TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO BÀI TẬP MÔN HỌC

**LẬP TRÌNH GIAO DIỆN**

**Đề tài:**

**LẬP TRÌNH GAME CỜ CARO**

Giảng viên: VÕ THỊ HỒNG TUYẾT

Sinh viên thực hiện:

NGUYỄN ĐỨC HIẾU

Mã số sinh viên: 1954052027

Lớp: DH19IT03

TP. Hồ Chí Minh, 08–2021

**Mục lục**

1. Phần mở đầu1
2. Nội dung đề tài2
3. Đối tượng sử dụng chương trình2
4. Phần phân tích4
5. Các chức năng của chương trình5
6. Các thành phần dữ liệu trong chương trình5
7. Form06
8. Form16
9. Form26
10. Phần thiết kế4
11. Cấu trúc chương trình5

Cách tổ chức cây thư mục trong chương trình6

1. Chi tiết chương trình5
2. Chức năng của từng Form6
3. Các hàm thành viên trong từng Form6
4. Phần kết quả4
5. Kết quả chạy chương trình5
6. Hướng dẫn sử dụng chương trình5
7. Phần kết luận4
8. Kết quả đạt được5
9. Hạn chế của chương trình5
10. Hướng phát triển tiếp t heo5
11. **Phần mở đầu**
12. **Nội dung đề tài**

Lập trình Game cờ caro với các yêu cầu:

- Game có nhiều màn (cấp độ chơi)

- Chơi giữa hai đối thủ

- Xử lý đúng kỹ thuật chơi và giao diện

- Lưu điểm

1. **Đối tượng sử dụng chương trình:**

Mọi người đều có thể sử dụng chương trình cho mục đích giải trí.

1. **Phần phân tích**
2. **Các chức năng của chương trình**

**-** Xác định ai là người đi nước tiếp theo: Khi người chơi đánh nước cờ X hoàn thành lượt đánh của mình, chương trình sẽ hiển thị lượt chơi tiếp theo của người chơi đánh nước cờ O bằng cách chuyển ô lượt chơi của người chơi đánh nước cờ O sang màu cam và ngược lại khi đến lượt chơi của người chơi đánh nước cờ X.

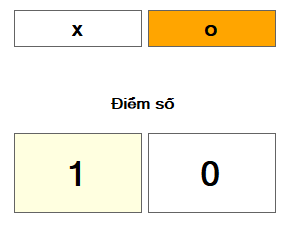


*Lượt chơi của người chơi có nước cờ X*



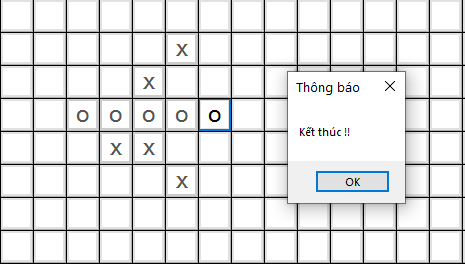
*Lượt chơi của người chơi đánh nước cờ O*

**-** Cộng và lưu điểm người chơi: Khi một trong hai người chơi dành chiến thắng ở nước đi của họ, người chơi sẽ được cộng 1 điểm và cập nhật lên màn hình hiển thị



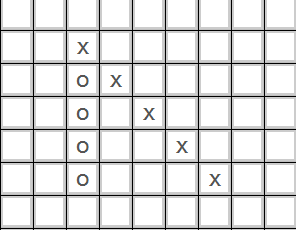
*Màn hình hiển thị điểm số của hai người chơi*

- Thông báo kết thúc ván chơi: Khi một trong hai người chơi thắng, chương trình sẽ hiện thị thông báo “Kết thúc” báo hiệu đã kết thúc ván chơi.



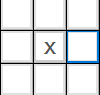
*Thông báo kết thúc ván chơi*

- Chức năng khóa toàn bộ bàn cờ: Sau khi chương trình hiện thị thông báo kết thúc ván chơi, toàn bộ các ô trên bàn cờ sẽ được khóa lại. Mục đích là không cho người chơi thực hiện thao tác đánh tiếp nước cờ khi ván vờ đã kết thúc.



*Toàn bộ các ô trên bàn cờ bị khóa lại, làm mờ đi các ký tự nước cờ khi ván cờ kết thúc*

- Chức năng khóa ô cờ tại vị trí vừa đánh: Sau khi người chơi đánh nước cờ tại một ô bất kỳ thì ô cờ này sẽ bị khóa lại. Mục đích là không cho một trong hai người chơi thay đổi giá trị nước cờ tại ô cờ đã được đánh

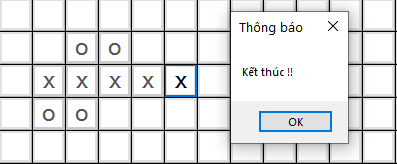


*Ô cờ dã được đánh sẽ bị khóa lại, làm mờ đi ký tự nước cờ*

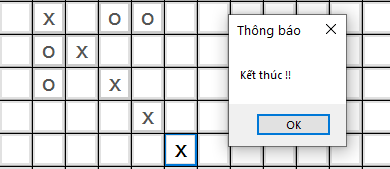
- Chức năng kiểm tra nước cờ thằng: Sau khi một trong hai người chơi đánh nước cờ tại ô cờ bất kỳ, chương trình sẽ kiểm tra các trường hợp nước cờ thằng là 5 ô cờ liên tiếp có giá trị bằng nhau theo các hướng: Dọc, ngang, chéo trái, chéo phải để xác định được người chơi dành chiến thắng



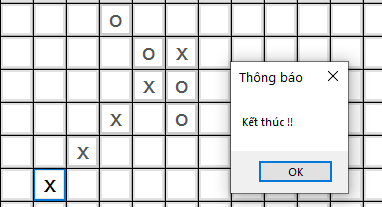
*Nước cờ x thắng, được kiểm tra và thỏa mãn điều kiện thắng theo hàng dọc*



*Nước cờ x thắng, được kiểm tra và thỏa mãn điều kiện thắng theo hàng ngang*



*Nước cờ x thắng, được kiểm tra và thỏa mãn điều kiện thắng theo hàng chéo trái*



*Nước cờ x thắng, được kiểm tra và thỏa mãn điều kiện thắng theo hàng chéo phải*

- Chức năng tiếp tục cuộc đấu giữa hai người: Sau khi kết thúc ván đấu trước, người chơi có thế nhấn nút “Tiếp tục” để tạo bàn cờ mới đồng thời điểm số hiện tại của hai người chơi vẫn được giữ nguyên để sử dụng cho các ván chơi tiếp theo



*Nút Tiếp tục*

- Chức năng chơi lại từ đầu: Sau khi kết thúc ván đấu trước, người chơi có thế nhấn nút “Chơi lại” để tạo bàn cờ mới và điểm số hiện tại của hai người chơi trước đó sẽ được chuyển về 0 để bắt đầu cuộc đấu mới

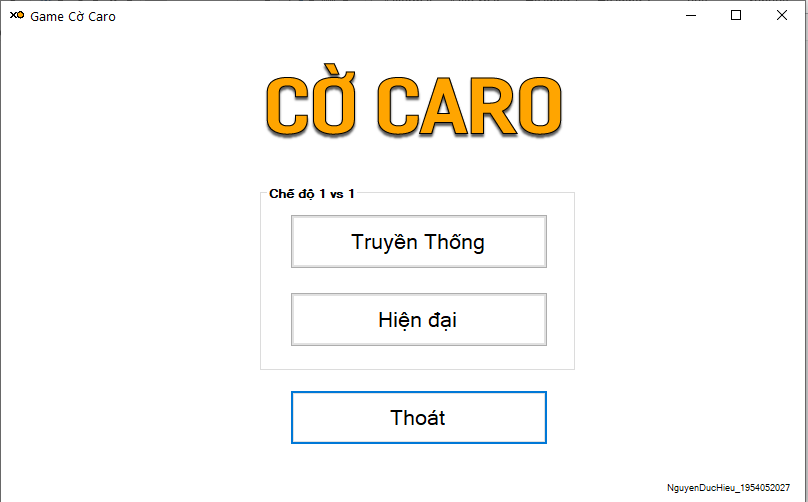


*Nút Chơi lại*

- Chức năng trở về mản hình chính: Trong quá trình chơi, nếu muốn thay đổi chế độ chơi thì người chơi có thể nhấn nút “Trở về” để quay lại màn hình chính và chọn lại chế độ chơi



