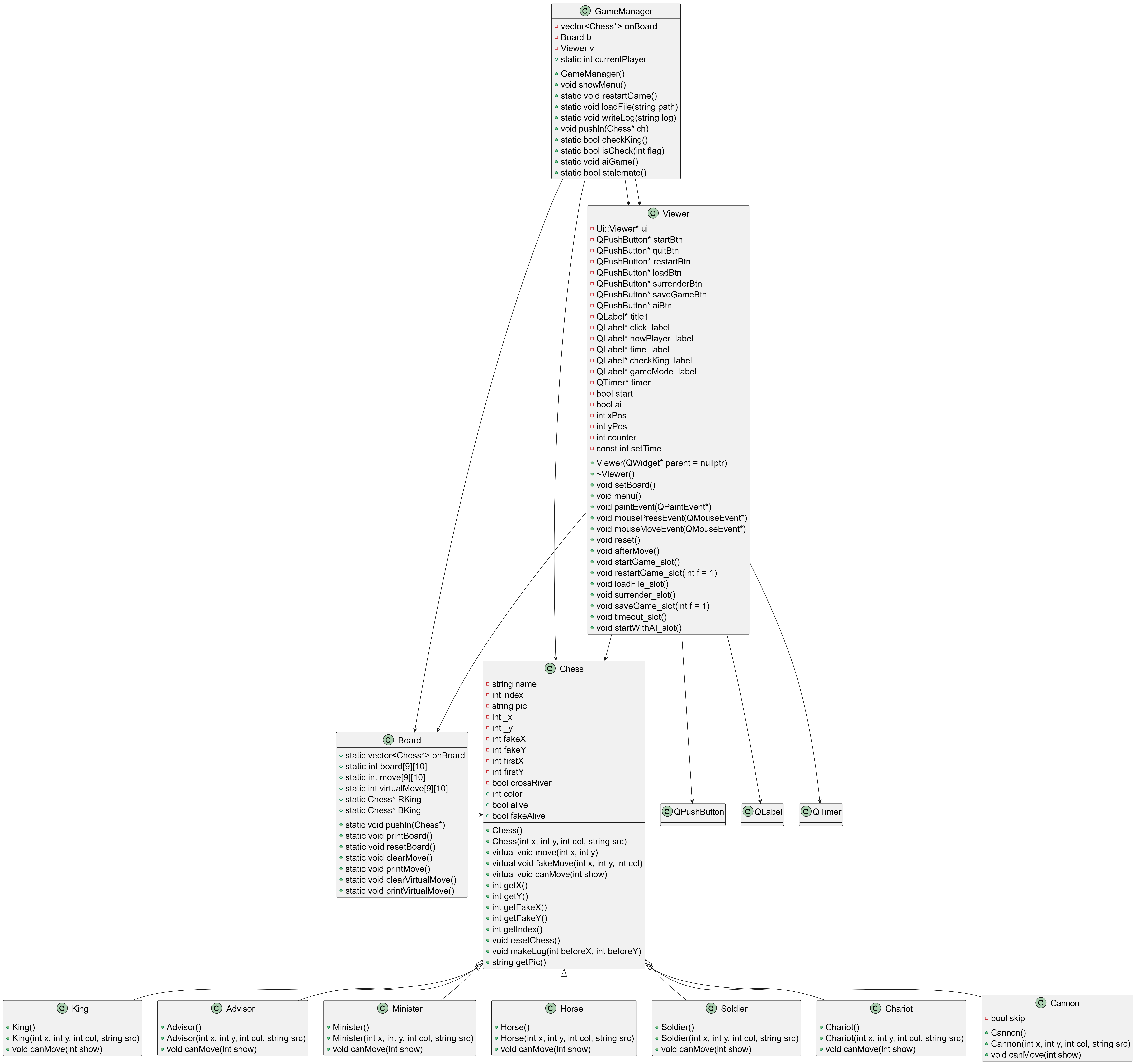
* Người chơi thực hiện di chuyển quân cờ của mình trên bàn cờ. Hệ thống sẽ làm nổi bật các ô mà quân cờ có thể di chuyển đến bằng cách hiện thị ô vuông màu đỏ quanh chúng.
* Hệ thống tự động kiểm tra tính hợp lệ của nước đi và cập nhật trạng thái bàn cờ sau mỗi lượt.
* Hệ thống có thời gian đếm giờ mỗi bước di chuyển được giới hạn trong 40 giây. Sau 40 giây mà không di chuyển bên đó sẽ bị xử thua.

