|  |
| --- |
| ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINHTRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN **KHOA CÔNG NGHỆ THÔNG TIN**    **BÁO CÁO ĐỒ ÁN**  **MÔN: Phương pháp lập trình hướng đối tượng**  **Lớp: 21\_2**  **Đề tài: Game cờ vua**  *Giảng viên hướng dẫn:*  Thầy Lê Khánh Duy  Thầy Lê Trung Nghĩa  Thầy Lý Duy Nam  *Thành viên nhóm:*  Nguyễn Chí Cường – 21120421  Đặng Thế Sinh – 21120611  Nguyễn Đình Phương Đại – 21120424 |

A.Thông tin chung:

- Hệ điều hành: Windows

- IDE: Visual Studio 2022

- Ngôn ngữ: C/C++

- Thư viện sử dụng: Thư viện chuẩn của C/C++, thư viện SFML

**1. Thư viện SFML:**

*1.1) Giới thiệu thư viện SFML:*

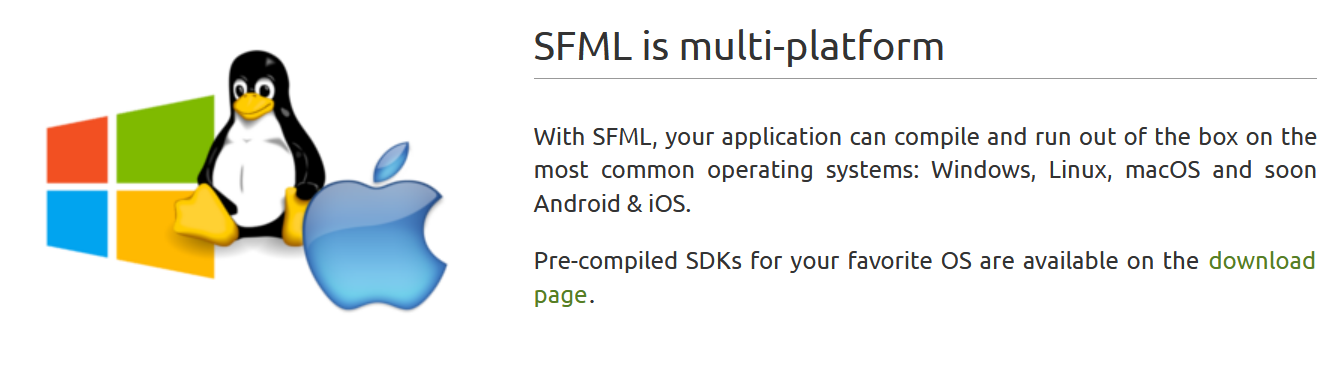
SFML (Simple and Fast Multimedia Library) là một thư viện đa phương tiện chủ yếu được viết bằng C/C++ và áp dụng phương pháp lập trình hướng đối tượng để tạo ra những ứng dụng có giao diện đẹp mắt cho người dùng.



Logo của thư viện SFML

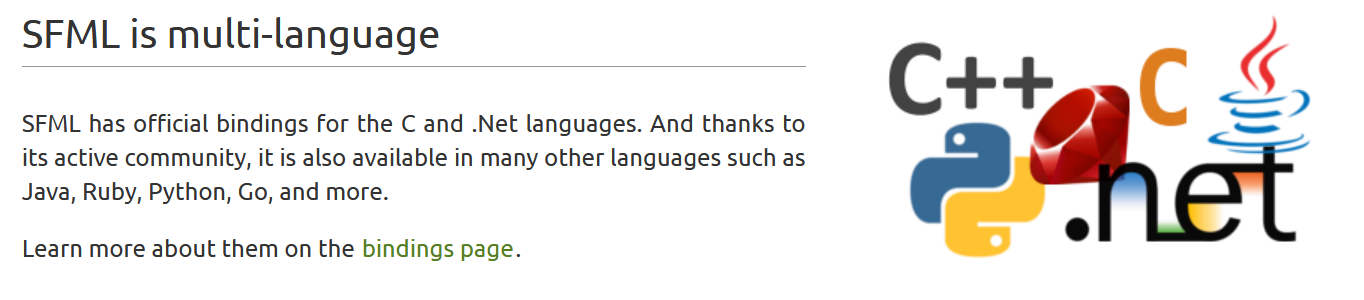
*1.2) Các nền tảng hỗ trợ:*

Thư viện SFML có thể được dùng để viết những ứng dụng có thể chạy được trên nhiều nền tảng như Windows, Linux, macOS,…



*1.3) Các ngôn ngữ lập trình hỗ trợ:*

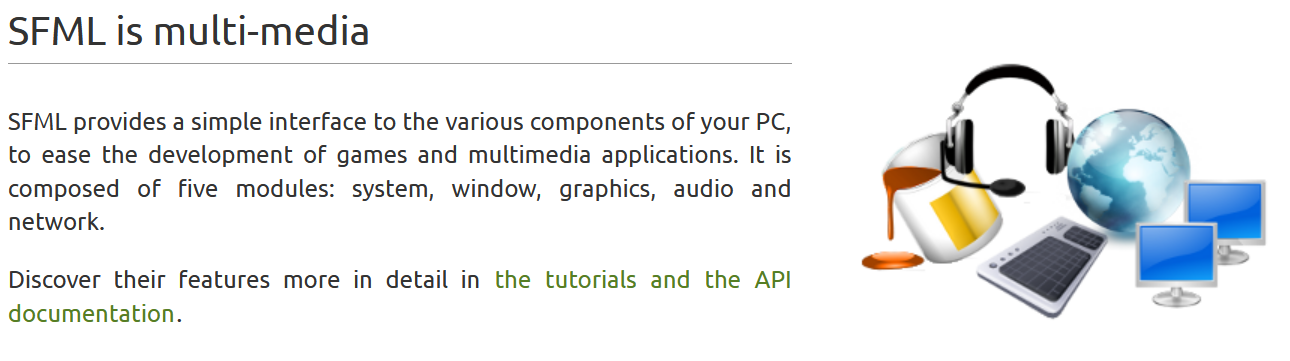
Thư viện này có thể được sử dụng cho nhiều ngôn ngữ lập trình khác ngoài C như Java, Python, … tuy nhiên 2 ngôn ngữ chính thức của thư viện này là C và .Net. Các ngôn ngữ khác được thêm vào là do cộng đồng hỗ trợ.



