

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 đồ 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	5
2	Thư viện SFML	6
2.1	Giới thiệu về thư viện SFML	6
2.2	Các modules của thư viện SFML	6
2.3	Các trang web về thư viện SFML	7
3	Mô tả thiết kế phần mềm	8
3.1	Sơ đồ UML của phần mềm	8
3.2	Mô tả các lớp trong phần mềm	8
3.2.1	Namespace utilities	8
3.2.2	Lớp Settings	8
3.2.3	Các lớp enum	9
3.2.4	Lớp ChessMove	11
3.2.5	Lớp Cell	11
3.2.6	Lớp Piece	12
3.2.7	Lớp King	13
3.2.8	Lớp Queen	14
3.2.9	Lớp Bishop	15
3.2.10	Lớp Knight	16
3.2.11	Lớp Rook	17
3.2.12	Lớp Pawn	18
3.2.13	Lớp AudioPlayer	19
3.2.14	Lớp GameUser	19
3.2.15	Lớp ChessBoard	19
4	Mô tả tính năng của phần mềm	21
4.1	Màn hình khởi động của game	21
4.2	Di chuyển quân cờ	21
4.3	Tính năng bắt (ăn) quân	22
4.4	Chiếu	23
4.5	Thăng cấp (Phong cấp) cho quân Tốt	23
4.6	Nhập thành	24
4.7	Kết thúc ván đấu (chiếu hết)	25
4.8	Thoát game	25
5	Hướng dẫn build phần mềm	26
6	Demo của phần mềm	27
7	Nhận xét về đồ án	28
7.1	Ưu điểm	28
7.1.1	Mô tả tính năng phần mềm	28
7.1.2	Mô tả thiết kế phần mềm	28
7.1.3	Các kỹ thuật của lập trình hướng đối tượng	28

7.1.4	Các cấu trúc dữ liệu	28
7.1.5	Giao diện	28
7.1.6	Độ hoàn thành	29
7.1.7	Trình bày	29
7.1.8	Design pattern	29
7.2	Khuyết điểm	29
8	Hướng phát triển	30
9	Phân công công việc của các thành viên	31
10	Phiếu tự đánh giá	33

Danh sách hình vẽ

1	Logo của thư viện SFML	6
2	Chi tiết thư viện SFML. Nguồn: gamedevspot.net	7
3	Sơ đồ UML của lớp <code>Settings</code>	8
4	Sơ đồ UML của lớp enum <code>GameSound</code>	9
5	Sơ đồ UML của lớp enum <code>CellStatus</code>	9
6	Sơ đồ UML của lớp enum <code>MoveType</code>	10
7	Sơ đồ UML của lớp enum <code>PieceType</code>	10
8	Sơ đồ UML của lớp enum <code>PieceDirection</code>	10
9	Sơ đồ UML của lớp enum <code>PieceColor</code>	11
10	Sơ đồ UML của lớp <code>ChessMove</code>	11
11	Sơ đồ UML của lớp <code>Cell</code>	11
12	Sơ đồ UML của lớp <code>Piece</code>	12
13	Sơ đồ UML của lớp <code>King</code>	13
14	Các nước đi của quân Vua. Nguồn: chesscorner.com	13
15	Sơ đồ UML của lớp <code>Queen</code>	14
16	Các nước đi của quân Hậu. Nguồn: chesscorner.com	14
17	Sơ đồ UML của lớp <code>Bishop</code>	15
18	Các nước đi của quân Tượng. Nguồn: chesscorner.com	15
19	Sơ đồ UML của lớp <code>Knight</code>	16
20	Các nước đi của quân Mã. Nguồn: chesscorner.com	16
21	Sơ đồ UML của lớp <code>Rook</code>	17
22	Các nước đi của quân Xe. Nguồn: chesscorner.com	17
23	Sơ đồ UML của lớp <code>Pawn</code>	18
24	Các nước đi của quân Tốt. Nguồn: chesscorner.com	18
25	Sơ đồ UML của lớp <code>AudioPlayer</code>	19
26	Sơ đồ UML của lớp <code>GameUser</code>	19
27	Sơ đồ UML của lớp <code>ChessBoard</code>	20
28	Màn hình khởi động của game	21
29	Di chuyển quân cờ trong game	22
30	Bắt quân cờ của đối thủ	22
31	Chiếu	23

32	Thăng cấp cho quân Tốt	24
33	Nhập thành	24
34	Kết thúc ván đấu (chiếu hết)	25
35	Thông báo người chiến thắng	25

Danh sách bảng

1	Bảng phân công thành viên	32
2	Bảng tự đánh giá	33

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

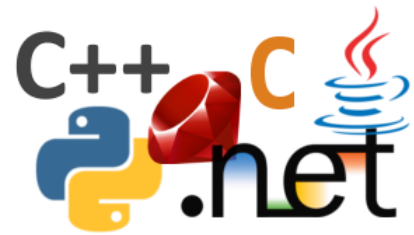
With SFML, your application can compile and run out of the box on the most common operating systems: Windows, Linux, macOS and soon Android & iOS.