*Nút Trở về*

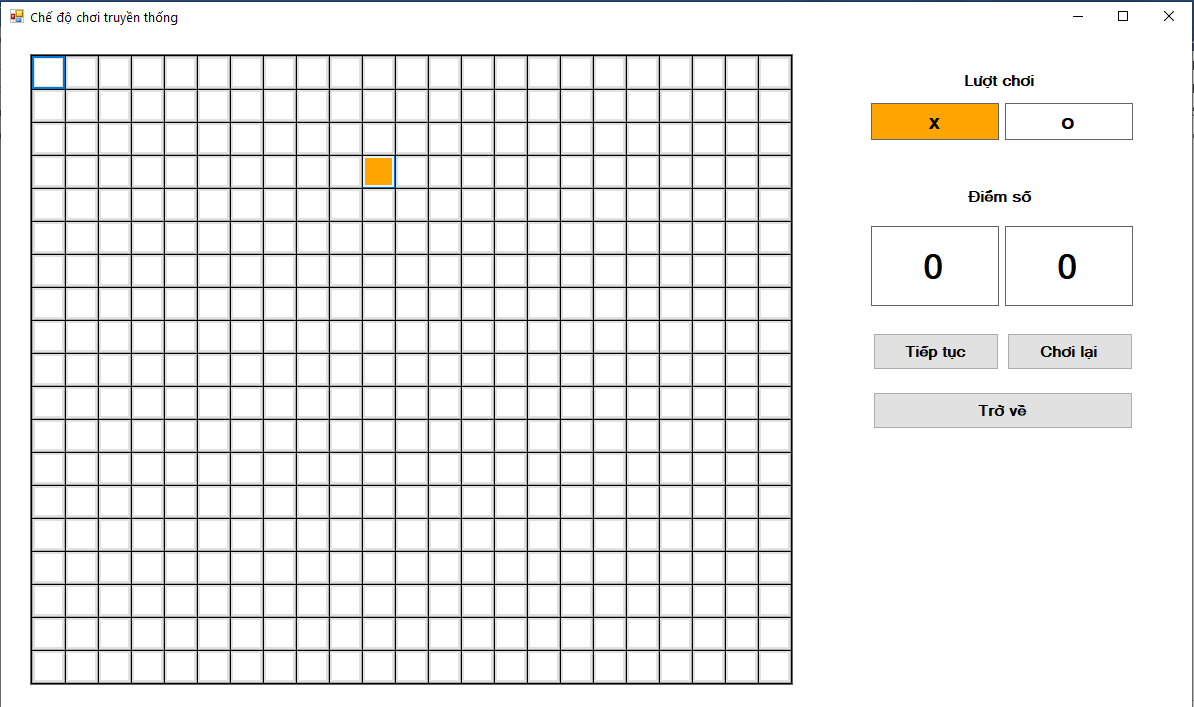


*Màn hình Menu của chương trình*

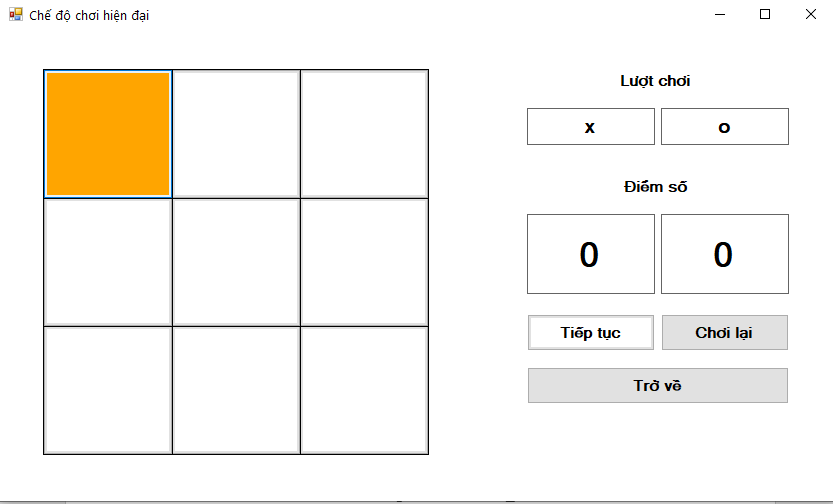
- Chức năng chọn chế độ chơi: Khi mới vào giao diện Menu, người chơi có thể chọn một trong hai chế độ chơi là Truyền thống hoặc Hiện đại. Ứng với mỗi lựa chọn, chương trình sẽ hiện thị giao diện chế độ chơi tương ứng



*Hai chế độ chơi Truyền thống và Hiện đại*



*Giao diện màn hình của chế độ chơi Truyền thống*



*Giao diện màn hình của chế độ chơi Hiện đại*

- Chức năng thoát khỏi trò chơi: Khi người chơi không muốn tiếp tục chơi thì có thể bấm vào nút “Thoát” ở giao diện Menu. Chương trình sẽ hiện thông báo xác nhận rằng người chơi có thật sự muốn thoát hay không, người chơi chọn “Yes” để thoát hoặc chọn “No” để quay lại màn hình Menu



*Nút Thoát*



*Thông báo xác nhận người chơi có thực sự muốn thoát khỏi trò chơi*

1. **Các thành phần dữ liệu trong chương trình**
2. **Form0**

- Form được khai báo trong sự kiện Button\_Click nhằm mục đích chuyển đổi qua lại từ Form0 sang Form1: f1 (kiểu Form)

- Form được khai báo trong sự kiện Button\_Click nhằm mục đích chuyển đổi qua lại từ Form0 sang Form2: f2 (kiểu Form)

1. **Form1**

- Mảng hai chiều chứa số các nguyên: a (kiểu int[,])

- Mảng hai chiều chứa các Button: buttonArray (kiểu Button[,])

- Control: btn (kiểu Button)

- Vị trí x trong Location khi khởi tạo Button: xLocation (kiểu int)

- Vị trí y trong Location khi khởi tạo Button: yLocation (kiểu int)

- Biến cờ xác định lượt chơi của người tiếp theo: checkNextTurnFlag (kiểu int)

- Điểm của người chơi nước cờ X: xScore (kiểu int)

- Điểm của người chơi nước cờ O: oScore (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng dọc: count (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng ngang: countNgang (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng chéo trái: countCheoTrai (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng chéo phải: countCheoPhai (kiểu int)

- Form được khai báo trong sự kiện Button\_Click nhằm mục đích chuyển đổi qua lại từ Form1 sang Form0: f (kiểu Form)

1. **Form2**

- Mảng hai chiều chứa số các nguyên: a (kiểu int[,])

- Mảng hai chiều chứa các Button: btnArray (kiểu Button[,])

- Control: btn (kiểu Button)

- Vị trí x trong Location khi khởi tạo Button: xLocation (kiểu int)

- Vị trí y trong Location khi khởi tạo Button: yLocation (kiểu int)

- Biến cờ xác định lượt chơi của người tiếp theo: checkNextTurnFlag (kiểu int)

- Điểm của người chơi nước cờ X: xScore (kiểu int)

- Điểm của người chơi nước cờ O: oScore (kiểu int)

- Biến cờ xác định kết quả có hòa hay không: checkDrawFlag (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng dọc: countDoc (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng ngang: countNgang (kiểu int)

