

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
PHƯƠNG PHÁP LẬP TRÌNH
HƯỚNG ĐỐI TƯỢNG
ĐỀ TÀI: Game Cờ vua 2 người

Giảng viên hướng dẫn: ThS. Trần Anh Duy

Lớp: 20CTT1TN2

Thành viên thực hiện:

- 20120131 – Nguyễn Văn Lộc
- 20120209 – Nguyễn Nhật Tiến
- 20120310 – Trà Như Khuyên
- 20120368 – Nguyễn Minh Tâm

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 1 NĂM 2022

Lời nói đầu

Cờ vua là một môn thể thao trí tuệ có nguồn gốc từ Ấn Độ với luật chơi rất đơn giản, dễ hiểu cho người mới chơi nhưng lại đem đến những ván cờ đầy sự thử thách, hấp dẫn từ những người bắt đầu cho đến những đại kiện tướng thế giới. Hằng năm, những giải đấu cờ vua luôn được tổ chức khắp nơi trên thế giới, thu hút lượng người tham gia đông đảo. Nhận thấy sự lôi cuốn của môn thể thao này, với sự hướng dẫn của ThS. Trần Anh Duy, nhóm chúng em quyết định chọn đề tài: **Game Cờ vua 2 người** cho đề án môn học Phương pháp lập trình hướng đối tượng, nhằm tạo ra một phần mềm giúp 2 người có thể luyện tập cờ vua với nhau mọi lúc, mọi nơi mà không cần phải mang bàn cờ theo, qua đó tạo sức hút cho nhiều người đến với môn thể thao trí tuệ này hơn.

Mục lục

1	Thông tin chung về đồ án	4
2	Thư viện SFML	5
2.1	Giới thiệu về thư viện SFML	5
2.2	Các modules của thư viện SFML	5
2.3	Các trang web về thư viện SFML	6
3	Mô tả thiết kế phần mềm	7
3.1	Sơ đồ UML của phần mềm	7
3.2	Mô tả các lớp trong phần mềm	7
3.2.1	Namespace utilities	7
3.2.2	Lớp Settings	7
3.2.3	Các lớp enum	8
3.2.4	Lớp ChessMove	10
3.2.5	Lớp Cell	10
3.2.6	Lớp Piece	11
3.2.7	Lớp King	12
3.2.8	Lớp Queen	13
3.2.9	Lớp Bishop	14
3.2.10	Lớp Knight	15
3.2.11	Lớp Rook	16
3.2.12	Lớp Pawn	17
3.2.13	Lớp AudioPlayer	18
3.2.14	Lớp GameUser	18
3.2.15	Lớp ChessBoard	18

Danh sách hình vẽ

1	Logo của thư viện SFML	5
2	Chi tiết thư viện SFML. Nguồn: gamedevspot.net	6
3	Sơ đồ UML của lớp Settings	7
4	Sơ đồ UML của lớp enum GameSound	8
5	Sơ đồ UML của lớp enum CellStatus	8
6	Sơ đồ UML của lớp enum MoveType	9
7	Sơ đồ UML của lớp enum PieceType	9
8	Sơ đồ UML của lớp enum PieceDirection	9
9	Sơ đồ UML của lớp enum PieceColor	10
10	Sơ đồ UML của lớp ChessMove	10
11	Sơ đồ UML của lớp Cell	10
12	Sơ đồ UML của lớp Piece	11
13	Sơ đồ UML của lớp King	12
14	Các nước đi của quân Vua. Nguồn: chesscorner.com	12
15	Sơ đồ UML của lớp Queen	13

16	Các nước đi của quân Hậu. Nguồn: <code>chesscorner.com</code>	13
17	Sơ đồ UML của lớp <code>Bishop</code>	14
18	Các nước đi của quân Tượng. Nguồn: <code>chesscorner.com</code>	14
19	Sơ đồ UML của lớp <code>Knight</code>	15
20	Các nước đi của quân Mã. Nguồn: <code>chesscorner.com</code>	15
21	Sơ đồ UML của lớp <code>Rook</code>	16
22	Các nước đi của quân Xe. Nguồn: <code>chesscorner.com</code>	16
23	Sơ đồ UML của lớp <code>Pawn</code>	17
24	Các nước đi của quân Tốt. Nguồn: <code>chesscorner.com</code>	17
25	Sơ đồ UML của lớp <code>AudioPlayer</code>	18
26	Sơ đồ UML của lớp <code>GameUser</code>	18
27	Sơ đồ UML của lớp <code>ChessBoard</code>	19

Danh sách bảng

1 Thông tin chung về đề án

Môi trường phát triển: Ubuntu 20.04

Thư viện sử dụng: các thư viện chuẩn của C++ cho phần xây dựng lõi của chương trình, thư viện SFML cho phần giao diện người dùng.