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.

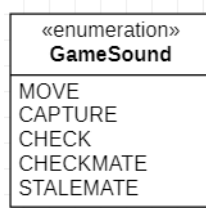
Các phương thức (methods) của lớp **Settings** cung cấp các hàm getters và setters cho các thuộc tính của lớp.

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).

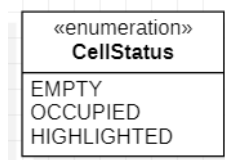
Sơ đồ UML của lớp enum **GameSound** như sau:



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).

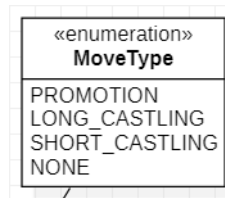
Sơ đồ UML của lớp enum **CellStatus** như sau:



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).

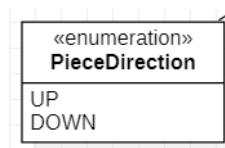
Sơ đồ UML của lớp enum PieceType như sau:



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.

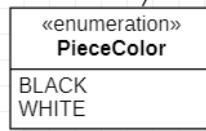
Sơ đồ UML của lớp enum PieceDirection như sau:



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).

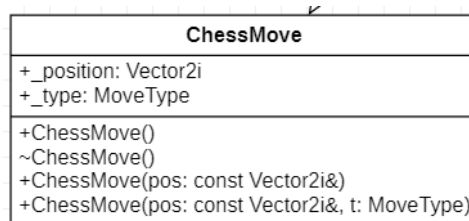
Sơ đồ UML của lớp enum PieceColor như sau:



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:

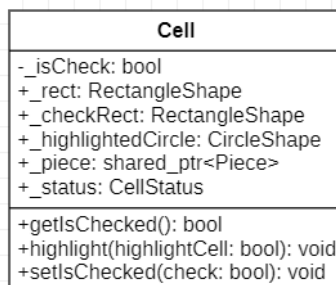


Hình 10: Sơ đồ UML của lớp ChessMove

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

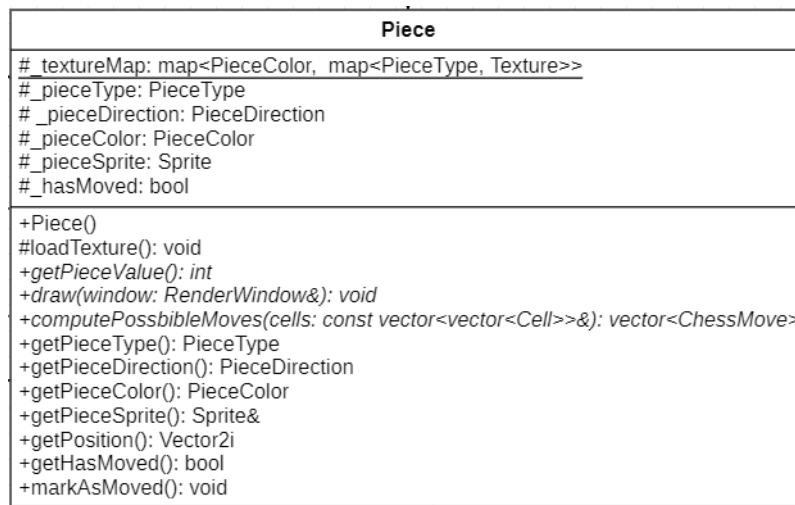
Các thuộc tính chính của lớp **Cell** gồm có thuộc tính để đánh dấu xem ô này có đang bị chiếu hay không, một con trỏ **shared_ptr** trỏ đến một đối tượng thuộc lớp **Piece** (sẽ

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.

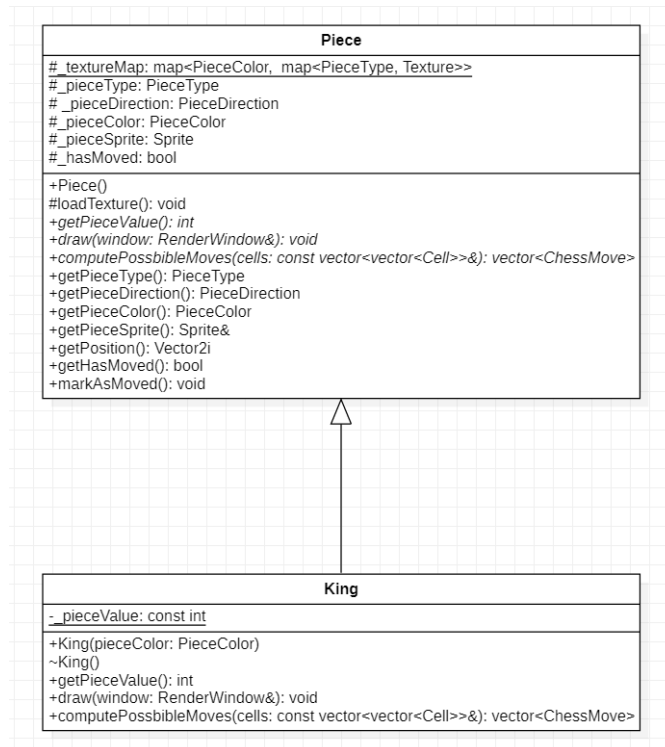
Các thuộc tính khác thể hiện màu sắc, loại quân cờ, hướng di chuyển cùng với một thuộc tính để kiểm tra xem quân cờ này đã di chuyển hay chưa.