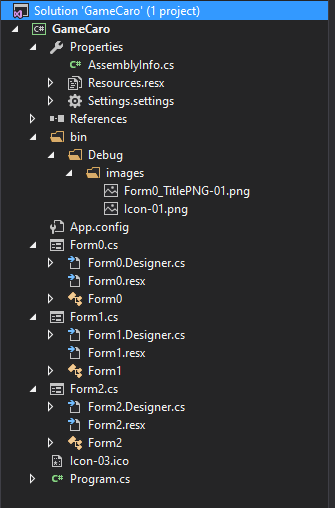
- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng chéo trái: countCheoTrai (kiểu int)

- Biến đếm xác định các giá trị phần tử trong mảng a bằng nhau theo hàng chéo phải: countCheoPhai (kiểu int)

- Form được khai báo trong sự kiện Button\_Click nhằm mục đích chuyển đổi qua lại từ Form2 sang Form0: f0 (kiểu Form)

1. **Phần thiết kế**
2. **Cấu trúc chương trình**

**Cách tổ chức cây thư mục trong chương trình**



*Tổ chức thư mục trong chương trình dạng Solution Explorer*

Chú giải:

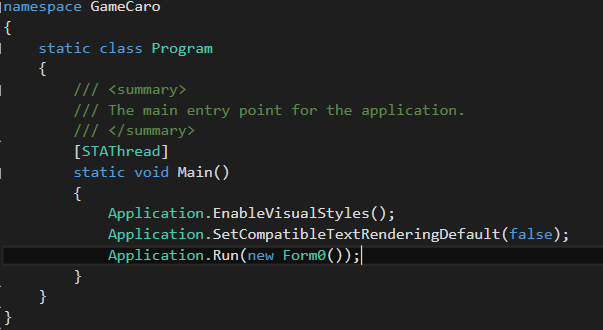
Thư mục images: Chứa các ảnh được sử dụng trong chương trình, được lưu dưới dạng .png và các file hình có đuôi .ico để thiết lập icon là hình ảnh tự thiết kế cho chương trình.



*Tổ chức thư mục image trong máy tính*

1. **Chi tiết chương trình**
2. **Chức năng của từng Form**

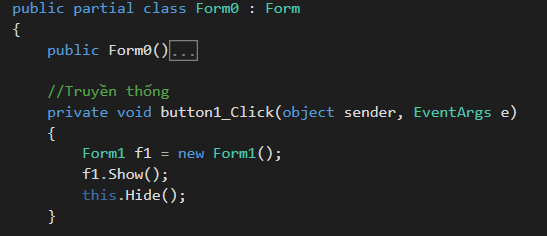
- Form0: Là Form được sử dụng làm màn hình giao diện Menu cho chương trình, được thiết lập hiển thị đầu tiên khi chương trình được thực thi. Thực hiện thiết lập này trong class Program trong Program.cs, thay đổi đối tượng mặc định trong *Application.Run(Đối tượng mặc định)* thành *new Form0()*



*Class Program trong Program.cs*

- Form1: Là Form được sử dụng làm màn hình giao diện chế độ chơi Truyền thống. Được gọi thực thi khi bấm vào nút “Truyền thống” (button1) trên màn hình Menu.

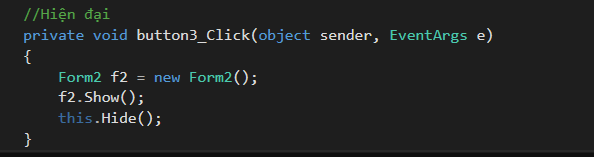
Ở Form0, trong sự kiện button1\_Click thực hiện *new Form1()* với tên biến f1. Sau đó, gọi phương thức *show()* của f1 cùng lúc đó ẩn đi Form0 bằng *this.Hide()*



*Sự kiện gọi Form1 hiển thị*

- Form2: Là Form được sử dụng làm màn hình giao diện chế độ chơi Hiện đại. Được gọi thực thi khi bấm vào nút “Hiện đại” trên màn hình Menu.

Ở Form0, trong sự kiện button3\_Click thực hiện *new Form2()* với tên biến f2. Sau đó, gọi phương thức *show()* của f2 cùng lúc đó ẩn đi Form0 bằng *this.Hide()*

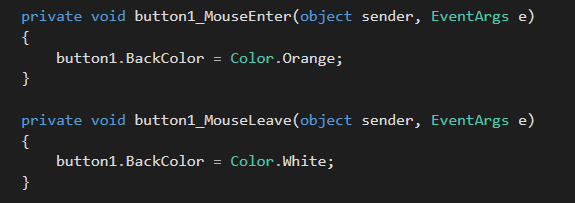


*Sự kiện gọi Form2 hiển thị*

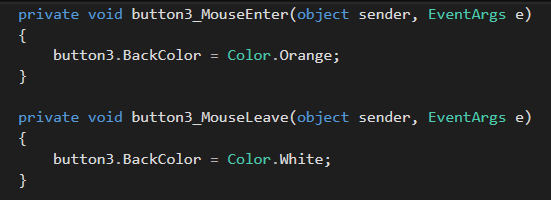
1. **Các hàm thành viên trong từng Form**

**Form0**

- Hàm xử lý sự kiện MouseEnter và MouseLeave của hai nút Truyền thống(button1), Hiện đại(button3) sẽ thực hiện đổi sang màu cam để nhận biết người chơi đang di chuyển chuột vào phạm vi Button và sẽ trả lại màu trắng khi người chơi di chuyển chuột ra khỏi phạm vi Button

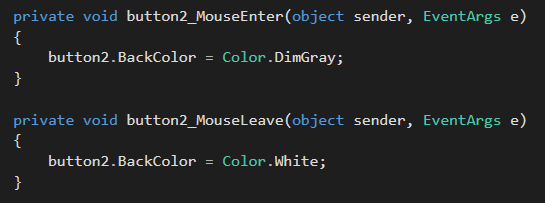


*Button1*



*Button3*

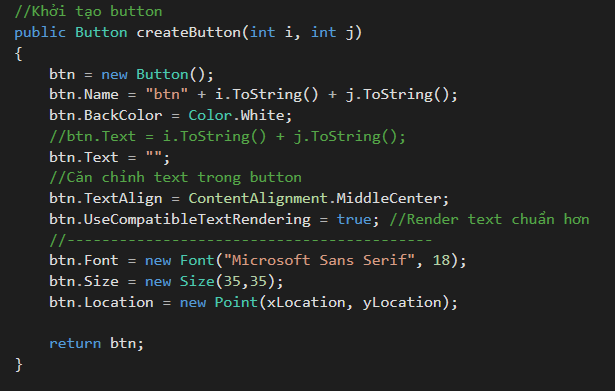
- Hàm xử lý sự kiện MouseEnter và MouseLeave của nút Thoát (button2) sẽ thực hiện đổi sang màu xám để nhận biết người chơi đang di chuyển chuột vào phạm vi Button và sẽ trả lại màu trắng khi người chơi di chuyển chuột ra khỏi phạm vi Button



*Button2*

**Form1**

- Hàm khởi tạo một Button lúc chương trình được thực thi, mục đích là để tạo ô cờ trên bàn cờ. Một bàn cờ gồm 437 ô cờ nên không thể tạo ô cờ theo cách Design thông thường mà phải thực hiện tạo bằng Code

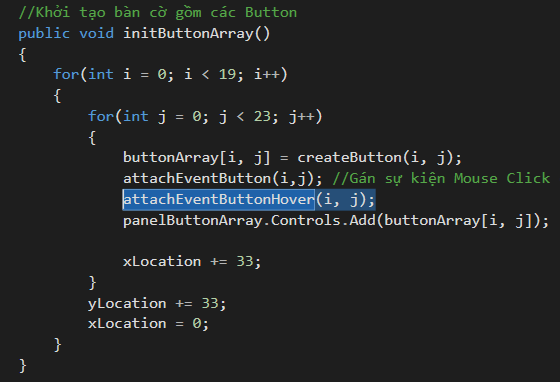


*Hàm khởi tạo Button lúc Runtime*

- Hàm khởi tạo bàn cờ gồm các Button sử dụng mảng hai chiều buttonArray đã được khai báo toàn cục. Thực hiện duyệt hai vòng lặp for, lần lượt gán buttonArray[i, j] bằng giá trị trả về là Button của hàm *createButton(i, j).*

