**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**ĐỒ ÁN 1**

**GAME CARO HAI NGƯỜI CHƠI QUA LAN**

**(STACK)**

**SVTH1 : TRẦN GIA HÂN - 17110133**

**SVTH2 : VÕ THỊ KIỀU DIỄM - 17110109**

**GVPT: ThS. TRẦN CÔNG TÚ**

**TP. Hồ Chí Minh, tháng 11 năm 2019**

# MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc24880710)

[CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN 2](#_Toc24880711)

[1.1. Lịch sử hình thành 2](#_Toc24880712)

[1.2. Giới thiệu game caro hai người chơi 2](#_Toc24880713)

[1.3. Lý do, mục đích, mục tiêu chọn đề tài 2](#_Toc24880714)

[1.3.1. Lý do, mục đích chọn đề tài 2](#_Toc24880715)

[1.3.2. Mục tiêu chọn đề tài 3](#_Toc24880716)

[1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng 3](#_Toc24880717)

[CHƯƠNG 2 : NỘI DUNG 4](#_Toc24880718)

[2.1. Luật chơi 4](#_Toc24880719)

[2.2. Khái quát thuật toán 4](#_Toc24880720)

[2.2.1. Giới thiệu ngăn xếp (stack) 4](#_Toc24880721)

[2.2.2. Ứng dụng 4](#_Toc24880722)

[2.2.3. Sơ lược về LAN 5](#_Toc24880723)

[2.3. Thiết kế giao diện 5](#_Toc24880724)

[2.4. Cách cài đặt và viết chương trình 6](#_Toc24880725)

[2.4.1. Một số hàm xử lý chức năng 6](#_Toc24880726)

[2.4.2. Ứng dụng của STACK 13](#_Toc24880727)

[2.4.3. Thiết lập chơi cờ qua mạng LAN 15](#_Toc24880728)

[2.5. Sản phẩm của đồ án 22](#_Toc24880729)

[CHƯƠNG 3 : KẾT LUẬN 24](#_Toc24880730)

[3.1. Kết luận 24](#_Toc24880731)

[3.2. Ưu điểm 24](#_Toc24880732)

[3.3. Nhược điểm 24](#_Toc24880733)

[3.4. Đề xuất phương án cải thiện 24](#_Toc24880734)

[PHỤ LỤC 25](#_Toc24880735)

[TÀI LIỆU THAM KHẢO 28](#_Toc24880736)

# Danh mục các hình

[Hình 1: Giao diện chính 5](#_Toc24886257)

[Hình 2: Generic 7](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886258)

[Hình 3: Push() 7](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886259)

[Hình 4: Pop() 8](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886260)

[Hình 5: Top() 8](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886261)

[Hình 6: Clear() 8](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886262)

[Hình 7: VeBanCo() 9](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886263)

[Hình 8: ButtonCu (1) 9](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886264)

[Hình 9: ButtonCu (2) 10](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886265)

[Hình 10: btn\_Click() 10](#_Toc24886266)

[Hình 11: Mark() 11](#_Toc24886267)

[Hình 12: isEndGame() 11](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886268)

[Hình 13: isEndGame() 11](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886269)

[Hình 14: ktHangNgang() 12](#_Toc24886270)

[Hình 15: LayViTri() 13](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886271)

[Hình 16: UndoAStep() 14](#_Toc24886272)

[Hình 17: Undo() 14](#_Toc24886273)

[Hình 18: Xoabanco() 14](file:///D:\Studying\Năm%203\Đồ%20án%201\doan1\Báo%20cáo%20Đồ%20án%201%20-%20Game%20Caro%20(LAN).docx#_Toc24886274)

[Hình 19: Client & Server 15](#_Toc24886275)

[Hình 20: SerializeData() & DeserializeData() 16](#_Toc24886276)

[Hình 21: ConnectServer() 16](#_Toc24886277)

[Hình 22: CreateServer() 17](#_Toc24886278)

[Hình 23: Send() 17](#_Toc24886279)

[Hình 24: Receive() 17](#_Toc24886280)

[Hình 25: Form1\_Shown() 18](#_Toc24886281)

[Hình 26: ProcessData() 19](#_Toc24886282)

[Hình 27: ChessBoard\_PlayerMarked() 20](#_Toc24886283)

[Hình 28: OtherPlayerMark() 20](#_Toc24886284)

[Hình 29: SEND\_POINT 21](#_Toc24886285)

[Hình 30: Class ButtonClickEvent 21](#_Toc24886286)

[Hình 31: Listen() 22](#_Toc24886287)

[Hình 32: btnLan\_Click() 22](#_Toc24886288)

[Hình 33: Giao diện chính 23](#_Toc24886289)

# LỜI MỞ ĐẦU

Nhóm thực hiện xin chân thành cảm ơn sự hướng dẫn tận tình của thầy Trần Công Tú, cả về chuyên môn lẫn định hướng cho nhóm. Vì kiến thức còn hạn hẹp và thời gian có hạn nên nhóm chưa thể tối ưu hết các thuật toán sử dụng trong game cũng như không thể tránh khỏi các thiếu sót. Vì vậy rất mong nhận được sự góp ý của thầy để đồ án của nhóm có thể hoàn thiện hơn.

# CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN

1.1. Lịch sử hình thành

Ban đầu loại cờ này được chơi bằng các quân cờ vây (quân cờ màu trắng và đen) trên một bàn cờ vây (19x19). Quân đen đi trước và người chơi lần lượt đặt một quân cờ của họ trên giao điểm còn trống. Người thắng là người đầu tiên có được một chuỗi liên tục gồm 5 quân hàng ngang, hoặc dọc, hoặc chéo. Tuy nhiên, vì một khi đã đặt xuống, các quân cờ không thể di chuyển hoặc bỏ ra khỏi bàn, do đó loại cờ này có thể chơi bằng giấy bút. Ở Việt Nam, cờ này thường chơi trên giấy tập học sinh (đã có sẵn các ô ca-rô), dùng bút đánh dấu hình tròn (O) và chữ X để đại diện cho 2 quân cờ.[1]

1.2. Giới thiệu game caro hai người chơi

Game Caro (hay còn gọi là Gomuku) là một game thuộc thể loại game đối kháng, trí tuệ giữa hai người chơi, bạn có thể chơi nó ở bất cứ đâu, bất cứ lúc nào.

Trước đây trò chơi này cần có giấy kẻ vuông và cây bút viết, nhưng nay Caro được phát triển trên các thiết bị điện tử và không còn xa lạ với bất cứ ai.

1.3. Lý do, mục đích, mục tiêu chọn đề tài

1.3.1. Lý do, mục đích chọn đề tài

Sau khi học môn cấu trúc dữ liệu, lập trình hướng đối tượng và mạng máy tính, nhóm được tiếp xúc nhiều hơn với các khối kiến thức cơ sở, điều đó giúp nhóm có thể xây dựng các chương trình trên máy tính, áp dụng các kiểu dữ liệu trong thực tế. Thông qua đó, nhóm vận dụng kiến thức về ngăn xếp (stack) và truyền dữ liệu qua mạng LAN để demo ***Game caro hai người chơi qua LAN***. Với mục đích xây dựng game caro với tính tiện dụng, gần gũi, dễ chơi và tính giải trí cao, phù hợp với mọi lứa tuổi, bám sát kiến thức đã học. Và quan trọng là hiểu rõ hơn về các thuật toán sử dụng, tìm hiểu thêm về thiết kế giao diện trong game, hiểu sâu về kết nối LAN, rèn luyện kỹ năng lập trình, làm tiền đề cho các môn học sau này và ứng dụng mai sau. Vì vậy nhóm quyết định chọn đề tài Game caro hai người chơi qua LAN, dùng STACK để undo để thực hiện đồ án 1.

1.3.2. Mục tiêu chọn đề tài

Với mục tiêu tạo ra một game caro có đầy đủ chức năng giúp người chơi dễ dàng sử dụng như undo, newgame, lựa chọn luật chơi, kết nối LAN… Đặc biệt, áp dụng stack để lưu trữ nước cờ của người chơi, giúp người chơi có thể hoàn lại nước đã đi.

1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng

Vì chương trình mô phỏng của nhóm biễu diễn thuật toán dưới dạng các đối tượng thuộc tính, hành vi rõ ràng nên nhóm quyết định sử dụng ngôn ngữ Microsoft C#.NET với phiên bản .NET Framework 4.5.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2019 để cài đặt mô phỏng này.

# CHƯƠNG 2 : NỘI DUNG

2.1. Luật chơi

Khi kết nối LAN, mỗi người sẽ lần lượt đi từng nước một, ký hiệu hai bên khác nhau, ngừng khi đạt điều kiện thắng hoặc hết giới hạn bàn cờ.

Có hai chế độ chơi cho phép người dùng chọn : chặn hai đầu và không chặn hai đầu.

Điều kiện thắng :

+ Đối với chế độ không chặn hai đầu : người chơi có được một hàng 5 nước dọc, ngang hoặc chéo liên tiếp thì sẽ là người chiến thắng!

+ Đối với chế độ chặn hai đầu : người chơi có được 5 nước dọc, ngang hoặc chéo liên tiếp mà không bị đối phương chặn hai đầu thì thắng!

+ Nếu hết thời gian mà người chơi vẫn chưa đánh thì bên người chơi còn lại sẽ thắng.

2.2. Khái quát thuật toán

2.2.1. Giới thiệu ngăn xếp (stack)

Trong khoa học máy tính, một ngăn xếp (còn gọi là bộ xếp chồng, tiếng Anh: stack) là một cấu trúc dữ liệu trừu tượng hoạt động theo nguyên lý "vào sau ra trước" (Last In First Out (LIFO)).

Ngăn xếp là một cấu trúc dữ liệu dạng thùng chứa (container) của các phần tử (thường gọi là các nút (node)) và có hai phép toán cơ bản: push and pop. *Push* bổ sung một phần tử vào đỉnh (top) của ngăn xếp, nghĩa là sau các phần tử đã có trong ngăn xếp. *Pop* giải phóng và trả về phần tử đang đứng ở đỉnh của ngăn xếp. Trong stack, các đối tượng có thể được thêm vào stack bất kỳ lúc nào nhưng chỉ có đối tượng thêm vào sau cùng mới được phép lấy ra khỏi stack.[2]

2.2.2. Ứng dụng

Ngăn xếp có nhiều ứng dụng trong khoa học máy tính và cài đặt game. Trong đó, Game caro cũng là một ứng dụng được cài đặt từ stack để lưu trữ dữ liệu bàn cờ.