2 Thư viện SFML

Trong đề án này, thư viện SFML được sử dụng để thiết kế giao diện người dùng.

2.1 Giới thiệu về thư viện SFML

SFML (Simple and Fast Multimedia Library) là một thư viện đa phương tiện được đóng góp từ nhiều người ở cộng đồng, được viết chủ yếu bằng ngôn ngữ C++.



Hình 1: Logo của thư viện SFML

Thư viện SFML có vài điểm tương đồng với thư viện SDL2 (Simple DirectMedia Layer 2), nhưng được viết chủ yếu theo phương pháp hướng đối tượng nên việc tiếp cận cho các phần mềm hướng đối tượng sẽ dễ dàng hơn nhiều so với SDL2.

Sử dụng thư viện SFML giúp ta viết được các chương trình có thể chạy trên nhiều nền tảng.

2.2 Các modules của thư viện SFML

Hiện tại, thư viện SFML cung cấp cho người dùng 5 modules:

- **Audio:** cung cấp các lớp giúp xử lý về âm thanh như: phát một tập tin nhạc hoặc tập tin ghi âm...
- **Graphics:** cung cấp các lớp giúp xử lý đồ họa như vẽ hình...
- **Network:** cung cấp các lớp giúp xử lý các giao thức mạng như HTTP, FTP...
- **System:** cung cấp các lớp giúp xử lý các vấn đề hệ thống như thời gian, Unicode...
- **Window:** cung cấp các lớp giúp xử lý cửa sổ sự kiện.



SFML is multi-platform

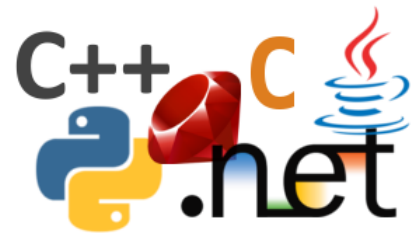
With SFML, your application can compile and run out of the box on the most common operating systems: Windows, Linux, macOS and soon Android & iOS.

Pre-compiled SDKs for your favorite OS are available on the [download page](#).

SFML is multi-language

SFML has official bindings for the C and .Net languages. And thanks to its active community, it is also available in many other languages such as Java, Ruby, Python, Go, and more.

Learn more about them on the [bindings page](#).



Hình 2: Chi tiết thư viện SFML. Nguồn: gamedevspot.net

2.3 Các trang web về thư viện SFML

Trang web chính thức của thư viện SFML, nơi người dùng có thể tìm kiếm tài liệu về các lớp của thư viện cũng như đặt các câu hỏi trong quá trình sử dụng: <https://www.sfml-dev.org/>.

GitHub của thư viện: <https://github.com/SFML/SFML>

3 Mô tả thiết kế phần mềm

3.1 Sơ đồ UML của phần mềm

Do kích thước chiều ngang của giấy có hạn nên sơ đồ UML của phần mềm được đặt trong thư mục **UMLDiagram**, đính kèm với bản báo cáo này.

3.2 Mô tả các lớp trong phần mềm

3.2.1 Namespace utilities

Namespace utilities cung cấp các hàm hỗ trợ cho quá trình xử lý, ví dụ như hàm cung cấp địa chỉ (location) của các hình ảnh, âm thanh, hay hàm kiểm tra vị trí hợp lệ...

3.2.2 Lớp Settings

Lớp Settings cung cấp một vài thông tin cài đặt cơ bản của game. Sơ đồ UML của lớp Settings như sau:



Hình 3: Sơ đồ UML của lớp Settings

Các thuộc tính (attributes) của lớp Settings là thông tin về màu của các ô vuông trên bàn cờ, màu của ô vuông lúc được làm nổi bật (highlight), màu của ô vuông khi bị chiếu. Các màu này được dựa trên mã màu RGBA. Mã màu RGBA trong đồ án được lấy từ trang web: <https://rgbacolorpicker.com>. Ngoài ra, còn có các thuộc tính mô

tả về tỉ lệ, độ dày mỏng của các nét highlight, cũng như kích thước và offset của ô vuông trên bàn cờ.

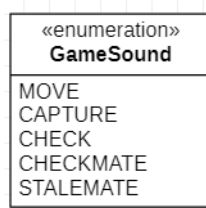
Các phương thức (methods) của lớp **Settings** cung cấp các hàm getters và setters cho các thuộc tính của lớp.