1.4) Các modules chính của thư viện SFML:

Gồm 5 modules:

* System: gồm các class liên quan với hệ thống như làm thời gian, xử lí Unicode…
* Window: liên quan tới việc tạo, đóng và xử lí sự kiện cửa sổ…
* Graphics: bao gồm các class về việc render đồ họa…
* Audio: bao gồm các class về xử lí âm thanh, ta có thể dùng để phát một file nhạc hoặc ghi âm cho máy tính…
* Network: liên quan đến việc xử lý các giao thức mạng…



B.Mô tả thiết kế phần mềm:

**1. Lớp Setting:**

Lớp Setting dùng để cấu hình cho các thông số của game (kích thước window, kích thước của bàn cờ, các button…) và đường dẫn các tài nguyên (hình ảnh các quân cờ, bàn cờ, background, âm thanh, phông chữ).

Các thuộc tính của lớp Setting:

screenTitle : Tên cửa sổ

screenWidth : Chiều dài cửa sổ  
 screenHeight : Chiều rộng cửa sổ

Các thuộc tính này được dùng để khởi tạo cửa sổ ứng dụng trong Windows.

squareSize : Kích thước 1 ô cờ

startX, startY : Tọa độ góc trái trên cùng của bàn cờ

pickColorBtnSize : Kích thước của ô chọn bên quân cờ

buttonSize : Kích thước các button (Undo, Redo…)

pieceURL : Đường dẫn đến ảnh của các quân cờ (xem trong thư mục Resources\Images)

promoteURL : Đường dẫn đến ảnh các quân cờ để chọn khi người chơi phong hậu

fontName : Đường dẫn đến font chữ sử dụng

soundURL : Đường dẫn dến âm thanh

Các thuộc tính này được dùng để hỗ trợ làm giao diện cho người dùng (GUI).

**2. Lớp Vec2**

*2.1) Chức năng lớp:*

Biểu diễn tọa độ của 1 điểm trên màn hình.

*2.2) Sơ đồ UML:*



*2.3) Thuộc tính:*

Các thuộc tính của lớp Vec2 bao gồm tọa độ x và y với kiểu dữ liệu đã được tham số hóa.

*2.4) Phương thức:*

Lớp Vec2 cung cấp các phương thức constructor, destructor, setter, getter. Ngoài ra, lớp còn cung cấp các phương thức nạp chồng toán tử để thuận tiện cho việc tinh toán hơn.

Toán tử + - \* với 1 vector rhs được quy ước rằng ta sẽ cộng, trừ, nhân giữa 2 tọa độ của 2 Vec2. Đối với các vector +=, -=, \*= thì cũng tương tự như vậy.

Toán tử ==, != với 1 vector rhs được quy ước rằng ta sẽ so sánh tọa độ giữa 2 Vec2. Nếu tọa độ x và y giữa hai Vec2 này bằng nhau thì sẽ trả về true, ngược lại là false.

**3. Namespace Utility**

*\* Chức năng:*

Utility là một namespace được sử dụng để gọi một số phương thức xử lý.

- isOnBoard(const Vec2<int> position): Kiểm tra xem quân cờ có ở vị trí hợp lệ trong bàn cờ 8 x 8 hay không.

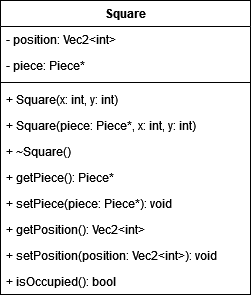
- playSound(soundURL: const char\*): Thêm nhạc vào game với địa chỉ là thông số đầu vào.

**4. Lớp Square:**

*4.1) Chức năng lớp:*

Lớp Square đại diện cho 1 ô vuông trong bàn cờ. Lớp này dùng để chứa các thông tin quân cờ và tọa độ của quân cờ đó.

*4.2) Sơ đồ UML:*



*4.3) Thuộc tính:*

position: Chứa thông tin tọa độ (Là thực thể của lớp Vec2)

piece: Chứa thông tin quân cờ (Là con trỏ trỏ tới thực thể Piece)

Các thuộc tính này chứa thông tin cần có của một ô trong bàn cờ.

*4.4) Phương thức:*

Lớp Square cung cấp các phương thức constructor, destructor, setter, getter:

~Square(): Hủy object Square nhưng chỉ cần xóa thuộc tính piece.

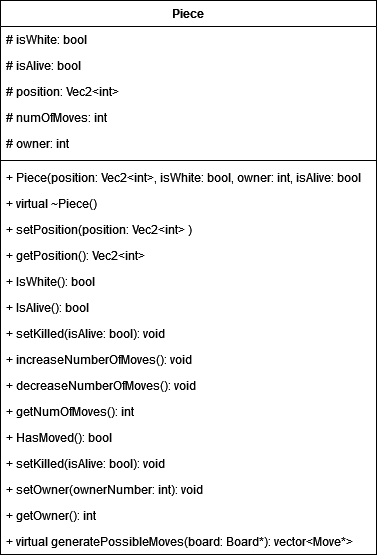
isOccupied(): Kiểm tra xem có quân cờ nào ở ô đó không.

**5. Lớp Piece:**

*5.1) Chức năng lớp:*

Lớp Piece là một lớp được trừu tượng hóa, tượng trưng cho quân cờ trên bàn cờ. Đây là lớp áp dụng tính đa hình (polymorphism) và là lớp dùng để cho các lớp liên quan đến quân cờ kế thừa (inheritance).

