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

Khoa Công nghệ Thông tin
Đại học Khoa học Tự nhiên, ĐHQG-HCM
Tháng 01/2020

MỤC LỤC

1 Giới thiệu chung

1.1 Nhóm

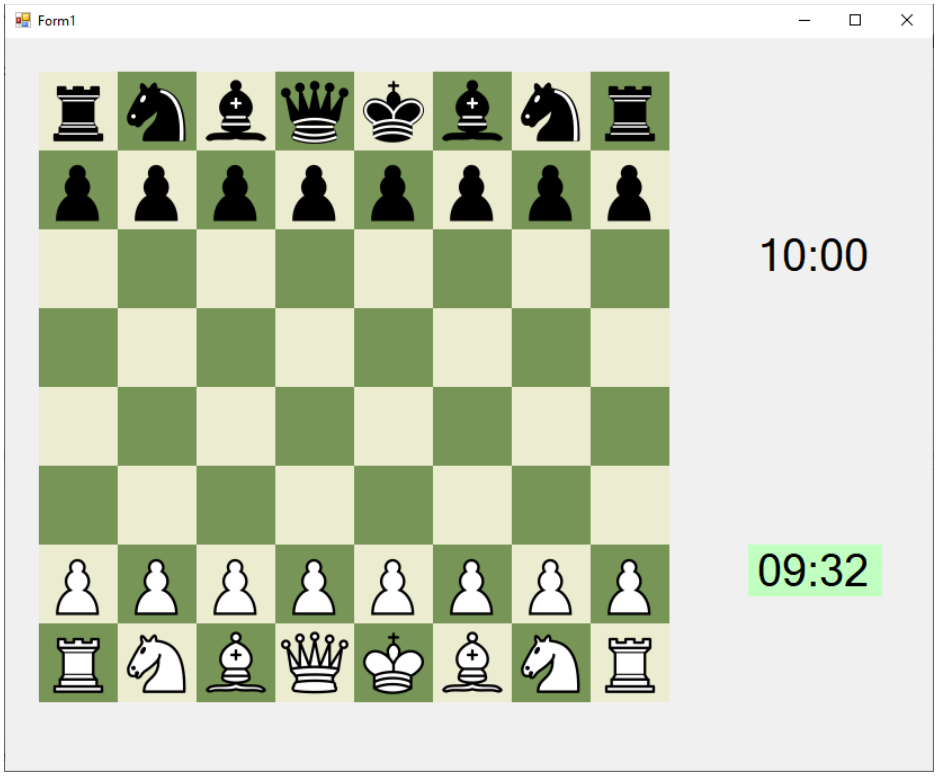
Thông tin thành viên nhóm: Nhóm gồm 3 thành viên:

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

1.2 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.



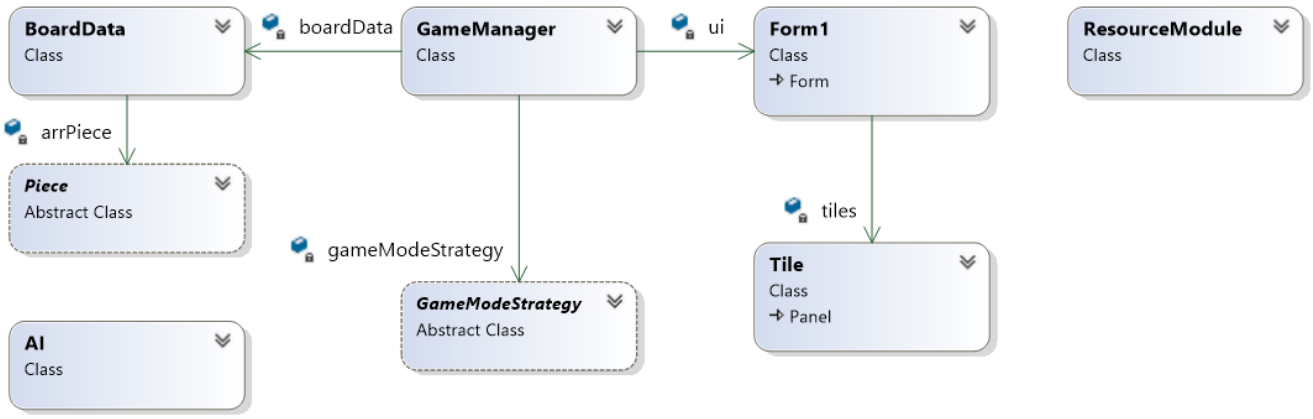
Hình 1. Giao diện chính của Game

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.