Hầu hết các thuộc tính và phương thức của lớp **Settings** (trừ hàm tạo – constructor và hàm hủy – destructor) đều được khai báo dưới dạng thuộc tính/phương thức tĩnh (method), tạo ra sự tiện lợi khi ta cần gọi chúng, giúp ta không phải tạo một đối tượng mới mỗi khi muốn sử dụng đến các tính năng này.

3.2.3 Các lớp enum

Lớp enum GameSound Lớp **GameSound** cung cấp các loại âm thanh của game: âm thanh khi di chuyển quân cờ (MOVE), khi bắt quân (CAPTURE), khi chiếu (CHECK), khi chiếu hết (CHECKMATE) hay khi stalemate (trạng thái mà cả hai bên đều không còn nước nào có thể đi được).

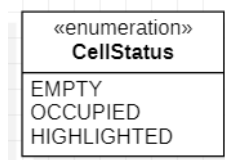
Sơ đồ UML của lớp enum **GameSound** như sau:



Hình 4: Sơ đồ UML của lớp enum **GameSound**

Lớp enum CellStatus Lớp **CellStatus** cung cấp các trạng thái của một ô vuông trên bàn cờ: đã có quân (OCCUPIED), chưa có quân (EMPTY) hay đang được highlight (HIGHLIGHTED).

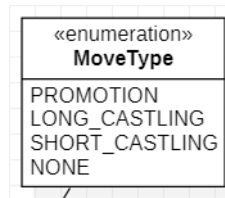
Sơ đồ UML của lớp enum **CellStatus** như sau:



Hình 5: Sơ đồ UML của lớp enum **CellStatus**

Lớp enum MoveType Lớp **MoveType** cung cấp thể loại của các nước di chuyển trong game: thăng cấp cho quân Tốt (PROMOTION), nhập thành ngắn (SHORT_CASTLING), nhập thành dài (LONG_CASTLING), hoặc không phải các nước đi trên (NONE).

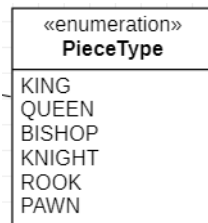
Sơ đồ UML của lớp enum **MoveType** như sau:



Hình 6: Sơ đồ UML của lớp enum MoveType

Lớp enum PieceType Lớp PieceType cung cấp các loại quân cờ: quân Vua (KING), quân Hậu (QUEEN), quân Tượng (BISHOP), quân Mã (KNIGHT), quân Xe (ROOK) và quân Tốt (PAWN).

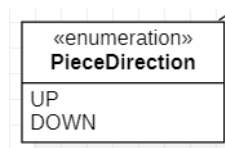
Sơ đồ UML của lớp enum PieceType như sau:



Hình 7: Sơ đồ UML của lớp enum PieceType

Lớp enum PieceDirection Lớp PieceDirection cung cấp hướng di chuyển thẳng về phía trước cho quân Tốt: đi lên (UP) đối với Tốt trắng và đi xuống (DOWN) đối với Tốt đen.

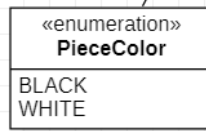
Sơ đồ UML của lớp enum PieceDirection như sau:



Hình 8: Sơ đồ UML của lớp enum PieceDirection

Lớp enum PieceColor Lớp PieceColor cho biết hai màu của người chơi/quân cờ: màu trắng (WHITE) và màu đen (BLACK).

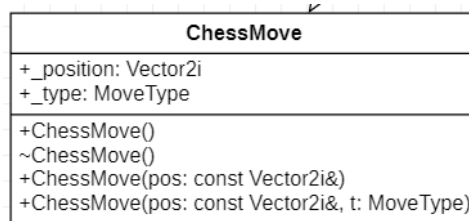
Sơ đồ UML của lớp enum PieceColor như sau:



Hình 9: Sơ đồ UML của lớp enum PieceColor

3.2.4 Lớp ChessMove

Lớp **ChessMove** cung cấp thông tin về nước đi trên bàn cờ.
Sơ đồ UML của lớp **ChessMove** như sau:

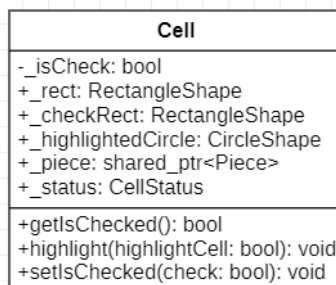


Hình 10: Sơ đồ UML của lớp ChessMove

Các thuộc tính của lớp **ChessMove** cho biết vị trí và thể loại của nước đi đó.
Các phương thức khởi tạo của lớp **ChessMove** cho phép ta khởi tạo mặc định, khởi tạo với một tham số và khởi tạo với đầy đủ tham số.