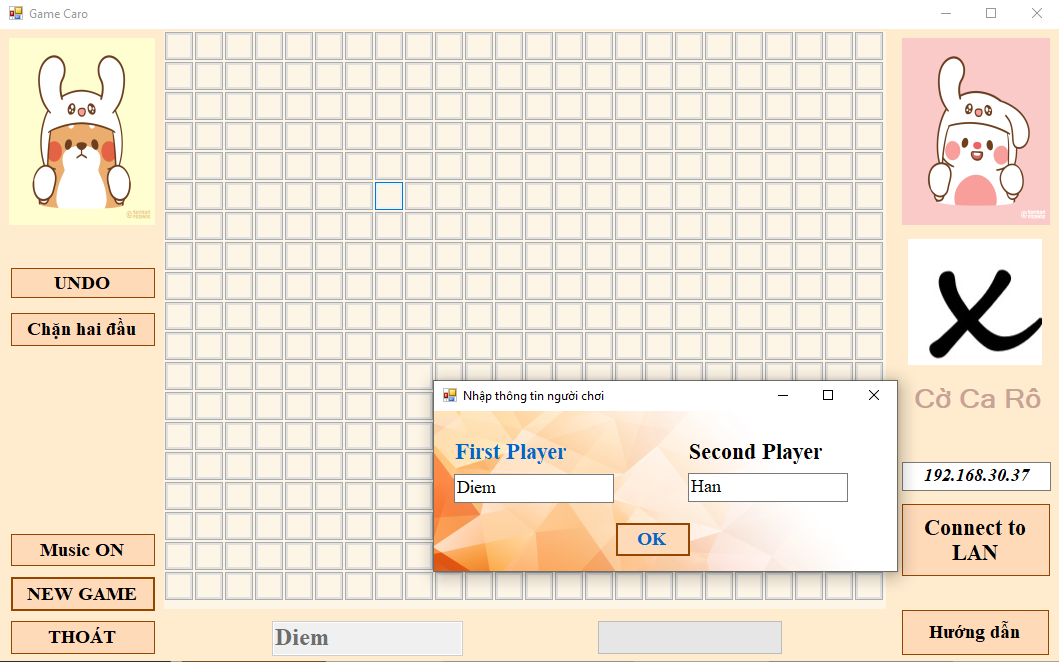
2.2.3. Sơ lược về LAN

LAN (Local Network Area - Mạng cục bộ) là sự kết hợp của nhiều thiết bị kết nối lại với nhau trong một hệ thống mạng tại một khu vực nhất định. Việc này cho phép chúng trao đổi dữ liệu với nhau một cách nhanh chóng, dễ dàng.

Cùng với sự phát triển công nghệ mỗi ngày đều tăng cao, mạng LAN ngày càng được phát triển và hoạt động với nhiều hình thức khác, ví dụ như WAN - các mạng LAN được kết nối lại với nhau thông qua thiết bị định tuyến (Router) hay WLAN (Wireless Local Network Area) mạng LAN không dây - các máy tính kết nối vào mạng thông qua Wi-Fi.

Một mạng LAN cơ bản cần có một máy chủ (Server), các máy trạm (client), các thiết bị ghép nối (Switch, Hub, Repeater, Access Point), cáp mạng (cable).[3]

2.3. Thiết kế giao diện



Hình 1: Giao diện chính

Chương trình cho phép người chơi có thể chọn trò chơi mới (new game), đi lại (undo), chọn bật nhạc hay tắt nhạc, hướng dẫn chơi game, chọn luật chơi, kết nối LAN. Phía dưới có thể hiện trạng thái đến lượt người nào đi.

Khi bắt đầu vào game sẽ cho phép chọn người đi trước và nhập tên của hai người chơi.

Chức năng kết nối LAN cho phép hai người chơi cùng một lúc trên cùng một thiết bị hay hai thiết bị (kết nối dựa trên IP của máy nhấn kết nối trò chơi trước (Server)).

Khi có người thắng thì hiển thị ra thông báo thắng hiệp và cho phép chọn các tùy chọn chơi tiếp hay thoát.

Bên cạnh đó còn bao gồm phần hướng dẫn cho người chơi giúp người chơi dễ dàng nắm được luật chơi.

2.4. Cách cài đặt và viết chương trình

2.4.1. Một số hàm xử lý chức năng

Đồ án của nhóm gồm 5 form và 8 class gồm:

+ FormNhapThongTin (Cho phép nhập tên người chơi).

+ FormGame (Form chính, bao gồm bàn cờ và những phím chức năng).

+ FormChienThang (Thông báo đã kết thúc game và một số hiệu ứng).

+ FormHuongDan (Hướng dẫn chơi game).

+ FormThoat (Hiệu ứng và quyết định có thoát game hay không) .

+ Class MyStack: tạo stack và các hàm trong nó.

+ Class Player: lớp người chơi bao gồm tên ký hiệu và âm thanh

+ Class XuLyBanCo: gồm các hàm xử lý trên một bàn cờ.

+ Class Cons: gồm các hằng được tạo (chiều dài, rộng bàn cờ, button…)

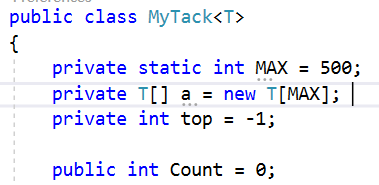
+ Class Program: chương trình chính.

+ Class PlayInfo: lớp thông tin gồm point và nguoichoihientai.

+ Class SocketData: tạo Socketdata và tạo các command để ghi nhớ.

+ Class SocketManager: tạo Client, server và các chức năng.

Hầu hết các hàm xử lý đều được viết trong class XuLyBanCo, và sau đây là một số hàm quan trọng.