• **Tướng bị bắt:**

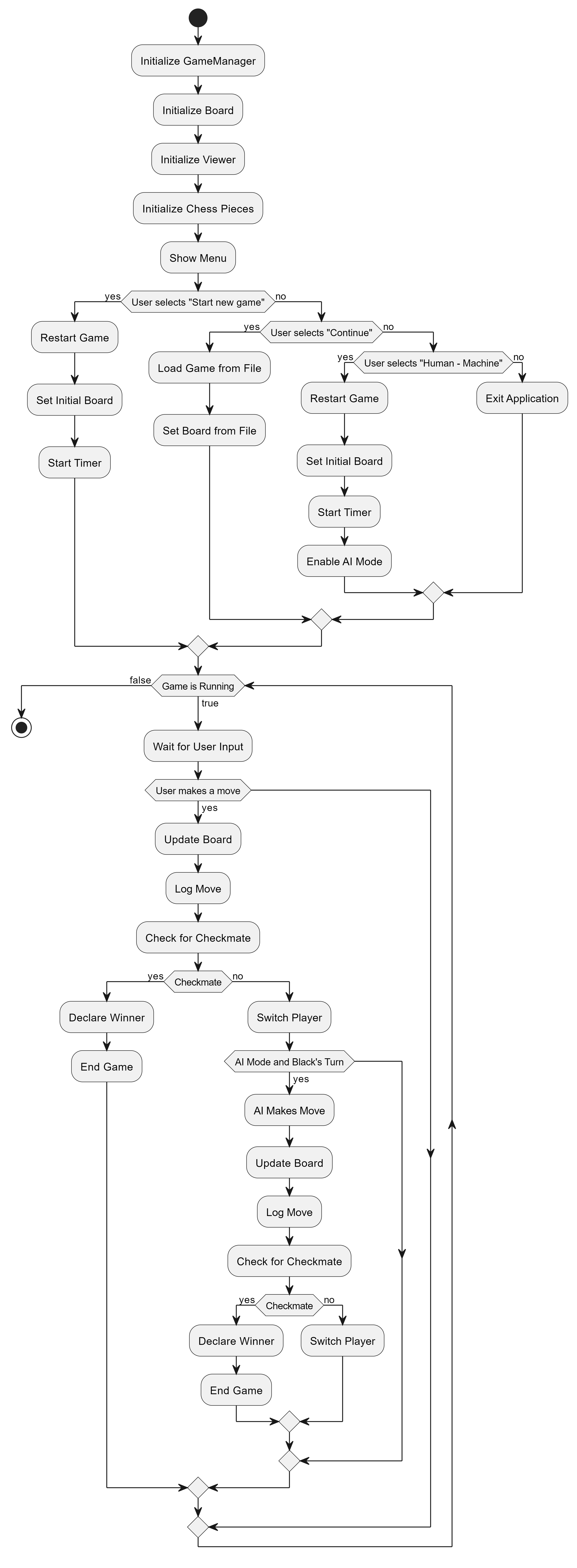
* Khi một bên bị tướng đối phương đe dọa, hệ thống sẽ hiển thị thông báo "Check mate".

• **Có bên thắng cuộc:**

* Khi một bên thắng cuộc (tướng đối phương bị bắt), hệ thống tự động thông báo kết quả và chuyển sang ván mới, giúp người chơi tiếp tục trận đấu mà không cần thao tác thêm.
  + 1. Mô hình Use-case
* Giao diện chơi được chia thành hai phía màn hình.
* Mỗi phía màn hình chứa bàn cờ và các quân cờ của người chơi tương ứng.
  + 1. Mô hình hướng đối tượng