3.2.5 Lớp Cell

Lớp **Cell** thể hiện mỗi một ô vuông trên bàn cờ vua.
Sơ đồ UML của lớp **Cell** như sau:



Hình 11: Sơ đồ UML của lớp Cell

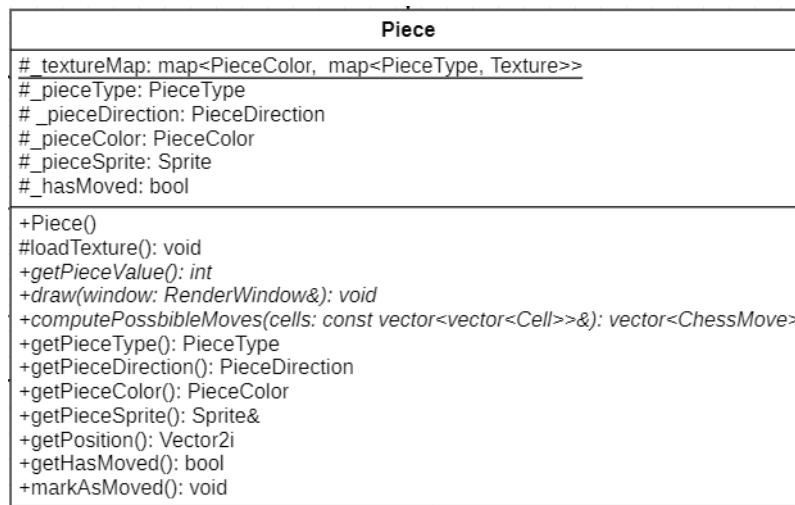
Các thuộc tính chính của lớp **Cell** gồm có thuộc tính để đánh dấu xem ô này có đang bị chiếu hay không, một con trỏ **shared_ptr** trỏ đến một đối tượng thuộc lớp **Piece** (sẽ

được trình bày trong phần sau) – quân cờ hiện tại ở ô này (nếu ô này đang không bị chiếm giữ bởi quân cờ nào thì trả về `nullptr`), hay thuộc tính thể hiện trạng thái hiện tại của ô vuông này. Ngoài ra, các đối tượng từ lớp `RectangleShape` và `CircleShape` của thư viện SFML cũng được sử dụng để tạo các thuộc tính nhằm phục vụ cho việc thiết kế giao diện người dùng (GUI).

Thuộc tính `_piece` có kiểu con trỏ `shared_ptr` của thư viện STL để hạn chế tình trạng rò rỉ bộ nhớ (memory leak).

3.2.6 Lớp Piece

Lớp `Piece` là một lớp trừu tượng, để biểu diễn một quân cờ tổng quát trên bàn cờ vua. Sơ đồ UML của lớp `Piece` như sau:



Hình 12: Sơ đồ UML của lớp `Piece`

Các thuộc tính của lớp `Piece` được khai báo **protected** để cho các lớp kế thừa từ nó có thể truy cập vào các thuộc tính.

Lớp `Piece` có thuộc tính tĩnh `_textureMap` lưu trữ các `Texture` của lần lượt từng quân cờ của mỗi màu. Lớp `Texture` và `Sprite` được sử dụng từ thư viện SFML.

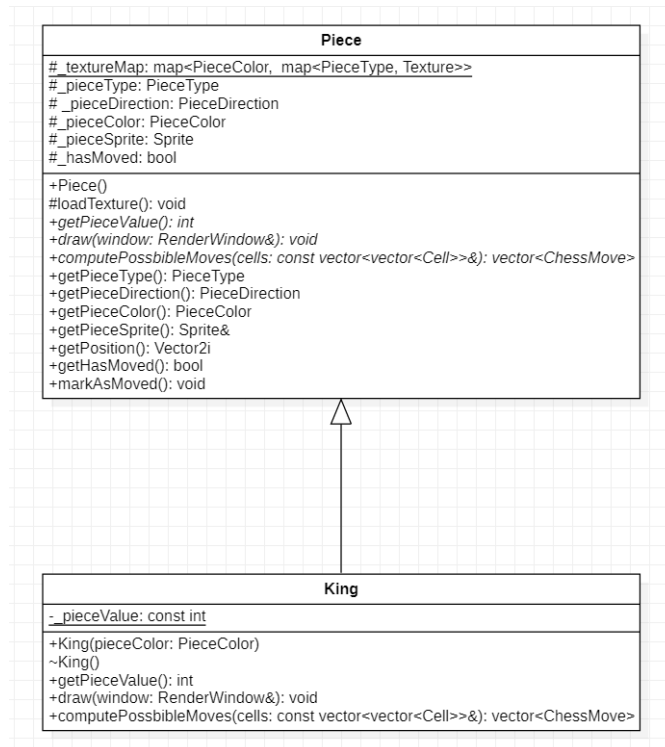
Các thuộc tính khác thể hiện màu sắc, loại quân cờ, hướng di chuyển cùng với một thuộc tính để kiểm tra xem quân cờ này đã di chuyển hay chưa.