Class Stack:

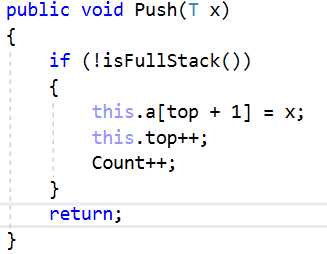
Hình 2: Generic

Vận dụng kiến thức đã học từ môn cấu trúc dữ liệu và giải thuật và môn lập trình hướng đối tượng, nhóm đã tạo stack từ Generic.

Các phần tử được lưu vào mảng một chiều với tối đa 500 phần tử.

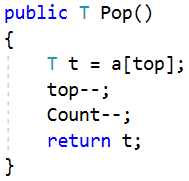
Khởi tạo rỗng top = -1;

Biến Count: để biết số lượng phần tử trong stack.

Hàm Push():

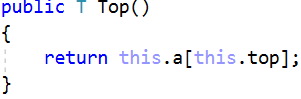
Hình 3: Push()

Kiểm tra xem đã tràn stack chưa, nếu chưa thì thêm vào stack và tăng biến đếm.

Hàm Pop():

Hình 4: Pop()

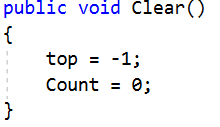
Lấy ra và trả về giá trị cuối cùng của stack.

Hàm Top():

Hình 5: Top()

Trả về (không lấy ra) giá trị cuối của stack.

Hình 6: Clear()

Hàm Clear():

Đặt lại stack.

Tạo bàn cờ:

Đầu tiên tạo ra một class player gồm các thuộc tính như Tên, ký hiệu, âm thanh khi đánh và các properties.

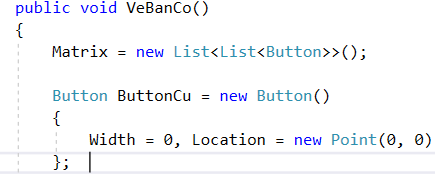
Sau đó tạo một list player gồm 2 người chơi (sử dụng list có thể phát triển game cao hơn, như tăng số lượng người chơi…).

Lấy thông tin từ form nhập, nếu không nhập thì sẽ không cập nhật tên người chơi (mặc định là *first player* và *second player).*

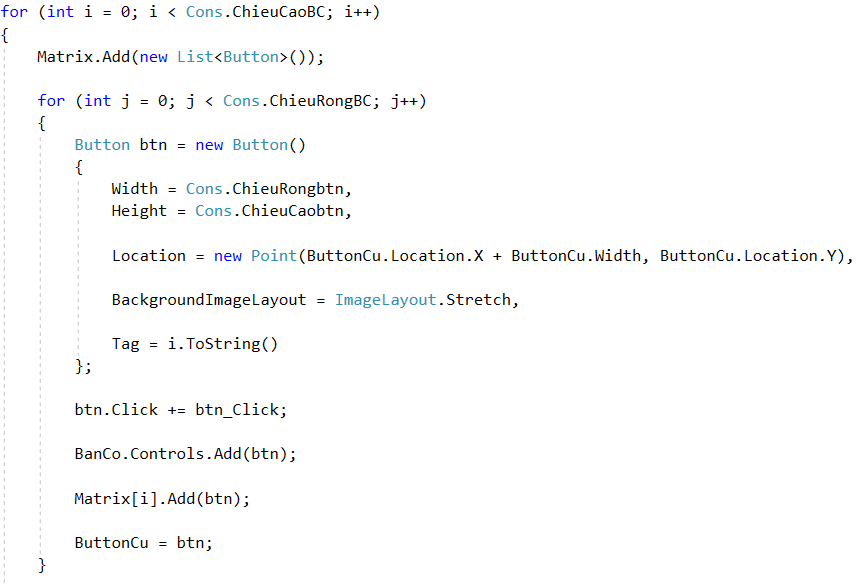
Mặc định người chơi thứ nhất đi trước (X đi trước).

Sử dụng hàm đổi người chơi để picture mark hiện ảnh khi mới vào game và cho biết ai đi trước.

Tạo bàn cờ: đây là bước quan trọng và tương đối khó.

Đầu tiên, khởi tạo matrix (2 list button lồng nhau).

Hình 7: VeBanCo()

Khởi tạo ButtonCu làm mốc để tạo ra các button (ô đánh) trên panel bàn cờ.

Hình 8: ButtonCu (1)

Sử dụng hai vòng lặp để tạo ra bàn cờ, mỗi ô đánh là một button.

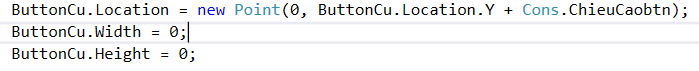
Vị trí ô được tạo sẽ bằng tọa độ ô cũ cách sang bên phải ô cũ đúng bằng chiều rộng của nó.

+ Vị trí hàng của button sẽ được lưu vào Tag của button.

+ Tạo event khi click chuột vào button.

+ Thêm button mới tạo vào panel bàn cờ và thêm vào Matrix.

+ Gán button cũ thành button mới tạo để tạo ra các button tiếp theo.

Tiếp tục làm như vậy cho đến khi hết bài cờ.

Hình 9: ButtonCu (2)

Sau khi tạo xong một hàng, tiếp tục tạo ra một button ở đầu hàng tiếp theo làm gốc và tiếp tục làm như trên cho đến khi vẽ hết bàn cờ.

Tạo event cho mỗi button trên bàn cờ:



Hình 10: btn\_Click()

Lúc đầu khởi tạo win = 0 (chưa thắng).

