

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG KHOA CÔNG NGHỆ THÔNG TIN**

**—————————————–**

BÁO CÁO THỰC TẬP CƠ SỞ

**VIẾT CHƯƠNG TRÌNH CHƠI CỜ CARO 2 NGƯỜI CHƠI THEO PHƯƠNG PHÁP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn** | **:** | **TS. Nguyễn Đình Hưng** |
| **Sinh viên thực hiện** | **:** | **Nguyễn Tuấn Kiệt** |
| **Lớp học phần** | **:** | **62.CNTT-2** |
| **Mã sinh viên** | **:** | **62130887** |
|  |  |  |

**Nha Trang – 12/2022**

**Mục lục**

[Chương 1: Giới thiệu 1](#_Toc123633105)

[1. Sơ lược về game cờ Caro 1](#_Toc123633106)

[1.1. Lịch sử game cờ Caro 1](#_Toc123633107)

[1.2. Luật chơi cơ bản của cờ Caro 1](#_Toc123633108)

[2. Sơ lược về mạng LAN 1](#_Toc123633109)

[2.1. Khái niệm mạng LAN 1](#_Toc123633110)

[2.2. Cách kết nối cơ bản của mạng LAN 1](#_Toc123633111)

[3. Yêu cầu đề tài 2](#_Toc123633112)

[3.1. Nội dung 2](#_Toc123633113)

[3.2. Phương pháp và công cụ phát triển 2](#_Toc123633114)

[Chương 2: Cơ sở lý thuyết 5](#_Toc123633115)

[1. Phương pháp lập trình hướng đối tượng 5](#_Toc123633116)

[1.1. Các khái niệm trong lập trình hướng đối tượng 5](#_Toc123633117)

[1.2. Đặc điểm cơ bản của lập trình hướng đối tượng 5](#_Toc123633118)

[1.3. Các tính chất trong lập trình hướng đối tượng 5](#_Toc123633119)

[2. Ngôn ngữ lập trình C# 6](#_Toc123633120)

[1.2. Đặc điểm của C# 6](#_Toc123633121)

[3. Tìm hiểu Winform 6](#_Toc123633122)

[1.1. Tổng quan 6](#_Toc123633123)

[1.2. Cơ bản về Winform 7](#_Toc123633124)

[4. Truyền dữ liệu trong mạng LAN 13](#_Toc123633125)

[4.1. Giao thức TCP/IP 13](#_Toc123633126)

[4.2. Socket 13](#_Toc123633127)

[4.3. Server và Client 13](#_Toc123633128)

[5. Giải thuật “Kiểm tra thắng thua” 15](#_Toc123633129)

[5.1. Lấy tọa độ quân cờ 15](#_Toc123633130)

[5.2. Duyệt theo chiều ngang 15](#_Toc123633131)

[5.3. Duyệt theo chiều dọc 16](#_Toc123633132)

[5.4. Duyệt theo đường chéo chính 17](#_Toc123633133)

[5.5. Duyệt theo đường chéo phụ 18](#_Toc123633134)

[Chương 3: Cài đặt chương trình 21](#_Toc123633135)

[1. Giao diện 21](#_Toc123633136)

[2. Các lớp và một số hàm cơ bản trong lớp 21](#_Toc123633137)

[1.1. ChessBoardCaro.cs [Design] 21](#_Toc123633138)

[1.2. Cons.cs 21](#_Toc123633139)

[1.3. Player.cs 21](#_Toc123633140)

[1.4. PlayInfo.cs 21](#_Toc123633141)

[1.5. ChessBoardManager.cs 22](#_Toc123633142)

[1.6. SocketData.cs 23](#_Toc123633143)

[1.7. SocketManager.cs 23](#_Toc123633144)

[1.8. ChessBoardCaro.cs 24](#_Toc123633145)

[3. Mã nguồn 26](#_Toc123633146)

[Chương 4: Kết luận 27](#_Toc123633147)

[1. Đánh giá chung về đề tài 27](#_Toc123633148)

[1.1. Những kết quả đạt được 27](#_Toc123633149)

[1.2. Một số hạn chế của đề tài 27](#_Toc123633150)

[2. Hướng phát triển của đề tài 27](#_Toc123633151)

[TÀI LIỆU THAM KHẢO 28](#_Toc123633152)

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

**NHẬN XÉT**

**(Của giảng viên phản biện)**

# Chương 1: Giới thiệu

## Sơ lược về game [cờ Caro](https://docs.google.com/document/d/1GMxtApV8ll5giVbwsGKQAIXZFffxdqY3/edit#heading=h.30j0zll)

### Lịch sử game cờ Caro

Cờ ca-rô (hay gomoku) là một trò chơi dân gian. 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 chuỗi 5 quân liên tục mớ. 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.