*5.2) Sơ đồ UML:*



*5.3) Thuộc tính:*

isWhite: Quân cờ có phải màu trắng không

isAlive: Quân cờ đã bị ăn hay chưa

position: Tọa độ quân cờ trong bàn cờ

numOfMoves: Số cách đi mà quân cờ có thể đi

owner: Xác định chủ của quân cờ là ai

*5.4) Phương thức:*

Lớp Piece có cung cấp các constructor, destructor, các hàm getter và setter, một số hàm thuần ảo và một số hàm khác:

virtual ~Piece(): Hàm thuần ảo cho phương thức hủy.

isWhite(), isAlive(): Đây là các hàm getter để lấy các thuộc tính isWhite, isAlive

HasMoved(): Kiểm tra quân cờ này đã được di chuyển hay chưa

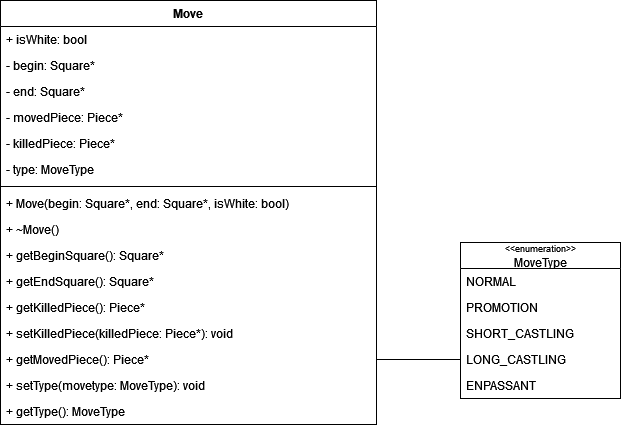
virtual generatePossibleMoves(board: Board\*) = 0: Đây là hàm thuần ảo cho phương thức sinh những đường đi hợp lệ.

**6. Lớp Move**

*6.1) Chức năng lớp:*

Lớp Move đại diện cho 1 nước đi của quân cờ.

*6.2) Sơ đồ UML:*



*6.3) Thuộc tính:*

Các thuộc tính sau đã cấu thành nên 1 nước đi trong quân cờ:

isWhite: Có phải là quân trắng hay không.

begin: Ô bắt đầu (Có thể nói là ô mà quân cờ đang đứng).

end: Ô cuối (Có thể nói là ô mà quân cờ sẽ di chuyển tới).

movedPiece: Quân cờ được di chuyển.

killedPiece: Quân cờ bị ăn.

*6.4) Phương thức:*

Lớp Move có cung cấp các constructor, destructor, các hàm getter và setter.

*6.5) Lớp enum MoveType:*

Lớp enum này cung cấp các loại đường đi có thể có của 1 quân cờ.

Normal: Nước đi thường.

Promotion: Nước đi “phong cấp”.

Short Castling, Long Castling: Nước đi “nhập thành”.

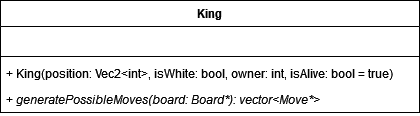
Empassant: Nước đi “bắt Tốt qua đường”

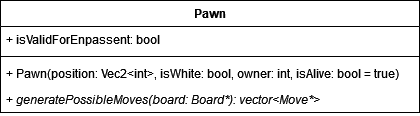
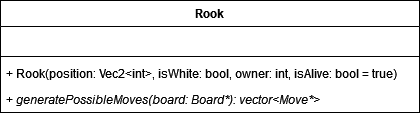
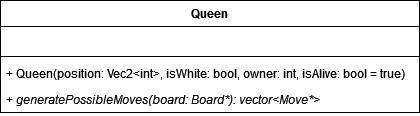
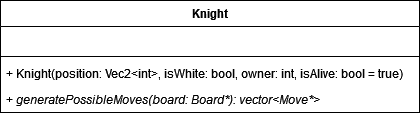
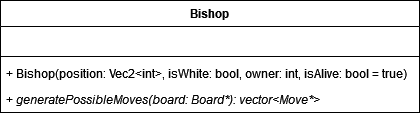
**7. Lớp King, Knight, Queen, Bishop, Rook, Pawn:**

*7.1) Chức năng lớp:*

Đây là các lớp đại diện cho các quân cờ King, Knight, Queen, Bishop, Rook, Pawn, là đặt biệt hóa của lớp quân cờ (Piece).

*7.2) Sơ đồ UML cho từng quân cờ:*





*7.3) Thuộc tính:*

Các thuộc tính của lớp các quân cờ tương ứng cũng là các thuộc tính của lớp Piece.

*7.4) Phương thức:*

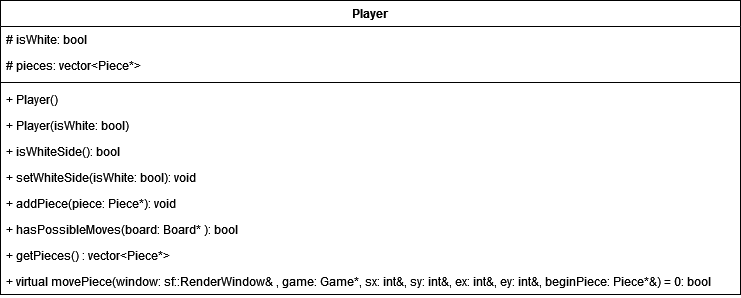
Constructor được kế thừa từ lớp Piece, generatePossibleMoves đã được đặc biệt hóa từ hàm thuần ảo của lớp Piece được dùng để sinh ra những nước đi có thể xảy ra cho quân cờ tương ứng.

**8. Lớp Player**

*8.1) Chức năng lớp:*

Lớp Player là một lớp được trừu tượng hóa, tượng trưng cho người chơi. Đây là lớp áp dụng tính đa hình (polymorphism) và là lớp dùng để cho các lớp liên quan đến kế thừa (inheritance).

*8.2) Sơ đồ UML*



*8.3) Thuộc tính:*

isWhite: Chỉ định bên người chơi nào sẽ là trắng.

pieces: Mảng các quân cờ mà người chơi có.