Lớp này cung cấp các getters, setters cho các thuộc tính, phương thức dùng để tải (load) các `Texture`. Ngoài ra, lớp `Piece` còn cung cấp các phương thức thuần ảo nhằm tận dụng tính đa hình (polymorphism). Các phương thức thuần ảo này sẽ được cài đặt trong các lớp kế thừa cho phù hợp.

Trong các lớp kế thừa từ lớp `Piece` (6 lớp được trình bày tiếp theo), các thuật toán tìm nước đi khả dĩ (possible moves) của quân cờ được lấy từ: <https://github.com/mbusy/chess/tree/master/src>.

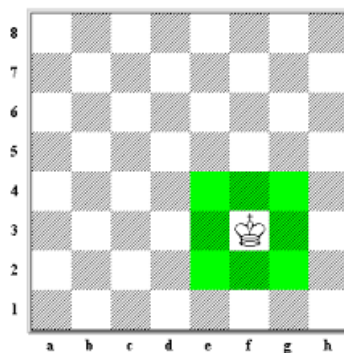
3.2.7 Lớp King

Lớp **King** là một lớp con kế thừa từ lớp **Piece**, thể hiện quân Vua trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp **Piece**. Các phương thức thuần ảo của lớp **Piece** được viết lại trong lớp **King**. Sơ đồ UML của lớp **King** như sau:



Hình 13: Sơ đồ UML của lớp King

Hình sau mô tả các nước đi có thể của quân Vua, được viết trong phương thức `computePossibleMoves`.

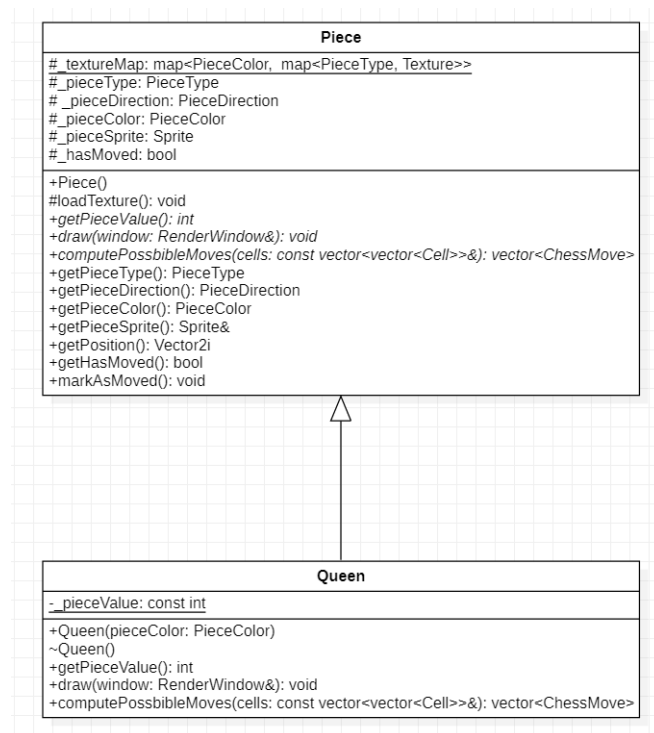


Hình 14: Các nước đi của quân Vua. Nguồn: chesscorner.com

3.2.8 Lớp Queen

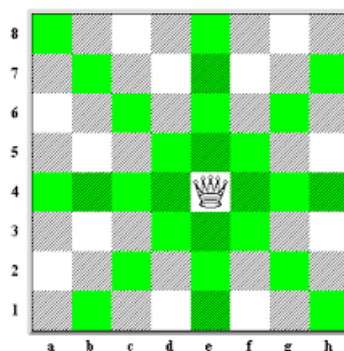
Lớp Queen là một lớp con kế thừa từ lớp Piece, thể hiện quân Hậu trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Queen.