### [Luật chơi cơ bản của cờ Caro](https://docs.google.com/document/d/1GMxtApV8ll5giVbwsGKQAIXZFffxdqY3/edit#heading=h.1fob9te)

Bàn cờ có kích thước tuỳ ý , chọn : Nếu tồn tại đúng 5 con liên tiếp trên 1 hàng là thắng (chéo,ngang,dọc).

## Sơ lược về mạng LAN

### Khái niệm mạng LAN

LAN là viết tắt của Local Area Network tạm dịch là mạng máy tính nội bộ, giao tiếp này cho phép các máy tính kết nối với nhau để cùng làm việc và chia sẻ dữ liệu.Nó chỉ có thể sử dụng được trong một phạm vi giới hạn như phòng làm việc, trong nhà, trường học…

### Cách kết nối cơ bản của mạng LAN

* Các thiết bị trong cùng mạng LAN có thể kết nối với nhau thông qua sợi cáp mạng.
* Ngoài ra, mạng LAN còn có thể được thiết lập bằng cổng kết nối không dây (Wireless) và được gọi chung là WLAN (Wireless LAN), hay chúng ta thường gọi là Wifi.

## [Yêu cầu đề tài](https://docs.google.com/document/d/1GMxtApV8ll5giVbwsGKQAIXZFffxdqY3/edit#heading=h.3znysh7)

### Nội dung

Xây dựng một ứng dụng chơi cờ Caro 2 người chơi qua mạng LAN theo phương pháp lập trình hướng đối tượng trong môi trường đồ họa.

### Phương pháp và công cụ phát triển

#### Phương pháp lập trình hướng đối tượng

* Chia chương trình thành các đối tượng.
* Tập trung vào dữ liệu hơn là thuật toán.
* Bảo mật hơn (phạm vi truy nhập).
* Dễ chỉnh sửa chương trình.

#### Công cụ phát triển

##### .Net Framework

* .NET Framework là một khung phần mềm độc quyền do Microsoft phát triển, chạy chủ yếu trên Microsoft Windows.
* .NET là một nền tảng dành cho nhà phát triển đa nền tảng để xây dựng nhiều loại ứng dụng khác nhau (web, mobile, desktop, games, and IoT).
* Cung cấp một môi trường hướng đối tượng nhất quán cho nhiều loại ứng dụng.
* Cung cấp một nền tảng phát triển chung cho nhiều ngôn ngữ lập trình khác nhau của Microsoft: C#, Visual J#, Visual Basic…
* .Net Framework: là 1 trong 3 phiên bản .NET hỗ trợ phát triển các ứng dụng (website, service, desktop app, ..) trên Windows.

##### Visual Studio

Microsoft Visual Studio (VS) là một IDE cung cấp:

* Các chức năng cơ bản: viết mã, build và debug.
* Làm việc nhóm thông qua Team Foundation Server của Microsoft.
* Các phím tắt và plugins hỗ trợ người dùng thao tác nhanh trong việc viết mã.
* Tùy chỉnh liên kết các project và thư viện, tập tin liên quan.

##### Ngôn ngữ lập trình C#

* C# được phát triển bởi nhóm tác giả điều hành bởi Anders Hejlsberg và Scott Wiltamuth.
* Ngôn ngữ C# là một ngôn ngữ lập trình hướng đối tượng được xây dựng trên nền tảng của hai ngôn ngữ mạnh là C++ và Java.

##### Một số thư viện sử dụng

###### System.Windows.Forms

Windows Forms (WinForms) là thư viện lớp đồ họa (GUI) mã nguồn mở và miễn phí được bao gồm như một phần của Microsoft.NET Framework hoặc Mono Framework, cung cấp nền tảng để viết các ứng dụng khách phong phú cho máy tính để bàn, máy tính xách tay và máy tính bảng.