Ép kiểu lại với button để biết đã click button nào.

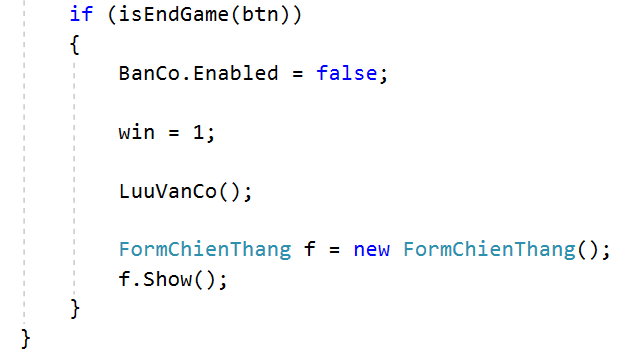
Kiểm tra xem ô này đã được đánh chưa, nếu rồi thì không được đánh vào.

Nếu chưa, thì đánh vào ô đó bằng hàm Mark():



Hình 11: Mark()

Hàm Mark thực hiện việc đổi Image của button theo người chơi hiện tại, chạy âm thanh của người đánh và đổi NguoiChoiHienTai sau mỗi lần đánh.

Mỗi lần đánh, lưu vị trí button đó vào STACK, Stack2 và đổi người chơi.

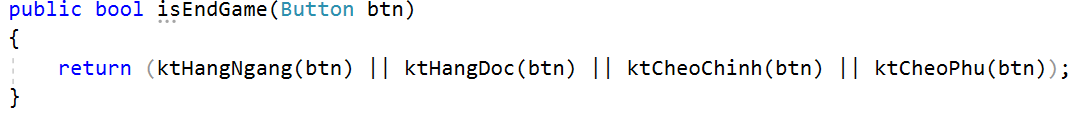
Hình 12: isEndGame()

Kiểm tra nếu kết thúc game:

+ Disable bàn cờ.

+ Trạng thái thắng = 1;

+ Hiện Form chiến thắng lên.

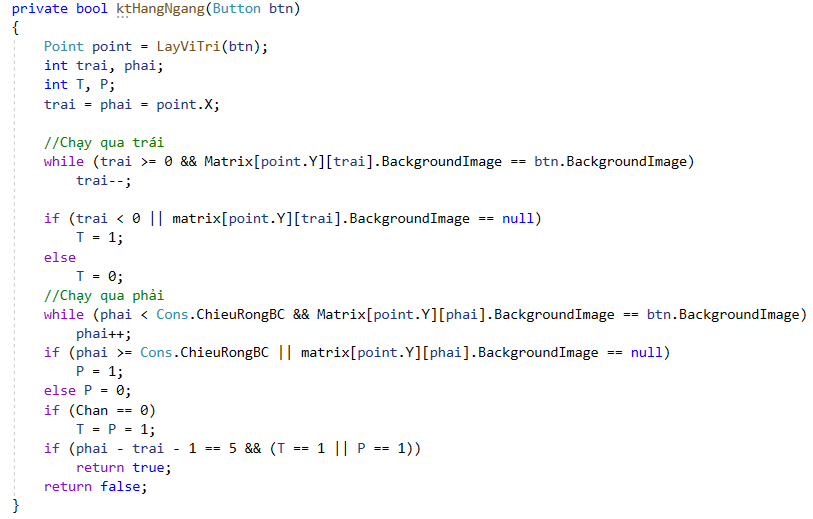
Hàm isEndGame():

Hình 13: isEndGame()

Kiểm tra xem đã kết thúc game hay chưa, nếu có 1 trong 4 điều kiện là có 5 nước đi liên tục trên hàng dọc, ngang, chéo chính và chéo phụ thì kết thúc.

Sau đây là các hàm kiểm tra kết thúc game, gồm kiểm tra hàng dọc, hàng ngang, đường chéo chính và đường chéo phụ, nếu đánh liên tiếp 5 nước liên tục trên một đường thẳng thì kết thúc game.

Kiểm tra hàng ngang:



Hình 14: ktHangNgang()

Luật không chặn hai đầu: ý tưởng là từ ô (button) mới đánh, ta kiểm tra bên trái, và bên phải ô đó có Mark giống với ô này không, nếu giống thì tiếp tục kiểm tra ô kế bên, nếu tổng số ô liên tục đúng bằng 5 thì kết thúc game.

Luật chặn hai đầu: Kiểm tra thêm 2 ô ở 2 đầu hàng 5 xem có bị chặn bởi người khác không, nếu bị chặn thì không tính thắng.

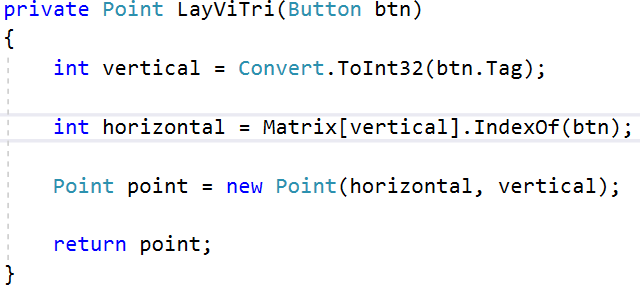
Từ button trên Form chính ta chọn chế độ chơi, nếu không chặn hai đầu (Chặn = 0) thì ta bỏ qua điều kiện chặn hai đầu.

Ta kiểm tra hàng dọc, đường chéo chính, chéo phụ tương tự như kiểm tra hàng ngang.