*8.4) Phương thức:*

Lớp Player cung cấp các constructor, getter, setter và một số hàm xử lý khác.

addPiece: Thêm 1 quân cờ vào mảng pieces

hasPossibleMoves: Xét xem có đường đi hợp lệ không

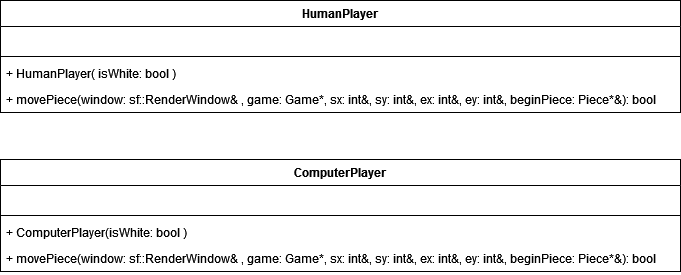
movePiece: Hàm thuần ảo, dùng để di chuyển quân cờ

**9. Lớp HumanPlayer, ComputerPlayer**

*9.1) Chức năng lớp:*

Đây là các lớp đại diện cho các người chơi (Người chơi, Máy chơi) là đặt biệt hóa của lớp Player.

*9.2) Sơ đồ UML:*



*9.3) Thuộc tính:*

Các thuộc tính của lớp HumanPlayer hay ComputerPlayer tương ứng cũng là các thuộc tính của lớp Player.

*9.4) Phương thức:*

Constructor được kế thừa từ lớp Piece, movePiece đã được đặc biệt hóa từ hàm thuần ảo của lớp Player được dùng để điều khiển hành vi di chuyển quân cờ.

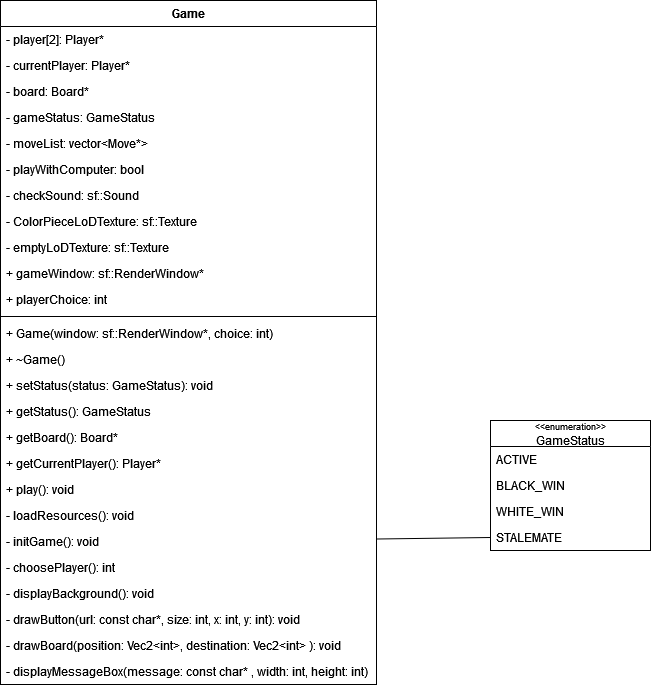
Ở lớp Computer Player, nước đi sẽ được quyết định bằng cách ngẫu nhiên.

**10. Lớp Game:**

*10.1) Chức năng lớp:*

Đây là lớp dựng kịch bản cho trò chơi.

*10.2) Sơ đồ UML:*



*10.3) Thuộc tính:*

player: Chứa đối tượng người chơi (2 người chơi).

currentPlayer: Người chơi hiện tại là ai (Theo lượt).

board: Chứa đối tượng bàn cờ.

gameStatus: Tiến trình của game

moveList: Danh sách các lượt đi của 2 người chơi.

ColorPieceLoDTexture: Texture của quân cờ: Màu (trắng / đen), quân cờ, LoD (ô vuông Light(trắng) or Dark(đen)). Do có quá nhiều thuộc tính tương tự nên đây là quy luật đặt tên để khái quát.

emptyLoDTexture: Texture của các ô vuông không có quân cờ

gameWindow: Màn hình window của trò chơi

*10.4) Phương thức:*

Lớp Move có cung cấp các constructor, destructor, các hàm getter và setter, một số hàm xử lý đồ họa và các hàm khác. Tùy vào mục đích sử dụng thì các phương thức có thể public hay private:

play: Chứa kịch bản của trò chơi

initGame: Khởi tạo trò chơi

choosePlayer: Người chơi sẽ chọn quân đen hay trắng.

*10.5) Lớp enum MoveType:*

Lớp enum này cung cấp các tiến trình của trò chơi.

Active: Hoạt động.

Black Win, White Win: Quân đen thắng, Quân trắng thắng.

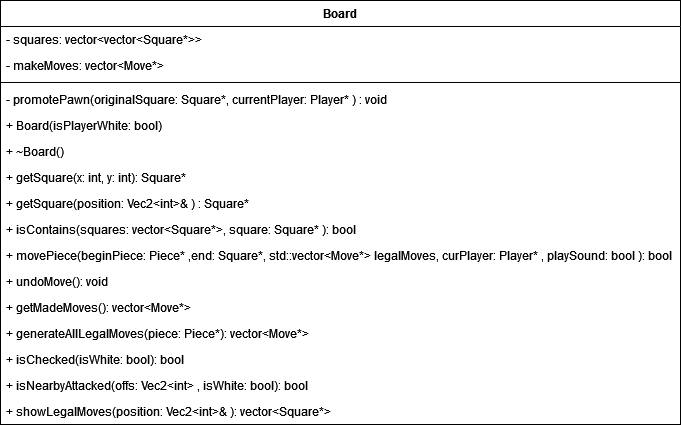
Stalemate: Hòa.

**11. Lớp Board:**

*11.1) Chức năng lớp:*

Lớp Board cung cấp các xử lý liên quan đến bàn cờ.

*11.2) Sơ đồ UML:*



*11.3) Thuộc tính:*

squares: Mảng 2 chiều chứa các đối tượng Square (tượng trưng cho 1 bàn cờ)

makeMoves: Chứa thông tin đường đi của các quân cờ.

*11.4) Phương thức:*

Có 1 phương thức xử lý riêng (private):

promotePawn: Hàm thực hiện nước đi “Thăng cấp”

Các phương thức public:

Board(isPlayerWhite): Khởi tạo bàn cờ với các quân cờ ở vị trí ban đầu của nó, isPlayerWhite dùng để tạo bàn cờ dựa trên người chơi chọn bên nào.