2 Kiến trúc của game

2.1 Các thành phần chính trong game



Hình 2. Sơ đồ các lớp chính trong game

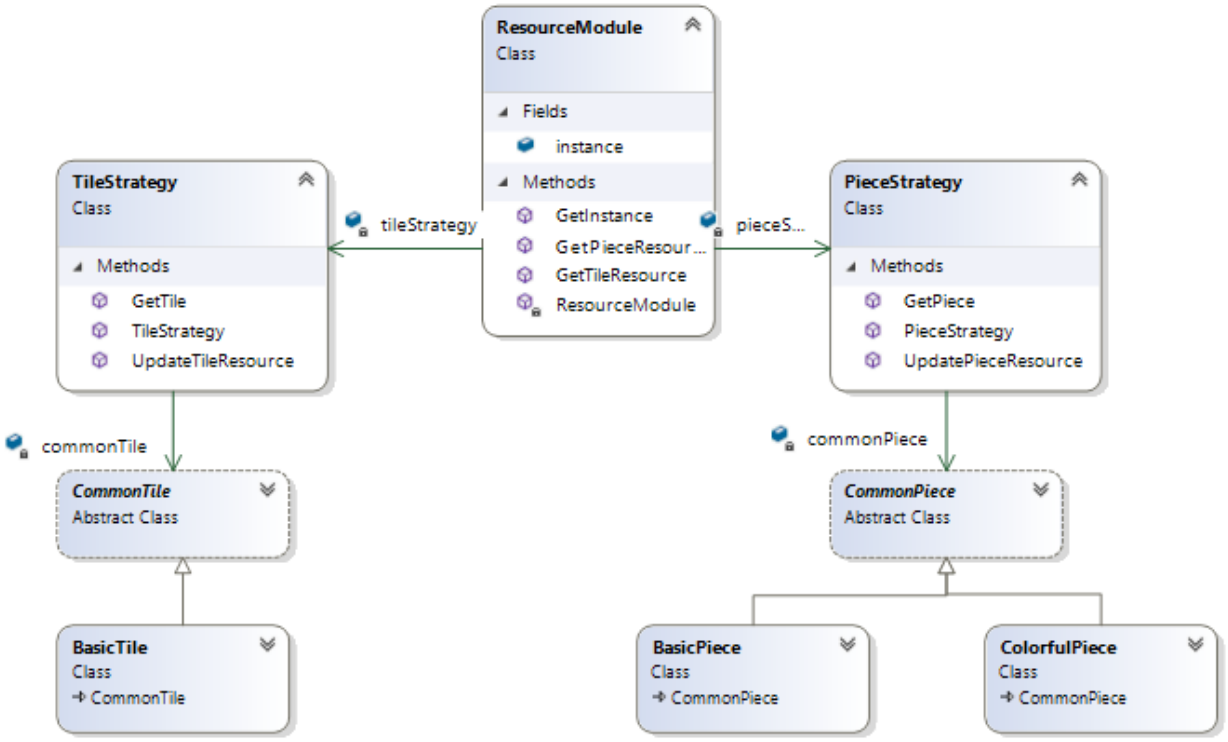
Kiến trúc của game bao gồm các lớp chính:

- 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

2.2 Các mẫu thiết kế được áp dụng trong đồ án

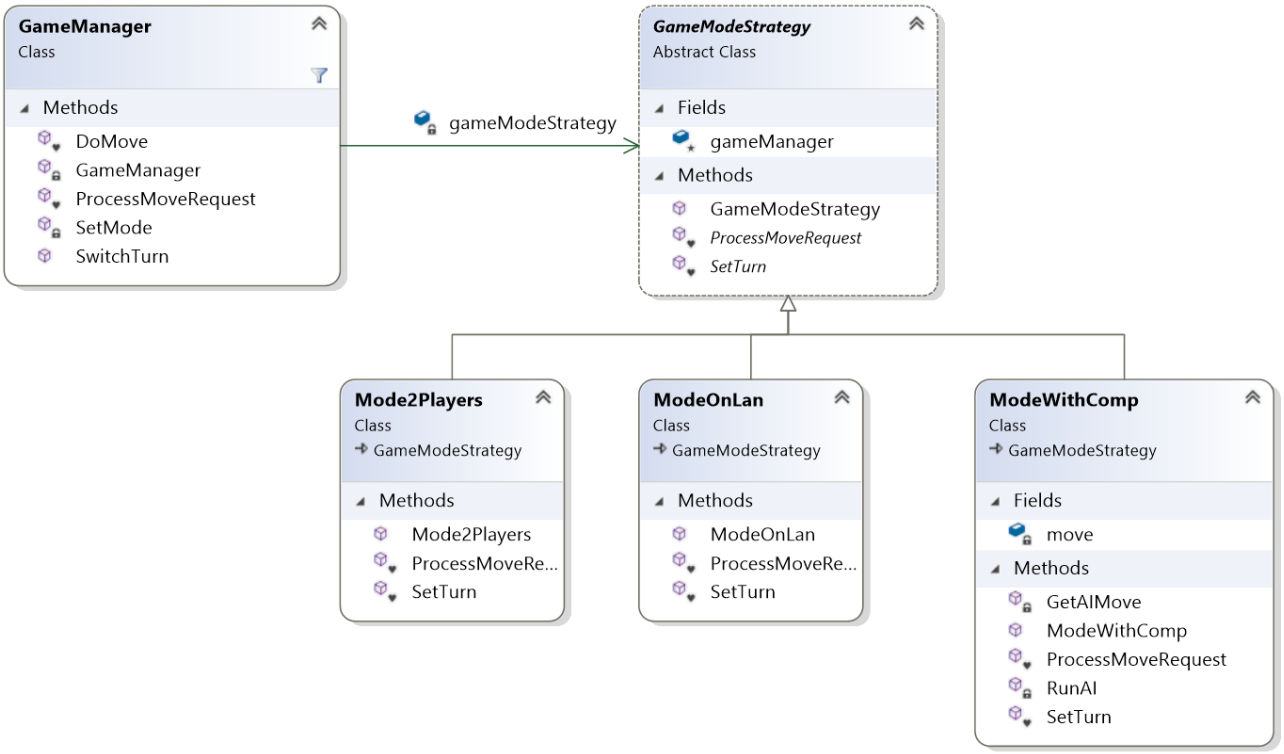
2.2.1. Áp dụng mẫu thiết kế Strategy vào lấy resources



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.

2.2.2. Áp dụng mẫu Strategy trong xử lý lượt di chuyển

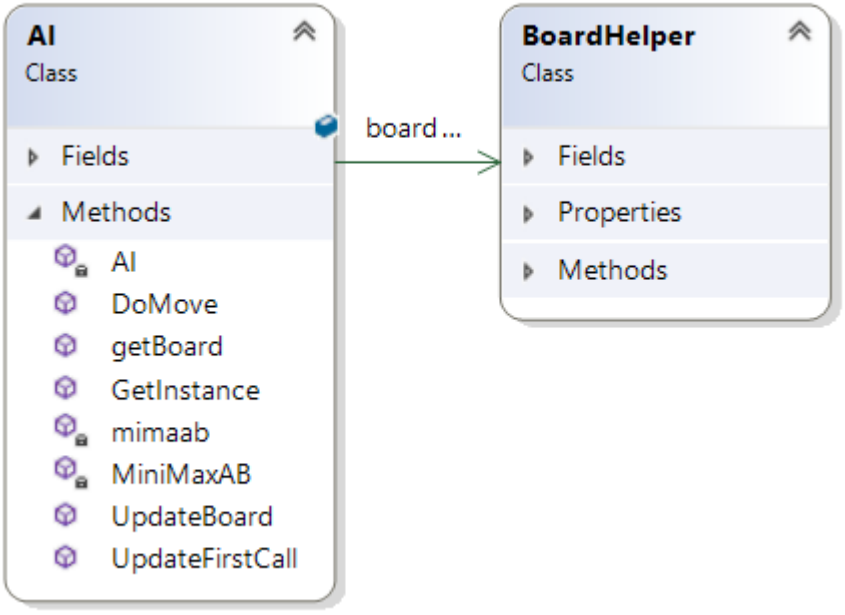


Tùy từng chế độ chơi mà lớp GameManager sẽ có các xử lý một yêu cầu di chuyển quân cờ từ người dùng khác nhau. Với chế độ chơi Offlline, khi nhận yêu cầu di chuyển quân cờ, hệ thống cần kiểm tra lại nước đi có hợp lệ hay không trước khi thực thi nước đi, sau đó chuyển lượt đi cho người chơi thứ hai (trong chế độ 2 người chơi) hoặc gọi AI tính toán tìm nước đi cho phía máy tính (chế độ chơi với máy). Trong chế độ chơi Online, game được thiết kế theo mô hình Peer-2-Peer, nên cần có sự đồng thuận của các client trong mạng, trước khi nước đi được thực thi, nên trong chế độ này, nước đi ngoài việc được kiểm tra ở client của người chơi, nước đi cần được gửi đến client của đối thủ, và đợi kết quả trả về. Khi có kết quả trả về là nước đi hợp lệ, thì nước đi mới thật sự được ghi vào dữ liệu và hiển thị lên màn hình.