###### System.Drawing

Lớp Đồ họa cung cấp các phương pháp để vẽ đến thiết bị hiển thị. Các lớp như Hình chữ nhật và Điểm đóng gói các nguyên thủy GDI +. Lớp Pen được sử dụng để vẽ các đường thẳng và đường cong, trong khi các lớp có nguồn gốc từ lớp trừu tượng Brush được sử dụng để lấp đầy nội thất của các hình dạng.

###### System.Runtime.Serialization.Formatters.Binary

Tuần tự hóa và deserializes (giải mả) một đối tượng, hoặc toàn bộ một đồ thị của các đối tượng được kết nối, ở định dạng nhị phân.

###### System.Net.NetworkInformation

Cung cấp quyền truy cập vào dữ liệu lưu lượng mạng, thông tin địa chỉ mạng và thông báo về các thay đổi địa chỉ cho máy tính cục bộ. Không gian tên cũng chứa các lớp thực hiện tiện ích Ping. Bạn có thể sử dụng Ping và các lớp liên quan để kiểm tra xem máy tính có thể truy cập được trên mạng hay không.

###### System.Net

Cung cấp một giao diện lập trình đơn giản cho nhiều giao thức được sử dụng trên các mạng hiện nay. Các lớp trong không gian tên System.Net có thể được sử dụng để phát triển các ứng dụng Windows Store hoặc ứng dụng dành cho máy tính để bàn.

###### System.Net.Sockets

Cung cấp quản lý của giao diện Windows Sockets (Winsock) cho các nhà phát triển cần kiểm soát chặt chẽ quyền truy cập vào mạng.

# Chương 2: Cơ sở lý thuyết

## Phương pháp lập trình hướng đối tượng

### Các khái niệm trong lập trình hướng đối tượng

* **Lập trình hướng đối tượng** (OOP - Object Oriented Programming): là một phương pháp lập trình sử dụng các đối tượng để xây dựng hệ thống, ứng dụng.
* **Đối tượng** (Object): Một đối tượng là một thực thể trong thế giới thực có định danh, trạng thái và các hành vi.
* **Lớp** (Class): Lớp là khái niệm dùng để mô tả một tập hợp các đối tượng có cùng một cấu trúc, cùng hành vi và có cùng những mối quan hệ với các đối tượng khác.
* **Thông điệp** (message): Một thông điệp là một lời yêu cầu thực hiện một hành vi. Trong lập trình, một thông điệp là lời gọi một phương thức của một đối tượng.

### Đặc điểm cơ bản của lập trình hướng đối tượng

* Chương trình được chia thành các đối tượng.

Đối tượng = Dữ liệu + Phương thức

* Dữ liệu của các đối tượng được truy cập chỉ bởi các phương thức liên quan đến đối tượng đó.
* Các phương thức của một đối tượng có thể truy nhập đến các phương thức của các đối tượng khác.

### Các tính chất trong lập trình hướng đối tượng

#### Tính trừu tượng (Abstraction)

* Là quá trình loại bỏ đi các thông tin cụ thể và giữ lại những thông tin chung.
* Cung cấp một cái nhìn tổng quát về các lớp hoặc đối tượng.

#### Tính đóng gói (Encapsulation)

* Là cơ chế ràng buộc dữ liệu và thao tác trên dữ liệu thành một thể thống nhất.
* Ẩn đi các chi tiết bên trong của đối tượng.

#### Tính đa hình (Polymorphism)

* Một tên nhưng có thể có nhiều dạng thức khác nhau.
* Hành động cùng tên có thể được thực hiện theo các cách khác nhau đối với các đối tượng/ lớp khác nhau (ngữ cảnh khác kết quả khác).

#### Tính kế thừa (Inheritance)

Cho phép chúng ta tạo các lớp mới sử dụng dựa trên lớp có sẵn và có thể mở rộng chúng.

## Ngôn ngữ lập trình C#

#### Cơ bản về ngôn ngữ lập trình C#

* C# là ngôn ngữ hiện đại.
* C# là một ngôn ngữ lập trình thuần hướng đối tượng.
* C# là một ngôn ngữ ít từ khóa.

### Đặc điểm của C#

* Hỗ trợ đầy đủ các cơ chế đặc trưng của lập trình hướng đối tượng.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.