~Board(): Hủy đối tượng Board tránh rò rỉ bộ nhớ.

getSquare(x: int, y: int): Lấy ô vuông từ vị trí ô vuông trong mảng 2 chiều.

getSquare(position: Vec2<int>&): Lấy ô vuông từ cặp tọa độ x, y

isContains(): Xét xem 1 ô vuông có nằm trong bàn cờ không

movePiece: Xử lý nước đi của quân cờ.

undoMove: Đi lại 1 nước đi

getMadeMove: Lấy những nước đi đã đi

showAllLegalMoves: Vẽ cho người chơi biết những nước đi hợp lệ

isChecked: Kiểm tra nước đi chiếu tướng.

isNearbyAttacked: Hiển thị cho người chơi biết quân nào sắp bị ăn

generateAllLegalMoves: Sinh những nước đi hợp lệ

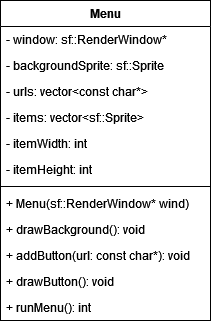
draw: Hàm vẽ hỗ trợ giao diện cho người dùng

**12. Lớp Menu:**

*12.1) Chức năng lớp:*

Lớp Menu sẽ cung cấp menu cho người dùng thao tác trong game.

*12.2) Sơ đồ UML*



*12.3) Thuộc tính, phương thức:*

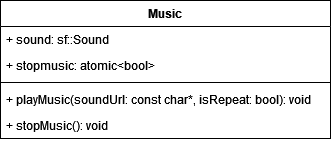
Các thuộc tính và phương thức cung cấp ở đây giúp cho việc tạo menu một cách đơn giản. Menu sẽ có background, đường dẫn dẫn đến tài nguyên sprite, kích cỡ tài nguyên, … và các phương thức dùng để gắn kết các phần của thuộc tính để tạo menu.

**13. Lớp Music**

*13.1) Chức năng lớp:*

Lớp Music sẽ cung cấp các phương thức xử lý âm thanh trong game.

*13.2) Sơ đồ UML*



*13.3) Thuộc tính, phương thức:*

Thuộc tính có kiểu dữ liệu atomic có nghĩa là lớp này được xử lý đa luồng, có thể phát nhạc song song với lúc chơi game. Các phương thức ở dưới để chơi nhạc hoặc dừng nhạc.

C.Mô tả tính năng game:

*\*Đọc phần [3] trong mục F để biết cách cài đặt thư viện SFML trước khi có thể chơi.\**

1) Menu:



Đây là màn hình lúc bắt đầu trò chơi. Có các chế độ chơi với người (Play with Human) và chơi với máy (Play with AI).

2) Chọn quân cờ:



Ta được phép chọn quân cờ đen hoặc trắng tùy theo ý của mình. Sau khi chọn xong, chúng ta sẽ bắt đầu chơi.

3) Bắt đầu trò chơi:

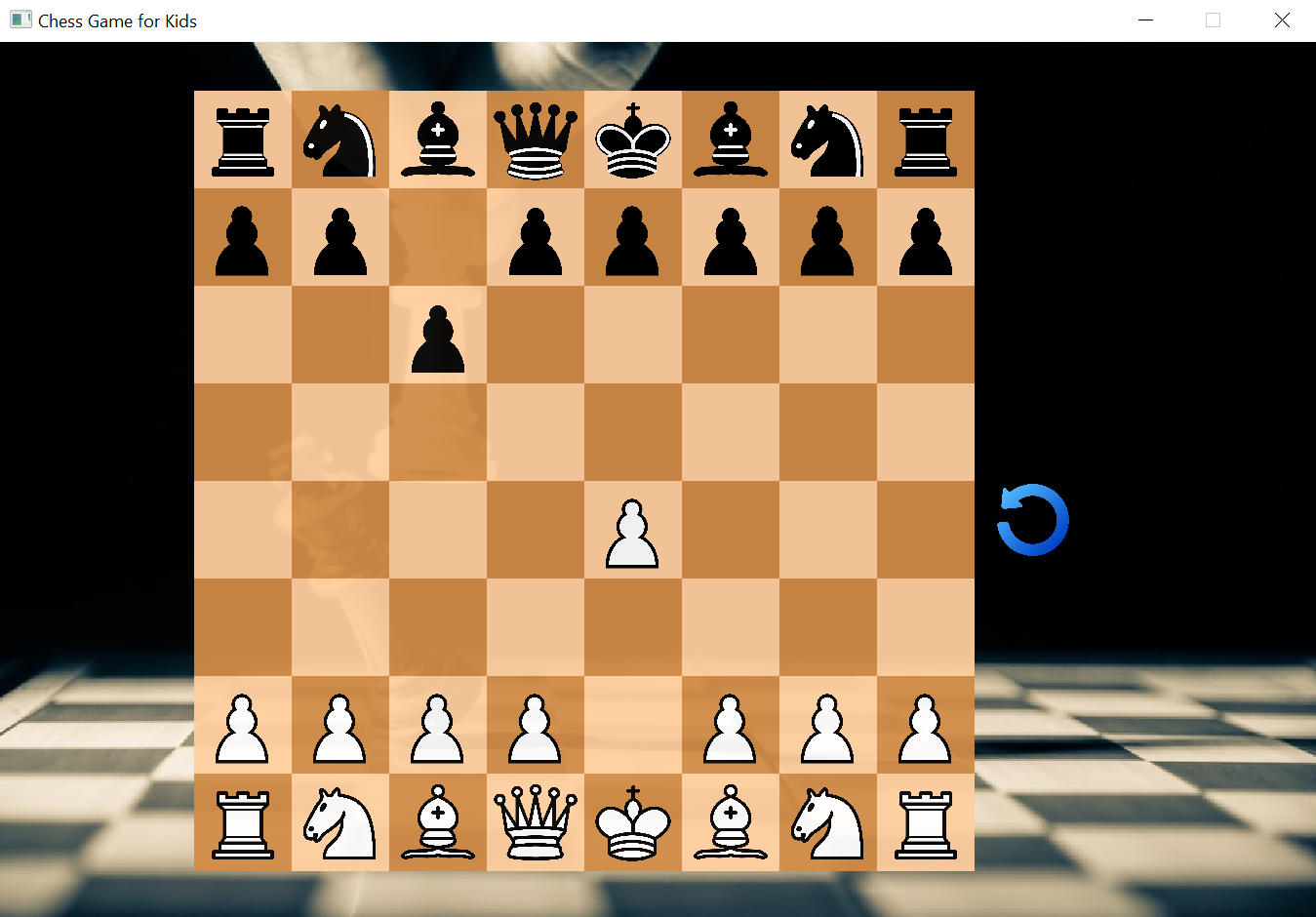
Giả sử ở đây ta chọn chế độ chơi với máy, quân trắng, màn hình sẽ hiển thị như thế này.