Để đáp ứng được yêu cầu đó, nhóm áp dụng mẫu Strategy, tạo lớp GameModeStrategy xử lý MoveRequest, có 3 lớp con kế thừa lớp này, tương ứng với 3 chế độ chơi, mỗi lớp sẽ override lại

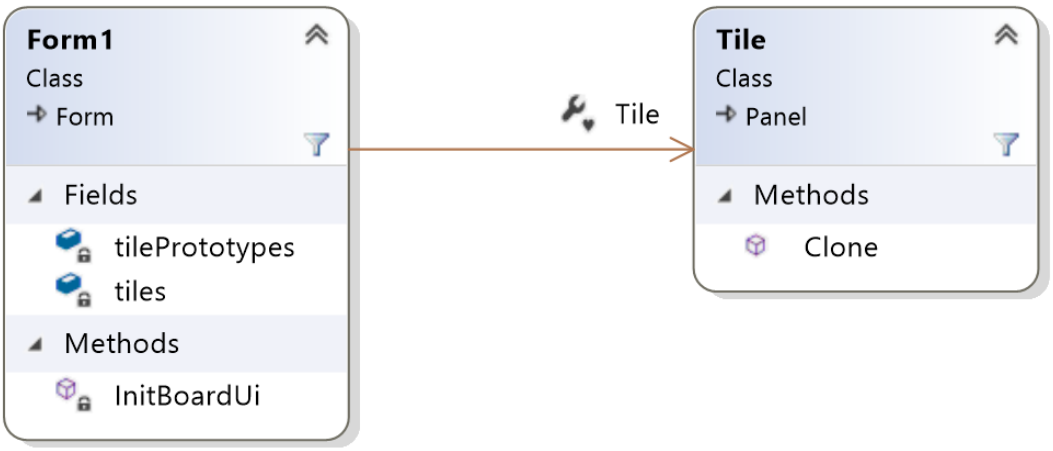
hàm của lớp cha cho phù hợp. Khi đó, lớp GameManager chứa một đối tượng của lớp GameModeStrategy, và tùy chế độ chơi người dùng chọn, đối tượng này được khởi tạo là 1 trong 3 lớp con. Khi có một yêu cầu di chuyển từ người dùng, chỉ cần gọi hàm xử lý MoveRequest của đối tượng GameModeStrategy này.

2.2.3. Áp dụng mẫu thiết kế Adapter cho AI trong game



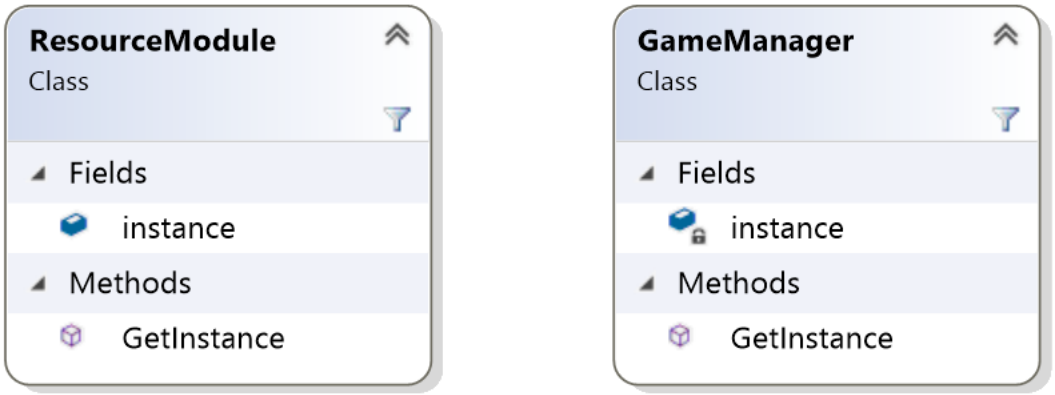
Giải thích: Thuật toán Minimax được áp dụng cho AI trong game, một thư viện có sẵn chứa thuật toán Minimax được tái sử dụng cho chương trình. Áp dụng mẫu Adapter để chuẩn hóa input và output của chương trình với thư viện có sẵn để có thể tái sử dụng.