## Tìm hiểu Winform

### Tổng quan

Windows Form là nền tảng GUI cho ứng dụng desktop được viết dùng .NET FrameWork.

### Cơ bản về Winform

#### Form

Form dùng để tạo giao diện cho chương trình. Các loại Form:

* Multiple Document Interface (MDI).
* Graphical user interface, application

  Description automatically generatedGraphical user interface, application, Word

  Description automatically generatedSingle Document Interface (SDI).

Hình 2. 1: Single Document Interface (SDI)

Hình 2. 2: Multiple Document Interface (MDI)

#### Button

* Button: Đối tượng nút ấn cho phép thực hiện một chức năng.
* Graphical user interface, application

  Description automatically generatedCó thể hiển thị hình ảnh kết hợp với văn bản.

Button

Hình 2. 3: Button

* Các thuộc tính thường dùng:
* Text: Văn bản hiển thị trên đối tượng.
* Image: Hình ảnh hiển thị trên đối tượng.
* Các sự kiện thường dùng:
* Click: Xảy ra khi nhấn con trỏ chuột hoặc gõ Enter trên đối tượng (ngầm định khi nháy đúp chuột trong chế độ thiết kế).

#### Label

* Label: Đối tượng hiển thị văn bản kết hợp hình ảnh.
* Logo

  Description automatically generatedKhông sửa được văn bản hiển thị.

Label

Hình 2. 4: Label

* Các thuộc tính thường dùng:
* AutoSize: Tự thay đổi kích thước của đối tượng.
* Fonts: Font chữ của đối tượng Label.
* ForeColor: Màu chữ của đối tượng.
* Image: Ảnh của đối tượng.
* Text: Văn bản xuất hiện trên đối tượng.
* TextAlign: Lề của văn bản.

#### Textbox

* TexBox: Đối tượng dùng để nhập dữ liệu từ bàn phím.

TextBox

Hình 2. 5: TextBox

* Các thuộc tính thường dùng:
* Enabled: Có/không cho phép thao tác đối tượng.
* Multiline: Có/không cho phép nhập dữ liệu nhiều dòng (ngầm định là không).
* PasswordChar: Nhập ký tự làm mật khẩu.
* ReadOnly: Có/không cho phép sửa dữ liệu của đối tượng (ngầm định là có).
* Text: Văn bản nhập (hiển thị) của đối tượng.
* Các sự kiện thường dùng:
* TextChanged: Xảy ra khi nhập hoặc xoá các ký tự (ngầm định khi nháy đúp chuột trong chế độ thiết kế).
* KeyDown: Xảy ra khi ấn một phím bất kỳ trên đối tượng.
* KeyUp: Xảy ra khi thả một phím ấn trên đối tượng.

#### PictureBox

* Sử dụng để hiển thị ảnh dạng bitmap, icon, JPEG, GIF,…
* Icon

  Description automatically generated with medium confidenceSử dụng thuộc tính Image để thiết lập ảnh lúc design hoặc runtime.

Hình 2. 6: PictureBox

* Các thuộc tính:
* Image: ảnh cần hiển thị.
* SizeMode: Normal, StretchImage, AutoSize.

#### MessageBox

* Dùng để hiển thị hộp thọai với tiêu đề, các loại nút và biểu tượng khác nhau.
* Dùng để hiển thị một thông báo.
* Graphical user interface, text, application, chat or text message

  Description automatically generatedDùng để xác nhận một hành động.

Hình 2. 7: MessageBox

#### ProgressBar

* A picture containing shape

  Description automatically generatedProgressBar được sử dụng để hiển thị thời gian thực hiện của một công việc nào đó.

ProgressBar

Hình 2. 8: ProgressBar

* Các thuộc tính:
* Maximum: Giá trị tối đa của ProgressBar. Khi ProgressBar được lấp đầy nghĩa là ProgressBar đã đạt giá trị Maximum.
* Minimum: Giá trị nhỏ nhất của ProgressBar. Khi ProgressBar trông rỗng nghĩa là ProgressBar đang có giá trị Minimum.
* Value: Giữ giá trị hiện tại của ProgressBar, giá trị này nằm trong đoạn Minimum và Maximum.
* Style: Kiểu hiển thị của ProgressBar.
* Step: Lượng giá trị thêm vào Value khi phương thức PerformStep() được gọi.
* Một số phương thức:
* PerformStep(): Phương thức giúp tăng ProgressBar. Giá trị tăng là giá trị được thiết lập trong thuộc tính Step.
* Increment(<giá trị>): Phương thức giúp tang ProgressBar. Giá trị tăng là tham số đầu vào <giá trị> của phương thức.