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>.

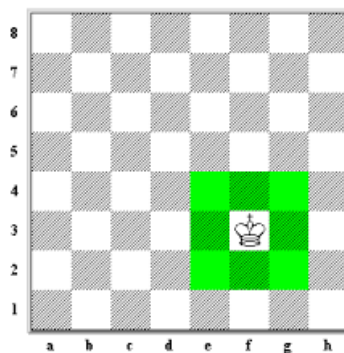
3.2.7 Lớp King

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

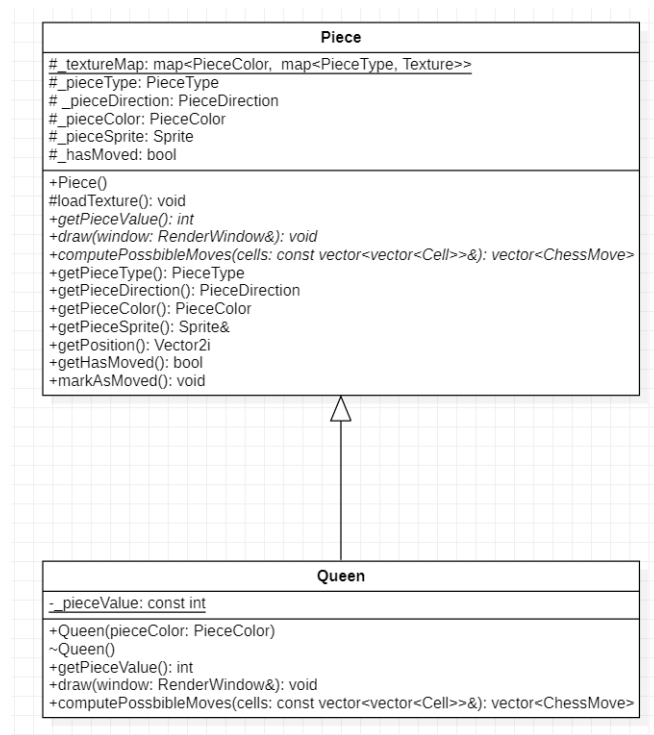
Hình sau mô tả các nước đi có thể của quân Vua, được viết trong phương thức `computePossibleMoves`.



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

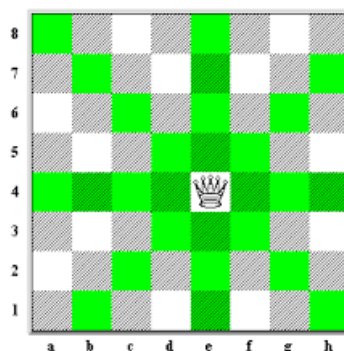
3.2.8 Lớp Queen

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

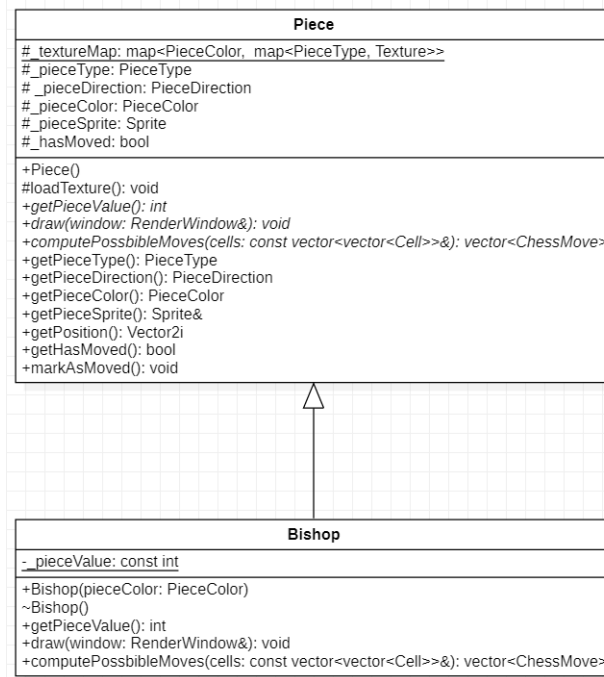
Hình sau mô tả các nước đi có thể của quân Hậu, được viết trong phương thức `computePossibleMoves`.