Quân trắng sẽ đi trước. Chúng ta sẽ thử di chuyển 1 nước cờ

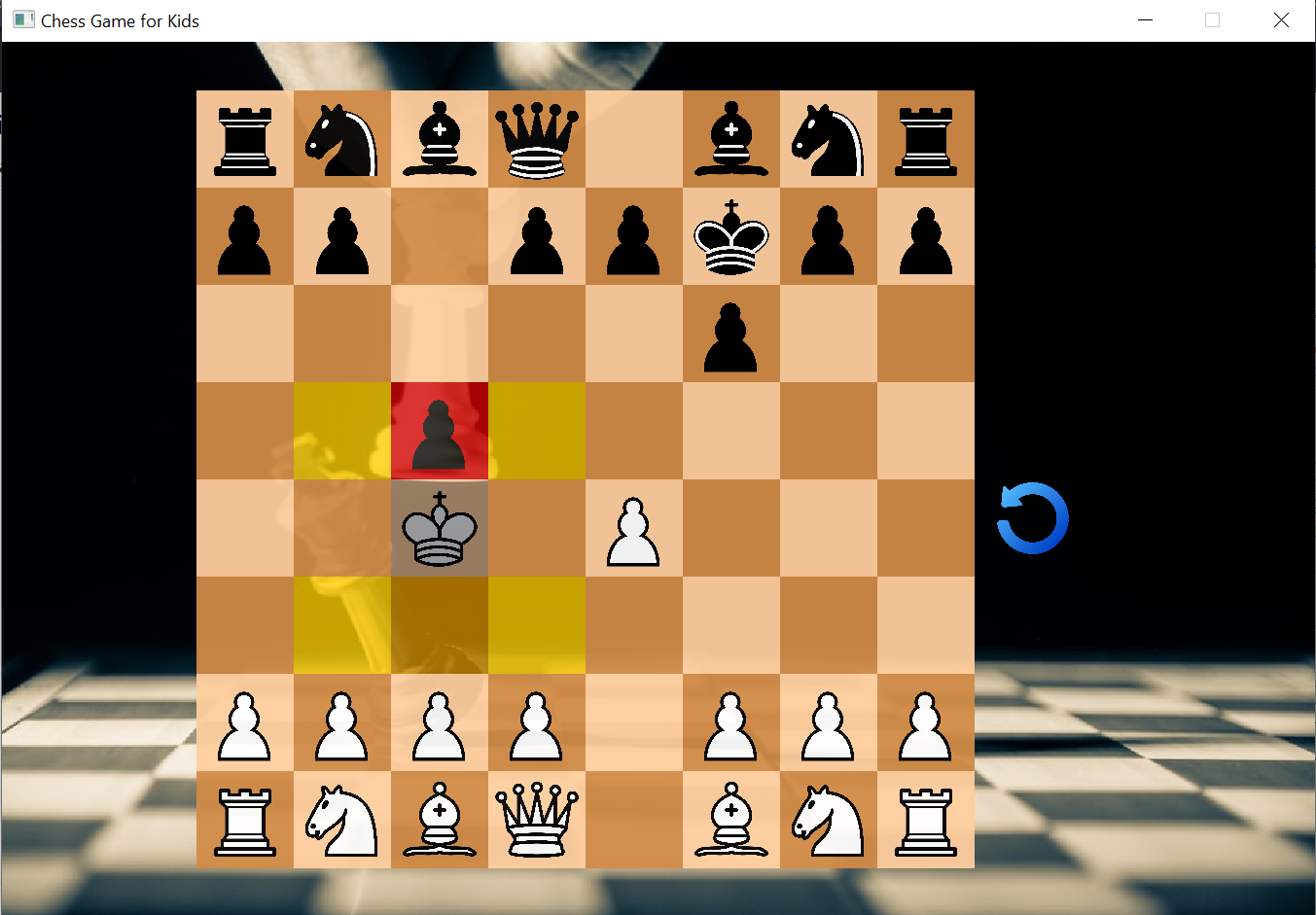


Khi nhấp chuột vào quân cờ, quân cờ được chọn sẽ mờ đi, và các nước đi có thể đi được sẽ xuất hiện, chỉ có thể nhấn vào các ô màu vàng mới có thể di chuyển.