#### MenuStrip

* Graphical user interface, text, application, table

  Description automatically generatedCho phép thiết kế menu trên Form.

Hình 2. 9: MenuStrip

* Các thuộc tính thường dùng:
* Name: Tên menu được dùng trong mã lệnh.
* ShortCutKey: Đặt phím nóng cho menu.
* ShowShortcut: Có/Không phím nóng hiển thị trên dòng menu. Ngầm định là true.
* Text: Xuất hiện trên dòng menu.
* Sự kiện thường dùng:
* Click: Xảy ra khi một dòng của menu được click chuột hoặc ấn phím nóng.

#### Timer

* Điều khiển Timer cho phép thực thi lại một hành động sau một khoảng thời gian xác định.
* Các thuộc tính:
* Interval: Thiết lập giá trị là một số nguyên. Giá trị nguyên này là thời lượng của một chu kì tính.
* Enable: Thiết lập giá trị True hoặc False. Nếu là giá trị True thì điều khiển Timer hoạt động, nếu là giá trị False thì điều khiển Timer không hoạt động.
* Một số phương thức thường dùng:
* Start(): Start() kích hoạt điều khiển Timer hoạt động. Phương thức này tương ứng với việc thiết lập giá trị thuộc tính Enable là True.
* Stop(): Dừng hoạt động của điều khiển Timer. Phương thức này tương ứng với việc thiết lập giá trị thuộc tính Enable là False.
* Sự kiện thường dùng
* Tick: Tick là sự kiện được gọi trong mỗi chu kỳ Interval.

## Truyền dữ liệu trong mạng LAN

### Giao thức TCP/IP

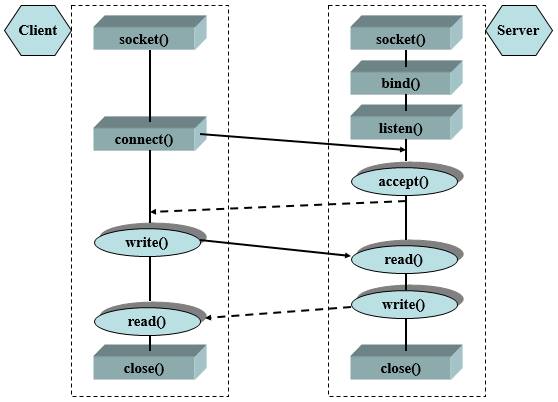
* TCP/IP hoặc Transmission Control Protocol/Internet Protocol (Giao thức điều khiển truyền vận/giao thức mạng) là một bộ các giao thức trao đổi thông tin được sử dụng để kết nối các thiết bị mạng trên Internet. TCP/IP có thể được sử dụng như là một giao thức trao đổi thông tin trong một mạng riêng (intranet hoặc extranet).
* TCP/IP sử dụng mô hình giao tiếp máy khách/máy chủ, trong đó người dùng hoặc thiết bị (máy khách) được một máy tính khác (máy chủ) cung cấp một dịch vụ (giống như gửi một trang web) trong mạng.

### Socket

Socket là điểm cuối end-point trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.

### Server và Client

#### Mô hình ứng dụng



Hình 2. 10: Mô hình ứng dụng Client-Server (sinhvientot.net, 2022)

t

#### Thao tác bên Server

* Tạo ra một Sockets.
* Gắn Sockets đó với một địa chỉ cụ thể (binding).
* Lắng nghe kết nối tới (Phương thức Listen()).
* Sau khi socket đã được gắn với một địa chỉ cụ thể, sử dụng phương thức Listen() để đặt socket ở trạng thái lắng nghe kết nối từ Client tới.
* Chấp nhận kết nối.

#### Thao tác bên Client

* Tạo ra một Sockets.
* Kết nối đến Server.
* Đợi phản hồi từ Server.

