**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**Game Cờ Vua**

**Môn: Kiến trúc phần mềm**

**GVLT : Trần Minh Triết**

**NGƯỜI THỰC HIỆN :** 1612078 - Nguyễn Đình Hoàng Đắc

1612173 – Đặng Anh Hào

1612543 – Phạm Anh Quốc

***Tp. Hồ Chí Minh - Tháng 1/2020***

ĐỒ ÁN CUỐI KÌ

Game Cờ Vua

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Khoa Công nghệ Thông tin

Đại học Khoa học Tự nhiên, ĐHQG-HCM

Tháng 01/2020

**MỤC LỤC**

[1 Kiến trúc của game 4](#_Toc30178742)

[1.1 Sơ đồ lớp 4](#_Toc30178743)

[1.2 Giải thích ý nghĩa 4](#_Toc30178744)

[2 Cấu trúc framework 4](#_Toc30178745)

# Giới thiệu chung

## Nhóm

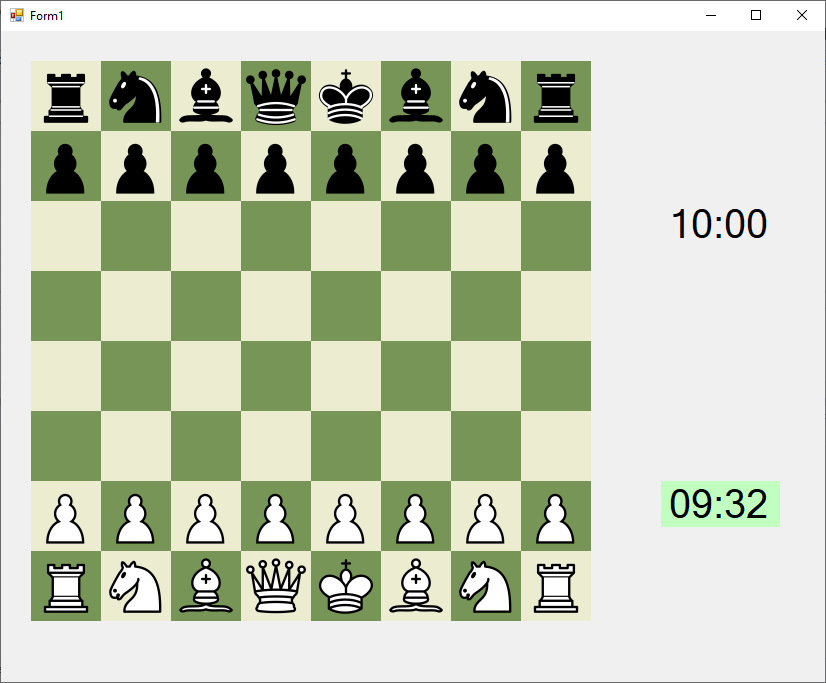
Thông tin thành viên nhóm: Nhóm gồm 3 thành viên:

|  |  |  |
| --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** |
| 1 | 1612078 | Nguyễn Đình Hoàng Đắc |
| 2 | 1612173 | Đặng Anh Hào |
| 3 | 1612543 | Phạm Anh Quốc |

## Đồ án

Game cờ vua **được xây dựng cho môn học Kiến trúc phần mềm**. Game cờ vua là game chơi cờ vua. Game có 2 chế độ chơi offline, chơi với người và chơi với máy. Nhóm em có phát triển tính năng chơi online qua mạng LAN nhưng chưa hoàn thành.

Mở đầu trò chơi, người chơi có thể chọn chế độ chơi, chọn cờ trắng hoặc cờ đen, chọn thời gian trận đấu. Trò chơi có giao diện đơn giản, bao gồm bàn cờ và 2 đồng hồ đếm giờ cho mỗi bên.



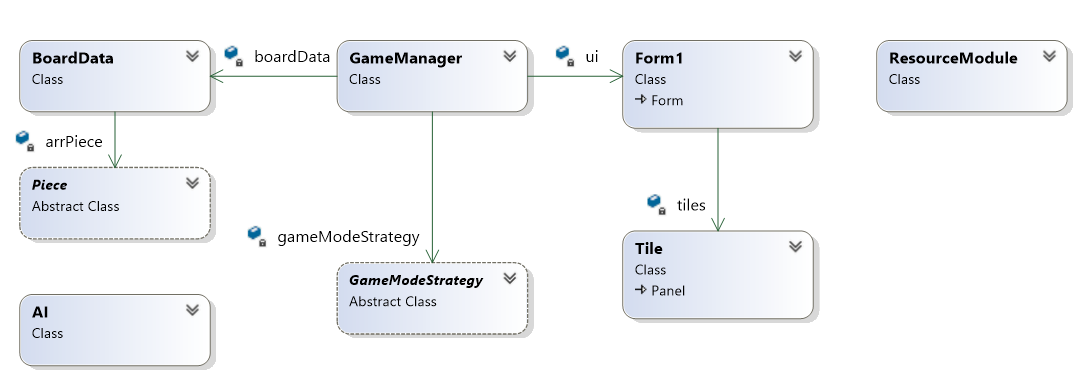
*Hình 1. Giao diện chính của Game*

Người chơi di chuyển quân cờ bằng cách click vào quân cờ muốn di chuyển, sau đó click vào ô muốn đi tới. Chỉ những nước đi hợp lệ mới được chấp nhận.