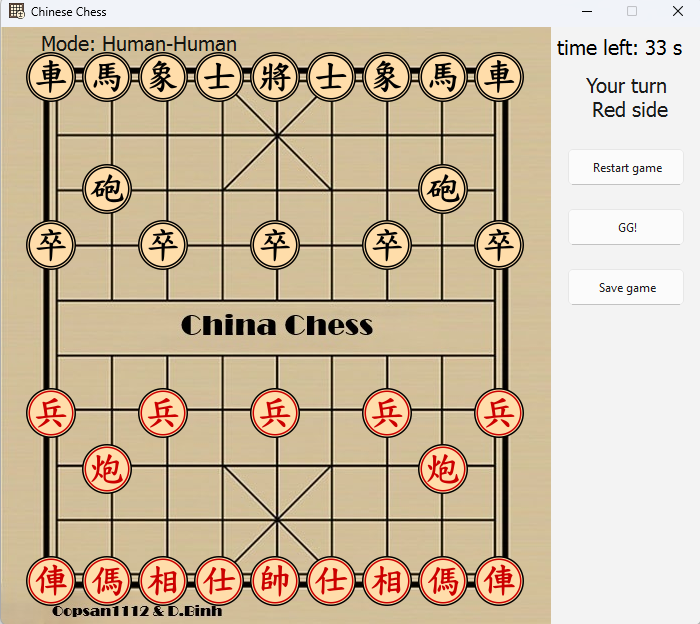


* **Lớp Viewer (viewer.cpp)**
* **Mô tả:**
* Lớp đại diện cho giao diện người dùng của trò chơi cờ tướng.
* **Thuộc tính:**
* [ui](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Giao diện người dùng.
* [timer](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Bộ đếm thời gian.
* [start](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [ai](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Trạng thái bắt đầu và AI.
* [title1](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [startBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [loadBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [aiBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [quitBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Các thành phần giao diện.
* [gameMode\_label](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [click\_label](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [time\_label](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [checkKing\_label](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [nowPlayer\_label](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Các nhãn hiển thị thông tin.
* [restartBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [surrenderBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [saveGameBtn](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Các nút điều khiển trò chơi.
* **Phương thức:**
* [Viewer(QWidget \*parent)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Constructor.
* ~Viewer(): Destructor.
* [menu()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Hiển thị menu.
* [setBoard()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Thiết lập bàn cờ ban đầu.
* [paintEvent(QPaintEvent \*)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Cập nhật bàn cờ sau mỗi nước đi.
* [mousePressEvent(QMouseEvent \*pos)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xử lý sự kiện nhấp chuột.
* [mouseMoveEvent(QMouseEvent \*pos)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xử lý sự kiện di chuyển chuột.
* [startGame\_slot()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Bắt đầu trò chơi.
* [reset()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Đặt lại trò chơi.
* [afterMove()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Cập nhật bàn cờ sau mỗi nước đi.
* [restartGame\_slot(int f)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Khởi động lại trò chơi.
* [loadFile\_slot()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tải trò chơi từ file.
* [surrender\_slot()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xử lý sự kiện đầu hàng.
* [saveGame\_slot(int f)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Lưu trò chơi.
* [timeout\_slot()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xử lý sự kiện hết thời gian.
* [startWithAI\_slot()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Bắt đầu trò chơi với AI.
* **Lớp GameManager (gamemanager.cpp)**
* **Mô tả:**
* Lớp quản lý trò chơi, xử lý logic và trạng thái của trò chơi.
* **Thuộc tính:**
* [currentPlayer](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Người chơi hiện tại.
* [onBoard](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Danh sách các quân cờ trên bàn cờ.
* [v](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Đối tượng Viewer.
* **Phương thức:**
* [GameManager()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Constructor.
* [showMenu()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Hiển thị menu.
* [restartGame()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Khởi động lại trò chơi.
* [loadFile(string path)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tải trò chơi từ file.
* [writeLog(string log)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Ghi log.
* [pushIn(Chess\* ch)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Thêm quân cờ vào trò chơi.
* [checkKing()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Kiểm tra trạng thái của các vua.
* [isCheck(int flag)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Kiểm tra trạng thái chiếu.
* [aiGame()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xử lý nước đi của AI.
* [stalemate()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Kiểm tra trạng thái hòa cờ.
* **Lớp Board (board.cpp)**
* **Mô tả:**
* Lớp đại diện cho bàn cờ và quản lý trạng thái của bàn cờ.
* **Thuộc tính:**
* [onBoard](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Danh sách các quân cờ trên bàn cờ.
* [board](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [move](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [virtualMove](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Mảng 2D đại diện cho trạng thái bàn cờ, các nước đi có thể và các nước đi ảo.
* [RKing](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [BKing](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Con trỏ đến các vua đỏ và đen.
* **Phương thức:**
* [pushIn(Chess\* ch)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Thêm quân cờ vào bàn cờ.
* [printBoard()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): In trạng thái bàn cờ.
* [resetBoard()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Đặt lại bàn cờ.
* [clearMove()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xóa mảng nước đi.
* [printMove()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): In trạng thái mảng nước đi.
* [clearVirtualMove()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xóa mảng nước đi ảo.
* [printVirtualMove()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): In trạng thái mảng nước đi ảo.
* **Lớp Chess (chess.cpp)**
* **Mô tả:**
* Lớp đại diện cho quân cờ và các hành động của quân cờ.
* **Thuộc tính:**
* [\_x](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [\_y](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tọa độ của quân cờ.
* [firstX](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [firstY](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tọa độ ban đầu của quân cờ.
* [pic](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Hình ảnh của quân cờ.
* [color](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Màu của quân cờ.
* [fakeX](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [fakeY](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Tọa độ ảo của quân cờ.
* [alive](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [fakeAlive](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Trạng thái sống của quân cờ.
* **Phương thức:**
* [Chess()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Constructor mặc định.
* [Chess(int x, int y, int col, string src)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Constructor có tham số.
* [move(int x, int y)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Di chuyển quân cờ.
* [fakeMove(int x, int y, int col)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Di chuyển ảo quân cờ.
* [getX()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [getY()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Lấy tọa độ của quân cờ.
* [getFakeX()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [getFakeY()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Lấy tọa độ ảo của quân cờ.
* [getIndex()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Lấy chỉ số của quân cờ.
* [resetChess()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Đặt lại quân cờ.
* [makeLog(int beforeX, int beforeY)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Ghi log nước đi của quân cờ.
* [canMove(int show)](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Xác định các nước đi hợp lệ của quân cờ.
* [getPic()](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Lấy hình ảnh của quân cờ.
* **Các lớp quân cờ cụ thể (King, Advisor, Minister, Horse, Soldier, Chariot, Cannon)**
* Các lớp này kế thừa từ lớp Chess và override phương thức [canMove](vscode-file://vscode-app/c:/Users/Admin/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) để xác định các nước đi hợp lệ cho từng loại quân cờ.
  + 1. Các biểu đồ tuần tự