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

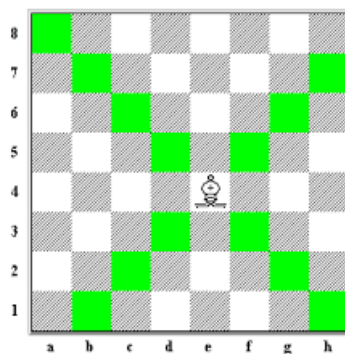
3.2.9 Lớp Bishop

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

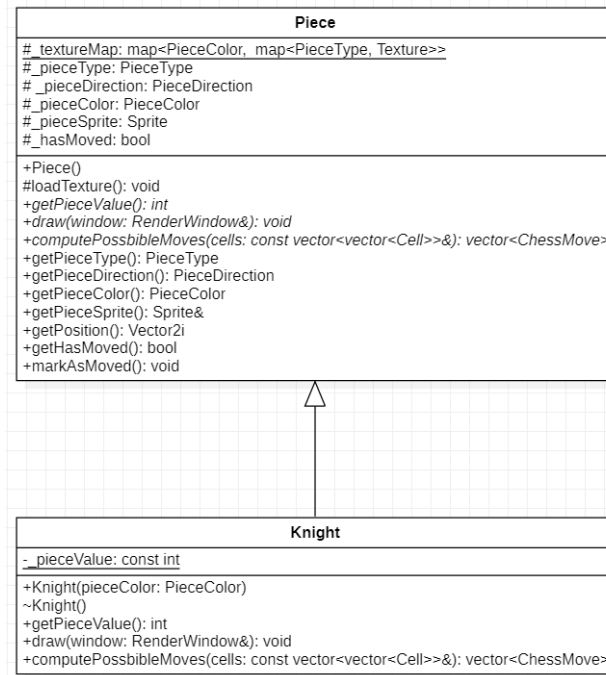
Hình sau mô tả các nước đi có thể của quân Tượng, được viết trong phương thức `computePossibleMoves`.



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

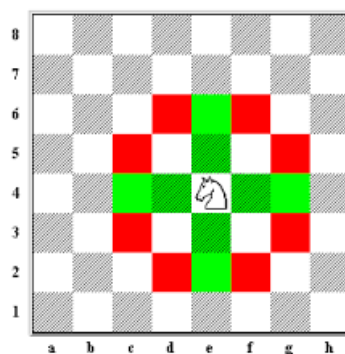
3.2.10 Lớp Knight

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`.

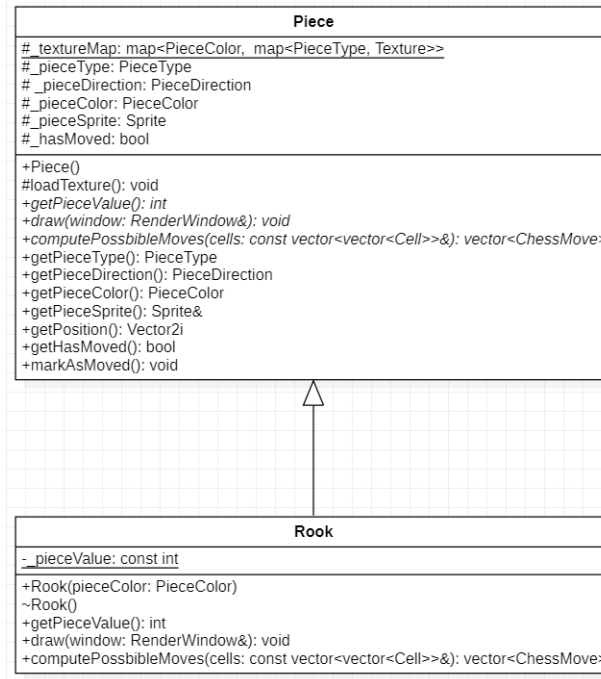


Hình 20: Các nước đi của quân Mã. Nguồn: chesscorner.com

3.2.11 Lớp Rook

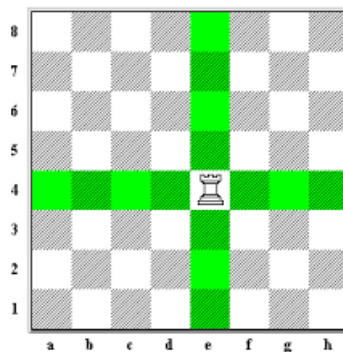
Lớp Rook là một lớp con kế thừa từ lớp Piece, thể hiện quân Xe trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Rook.

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`.