## Giải thuật “Kiểm tra thắng thua”

* Ý tưởng thuật toán: Đầu tiên sẽ lấy tọa độ quân cờ (button) mà người chơi vừa đánh (click) trên bàn cờ. Lấy tọa độ quân cờ (button) làm điểm giữa sau đó thực hiện duyệt bên trái và bên phải hoặc duyệt bên trên và bên dưới xem có bao nhiêu quân cờ giống với quân cờ vừa đánh nếu tổng quân cờ bên trái và bên phải cộng lại bằng 5 hoặc bên trên và bên dưới cộng lại bằng 5 thì trả về giá trị là true ngược lại là false.

### Lấy tọa độ quân cờ

* Ý tưởng thuật toán: Khi người chơi đánh quân cờ (click vào vị trí button trên bàn cờ) sẽ lấy cái Tag của button vừa đánh làm tọa độ x (vertical), với tọa độ y (horizontal) sẽ bằng vị trí xuất hiện đầu tiên của button thứ x trong matrix (bàn cờ).
* Mã giả:

#Lấy tọa độ x, y

vertical = Convert.ToInt32(button.Tag)

horizontal = Matrix[vertical].IndexOf(button)

#Tạo Point để lưu trữ tọa độ

Point = Point(horizontal, vertical)

return Point

### Duyệt theo chiều ngang

* Mã giả:

Point = Tọa độ button người dùng vừa đánh quân cờ

#Duyệt LEFT

countLeft = 0

FOR i in range (0; point.X):

IF Matrix[point.Y][i].BackgroundImage == Button.BackgroundImage:

countLeft++

ELSE: Break

#Duyệt RIGHT

countRight = 0

FOR i in range (point.X + 1; Chess Board Width):

IF Matrix[point.Y][i].BackgroundImage == Button.BackgroundImage:

countRight++

ELSE: Break

RETURN countLeft + countRight == 5

### Duyệt theo chiều dọc

* Mã giả:

Point = Tọa độ button người dùng vừa đánh quân cờ

#Duyệt TOP

countTOP = 0

FOR i in range (0; point.Y):

IF Matrix[i][point.X].BackgroundImage == Button.BackgroundImage:

countTop++

ELSE: Break

#Duyệt BOTTOM

countBottom = 0

FOR i in range (point.X + 1; Chess Board Height):

IF Matrix[i][point.X].BackgroundImage == Button.BackgroundImage:

countBottom++;

ELSE Break

Return countTop + countBottom == 5

### Duyệt theo đường chéo chính

* Mã giả:

Point = Tọa độ button người dùng vừa đánh quân cờ

#Duyệt TOP

countTop = 0

FOR i in range (0; point.X):

IF point.X - i < 0 OR point.Y - i < 0

Break

IF Matrix[point.Y-i][point.X-i].BackgroundImage == btn.BackgroundImage:

countTop++

ELSE: Break

#Duyệt BOTTOM

countBottom = 0

FOR i in range (1; Chess Board Width - point.X)

IF point.Y + i >= ChessBoardHeight OR point.X + i >= ChessBoardWidth:

Break

IF Matrix[point.Y + i][point.X + i].BackgroundImage

== Button.BackgroundImage:

countBottom++

ELSE: Break

return countTop + countBottom == 5

### Duyệt theo đường chéo phụ

* Mã giả:

Point = Tọa độ button người dùng vừa đánh quân cờ

#Duyệt TOP

countTop = 0

FOR in range (0; point.X):

IF point.X + i > ChessBoardWidth OR point.Y - i < 0:

Break

IF Matrix[point.Y - i][point.X + i].BackgroundImage

== Button.BackgroundImage:

countTop++

ELSE: Break

#Duyệt BOTTOM

countBottom = 0

FOR i in range (1; ChessBoardWidth - point.X):

IF point.Y + i >= ChessBoardHeight OR point.X - i < 0:

Break

IF Matrix[point.Y + i][point.X - i].BackgroundImage

== Button.BackgroundImage)

countBottom++

ELSE: Break

RETURN countTop + countBottom == 5

# Chương 3: Cài đặt chương trình

## A picture containing graphical user interface Description automatically generatedGiao diện

Hình 3. 1: Giao diện game Caro trên Winform

## Các lớp và một số hàm cơ bản trong lớp

### ChessBoardCaro.cs [Design]

Quản lý thiết kế giao diện UID cho game Caro.