Sau đó, gán các sự kiện *attachEventButton(i, j), attachEventButtonHover(i, j)* các phần tử trong mảng buttonArray

đồng thời gọi phương thức *Control.Add(buttonArray[i, j])* của panelButtonArray đã được Design sẵn



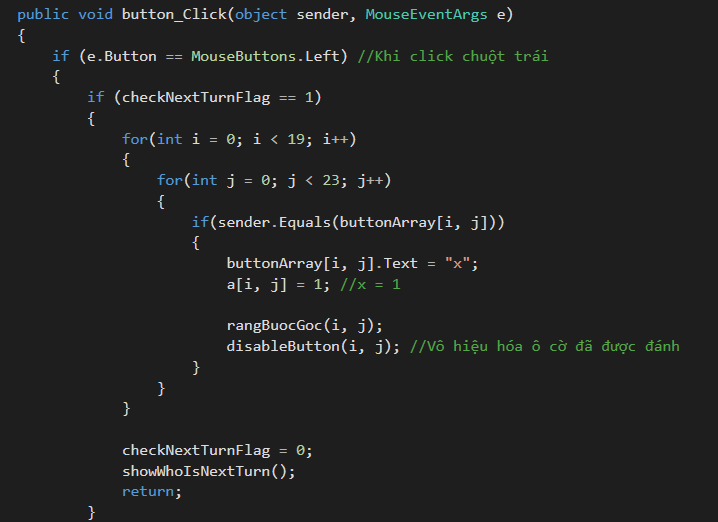
*Hàm khởi tạo bàn cờ lúc Runtime*

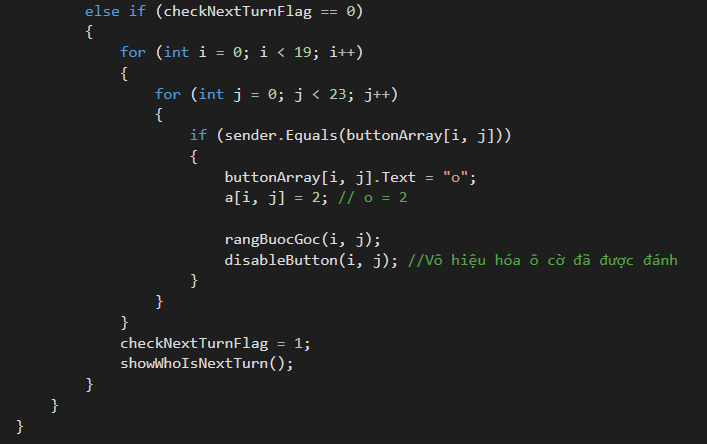
Hàm xử lý sự kiện khi click chuột trái lên vị trí đánh nước cờ. Khi người chơi click chuột trái lên ô cờ để đi nước cờ của mình, chương trình sẽ kiểm tra nếu checkNextTurnFlag = 1 tương ứng với lượt đi của nước cờ X thì thực hiện chạy hai vòng lặp for lồng nhau, với giới hạn theo chiều dài và chiều rộng bàn cờ để xác định ô cờ được đánh thông qua *sender.Equals(buttonArray[i, j]*. Sau đó chuyển butttonArray[i, j].Text từ rỗng ban đầu sang “x” đồng thời gán phần tử a[i, j] = 1 để tiện cho việc kiểm tra các trường hợp nước cờ thắng.

Tiếp theo, gọi hàm *rangBuocGoc(i, j)* để kiểm tra các trường hợp nước cờ thắng tại vị trí ô cờ đang được đánh. Gọi hàm *disableButton(i, j)* để vô hiệu hóa ô cờ đã được đánh nhằm tránh việc thay đổi giá trị quân cờ.

Cuối cùng, chuyển biến cờ checkNextTurnFlag sang 0 và gọi hàm *showWhoIsNextTurn()* để hiện thị lượt chơi tiếp theo là của nước cờ O trên thanh Lượt chơi.

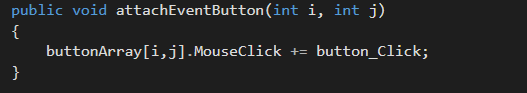
Thực hiện tương tự với khối lệnh trong else if{} ứng với nước cờ O.





*Sự kiện Button\_Click trên ô cờ*

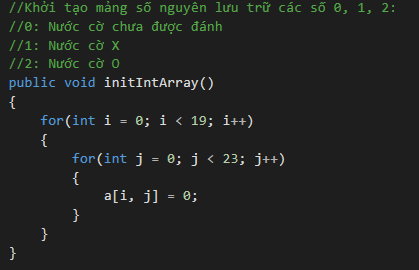
Thực hiện gán sự kiện button\_Click cho các ô button trên bàn cờ t thông qua hàm *attachEventButton(i, j)*



*Gán sự kiện cho các button*

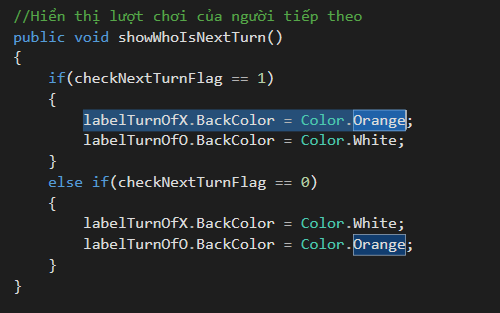
Hàm khởi tạo mảng hai chiều số nguyên ứng với giá trị của các ô cờ (button), thực hiện duyệt hai vòng lặp for lồng nhau và gán giá trị a[i, j] = 0 tức là chưa ô cờ nào được đi. Phần tử a[i, j] sẽ thay đổi thành 1 nếu ô cờ được đánh nước X và thành 2 nếu ô cờ được đánh nước 2

Mục đích của việc này là để thuận tiện cho việc kiểm tra các trường hợp thắng tại các nước cờ được đánh dễ dàng hơn



*Hàm khởi tạo mảng hai chiều các số nguyên bằng 0*

Hàm kiểm tra và hiện thị nước cờ của người chơi đánh lượt tiếp theo. Nếu biến cờ checkNextTurnFlag = 1 thì ô lượt chơi của người chơi nước cờ X sẽ chuyển thành màu cam, đồng thời ô lượt chơi của người chơi nước cờ O sẽ chuyển thành màu trắng. Ngược lại đối với trường hợp checkNextTurnFlag = 0.

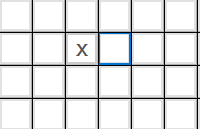


*Hàm kiêm tra và hiển thị lượt chơi của người chơi tiếp theo*

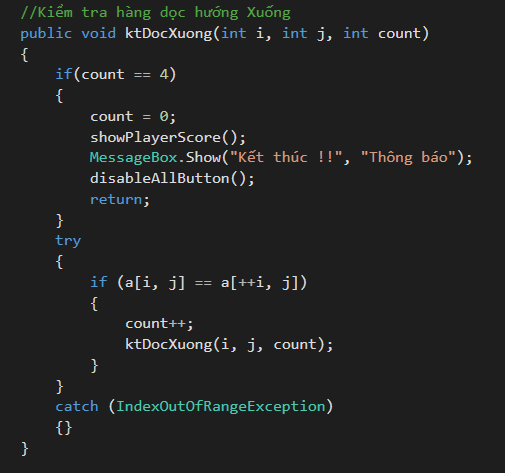
Hàm kiểm tra các nước cờ thắng theo hàng dọc hướng xuống kể từ vị trí ô cờ đã được đánh. Áp dụng thuật giải đệ quy để kiểm tra:

Trường hợp cơ sở là khi count = 4 (ban đầu count được khai báo toàn cục và được gán bằng 0) thì count được gán lại là 0, gọi hàm hiển thị điểm của hai người chơi *showPlayerScore(),* Hiển thị MessageBox thông báo kết thúc trò chơi và gọi hàm *disableAllButton()* để khóa toàn bộ bàn cờ.