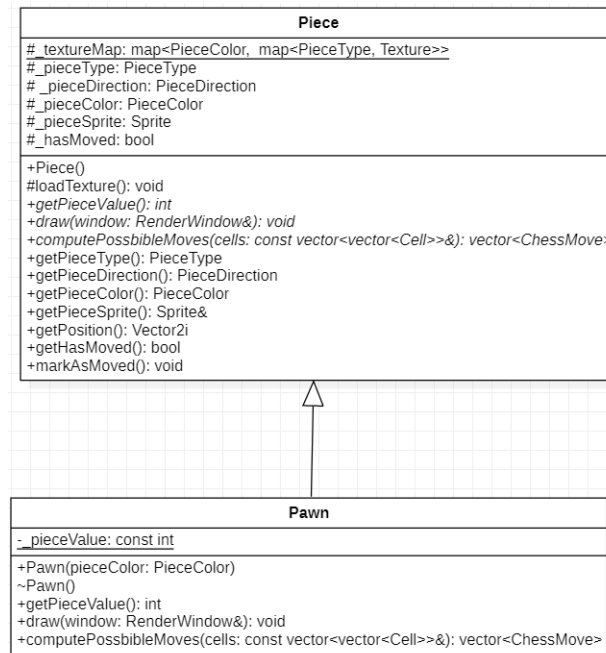


Hình 22: Các nước đi của quân Xe. Nguồn: chesscorner.com

3.2.12 Lớp Pawn

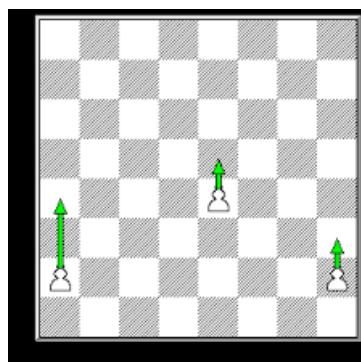
Lớp Pawn là một lớp con kế thừa từ lớp Piece, thể hiện quân Tốt trên bàn cờ. Do là một lớp kế thừa nên nó kế thừa lại tất cả các thuộc tính và phương thức của lớp Piece. Các phương thức thuần ảo của lớp Piece được viết lại trong lớp Pawn.

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.



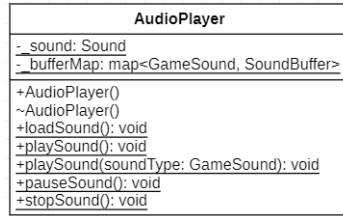
Hình 24: Các nước đi của quân Tốt. Nguồn: chesscorner.com

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:

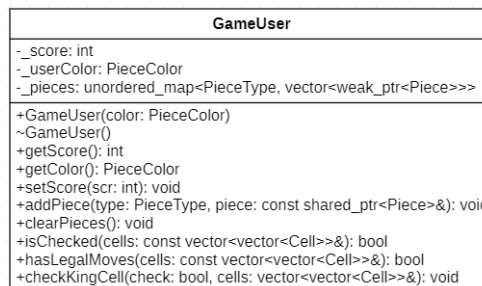


Hình 25: Sơ đồ UML của lớp `AudioPlayer`

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:



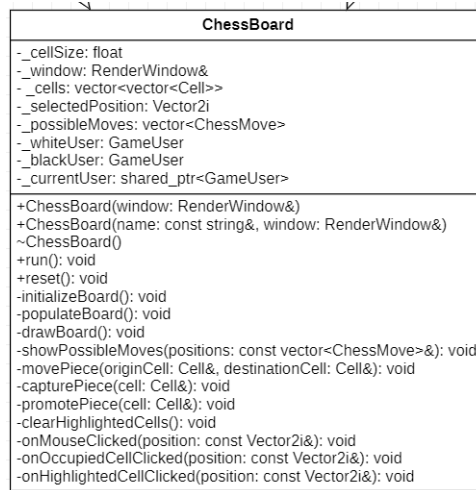
Hình 26: Sơ đồ UML của lớp `GameUser`

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.

Phần cài đặt của lớp **ChessBoard** sử dụng nhiều các lớp và phương thức của chúng trong thư viện SFML.

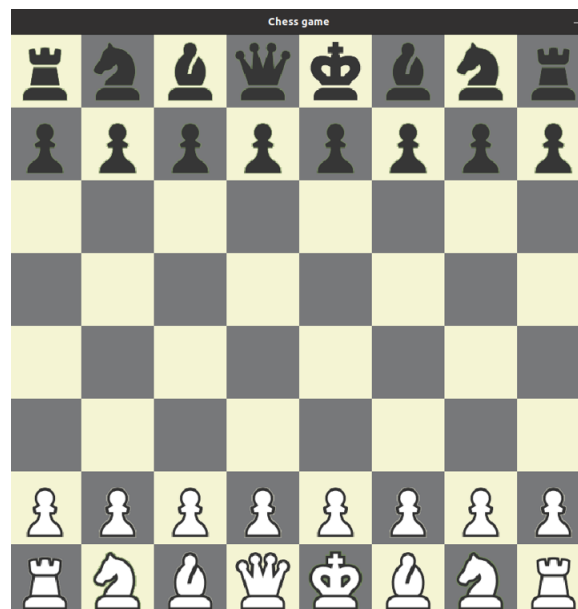
4 Mô tả tính năng của phần mềm

Đồ án Game Cờ vua 2 người cung cấp một phần mềm mô phỏng môn thể thao trí tuệ Cờ vua, với tính năng 2 người chơi. Phần mềm cung cấp đầy đủ các tính năng cơ bản của một game cờ vua như các bước di chuyển, tính năng bắt quân, chiếu, phong cấp cho quân Tốt, nhập thành...