2.4.2. Ứng dụng của STACK

2.4.2.1. Ứng dụng đầu tiên và cũng là quan trọng nhất trong đồ án của nhóm đó chính là ứng dụng vào chức năng Undo.

- Ý tưởng: mỗi khi click đánh vào một ô, ta sẽ lấy ra tọa độ của ô đó push vào stack, khi người chơi thực hiện Undo, theo nguyên tắc hoạt động LIFO, tọa độ của ô mới đánh sẽ được pull ra và ô cờ ứng với tọa độ đó sẽ được trả về null.

- Sau đây là một số hàm xử lý:

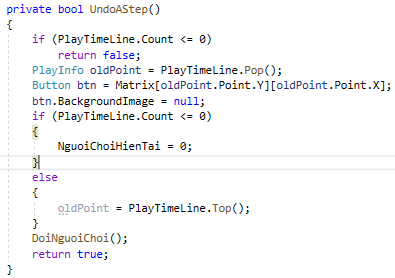
Hình 15: LayViTri()

Hàm lấy vị trí:

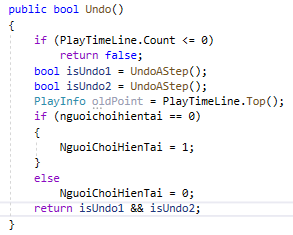
Lấy vị trí hàng của button từ Tag đã lưu khi vẽ bàn cờ.

Lấy vị trí cột của button từ hàng thứ vartical thông qua IndexOf(btn).

Đối với mạng LAN thì cần gọi undo hai lần nên cần viết một hàm undoAstep và gọi nó trong hàm undo chính.

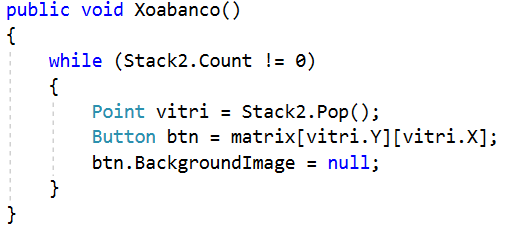


Hình 16: UndoAStep()



Hình 17: Undo()

2.4.2.2. Ứng dụng thứ hai về stack trong đồ án là ứng dụng vào việc xóa bàn cờ (New game).

Hàm XoaBanCo():

Hình 18: Xoabanco()

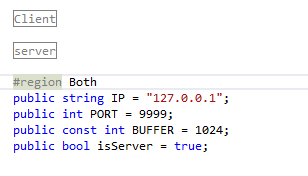
- Về cơ bản cũng chính là Undo, nhưng sẽ đưa tất cả background image của các ô đánh về null.

2.4.3. Thiết lập chơi cờ qua mạng LAN

2.4.3.1. Thiết lập hai người chơi qua mạng LAN

*2.4.3.1.1. Tạo Client và server*

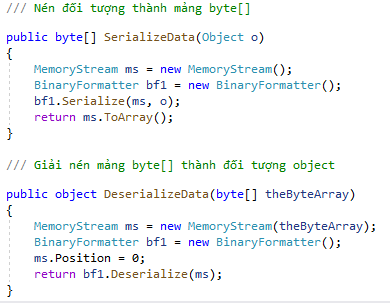
- Tạo IP: 127.0.0.1 và port 9999 . Hàm sử dụng chung cho sever và client. Dùng buffer là mỗi lần nhận tin được bao nhiêu.



Hình 19: Client & Server

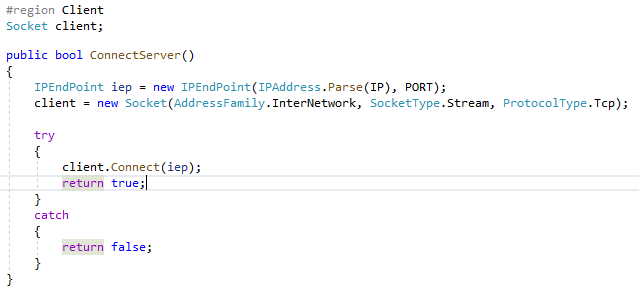
- Hàm GetLocalIPv4 lấy ra IPv4 của card mạng đang dùng (thay vì sử dụng ipcongfig để lấy IPv4).

- Cần hai hàm SerializeData (phân tích thành mảng byte và gửi đi) và DeserializeData (giải nén một mảng byte thành một đối tượng object).



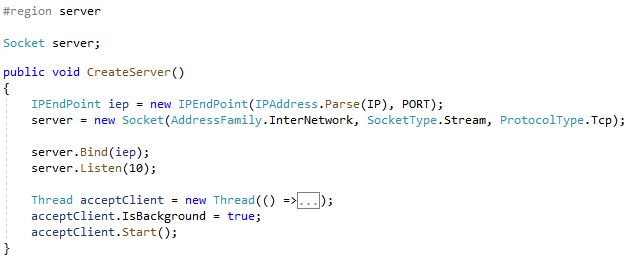
Hình 20: SerializeData() & DeserializeData()

- Trong client, tạo hàm kết nối với server: ConnectServer(), tạo một socket client, tạo một ipaddress (IPEndPoint) để kết nối.



Hình 21: ConnectServer()

- Trong server, tạo hàm kết nối với Client: ConnectClient(), tạo một socket server, tạo một ipaddress (IPEndPoint) để kết nối, server sẽ đợi kết nối của Client trong 10s nếu không có thì bỏ. Sau Listen() sẽ chấp nhận kết nối. Tạo một luồng chạy riêng để client mà server accept.