Sơ đồ UML của lớp Queen như sau:



Hình 15: Sơ đồ UML của lớp Queen

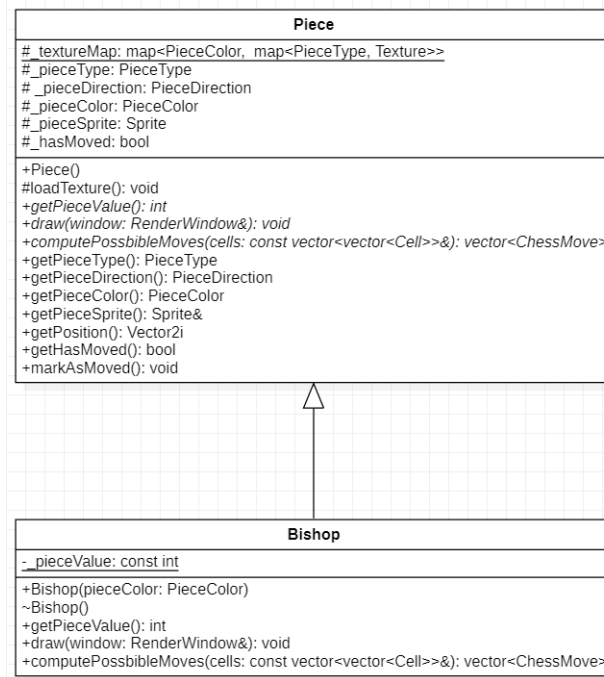
Hình sau mô tả các nước đi có thể của quân Hậu, được viết trong phương thức computePossibleMoves.



Hình 16: Các nước đi của quân Hậu. Nguồn: chesscorner.com

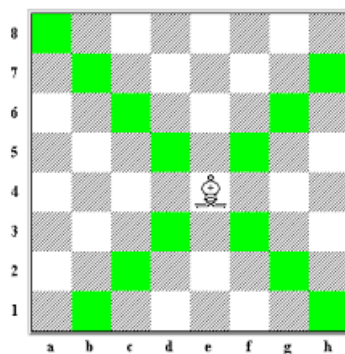
3.2.9 Lớp Bishop

Lớp Bishop là một lớp con kế thừa từ lớp Piece, thể hiện quân Tượng trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Bishop. Sơ đồ UML của lớp Bishop như sau:



Hình 17: Sơ đồ UML của lớp Bishop

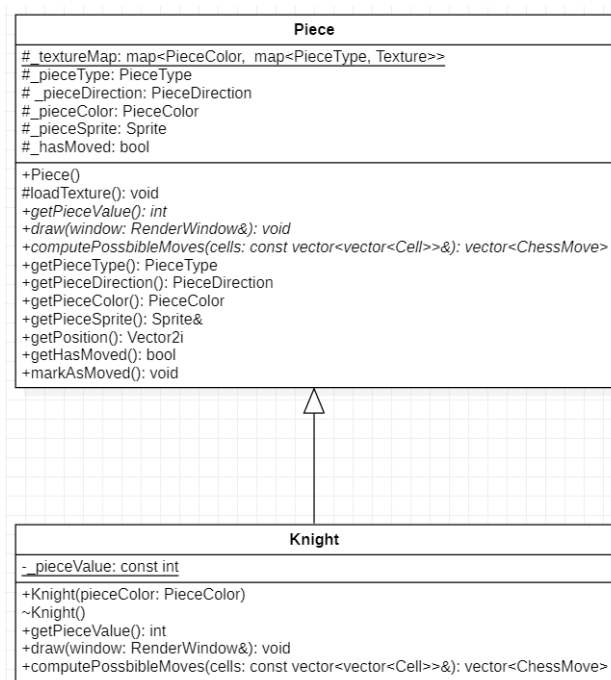
Hình sau mô tả các nước đi có thể của quân Tượng, được viết trong phương thức `computePossibleMoves`.



Hình 18: Các nước đi của quân Tượng. Nguồn: chesscorner.com

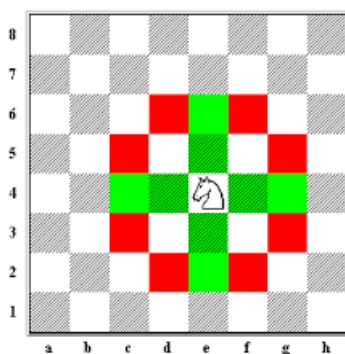
3.2.10 Lớp Knight

Lớp `Knight` là một lớp con kế thừa từ lớp `Piece`, thể hiện quân Mã trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp `Piece`. Các phương thức thuần ảo của lớp `Piece` được viết lại trong lớp `Knight`. Sơ đồ UML của lớp `Knight` như sau:



Hình 19: Sơ đồ UML của lớp `Knight`

Hình sau (các ô màu đỏ) mô tả các nước đi có thể của quân Mã, được viết trong phương thức `computePossibleMoves`.