Trò chơi tuân theo luật cờ vua thông thường, tuy nhiên vẫn còn thiếu một số nước đi đặc biệt: Bắt Tốt qua đường, Nhập Thành, Phong Hậu và luật Hòa Cờ.

# Kiến trúc của game

## Các thành phần chính trong game



*Hình 2. Sơ đồ các lớp chính trong game*

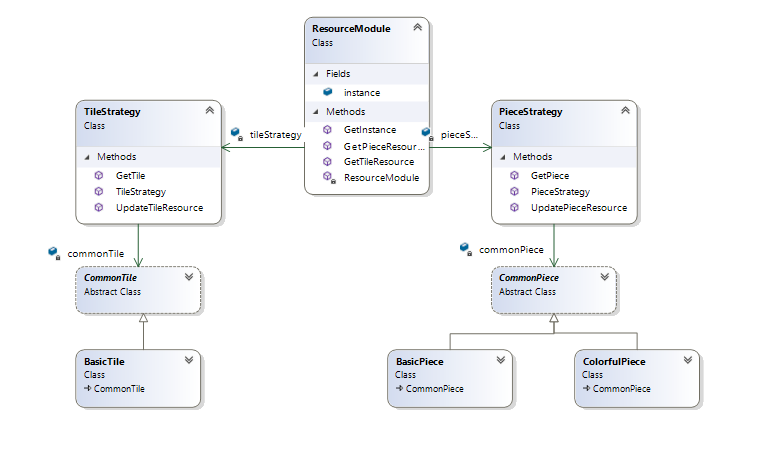
Kiến trúc của game bao gồm các lớp chính:

* Các lớp BoardData, Piece, AI: là các lớp quản lý về dữ liệu và xử lý logic cờ vua trong game. Lớp BoardData chứa dữ liệu về trạng thái bàn cờ, vị trí các quân cờ, xử lý logic về các nước đi trong game, kiểm tra tính hợp lệ của nước đi, kiểm tra nước chiếu, chiếu bí. Lớp Piece xử lý logic về các nước đi có thể có của một quân cờ. Lớp AI thực hiện các tính toán để tự động chọn ra nước đi tối ưu.
* Các lớp Form1, Tile: là các lớp xử lý về mặt giao diện và hiển thị, trong đó lớp Form1 quản lý giao diện chính, lớp Tile là lớp các ô cờ trên bàn cờ, xử lý các sự kiện click chuột.
* Lớp GameModeStrategy và các lớp con của nó: Xử lý về mặt logic của trò chơi tùy vào chế độ chơi (2 người, chơi với máy hoặc chơi online).
* Lớp ResourceModule là lớp phụ trách việc load và cung cấp hình ảnh các thành phần trong game.
* Cuối cùng là lớp GameManager là lớp trung gian, kết nối các thành phần dữ liệu, logic, giao diện.

Ngoài ra còn có kiến trúc về phần Network, đợi bạn ĐẮc gửi

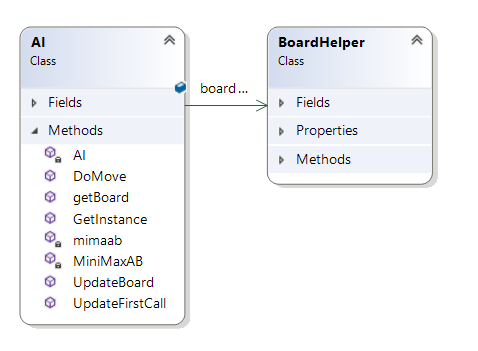
## Các mẫu thiết kế được áp dụng trong đồ án

### Áp dụng mẫu thiết kế Strategy vào lấy resources



***Giải thích:*** Các resource trong lúc khởi tạo bàn cờ và các con cờ được lớp **ResourceModule** quản lý và trả về theo yêu cầu. Lớp **ResourceModule** cung cấp các phương thức cần thiết (*GetPieceResource, GetTileResource*) để lấy resource. Trong lớp ResourceModule có chứa 1 thể hiện của lớp **PieceStrategy** và 1 thể hiện của lớp **TileStrategy**, khi cần lấy các resource thì sẽ hủy thác cho 2 thể hiện của 2 lớp này lấy. Hai lớp **PieceStrategy** và **TileStrategy** có thể lấy resource từ các nguồn khác nhau, miễn sao là trả về đúng yêu cầu. Khi cần thay đổi nguồn lấy resource, ta chỉ cần Update strategy, hoặc khi ta muốn thêm một resource khác, ta chỉ cần kế thừa lớp **CommonPiece (CommonTile)** và tạo 1 lớp mới, implement các phương thức đã có, mà không cần sử code, đảm bảo quy tắc Open-closed principle.

### Áp dụng mẫu thiết kế Adapter cho AI trong game



***Giải thích:*** Thuật toán Minimax được áp dụng cho AI trong game, một thư viện có sẵn chứa thuật toán Minimax được tái sử dụng cho chương trình. Áp dụng mẫu Adapter để chuẩn hóa input và output của chương trình với thư viện có sẵn để có thể tái sử dụng.

# Cấu trúc framework