Trường hợp tổng quát sẽ kiểm tra vị trí ô cờ hiện tại được đánh có giá trị trùng với ví trí ô cờ ngay bên dưới hay không. Nếu có thì thăng count lên 1 đơn vị và gọi đệ quy hàm *ktDocXuong(i, j, count).* Nếu không thì chuyển sang kiểm tra theo các hướng khác



*Kiểm tra hàng dọc hướng xuống*

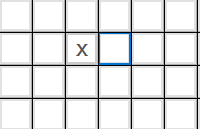


*Hàm kiểm tra hàng dọc hướng xuống*

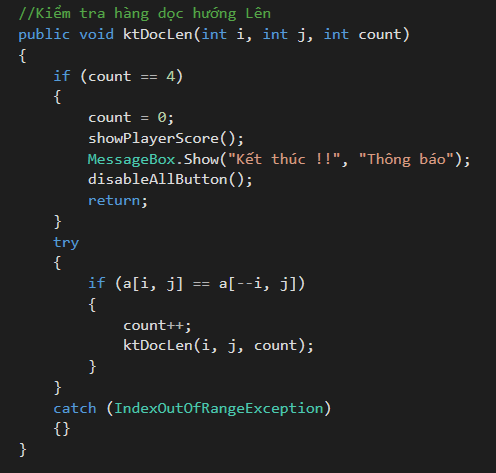
Hàm kiểm tra các nước cờ thắng theo hàng dọc hướng lên kể từ vị trí ô cờ đã được đánh. Áp dụng thuật giải đệ quy để kiểm tra:

Trường hợp cơ sở là khi count = 4 (ban đầu count được khai báo toàn cục và được gán bằng 0) thì count được gán lại là 0, gọi hàm hiển thị điểm của hai người chơi *showPlayerScore(),* Hiển thị MessageBox thông báo kết thúc trò chơi và gọi hàm *disableAllButton()* để khóa toàn bộ bàn cờ.

Trường hợp tổng quát sẽ kiểm tra vị trí ô cờ hiện tại được đánh có giá trị trùng với ví trí ô cờ ngay bên dưới hay không. Nếu có thì thăng count lên 1 đơn vị và gọi đệ quy hàm *ktDocXuong(i, j, count).* Nếu không thì chuyển sang kiểm tra theo các hướng khác



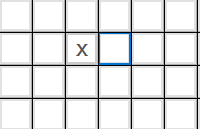
*Kiểm tra hàng dọc hướng lên*



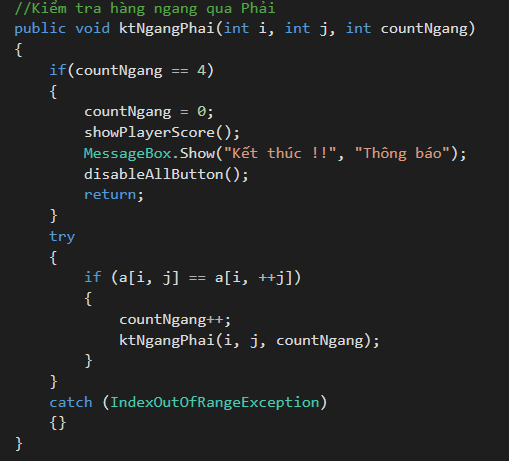
*Hàm kiểm tra hàng dọc hướng lên*

Hàm kiểm tra các nước cờ thắng theo hàng ngang hướng sang phải kể từ vị trí ô cờ đã được đánh. Áp dụng thuật giải đệ quy để kiểm tra:

Trường hợp cơ sở là khi countNgang = 4 (ban đầu countNgang được khai báo toàn cục và được gán bằng 0) cách thức hoạt động tương tự hai hàm kiểm tra dọc ở trên

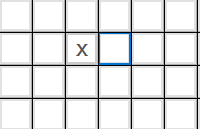


*Kiểm tra theo hàng ngang hướng sang phải*

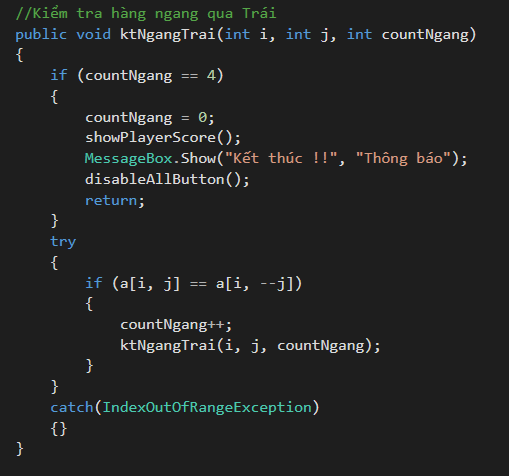


*Hàm kiểm tra hàng ngang hướng sang phải*

Hàm kiểm tra các nước cờ thắng theo hàng ngang hướng sang trái kể từ vị trí ô cờ đã được đánh

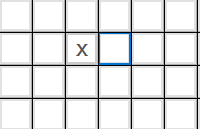


*Kiểm tra theo hàng ngang hướng sang trái*

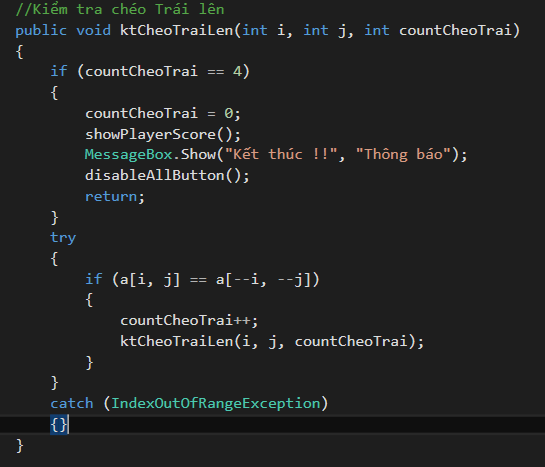


*Hàm kiểm tra hàng ngang hướng sang trái*

Hàm kiểm tra các nước cờ thắng theo hàng chéo hướng lên sang trái kể từ vị trí ô cờ đã được đánh

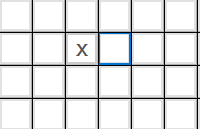


*Kiểm tra theo hàng chéo hướng lên sang trái*

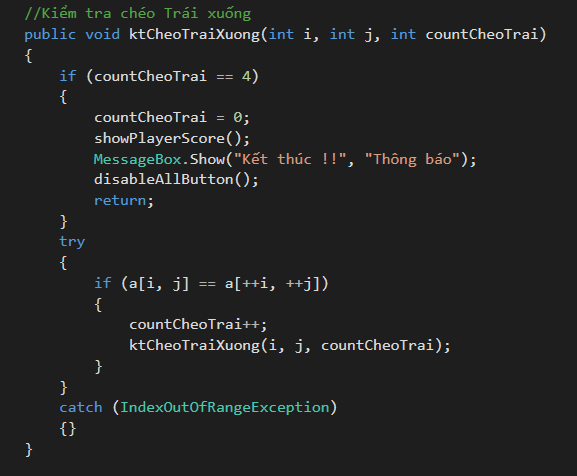


*Hàm kiểm tra hàng chéo hướng lên sang trái*

Hàm kiểm tra các nước cờ thắng theo hàng chéo hướng xuống sang phải kể từ vị trí ô cờ đã được đánh