2.2.4. Prototype Pattern



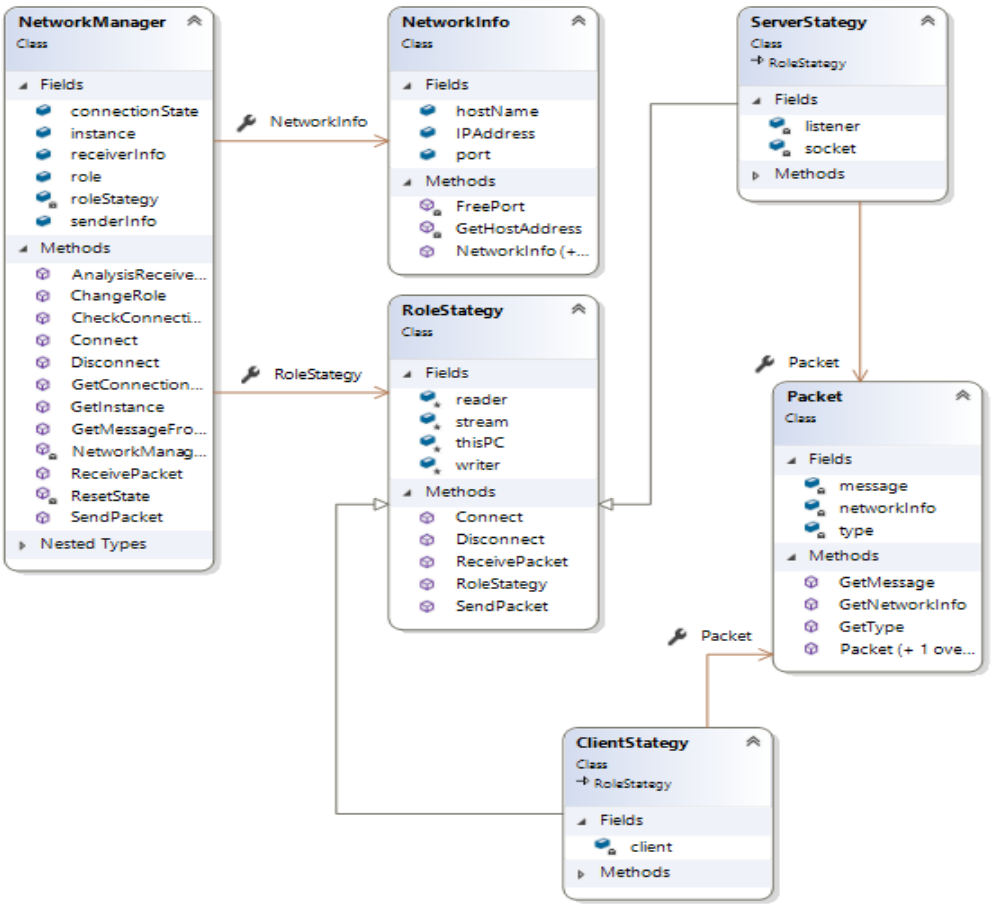
Thiết kế được sử dụng trong game không hoàn toàn là mẫu Prototype, mà chỉ mang tư tưởng tương tự. Bàn cờ vua có 64 ô cờ, trong đó có 32 ô đen và 32 ô trắng, các ô này nếu chưa hiện các quân cờ, thì chỉ khác nhau về vị trí, còn các thuộc tính khác là giống nhau nếu màu giống nhau. Vì vậy để tiết kiệm thời gian tạo các ô cờ, ta tạo sẵn 2 ô cờ màu đen và trắng, các ô cờ còn lại ta Clone từ 2 ô cờ đã tạo sẵn.

2.2.5. Singleton



Mẫu Singleton được áp dụng khá nhiều trong đồ án vì tính đơn giản và tiện dụng của nó, nhất là đối với các lớp quản lý, không biết chính xác thời điểm khởi tạo, và được truy cập bởi nhiều lớp khác nhau, như lớp ResourceModule hay GameManager. Tuy nhiên vì sự truy cập tiện dụng như vậy nên mẫu Singleton tạo ra các biến toàn cục, khó quản lý, đồng thời cũng góp phần làm tăng sự coupling giữa các lớp.

2.2.6. Áp dụng mẫu thiết kế Strategy cho kết nối qua Lan



Giải thích: Vì kết nối qua LAN nên hình thức kết nối sẽ là Peer-to-peer. Khi người chơi host game, họ sẽ trở thành server, ngược lại người chơi kết nối vào host sẽ là client nên sử dụng Strategy để có thể dễ dàng luân chuyển Role của người chơi để họ vừa có thể trở thành Client vừa có thể trở thành Server. Lớp Abstract RoleStategy có các method mà cả Client lẫn Server đều có chung như Connect, Disconnect, Receive, Send, … Ngoài ra lớp NetworkInfo để lưu thông tin của các máy tham gia trò chơi và lớp packet dùng để chuyển thông tin.