Phần mềm được thiết kế và xây dựng trên nền tảng Linux (Ubuntu 20.04), vì vậy, những tính năng sau đây cũng được ví dụ trên nền tảng này.

4.1 Màn hình khởi động của game

Hình sau đây là màn hình khởi động của game, với đầy đủ các quân cờ đã được sắp xếp vào vị trí, sẵn sàng cho một trận chiến mới.



Hình 28: Màn hình khởi động của game

4.2 Di chuyển quân cờ

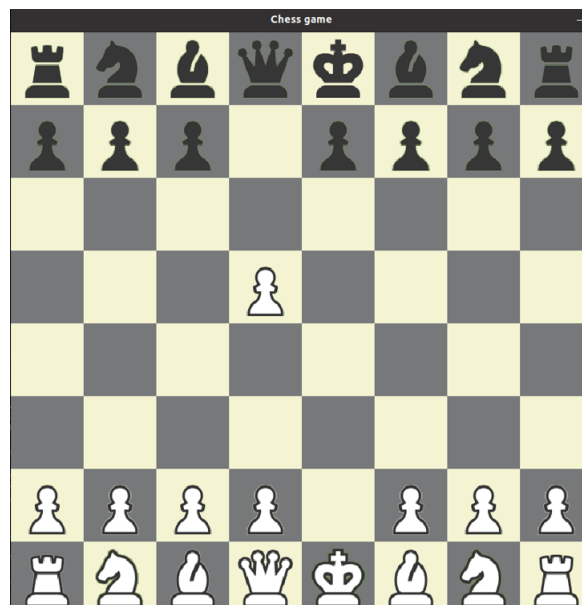
Hình sau đây là nước đi đầu tiên trong game, với quân Tốt trắng tiến lên hai bước.



Hình 29: Di chuyển quân cờ trong game

4.3 Tính năng bắt (ăn) quân

Hình sau đây là ví dụ về tính năng bắt quân theo đường chéo của quân Tốt trắng đối với quân Tốt đen, sau khi quân Tốt đen này tiến vào ô có thể bị bắt bởi quân Tốt trắng. Quân cờ bị bắt sẽ bị loại khỏi bàn chơi ngay lập tức.



Hình 30: Bắt quân cờ của đối thủ

4.4 Chiếu

Hình sau đây mô tả tình huống quân Vua bị chiếu, tức là quân Vua nằm trong một ô có thể bị bắt ở các quân cờ của đối thủ. Ô bị chiếu hiện tại sẽ được tô màu đỏ. Sau khi bị chiếu, quân Vua phải di chuyển đến nơi an toàn, hoặc những quân cờ khác phải di chuyển để "cắt" đường chiếu của quân cờ đối thủ, thậm chí có thể bắt quân cờ này để bảo vệ vua. Nếu không thể, ván đấu sẽ kết thúc và bên bị chiếu sẽ thua cuộc.



Hình 31: Chiếu

4.5 Thăng cấp (Phong cấp) cho quân Tốt

Khi quân Tốt tiến đến hàng cuối cùng trên "địa phận" của đối thủ, nó sẽ được thăng cấp thành một trong bốn quân: Hậu, Xe, Tượng, Mã. Hình sau đây mô tả một ví dụ về tính năng thăng cấp cho quân Tốt này.



Hình 32: Thăng cấp cho quân Tốt

4.6 Nhập thành

Khi quân Vua và quân Xe của một bên chưa di chuyển, các ô giữa chúng đều trở thành ô trống và ô đích đến không bị chiếu thì quân Vua có thể tiến hành nhập thành. Có hai loại nhập thành, ứng với hai quân Xe, lần lượt là long castling (tạm dịch là "nhập thành dài") – ứng với quân Xe ở xa Vua hơn và short castling (tạm dịch là "nhập thành ngắn") – ứng với quân Xe ở gần Vua hơn. Hình sau đây cho ta một ví dụ về nhập thành dài.