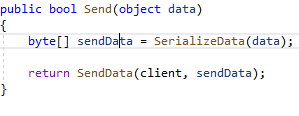


Hình 22: CreateServer()

- Gửi nếu sendData bằng 1 thì trả về true và ngược lại.

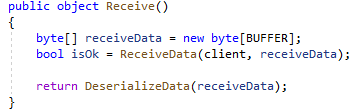
- Nhận data nếu receiveData bằng 1 thì trả về true và ngược lại.

- Hàm send(): phân tích gói tin, xét server và client (tất cả đều dùng client để gửi).



Hình 23: Send()

**-** Hàm Receive() đều receive bằng client, receive bằng object.



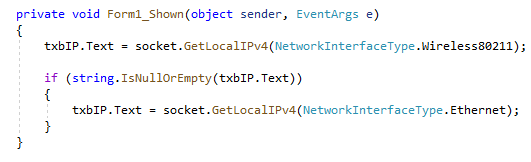
Hình 24: Receive()

- Tạo socketManager trong formcaro

- BtnLan lấy IP từ textbox lấy từ server, nếu như kết nối server không thành công thì tạo server nếu đã kết nối thành công thì Listen. Tạo một hàm Listen() để lắng nghe server hoặc gửi tin lên server.

- Khi lắng nghe sẽ receive lấy giá trị ra, show data.

- Xét ip trong form1\_shown kiểm tra kết nối, hiện IP.

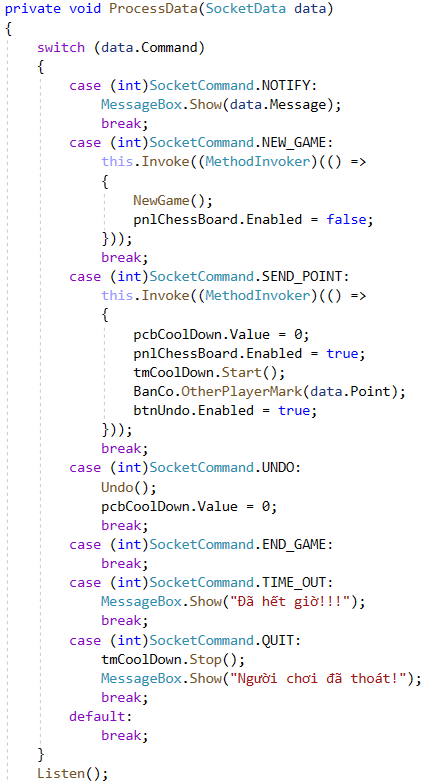


Hình 25: Form1\_Shown()

2.4.3.2. Hiện quân cờ trên mạng LAN

- Tạo Class SocketData để truyền dữ liệu gồm command và point. Xây dựng một hàm dựng và enum để ghi nhớ gồm SEND\_POINT, NEW\_GAME, UNDO, QUIT, NOTIFY, END\_GAME.

- Xây dựng ProcessData để xử lý sau khi Listen.



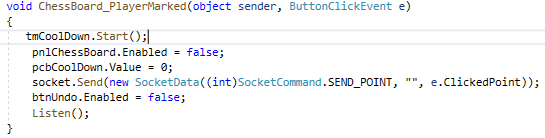
Hình 26: ProcessData()

Khi đánh sẽ gửi data đi với command là Send\_point và gửi đến vị trí click.

+ tmCoolDown sẽ bắt đầu chạy đếm thời gian, pcbCoolDown bằng 0.

+ pnlChessBoard sẽ false.

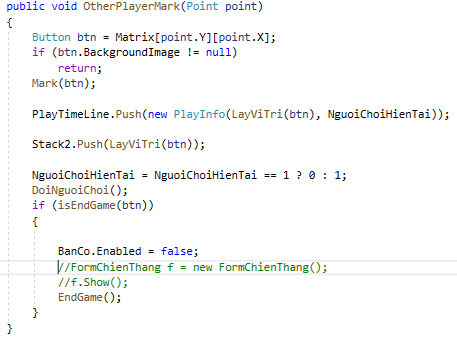
+ Button Undo false và tạo luồng cho server bắt đầu lắng nghe.



Hình 27: ChessBoard\_PlayerMarked()

Tạo một hàm để truyền vị trí của người ta đánh.

Cho button bằng Matrix. Gọi PlayerMarked trong btnClick và sử dụng chung.



Hình 28: OtherPlayerMark()

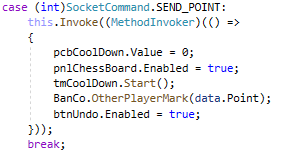
Gọi hàm OtherPlayerMark() trong Send\_Point để lấy Point.

+ Giá trị của pcbCoolDown bằng 0.

+ pnlChessBoard là true.

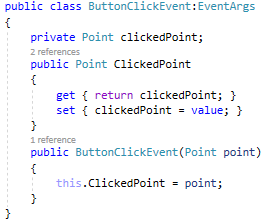
+ tmCoolDown bắt đầu chạy.

+ btnUndo cho là true.



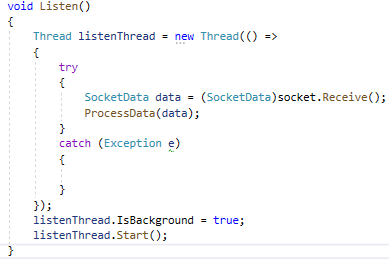
Hình 29: SEND\_POINT

Tạo một class ButtonClickEvent kế thừa EventArgs để lưu lại tọa độ của button click, và tạo hàm dựng để truyền Point vào. Sử dụng trong ChessBoard\_PlayerMarked.