* + 1. Các màn hình giao diện với người dùng

A screenshot of a game

AI-generated content may be incorrect.

Chương 3: Các yếu tố khác của dự án

* 1. Kế hoạch và phân chia nhiệm vụ
  2. Cách thức hoạt động để đảm bảo thực hiện đúng tiến trình
* Phân Công Công Việc: Mô tả rõ vai trò và nhiệm vụ của từng thành viên.
* Tương Tác Đội Nhóm: Lên kế hoạch họp định kỳ mỗi tuần 2 lần, sử dụng các kênh trực tuyến hoặc offline để thảo luận về tiến triển và vấn đề đang phát sinh.
  1. Các vấn đề về đạo đúc và tính chuyên nghiệp trong khi thực hiện
* Quy Tắc Ứng Xử: Xây dựng quy tắc ứng xử trong nhóm, giúp mọi người hiểu và tuân thủ quy định đạo đức.
* Đảm Bảo Đạo Đức Trong Lập Trình: Đảm bảo rằng mọi đoạn mã nguồn được viết tuân thủ các nguyên tắc và quy tắc lập trình đã được đề ra.
  1. Các yếu tố tác động đến xã hội
* Khía Cạnh Giáo Dục: Phát triển tính năng giáo dục trong trò chơi, như quy tắc cờ vua để người chơi có thể học hỏi.
* Tạo không gian giải trí, giao lưu: Chức năng chính của game là tạo ra không gian tương tác trực tiếp giữa hai người chơi, ngồi kế cạnh nhau trên cùng một máy.
  1. Chiến lược và kế hoạch học tập kiến thức mới
* Tổng kết và Phân Tích: Sau khi hoàn thành dự án lập trình hướng đối tượng về trò chơi cờ vua, dành thời gian để tổng kết và phân tích công việc bao gồm việc xem xét mã nguồn, kiểm tra các quyết định thiết kế, và đánh giá hiệu suất của chương trình.
* Học Từ Cộng Đồng và Dự Án Mở: Tham gia cộng đồng lập trình, diễn đàn trực tuyến hoặc dự án mã nguồn mở liên quan đến trò chơi cờ vua. Qua việc tương tác với người khác có thể học được nhiều từ kinh nghiệm của họ, nhận phản hồi xây dựng và có cơ hội tham gia vào các dự án lớn hơn. Điều này giúp phát triển kỹ năng cộng tác, hiểu rõ hơn về quy trình phát triển phần mềm, và mở rộng mạng lưới chuyên gia.