Hình 33: Nhập thành

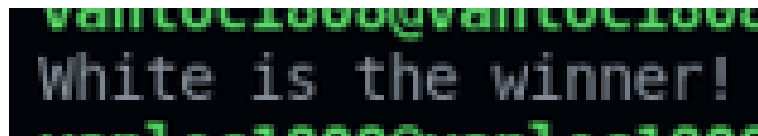
4.7 Kết thúc ván đấu (chiếu hết)

Khi quân Vua bị chiếu và không còn nước đi nào khác có thể "cứu" quân Vua ra khỏi ô bị chiếu này, ván đấu sẽ kết thúc với chiến thắng cho bên chiếu. Hình sau đây cho ta một ví dụ về bàn cờ lúc kết thúc ván đấu.



Hình 34: Kết thúc ván đấu (chiếu hết)

Khi tình huống này xảy ra, một thông báo sẽ hiện lên màn hình console, cho ta biết người chiến thắng.



Hình 35: Thông báo người chiến thắng

4.8 Thoát game

Ta có thể thoát game bằng cách tắt cửa sổ game hoặc nhấn phím Escape (Esc) ở góc trên bên trái bàn phím.

5 Hướng dẫn build phần mềm

Phần mềm được thiết kế và xây dựng cho môi trường Linux, cho nên phần hướng dẫn này chỉ tập trung hướng dẫn build phần mềm trên môi trường này.

Trước tiên, để build phần mềm trên môi trường này, chúng ta cần cài đặt thư viện SFML cho Linux bằng câu lệnh:

```
sudo apt-get install libsFML-dev
```

Nếu câu lệnh trên không hoạt động, ta có thể cài đặt thư viện này bằng những cách khác theo hướng dẫn trong link sau: <https://www.sFML-dev.org/tutorials/2.5/start-linux.php>

Do phần mềm đã hỗ trợ Makefile, nên sau khi cài đặt thư mục SFML thành công, ta chỉ cần mở thư mục chứa source code và dùng lệnh:

```
make
```

Sau khi hoàn thành, tập tin thực thi có tên là **chess** sẽ xuất hiện. Ta chỉ cần dùng lệnh sau để khởi chạy tập tin này:

```
./chess
```

6 Demo của phần mềm

Nhóm đã tiến hành quay video demo của phần mềm, gồm có sơ đồ UML, giới thiệu các file trong source code, hướng dẫn build phần mềm, demo một ván đấu cờ vua.

Link demo phần mềm: <https://youtu.be/Pev0-ZqRduw>

7 Nhận xét về đồ án

7.1 Ưu điểm

7.1.1 Mô tả tính năng phần mềm

Báo cáo đã mô tả đầy đủ các tính năng của phần mềm.

7.1.2 Mô tả thiết kế phần mềm

Sơ đồ UML của phần mềm và mô tả các lớp đã được trình bày trong báo cáo và hình ảnh sơ đồ UML đính kèm.

7.1.3 Các kỹ thuật của lập trình hướng đối tượng

Đồ án đã ứng dụng được nhiều kỹ thuật trong lập trình hướng đối tượng.

- Hàm dựng (constructor): hầu hết các lớp đều được xây dựng với ít nhất một hàm dựng.
- Hàm hủy (destructor): các lớp sử dụng các lớp con trở "an toàn" của thư viện STL nên hầu hết các hàm hủy đều được để ở dạng mặc định, nhưng chúng cũng được khai báo một cách tường minh trong lớp.
- Tính đóng gói (encapsulation): các lớp đều tuân thủ tính đóng gói của lập trình hướng đối tượng, tuân theo quy tắc hộp đen và quy tắc "Tell, don't ask".
- Tính kế thừa (inheritance): ứng dụng được mối quan hệ tổng quát hóa/đặc biệt hóa (IS-A, generalization), mối quan hệ bao hàm/bộ phận (association), đặc biệt là mối quan hệ bao hàm/bộ phận độc lập (aggregation).
- Tính đa hình (polymorphism): ứng dụng được tính đa hình trong việc xây dựng các phương thức của các quân cờ.
- Phương thức thuần ảo (pure virtual method): ứng dụng được hàm thuần ảo trong lớp `Piece`.
- Lớp trừu tượng (abstract class): tạo dựng lớp `Piece` là lớp trừu tượng, các lớp kế thừa từ `Piece` sẽ khai báo các phương thức thuần ảo của lớp `Piece`.
- Sử dụng kỹ thuật try-throw-catch, giúp phát hiện và xử lý lỗi và ngoại lệ (exception).

7.1.4 Các cấu trúc dữ liệu

Các cấu trúc dữ liệu sử dụng trong đồ án hầu hết đều từ thư viện STL, giúp ta xử lý nhanh chóng và tránh tình trạng rò rỉ bộ nhớ (memory leak).

7.1.5 Giao diện

Đồ án đã xây dựng một giao diện người dùng (GUI) trực quan, sinh động, minh họa đầy đủ các tính năng của môn Cờ vua.

7.1.6 Độ hoàn thành

Đề án đã hoàn thành hầu hết các chỉ tiêu đặt ra.