### Cons.cs

Lưu trữ các giá trị hằng số được sử dụng trong chương trình:

* Kích thước mỗi quân cờ.
* Kích thước bàn cờ.
* Giá trị cài đặt của ProgressBar bao gồm: step, time và interval.

### Player.cs

Chứa thuộc tính name và mark của đối tượng Player.

### PlayInfo.cs

Chứa thuộc tính point và currentPlayer của Player để truyền qua lại giữa Client và Server khi kết nối mạng LAN.

### ChessBoardManager.cs

* Thuộc tính:
* Panel chessBoard.
* List<Player> player.
* int currentPlayer.
* TextBox playerName.
* PictureBox playerMark.
* List<List<Button>> matrix.
* event EventHandler<ButtonClickEvent> playerMarked.
* event EventHandler endedGame.
* Phương thức:
* DrawChessBoard(): Vẽ bàn cờ và lưu trữ toàn bộ vị trí button trên bàn cờ vào mảng lồi lõm Matrix.
* btn\_Click(object sender, EventArgs e): Xử lý sự kiện khi người chơi click vào ô cờ trên bàn cờ sẽ kiểm tra xem ai đang đánh và đổi sang người chơi khác. Nếu button click đủ điều kiện thắng sẽ gọi phương thức EndGame() để kết thúc trò chơi.
* OtherPlayerMark(Point point): Kiểm tra xem button có tọa độ vừa truyền vào đã có BackgroundImage không nếu không sẽ gọi phương thức Mark(btn) để thay BackgroundImage sau đó đổi người chơi khác. Nếu button có tọa độ vừa truyền vào đủ điều kiện thắng sẽ gọi phương thức EndGame() để kết thúc trò chơi.
* EndGame(): Kiểm tra xem EventHandler endedGame đã được ủy thác chưa nếu chưa được ủy thác sẽ truyền vào class gọi nó và event endedGame.
* isEndGame(Button btn): Kiểm tra điều kiện thắng thua.
* GetChessPoint(Button btn): Lấy tọa độ người chơi vừa đánh quân cờ trên bàn cờ.
* isEndHorizontal(Button btn): Kiểm tra thắng thua theo chiều ngang.
* isEndVertical(Button btn): Kiểm tra thắng thua theo chiều dọc.
* isEndPrimary(Button btn): Kiểm tra thắng thua theo đường chéo chính.
* isEndSub(Button btn): Kiểm tra thắng thua theo đường chéo phụ.
* Mark(Button btn): Thay đổi kí hiệu người chơi.
* ChangePlayer(): Thay đổi tên và hình ảnh người chơi.
* class ButtonClickEvent : EventArgs : chứa thuộc tính clickedPoint để xác nhận button người dùng vừa nhấn trên bàn cờ để kích hoạt sự kiện.

### SocketData.cs

Chứa thuộc tính command, point, message của đối tượng Client và Server.

### SocketManager.cs

* Phương thức của Client:

ConnectServer(): Tạo kết nối đến Server.

* Phương thức của Server:

CreateServer(): Tạo Server và đợi chấp nhận yêu cầu kết nối từ Client.

* Phương thức chung của Client và Server (Both):
* Thuộc tính: IP, PORT, BUFER, isServer.
* Phương thức
* Send(object data): Gửi đối tượng ở dạng data.
* Receive(): Nhận data ở dạng đối tượng.
* SendData(Socket target, byte[] data): Kiểm tra xem có gửi data thành công hay không.
* ReceiveData(Socket target, byte[] data): Kiểm tra xem nhận data thành công hay không.
* SerializeData(Object o): Nén đối tượng thành mảng byte[].
* DeserializeData(byte[] theByteArray): Giải nén mảng byte[] thành đối tượng object.
* GetLocalIPv4(NetworkInterfaceType \_type): tự động Lấy ra IP V4 của card mạng đang dùng.