KẾT LUẬN

**Tóm tắt kết quả đã đạt được:** Dự án phát triển trò chơi cờ tướng bằng ngôn ngữ C++ đã thành công trong việc tạo ra một ứng dụng chơi cờ tướng hoàn chỉnh, hỗ trợ hai người chơi thi đấu trên cùng một máy. Trò chơi có giao diện đơn giản, dễ sử dụng và tối ưu hóa trải nghiệm chơi, giúp người chơi dễ dàng tham gia và thưởng thức cờ tướng truyền thống. Các tính năng chính như di chuyển quân cờ, hiển thị nước đi hợp lệ, và thông báo khi tướng bị bắt đã được triển khai đầy đủ và hoạt động ổn định.

**Những vấn đề còn hạn chế:** Mặc dù trò chơi đã hoàn thành với các tính năng cơ bản, vẫn còn một số vấn đề cần cải thiện:

1. **AI đối kháng:** Hiện tại, trò chơi đã có chế độ chơi với AI, nhưng hệ thống AI hiện tại còn khá đơn giản và chưa đủ sức thách thức người chơi.
2. **Chức năng lưu ván chơi:** Game hiện tại hỗ trợ lưu lại ván chơi. Nhưng giao diện để thực hiện chưa được trực quan
3. **Giao diện người dùng:** Mặc dù giao diện đơn giản, nhưng vẫn có thể cải thiện để trở nên trực quan và dễ dàng hơn, đặc biệt đối với những người mới làm quen với cờ tướng.

**Kiến nghị:**

1. **Phát triển thêm chế độ chơi với AI:** Mặc dù trò chơi hiện đã có chế độ chơi với máy tính (AI), hệ thống AI hiện tại còn khá đơn giản và chưa đủ sức thách thức người chơi có kinh nghiệm. Việc nâng cấp và cải tiến trí tuệ nhân tạo (AI) là rất cần thiết để mang lại trải nghiệm chơi game hấp dẫn và thú vị hơn, đặc biệt là cho những người chơi muốn luyện tập và thử thách bản thân với đối thủ máy.
2. **Tính năng lưu và tải ván chơi (mở rộng khả năng lưu trữ và tùy chỉnh):** Bổ sung tính năng lưu lại ván chơi dang dở và tải lại để tiếp tục là một tính năng quan trọng để tăng tính tiện lợi và trải nghiệm người dùng. Ngoài ra, để làm phong phú thêm trải nghiệm, nên cân nhắc mở rộng tính năng này để hỗ trợ lưu và tải các thế cờ đố (puzzle scenarios) hoặc cho phép người chơi đi lại nước cờ (undo move) để luyện tập và phân tích ván chơi.
3. **Cải tiến giao diện người dùng:** Cần cải thiện giao diện để người chơi dễ dàng tương tác và theo dõi diễn biến ván đấu, đồng thời có thể thêm các tính năng tùy chỉnh như thay đổi màu sắc bàn cờ hay quân cờ.
4. **Mở rộng chế độ chơi:** Bổ sung chế độ chơi trực tuyến hoặc đấu với bạn bè qua mạng cũng sẽ là một cải tiến đáng giá, mang lại nhiều sự lựa chọn cho người chơi.

CODE

#include "GameManager.h"

#include <sstream>

#include <fstream>

#include <ctime>

#include <cstdlib>

// Initialize the current player to 0

int *GameManager*::currentPlayer = 0;

// Constructor for the GameManager class

*GameManager*::GameManager()

{

    srand(time(NULL)); // Seed the random number generator

*Board*::onBoard = onBoard;

*fstream* file;

    file.open("log.txt", *ios*::out | *ios*::trunc);

    file.close();

}

// Function to show the menu

void *GameManager*::showMenu()

{

    v.menu();

    v.show();

}

// Function to restart the game

void *GameManager*::restartGame()

{

*Board*::clearVirtualMove();

*Board*::clearMove();

*Board*::resetBoard();

    for (int i = 0; i < *Board*::onBoard.size(); i++)

    {

*Board*::onBoard[i]->resetChess();

    }

*GameManager*::currentPlayer = 0;

*fstream* file;

    file.open("log.txt", *ios*::out | *ios*::trunc);

    file.close();

}

// Function to load a game from a file

void *GameManager*::loadFile(*string* *path*)