7.1.7 Trình bày

Báo cáo đã trình bày rõ ràng, chi tiết về đề án.

7.1.8 Design pattern

Đề án đã ứng dụng mẫu thiết kế Iterator trong quá trình duyệt qua các phần tử của nhiều cấu trúc dữ liệu khác nhau.

7.2 Khuyết điểm

Trong quá trình xây dựng, nhóm đã thử tính năng đánh với máy (sử dụng stockfish) nhưng đã gặp lỗi và chưa cung cấp được.

Chưa thiết kế được cơ sở dữ liệu hỗ trợ quản lý tài khoản, nên tính năng quản lý tài khoản chưa được đưa vào phần mềm.

Chưa cung cấp tính năng đi lại cho các nước đi trong ván cờ.

8 Hướng phát triển

Nghiên cứu áp dụng lại stockfish hoặc một engine khác để xây dựng tính năng chơi với máy, giúp việc luyện tập cờ vua trở nên dễ dàng hơn với người sử dụng.

Nghiên cứu xây dựng cơ sở dữ liệu cho việc quản lý tài khoản.

Nghiên cứu xây dựng tính năng cho phép người chơi đi lại sau mỗi nước đi.

Nghiên cứu cải tiến giao diện người dùng, cung cấp nhiều tính năng hơn nữa.

9 Phân công công việc của các thành viên

Bảng 1 dưới đây mô tả thông tin phân công các thành viên của nhóm.

Thành viên	Công việc	Phần trăm đóng góp
Nguyễn Văn Lộc	<ul style="list-style-type: none"> • Lên ý tưởng thiết kế cho phần mềm. • Xây dựng các lớp Pawn, Cell, AudioPlayer, GameSound, ChessMove, ChessBoard, Settings. • Thiết kế giao diện người dùng (GUI) cho phần mềm. • Tester chính cho phần mềm. • Viết báo cáo chính cho phần mềm. 	34
Nguyễn Nhật Tiến	<ul style="list-style-type: none"> • Tham gia thiết kế sơ đồ UML của chương trình. • Tham gia viết báo cáo cho phần mềm. • Xây dựng các lớp Knight, Rook, ChessBoard. • Tìm kiếm tài nguyên hình ảnh, âm thanh cho phần mềm. 	21
Trà Như Khuyên	<ul style="list-style-type: none"> • Tham gia thiết kế sơ đồ UML của chương trình. • Tham gia viết báo cáo cho phần mềm. • Xây dựng các lớp Piece, King, GameUser, ChessBoard. • Tham gia xây dựng namespace Utility. 	23

Nguyễn Minh Tâm	<ul style="list-style-type: none">• Tham gia thiết kế sơ đồ UML của chương trình.• Tham gia viết báo cáo cho phần mềm.• Xây dựng các lớp Queen, Bishop, ChessBoard.• Tham gia xây dựng namespace Utility.	22
-----------------	--	----

Bảng 1: Bảng phân công thành viên

10 Phiếu tự đánh giá

Bảng 2 dưới đây là phiếu tự đánh giá các tiêu chí của nhóm.

STT	Tiêu chí	Điểm	Đánh giá
1	Tài liệu mô tả phần mềm	0.5	0.5
2	Tài liệu thiết kế phần mềm	3	3
3	Ứng dụng được ít nhất 5 kỹ thuật đã học trong môn Phương pháp lập trình hướng đối tượng	3	3
4	Giao diện đẹp	1	0.75
5	Độ hoàn thiện	1	0.75
6	Trình bày	1	0.75
7	Cơ sở dữ liệu	0.5	0
8	Ứng dụng được ít nhất một mẫu design pattern	1	0.5
	Tổng điểm	11	10

Bảng 2: Bảng tự đánh giá

Lời cảm ơn

Những kiến thức về lập trình hướng đối tượng được ứng dụng trong đồ án của chúng em được Thầy Nguyễn Minh Huy giảng dạy rất nhiệt tình, tâm huyết. Trong quá trình thực hiện đồ án, chúng em đã nhận được những sự hướng dẫn, những góp ý tận tâm của Thầy Trần Anh Duy. Chúng em xin cảm ơn hai Thầy vì đồ án này nói riêng và vì những kiến thức mà hai Thầy đã cung cấp nói chung ạ.

Bên cạnh đó, trong quá trình thực hiện đồ án, nhóm đã nhận được những lời góp ý từ các bạn cùng lớp. Nhóm xin gửi lời cảm ơn chân thành đến các bạn.

Ngoài ra, nhóm cũng đã tham khảo từ nhiều trang web như GitHub, trang web chính thức của thư viện SFML. Những thuật toán, những ý kiến giải đáp trên các diễn đàn trên đã góp phần giúp nhóm thực hiện đồ án này.

Thành phố Hồ Chí Minh, tháng 1 năm 2022