### ChessBoardCaro.cs

Phương thức:

* EndGame(): Dừng thanh Pro và vô hiệu hóa bàn cờ.
* NewGame(): Dừng thanh Pro và làm mới value của nó sau đó vẽ lại bàn cờ.
* Quit(): Thoát khỏi ứng dụng Game Caro.
* ChessBoard\_PlayerMarked(object sender, ButtonClickEvent e): Xử lý sự kiện khi người chơi đánh vào bàn cờ sẽ kích hoạt Timer, vô hiệu hóa bàn cờ và đặt value của thanh ProgressBar về 0 sau đó gửi tọa độ quân cờ vừa đánh đến máy tính được kết nối mạng LAN và gọi phương thức Listen() để lắng nghe thông điệp từ Server.
* ChessBoard\_EndedGame(object sender, EventArgs e): Xử lý sự kiện endedGame gọi phương thức EndGame() để kết thúc game và gửi Command “END\_GAME” đến máy được kết nối mạng LAN để cùng kết thúc game.
* tmCoolDown\_Tick(object sender, EventArgs e): Xử lý sự kiện Tick của Timer sẽ cho thanh ProgressBar bắt đầu chạy. Nếu ProgressBar đạt giá trị maximum sẽ gọi phương thức EndGame() để kết thúc game.
* newGameToolStripMenuItem\_Click(object sender, EventArgs e): Xử lý sự kiện click vào New Game trong MenuStrip sẽ tiến hành gọi phương thức NewGame() để làm mới game, gửi command “NEW\_GAME” đến máy kết nối mạng LAN và kích hoạt bàn cờ để tiến hành đánh trước.
* quitToolStripMenuItem\_Click(object sender, EventArgs e): Xử lý sự kiện click vào Quit trong MenuStrip sẽ thoát khỏi ứng dụng Game.
* Form1\_FormClosing(object sender, FormClosingEventArgs e): Xử lý sự kiện đóng chương trình sẽ hiển thị thông báo và gửi command đến máy kết nối mạng LAN.
* btnLAN\_Click(object sender, EventArgs e): Xử lý sự kiện click vào button LAN sẽ lấy địa chỉ IP được ghi tại textBox IP trong game đưa vào làm địa chỉ IP của socket sau đó kiểm tra nếu nó chưa có Server sẽ tiến hành tạo Server và kích hoạt bàn cờ để đánh, nếu đã có Server thì sẽ vô hóa bàn cờ và gọi phương thức Listen() lắng nghe thông điệp từ Server.
* Form1\_Shown(object sender, EventArgs e): Xử lý sự kiện Shown của Form sẽ tự động lấy địa chỉ IPv4 của máy đang sử dụng hiển thị vào textBox địa chỉ IP trong Game Caro.
* Listen(): Lắng nghe thông tin từ Client gửi đến Server hoặc Server gửi đến Client mỗi 0,5 giây 1 lần.
* ProcessData(SocketData data): Xử lý command được gửi qua lại giữa Client và Server.

## Mã nguồn

Link mã nguồn: <https://github.com/Rain0809/GameCaro>

# Chương 4: Kết luận

## Đánh giá chung về đề tài

### Những kết quả đạt được

* Tạo được game Caro 2 người chơi qua mạng LAN theo phương pháp lập trình hướng đối tượng.
* Cơ bản có đầy đủ chức năng của 1 game caro hoàn chỉnh.
* Kết nối 2 máy thông qua IP thành công.

### Một số hạn chế của đề tài

* Chưa xử lí chặt chẽ luật chơi.
* Chưa có tính năng Undo để người dùng hủy bước chơi đánh nhầm.
* Khi ứng dụng game thoát khi chưa kết nối LAN sẽ bị lỗi.

## Hướng phát triển của đề tài

Cập nhật và phát triển thêm các tính năng mới như:

* Đánh với máy (sử dụng AI).
* Người đánh với người.
* Thêm các ràng buộc để luật chơi chặt chẽ hơn.
* Thêm tính năng Undo để người chơi có thể lùi lại bước đi nhầm.

# TÀI LIỆU THAM KHẢO

1. Phạm Thị Kim Ngoan. *Bài giảng Lập trình hướng đối tượng*. ĐH Nha Trang
2. Cấn Thị Phượng. *Bài giảng Mạng máy tính*. ĐH Nha Trang
3. CONGDONGLAPTRINGVIETNAM. <https://by.com.vn/thuat-toan-kiem-tra-thang-thua-game-caro>

Truy cập ngày 18/12/2022

1. APTECH. <https://by.com.vn/lap-trinh-socket-tcp-clientserver>

Truy cập ngày 11/12/2022

1. TUHOCICT. <https://by.com.vn/ky-thuat-lap-trinh-voi-tcp-socket>

Truy cập ngày 11/12/2022