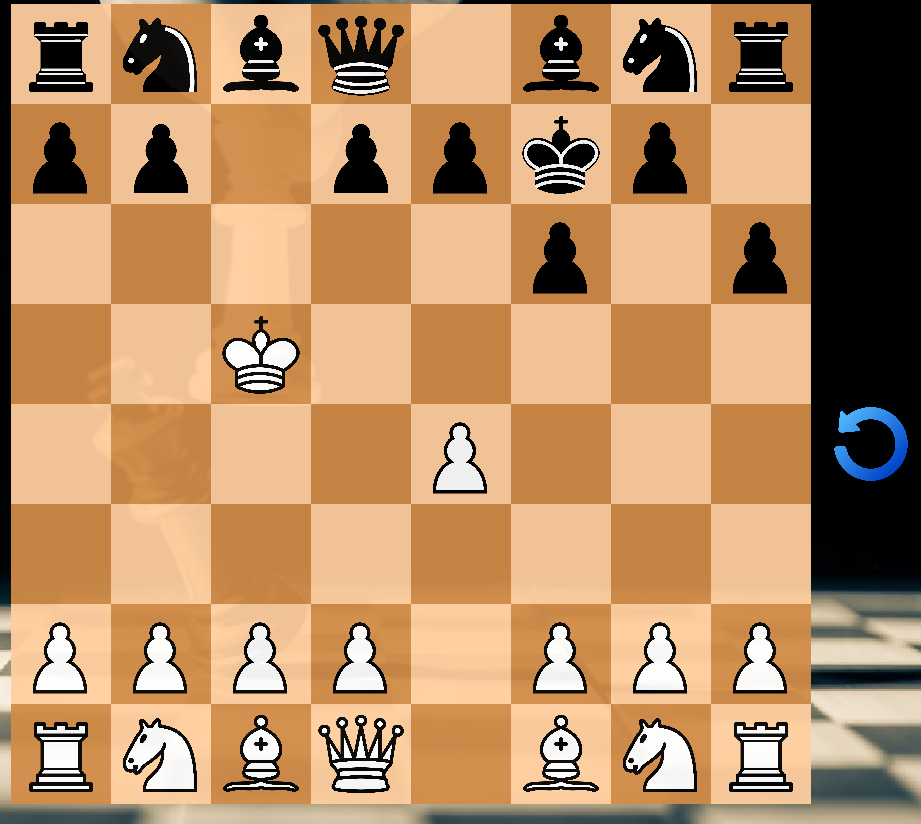


Sau khi thực hiện nước đi của mình, bên máy (Quân đen) sẽ ngay lập tức đi 1 nước cờ.

Chúng ta có thể sử dụng nút quay lại (được đặt ở phía bên phải của bàn cờ để quay lại)



Trong trường hợp này, nếu ta chọn nước đi có màu đỏ thì ta có thể ăn quân đó. Hình dưới biểu diễn cho việc ăn 1 quân cờ (quân Tốt) của quân Vua bên Trắng.



4) Biểu diễn 1 số nước đi:

4.1) Tốt đi 2 ô vào bước đầu:



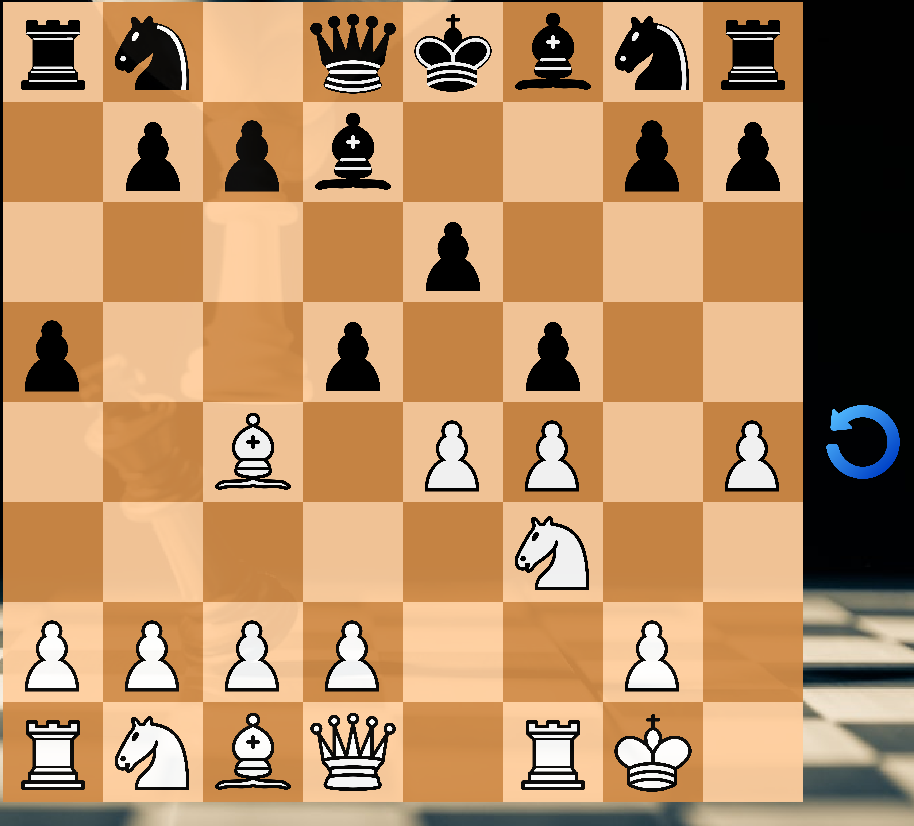
4.2) Tốt ăn chéo:



4.3) Nhập thành ngắn, nhập thành dài:

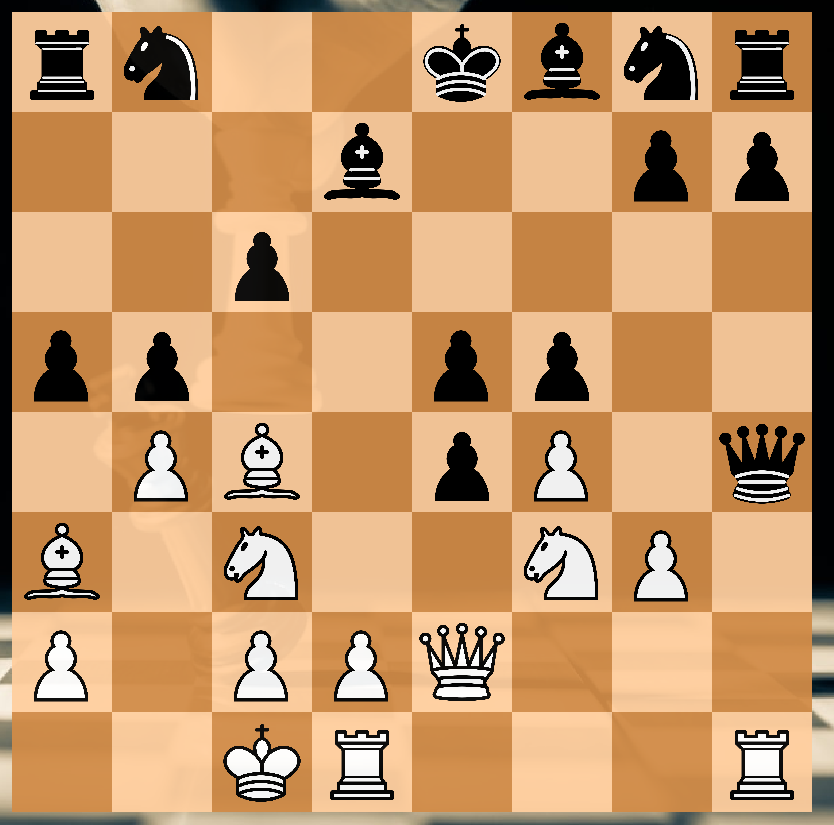
a) Nhập thành ngắn:



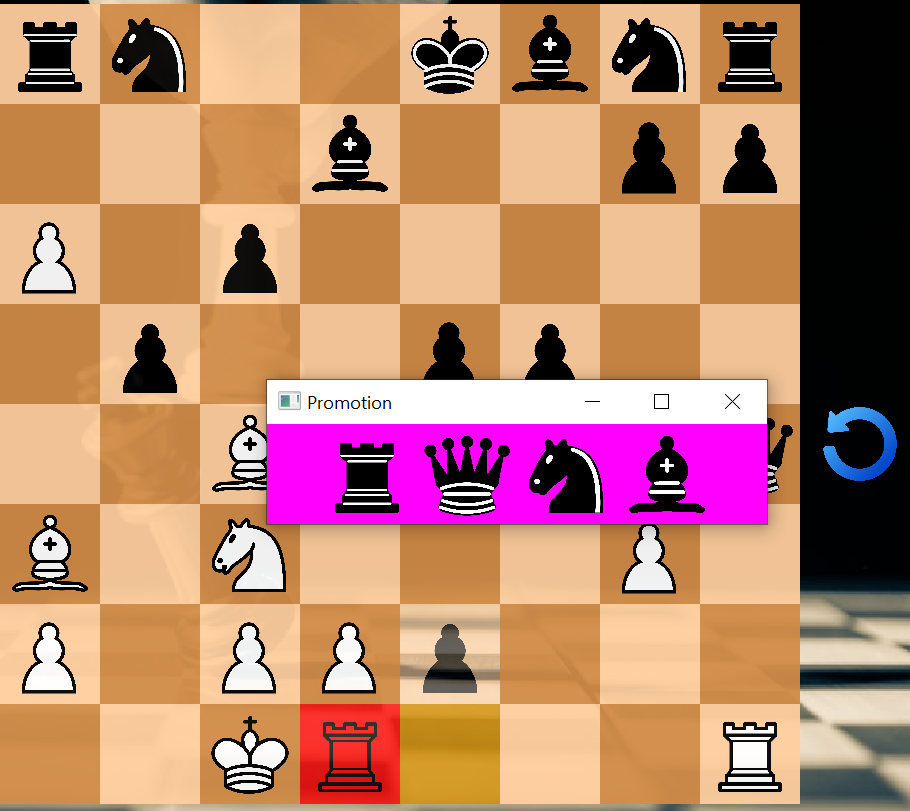


b) Nhập thành dài:

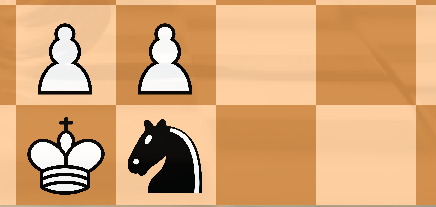




4.4) Phong hậu:



Sau khi quân tốt bên đen đi đến cuối bàn cờ bên quân màu trắng, một cửa sổ sẽ xuất hiện để tiến hành “phong Hậu” cho quân tốt. Ví dụ ở đây ta sẽ chọn quân Mã.



Quân tốt sẽ được “phong Hậu” trở thành quân Mã.

4.5) Thắng khi chiếu hết:

Ở đây ta sử dụng phương pháp Fool’s Mate để thực hiện chiếu tướng nhanh.



Đây là màn hình khi thắng.

D.Phân công công việc của các thành viên:

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Phân công công việc | Phần trăm công việc (100%) |
| 21120421 | Nguyễn Chí Cường | Viết lớp King, Pawn, Square, Board, Move, Game, Vẽ UML, Viết báo cáo | 45% |
| 21120424 | Nguyễn Đình Phương Đại | Viết lớp Rook, Knight, Bishop, Queen, Player, thêm âm thanh cho game, Vẽ UML, Viết báo cáo | 45% |
| 21120611 | Đặng Thế Sinh | Viết lớp Menu | 10% |

E.Tự đánh giá các yêu cầu:

|  |  |
| --- | --- |
| Các tiêu chí | Đánh giá |
| Sơ đồ lớp UML | 100% |
| Cho phép họ chọn quân cờ, nhập vị trí nước đi tiếp theo thông qua bàn phím. | 100% |
| Hiển thị bàn cờ và quân cờ lên màn hình console | 100% |
| Cài đặt được logic của trò chơi (điều kiện thắng, nước đi các quân cờ,…) | 90% (chưa cài đặt nước bắt tốt qua đường, điều kiện hòa) |
| Lưu và chơi lại ván cờ trước đó. | 0% |
| Thao tác thông qua click chuột trên console | 100% |
| Chế độ chơi với máy (random nước đi) | 100% |
| Undo/Redo | 50% (chỉ cài đặt Undo) |
| Replay: xem lại ván đấu vừa đấu | 0% |
| Giao diện người dùng | 100% |
| Âm thanh trò chơi (nhạc nền, di chuyển quân cờ, ăn quân cờ đối phương, kết thúc ván đấu,…) | 100% |

F.Tài nguyên:

[1] Xem các bài viết về thư viện SFML: <https://gamedevspot.net/index.php/forum/user/944-hardingadonis>

[2] Trang chính thức thư viện SFML: <https://www.sfml-dev.org/>

[3] Cách cài đặt thư viện SFML: <https://gamedevspot.net/index.php/forum/do-hoa/38-sfml-tutorial-ca-i-a-t-sfml-ep1-pha-n-1>

[4] Mã nguồn tham khảo: <https://github.com/vanloc1808/HCMUS-OOP-Project-ChessGame.git>