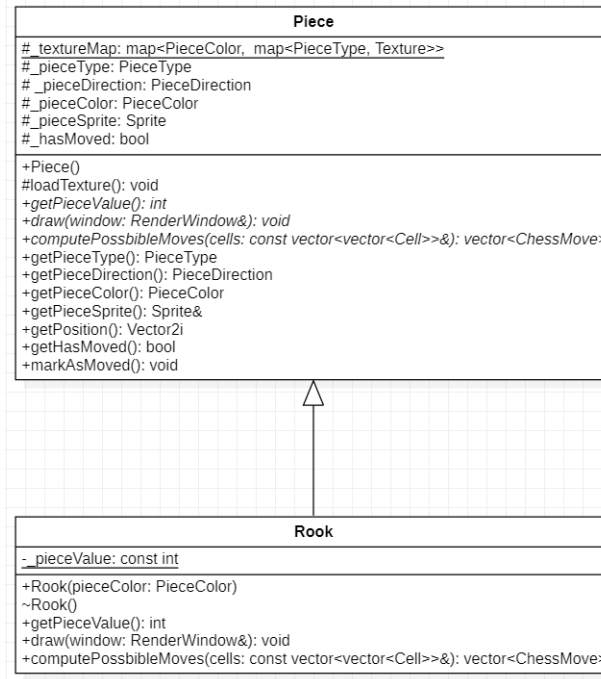


Hình 20: Các nước đi của quân Mã. Nguồn: chesscorner.com

3.2.11 Lớp Rook

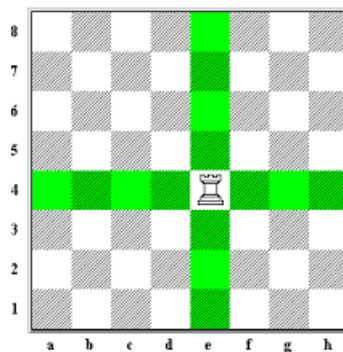
Lớp Rook là một lớp con kế thừa từ lớp Piece, thể hiện quân Xe trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Rook.

Sơ đồ UML của lớp Rook như sau:



Hình 21: Sơ đồ UML của lớp Rook

Hình sau mô tả các nước đi có thể của quân Xe, được viết trong phương thức `computePossibleMoves`.

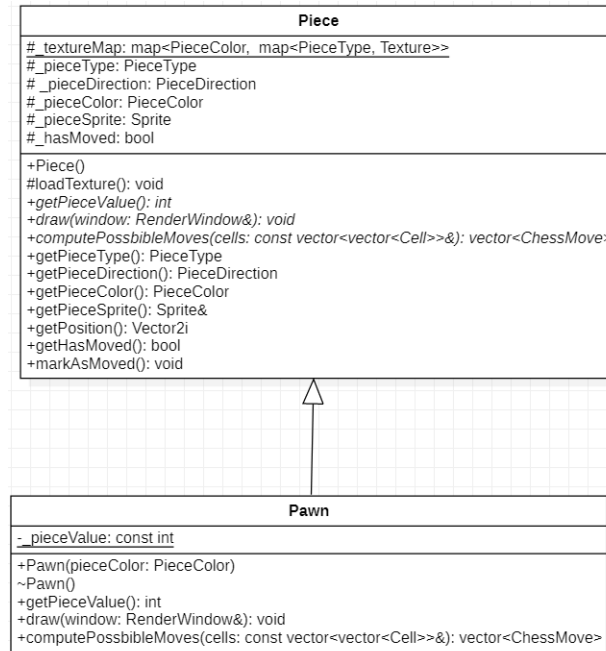


Hình 22: Các nước đi của quân Xe. Nguồn: chesscorner.com

3.2.12 Lớp Pawn

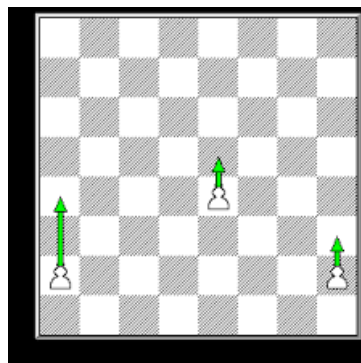
Lớp Pawn là một lớp con kế thừa từ lớp Piece, thể hiện quân Tốt trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Pawn.

Sơ đồ UML của lớp Pawn như sau:



Hình 23: Sơ đồ UML của lớp Pawn

Hình sau mô tả các nước đi có thể của quân Tốt, được viết trong phương thức `computePossibleMoves`.