{

*GameManager*::restartGame();

*fstream* file;

    file.open(*path*, *ios*::in);

*string* input;

    int nowX, nowY, nextX, nextY;

    while (getline(file, input))

    {

        for (int i = 0; i < input.length(); i++)

        {

            if (!isdigit(input[i]))

                input[i] = ' ';

        }

        cout << input << endl;

*stringstream* in(input);

        in >> nowX >> nowX >> nowY >> nextX >> nextY;

        for (int i = 0; i < *Board*::onBoard.size(); i++)

        {

            if (*Board*::onBoard[i]->getX() == nowX && *Board*::onBoard[i]->getY() == nowY && *Board*::onBoard[i]->alive)

            {

*Board*::onBoard[i]->canMove(0);

                cout << "find!\n";

                if (*Board*::board[nextX][nextY] != 0)

                {

                    for (int j = 0; j < *Board*::onBoard.size(); j++)

                    {

                        if (*Board*::onBoard[j]->getX() == nextX && *Board*::onBoard[j]->getY() == nextY && *Board*::onBoard[j]->alive)

                        {

*Board*::onBoard[j]->alive = false;

*Board*::onBoard[j]->fakeAlive = false;

                            break;

                        }

                    }

                }

*Board*::onBoard[i]->move(nextX, nextY);

*Board*::onBoard[i]->makeLog(nowX, nowY);

*Board*::clearMove();

                break;

            }

        }

    }

*Board*::clearMove();

    file.close();

}

// Function to write a log entry to the log file

void *GameManager*::writeLog(*string* *log*)

{

*fstream* file;

    file.open("log.txt", *ios*::out | *ios*::app);

    file << *log*;

    file.close();

}

// Function to add a chess piece to the game

void *GameManager*::pushIn(*Chess* \**ch*)

{

    onBoard.push\_back(*ch*);

*Board*::board[*ch*->getX()][*ch*->getY()] = *ch*->color;

*Board*::onBoard = onBoard;

}

// Function to check if both kings are still alive

bool *GameManager*::checkKing()

{

    if (*Board*::RKing->alive && *Board*::BKing->alive)

    {

        return true;

    }

    else

    {

        return false;

    }

}

// Function to check if a king is in check

bool *GameManager*::isCheck(int *flag*) // flag 1: actual check, 0: simulated check

{

    if (*flag* == 1)

    {

        for (int i = 0; i < *Board*::onBoard.size(); i++)

        {

            if (currentPlayer % 2 == 0) // red's turn

            {

                if (*Board*::onBoard[i]->color == -1 && *Board*::onBoard[i]->alive) // black chess

                {

*Board*::onBoard[i]->canMove(1);

                    int x = *Board*::RKing->getX();

                    int y = *Board*::RKing->getY();

                    if (*Board*::virtualMove[x][y] == 1)

                    {

                        return true;

                    }

                }

            }

            else // black's turn

            {

                if (*Board*::onBoard[i]->color == 1 && *Board*::onBoard[i]->alive) // red chess

                {

*Board*::onBoard[i]->canMove(1);

                    int x = *Board*::BKing->getX();

                    int y = *Board*::BKing->getY();

                    if (*Board*::virtualMove[x][y] == 1)

                    {

                        return true;

                    }

                }

            }

        }

        return false;

    }

    else if (*flag* == 0)

    {

        for (int i = 0; i < *Board*::onBoard.size(); i++)

        {

            if (currentPlayer % 2 == 0) // red's turn

            {

                if (*Board*::onBoard[i]->color == -1 && *Board*::onBoard[i]->fakeAlive) // black chess

                {

*Board*::onBoard[i]->canMove(1);

                    int x = *Board*::RKing->getFakeX();

                    int y = *Board*::RKing->getFakeY();

                    if (*Board*::virtualMove[x][y] == 1)

                    {

                        return true;

                    }

                }

            }

            else // black's turn

            {

                if (*Board*::onBoard[i]->color == 1 && *Board*::onBoard[i]->fakeAlive) // red chess

                {

*Board*::onBoard[i]->canMove(1);

                    int x = *Board*::BKing->getFakeX();

                    int y = *Board*::BKing->getFakeY();

                    if (*Board*::virtualMove[x][y] == 1)

                    {

                        return true;

                    }

                }

            }

        }

        return false;

    }

}

// Structure to represent a position on the board

typedef struct *Position*

{

    int x;

    int y;