Hình 30: Class ButtonClickEvent

Tạo Listen() để lắng nghe truyền data. Khi server nhấn btnLan để kết nối thì sẽ gửi kết nối qua client và client nhận dữ liệu thông qua hàm listen().

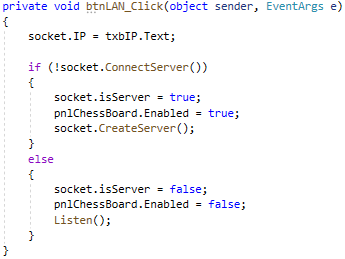


Hình 31: Listen()

BtnLan để tạo kết nối và quyết định người đánh trước.

+ Nếu không thể kết nối với Server tức là chưa có server thì tạo server.

+ Nếu như là server thì pnlChessBoard là true, còn client thì false.



Hình 32: btnLan\_Click()

2.5. Sản phẩm của đồ án

Một game caro hoàn chỉnh dành cho hai người chơi kết nối với nhau qua LAN.



Hình 33: Giao diện chính

# CHƯƠNG 3 : KẾT LUẬN

3.1. Kết luận

Với mục tiêu đã đặt ra thì nhóm tự tin đã hoàn thành được hơn 90% kế hoạch ban đầu. Nhóm đã xây dựng được một game caro hoàn chỉnh cho hai người chơi trên LAN.

- Mục tiêu đề ra:

+

- Đã hoàn thành:

+

3.2. Ưu điểm

Sử dụng hợp lý các chức năng của stack (ngăn xếp) vào cài đặt game caro hai người chơi với mục đích lưu trữ các nước cờ của người chơi kết hợp với kết nối LAN. Nhóm đã tạo ra nhiều chứ năng trong game caro giúp người chơi dễ dàng sử dụng và thích thú với game hơn.

Dựa trên kết nối LAN người chơi có thể chơi ở hai thiết bị khác nhau thông qua IP.

Giao diện dễ nhìn, phù hợp với mọi lứa tuổi.

3.3. Nhược điểm

- Vẫn có thể đánh tiếp ván cờ khi mở ván cờ đã kết thúc.

- Form chiến thắng xuất hiện nhiều lần khi kết thúc ván cờ.

3.4. Đề xuất phương án cải thiện

- Phát triển game có thể chơi trên web với những ngôn ngữ lập trình khác.

- Tạo thêm chức năng chơi với máy.

- Khi kết thúc game chỉ xuất hiện một form thông báo chiến thắng.

- Phát triển về mặt mô phỏng hay lưu ván cờ đã chơi.

- Có thể lưu tên của người chiến thắng.

# PHỤ LỤC

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Công việc | | Võ Thị Kiều Diễm | Trần Gia Hân | Ngày bắt đầu dự kiến | Ngày kết thúc dự kiến | Ngày bắt đầu thực tế | Ngày kết thúc thực tế |
| Thiết kế giao | Tạo background, tạo các khung sườn | X | X | 25/9 | 26/9 | 25/9 | 26/9 |
| Add vào các hình ảnh |  | X | 25/9 | 27/9 | 25/9 | 26/9 |
| Tạo bàn cờ | X |  | 25/9 | 25/9 | 25/9 | 26/9 |
| Xử lý đổi người chơi | Thiết lập ra hai người chơi | X |  | 27/9 | 28/9 | 25/9 | 26/9 |
| Xử lý đổi người |  | X | 27/9 | 28/9 | 26/9 | 27/9 |
| Xử lý thắng thua | Chuyển tất cả các button vào 2 list lồng nhau để xử lý như trên ma trận. |  | X | 30/9 | 3/10 | 26/9 | 27/9 |
| Kiểm tra hàng 5 (ngang, dọc, chéo chính, chéo phụ) |  | X | 1/10 | 3/10 | 27/9 | 27/9 |
| Xác định trạng thái thắng, kiểm tra người thắng. | X |  | 4/10 | 8/10 | 27/9 | 28/9 |
| Tạo chức năng Undo | Tìm hiểu stack | X | X | 9/10 | 13/10 | 29/9 | 2/10 |
| Tạo các phím chức năng | New game, hướng dẫn, mô phỏng, undo, exit, music, chế độ chơi | X | X | 17/10 | 20/10 | 2/10 | 22/10 |
| Kết nối LAN | Tìm hiểu và tạo kết nối LAN | X | X | 14/10 | 18/10 | 30/9 | 10/10 |
| Hiển thị quân cờ đối phương qua mạng LAN | X |  | 14/10 | 19/10 | 11/10 | 22/10 |
| Debug và thử nghiệm | | X | X | 22/10 | 23/10 | 23/10 | 25/10 |
| Viết báo cáo | File Word | X | X | 24/10 | 26/10 | 27/10 | 1/11 |
| Thuyết trình | File Power Point | X | X | 25/10 | 27/10 | 1/11 | 5/11 |

# TÀI LIỆU THAM KHẢO

[1] - https://vi.wikipedia.org/wiki/C%E1%BB%9D\_ca-r%C3%B4

[2] -https://vi.wikipedia.org/wiki/Ng%C4%83n\_x%E1%BA%BFp

[3] https://www.totolink.vn/article/151-lan-la-gi-tong-quan-kien-thuc-ve-lan-ma-ban-nen-biet.html