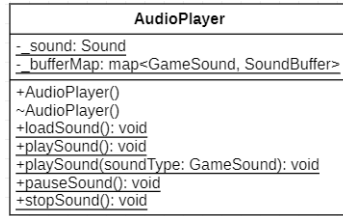
Hình 24: Các nước đi của quân Tốt. Nguồn: chesscorner.com

Việc phong cấp cho quân Tốt sẽ được xử lý ở lớp `ChessBoard`, được trình bày ở phần sau.

3.2.13 Lớp AudioPlayer

Lớp `AudioPlayer` chủ yếu cung cấp cho ta các phương thức liên quan đến việc xử lý âm thanh của game (load, play, pause, resume, stop).

Sơ đồ UML của lớp `AudioPlayer` như sau:

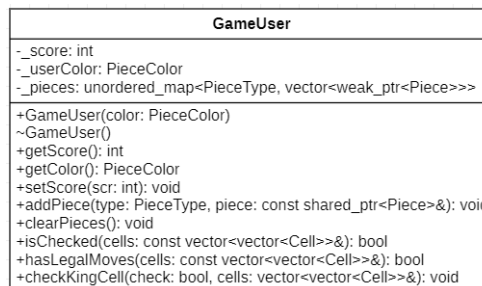


Hình 25: Sơ đồ UML của lớp `AudioPlayer`

3.2.14 Lớp GameUser

Lớp `GameUser` thể hiện người dùng game Cờ vua (bên trắng và bên đen).

Sơ đồ UML của lớp `GameUser` như sau:



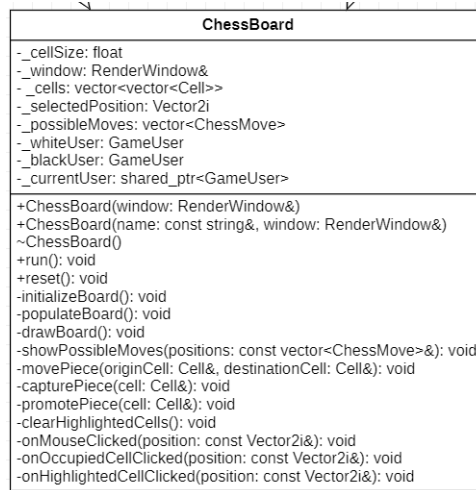
Hình 26: Sơ đồ UML của lớp `GameUser`

Các thuộc tính của lớp `GameUser` gồm có điểm, màu, và các quân cờ. Con trỏ `weak_ptr` của thư viện STL được sử dụng để tham chiếu đến đối tượng được quản lý bởi `shared_ptr`. Các thuộc tính của lớp `GameUser` bao gồm các getters, setters, các hàm giúp ta thêm các quân cờ, kiểm tra xem người dùng có đang bị chiếu hay không, kiểm tra xem người dùng còn nước đi hợp lệ nào hay không, hay phương thức dùng để chiếu người dùng.

3.2.15 Lớp ChessBoard

Lớp `ChessBoard` thể hiện bàn cờ của game Cờ vua trên cửa sổ game.

Sơ đồ UML của lớp `ChessBoard` như sau:



Hình 27: Sơ đồ UML của lớp ChessBoard

Các thuộc tính chủ yếu của lớp **ChessBoard** gồm cửa sổ game, mảng hai chiều các ô vuông trên bàn cờ, các nước đi có thể có, người chơi màu trắng, người chơi màu đen và người chơi hiện tại của game.

Các phương thức có tầm vực **public** của lớp **ChessBoard** gồm các phương thức **run()**, là phương thức chủ yếu để biểu diễn bàn cờ lên GUI, và phương thức **reset()** dùng để reset bàn cờ.

Ngoài ra, lớp **ChessBoard** còn gồm nhiều phương thức có tầm vực **private**:

- Phương thức **initializeBoard** dùng để khởi tạo một bàn cờ.
- Phương thức **populateBoard**, dùng để đặt các quân cờ vào vị trí trên bàn cờ.
- Phương thức **drawBoard**, dùng để hiển thị bàn cờ ra màn hình.
- Phương thức **showPossibleMoves**, dùng để hiển thị những nước đi khả dĩ của quân cờ được chọn hiện tại.
- Phương thức **movePiece**, dùng để di chuyển một quân cờ.
- Phương thức **capturePiece**, dùng để bắt (ăn) một quân cờ khác.
- Phương thức **promotePiece**, dùng để phong cấp cho quân Tốt khi đạt đủ điều kiện.
- Ngoài ra, còn các phương thức xử lý những cú click chuột trên cửa sổ game.