};

// Function to simulate an AI move

void *GameManager*::aiGame()

{

    int randNum = rand() % *Board*::onBoard.size() / 2 + 1;

    int count = 0;

    for (int i = 0; i < *Board*::onBoard.size(); i++)

    {

        if (*Board*::onBoard[i]->color == -1)

        { // black piece

            count++;

        }

        if (count == randNum && *Board*::onBoard[i]->alive)

        {

            int beforeX = *Board*::onBoard[i]->getX();

            int beforeY = *Board*::onBoard[i]->getY();

*Board*::onBoard[i]->canMove(0);

            vector<*Position*> list;

            list.clear();

            for (int j = 0; j < 9; j++)

            {

                for (int k = 0; k < 10; k++)

                {

                    if (*Board*::move[j][k] == 1)

                    {

                        list.push\_back({j, k});

                    }

                }

            }

            if (list.size() == 0)

            {

                randNum++;

            }

            else

            {

                randNum = rand() % list.size();

                if (*Board*::board[list[randNum].x][list[randNum].y] != 0)

                {

                    for (int j = 0; j < *Board*::onBoard.size(); j++)

                    {

                        if (*Board*::onBoard[j]->getX() == list[randNum].x && *Board*::onBoard[j]->getY() == list[randNum].y && *Board*::onBoard[j]->alive)

                        {

*Board*::onBoard[j]->alive = false;

*Board*::onBoard[j]->fakeAlive = false;

                            break;

                        }

                    }

                }

*Board*::onBoard[i]->move(list[randNum].x, list[randNum].y);

*Board*::onBoard[i]->makeLog(beforeX, beforeY);

*Board*::clearMove();

                break;

            }

        }

        else if (count == randNum && !*Board*::onBoard[i]->alive)

        {

            randNum++;

        }

    }

}

// Function to check for stalemate

bool *GameManager*::stalemate()

{

*Board*::clearVirtualMove();

    int nowTurn = currentPlayer % 2;

    if (nowTurn == 0)

        nowTurn = 1;

    else

        nowTurn = -1;

    for (int i = 0; i < *Board*::onBoard.size(); i++)

    {

*Board*::clearMove();

        if (*Board*::onBoard[i]->color == nowTurn && *Board*::onBoard[i]->alive)

        {

*Board*::onBoard[i]->canMove(0);

            vector<*Position*> list;

            list.clear();

            for (int j = 0; j < 9; j++)

            {

                for (int k = 0; k < 10; k++)

                {

                    if (*Board*::move[j][k] == 1)

                        list.push\_back({j, k});

                }

            }

            for (int k = 0; k < list.size(); k++)

            {

                int col = *Board*::board[list[k].x][list[k].y];

                int tmpIndex = -1;

                if (*Board*::board[list[k].x][list[k].y] != 0)

                {

                    for (int j = 0; j < *Board*::onBoard.size(); j++)

                    {

                        if (*Board*::onBoard[j]->getX() == list[k].x && *Board*::onBoard[j]->getY() == list[k].y && *Board*::onBoard[j]->alive)

                        {

*Board*::onBoard[j]->fakeAlive = false;

                            tmpIndex = j;

                        }

                    }

                }

*Board*::onBoard[i]->fakeMove(list[k].x, list[k].y, 0);

                if (!isCheck(0))

                {

*Board*::onBoard[i]->fakeMove(*Board*::onBoard[i]->getX(), *Board*::onBoard[i]->getY(), col);

                    if (tmpIndex != -1)

*Board*::onBoard[tmpIndex]->fakeAlive = true;

*Board*::clearVirtualMove();

                    return false;

                }

                else

                {

*Board*::onBoard[i]->fakeMove(*Board*::onBoard[i]->getX(), *Board*::onBoard[i]->getY(), col);

                    if (tmpIndex != -1)

*Board*::onBoard[tmpIndex]->fakeAlive = true;

*Board*::clearVirtualMove();

                }

            }

        }

    }

    return true;

}

Tài liệu tham khảo

1. https://doc.qt.io/qt-6/qgraphicsrectitem.html
2. https://doc.qt.io/qt-6/qgraphicspixmapitem.html
3. https://doc.qt.io/qt-6/qgraphicsview.html
4. https://doc.qt.io/qt-6/qpixmap.html
5. https://github.com/chenshuo/xiangqi
6. https://github.com/XMuli/ChineseChess