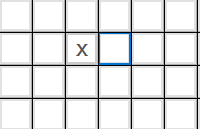


*Kiểm tra theo hàng chéo hướng xuống sang phải*

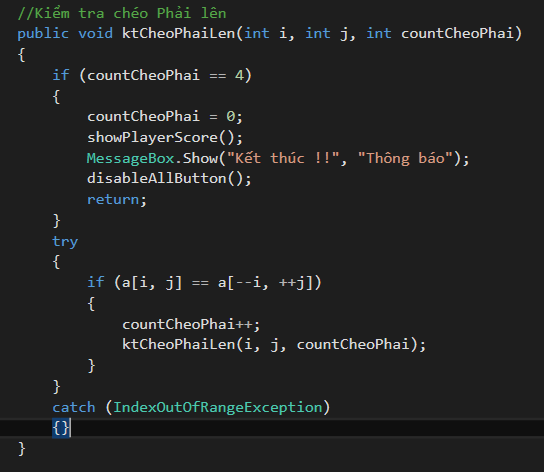


*Hàm kiểm tra hàng chéo hướng xuống sang phải*

Hàm kiểm tra các nước cờ thắng theo hàng chéo hướng lên sang phải kể từ vị trí ô cờ đã được đánh

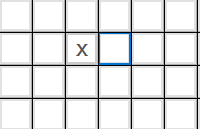


*Kiểm tra theo hàng chéo hướng lên sang phải*

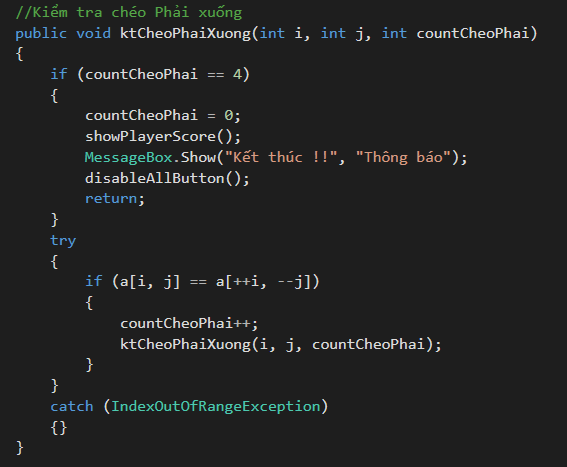


*Hàm kiểm tra hàng chéo hướng lên sang phải*

Hàm kiểm tra các nước cờ thắng theo hàng chéo hướng xuống sang trái kể từ vị trí ô cờ đã được đánh



*Kiểm tra theo hàng chéo hướng xuống sang trái*



*Hàm kiểm tra hàng chéo hướng xuống sang trái*

Hàm ràng buộc góc và các cạnh khi kiểm tra nước cờ thắng. Hàm thực hiện ràng buộc các vị trí đặc biệt trên bàn cờ

1

2

2

3

1

4

4

3

*Các vị trí đặc biệt trên bàn cờ*

Khi thực hiện kiểm tra các trường hợp thắng theo các hướng như lên, xuống, chéo trái, chéo phải thì phải có trường hợp ngoại lệ cho các vị trí đặc biệt trên.

Ví dụ:

- Tại vị trí ô 1, chỉ được thực hiện các hàm:

*ktDocXuong(i, j, count);*

*ktNgangPhai(i, j, countNgang);*

*ktCheoTraiXuong(i, j, countCheoTrai);*

Nếu thực hiện kiểm tra hàng dọc hướng lên, kiểm tra hàng ngang sang trái và các trường hợp kiểm tra hướng còn lại sẽ vi phạm giới hạn phạm vi của mảng buttonArray cho trước, xuất hiện lỗi tràn vị trí phần tử trong mảng (IndexOutOfRangeException)

- Tại cạnh 1 4 chỉ được thực hiện các hàm:

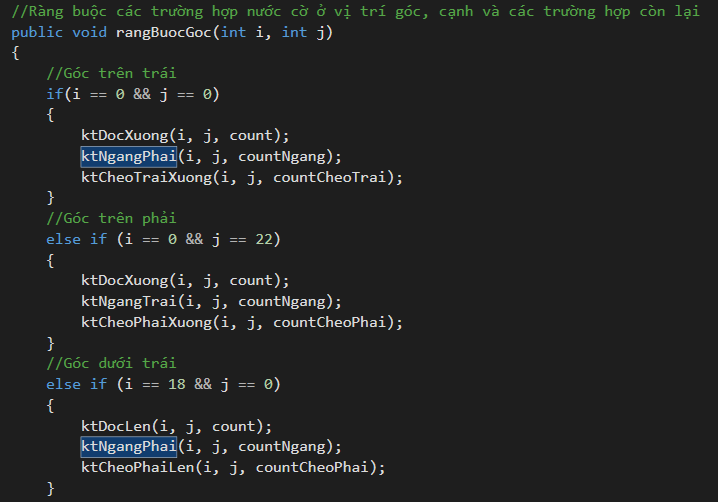
*ktDocXuong(i, j, count);*

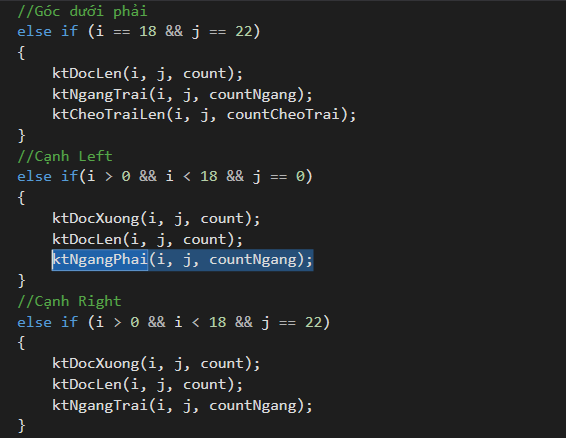
*ktDocLen(i, j, count);*

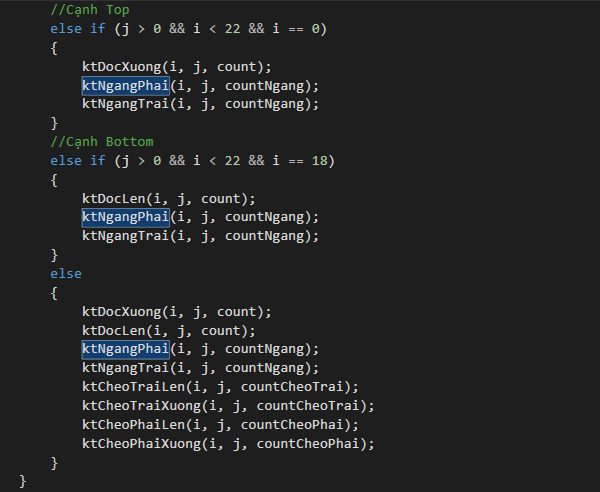
*ktNgangPhai(i, j, countNgang);*

Lý do chỉ thực hiện kiểm tra các hàm này giống như trường hợp tại ô số 1. Nếu kiểm tra theo các hướng còn lại thì sẽ vi phạm phạm vi của mảng buttonArray và xuất hiện lỗi tràn vị trí phần tử trong mảng

Tương tự sẽ có các ràng buộc khi kiểm tra đối với các vị trí đặc biệt còn lại là: ô 2, ô 3, ô 4, cạnh 1 2, cạnh 2 3, cạnh 3 4. Ràng buộc kiểm tra sẽ không xuất hiện đối với các ô cờ không thuộc các vị trí đặc biệt trên



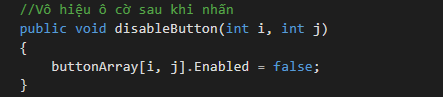




*Hàm ràng buộc các vị trí kiểm tra hướng nước cờ thắng*

Hàm vô hiệu hóa ô cờ sau khi nhấn bằng cách lấy vị trí index i, j tại thời điểm ô cờ được click và thực hiện lệnh:

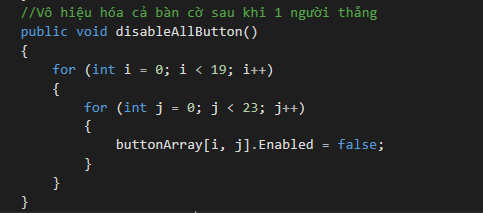
*buttonArray[i, j].Enabled = false;*



*Hàm vô hiệu hóa ô cờ đã được đánh*

Hàm vô hiệu hóa cả bàn cờ khi một trong hai người chơi chiến thắng. Hàm thực hiện duyệt hai vòng lặp for lồng nhau theo chiều chiều dài và chiều rộng của ô cờ, cùng lúc tại ví trị index i và j thực hiện lệnh:

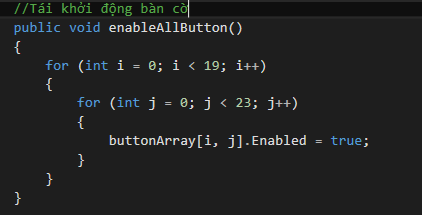
*buttonArray[i, j].Enabled = false;*



*Hàm vô hiệu hóa cả bàn cờ khi một trong hai người chơi thắng*

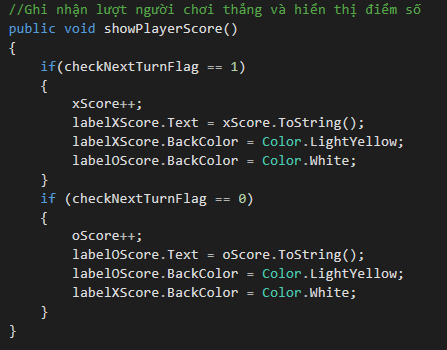
Hàm tái khởi động bàn cờ khi người chơi nhấn nút Chơi lại hoặc Tiếp tục chơi ván mới. Hàm thực hiện duyệt hai vòng lặp for lồng nhau theo chiều chiều dài và chiều rộng của ô cờ, cùng lúc tại ví trị index i và j thực hiện lệnh:

*buttonArray[i, j].Enabled = true;*



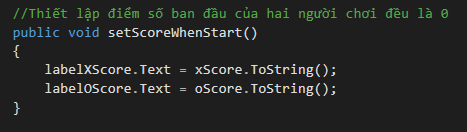
*Hàm tái khởi động bàn cờ*

Hàm tăng và hiển thị điểm số của người chơi thắng ở lượt chơi hiện tại, nếu checkNextTurnFlag hiện tại là 1 tức là người chơi nước cờ X vừa đánh xong và thỏa mãn điều kiện thắng thì thực hiện tăng xScore lên 1 đơn vị, hiển thị điểm số lên labelXScore. Ngược lại với trường hợp giá trị checkNextTurnFlag là 0 cho người chơi nước cờ O



Hàm hiển thị tăng và hiển thị điểm số người chơi thắng ở lượt hiện tại

Hàm thiết lập điểm số ban đầu cho hai người chơi, hàm thực hiện thiết lập điểm số của hai người chơi về 0 lúc Form được load lần đầu



*Hàm thiết lập điểm số ban đầu cho hai người chơi*

Hàm xừ lý sự kiện khi click vào nút Chơi lại đã được Design trên Form. Hàm thực hiện duyệt hai vòng lặp for lồng nhau theo chiều dài và chiều rộng của bàn cờ, thực hiện gán toàn bộ thuộc tính Text của các button (ô cờ) bằng rỗng và gán giá trị 0 cho các phần tử trong mảng số nguyên a:

buttonArray[i, j].Text = “”;

a[i, j] = 0;

Sau khi thoát khỏi hai vòng lặp, hàm thực hiện thiết lập lại điểm số của hai người chơi về 0:

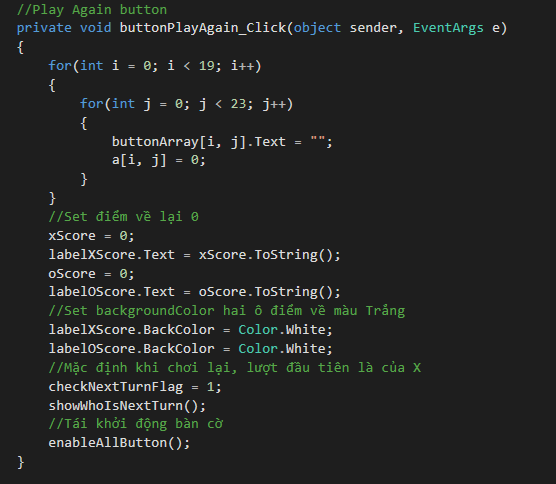
xScore = 0;

oScore = 0;

Tiếp theo, thiết lập lại biến cờ checkNextTurnFlag về là 1 để khi bắt đầu ván mới thì nước cờ đi đầu tiên luôn là nước cờ X và gọi hàm hiển thị lượt chơi *showWhoIsNextTurn()*:

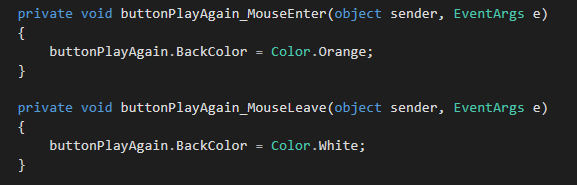
checkNextTurnFlag = 1;

Cuối cùng, gọi hàm *enableAllButton()* để tái khởi động bàn cờ sau khi đã vô hiệu hóa bàn cờ bởi lượt thắng của người chơi trước



*Hàm xử lý sự kiện khi click vào button Chơi lại*

Hàm xử lý sự kiện MouseEnter và MouseLeave của button Chơi lại. Khi chuột di chuyển đến phạm vi button Chơi lại, thuộc tính BackColor sẽ chuyển thành màu cam và chuyển thành màu trắng khi con trỏ chuột di chuyển ra khỏi phạm vi button



*Hàm xử lý sự kiện MouseEnter và MouseLeave của button Chơi lại*

Hàm xừ lý sự kiện khi click vào button Tiếp tục đã được Design trên Form. Hàm thực hiện duyệt hai vòng lặp for lồng nhau theo chiều dài và chiều rộng của bàn cờ thực hiện gán toàn bộ thuộc tính Text của các button (ô cờ) bằng rỗng và gán giá trị 0 cho các phần tử trong mảng số nguyên a:

buttonArray[i, j].Text = “”;

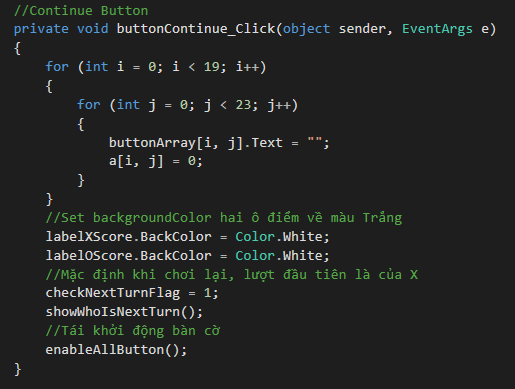
a[i, j] = 0;

Khác với sự kiện khi click vào button Chơi lại, điểm số của hai người chơi vẫn được giữ nguyên thay vì được thiết lập lại là 0

Tiếp theo, thiết lập lại biến cờ checkNextTurnFlag về là 1 để khi bắt đầu ván mới thì nước cờ đi đầu tiên luôn là nước cờ X và gọi hàm hiển thị lượt chơi *showWhoIsNextTurn()*:

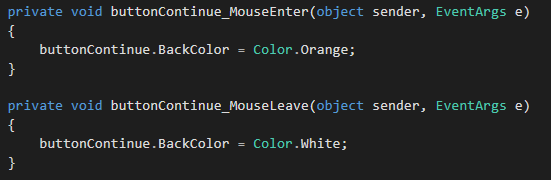
checkNextTurnFlag = 1;

Cuối cùng, gọi hàm *enableAllButton()* để tái khởi động bàn cờ sau khi đã vô hiệu hóa bàn cờ bởi lượt thắng của người chơi trước



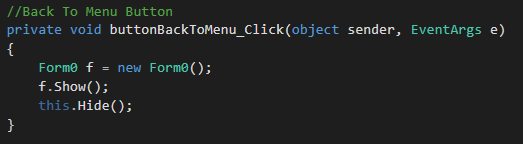
*Hàm xử lý sự kiện khi click vào button Tiếp tục*

Hàm xử lý sự kiện MouseEnter và MouseLeave của button Tiếp tục. Khi chuột di chuyển đến phạm vi button Tiếp tục, thuộc tính BackColor sẽ chuyển thành màu cam và chuyển thành màu trắng khi con trỏ chuột di chuyển ra khỏi phạm vi button



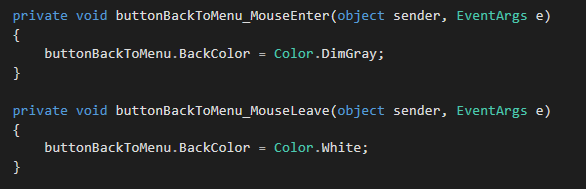
*Hàm xử lý sự kiện MouseEnter và MouseLeave của button Tiếp tục*

Hàm xừ lý sự kiện khi click vào button Trở về đã được Design trên Form. Hàm thực hiện gọi Form0 (Màn hình giao diện Menu) và ẩn đi Form1 hiện tại



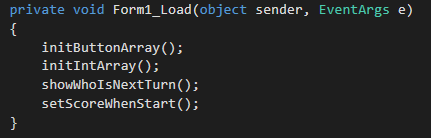
*Hàm xử lý sự kiện khi click vào button Trở về*

Hàm xử lý sự kiện MouseEnter và MouseLeave của button Trở về. Khi chuột di chuyển đến phạm vi button Trở về, thuộc tính BackColor sẽ chuyển thành màu xám và chuyển thành màu trắng khi con trỏ chuột di chuyển ra khỏi phạm vi button



*Hàm xử lý sự kiện MouseEnter và MouseLeave của button Trở về*

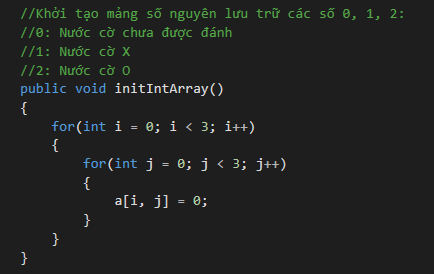
Hàm xử lý sự kiện Form được load lên lần đầu, thực hiện gọi các hàm: khởi tạo bàn cờ *initButtonArray(),* khởi tạo mảng số nguyên ứng với giá trị tại các ô cờ *initIntArray(),* hiển thị lượt chơi của người chơi (mặc định X chơi đầu tiên) *showWhoIsNextTurn(),* thiết lập điểm số ban đầu cho hai người chơi *setScoreWhenStart().*



*Hàm xử lý sự kiện Form load*

**Form2**

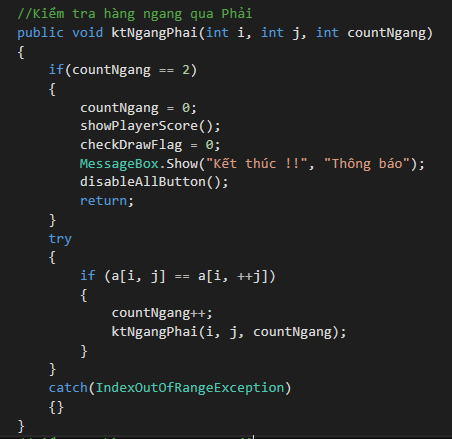
Các hàm thành viên trong Form2 có tên và chức năng tương tự các hàm thành viên trong Form1. Ở Form2, điểm khác biệt duy nhất là trong các hàm có duyệt hai vòng lặp for lồng nhau thì độ dài của chiều rộng và chiều cao của bàn cờ đều là 3 vì bàn cờ ở chế độ chơi Hiện đại có kích thước 3x3



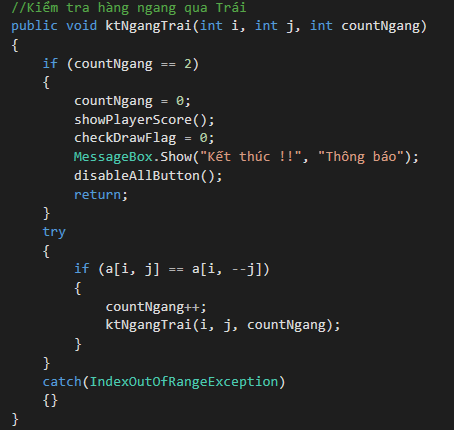
*Hai vòng lặp for lồng nhau trong hàm initIntArray(), chỉ số index i, j chỉ duyệt từ 0 đến 3 thay vì duyệt đến 19 và 23 như ở Form1*

Trong Form2, xuất hiện thêm một hàm mới có chức năng kiểm tra xem nước cờ hai bên có hòa nhau hay không. Vì số lượng ô cờ chỉ là 9 nên sẽ có trường hợp hai người đã đi hết các ô cờ nhưng vẫn chưa tìm ra người chiến thắng.

Biến cờ checkDrawFlag được khai báo toàn cục và được gán mặc định là 1 tức là hòa. Vị trí đặt biến cờ này là ở các hàm kiểm tra trường hợp nước cờ thắng theo các hướng dọc, ngang, chéo trái, chéo phải. Nếu trường hợp nước cờ thỏa mãn các điều kiện ràng buộc kiểm tra theo các hướng trên (tức là một trong hai bên đã thắng) thì biến cờ checkDrawFlag được gán bằng 0 (tức là hai bên không hòa)

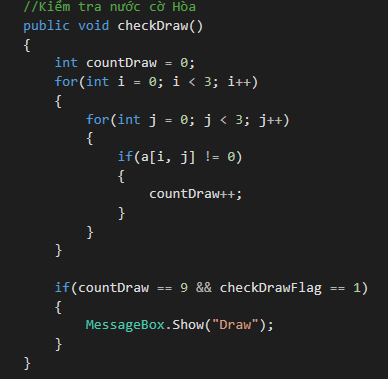


*Đặt biến cờ trong hàm ktNgangPhai(i, j, countNgang)*



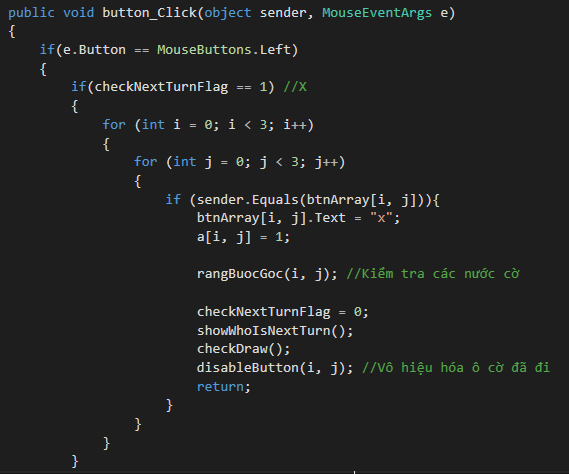
*Đặt biến cờ trong hàm ktNgangTrai(i, j, countNgang). Và tương tự trong các hàm kiểm tra nước cờ thắng theo các hướng khác*

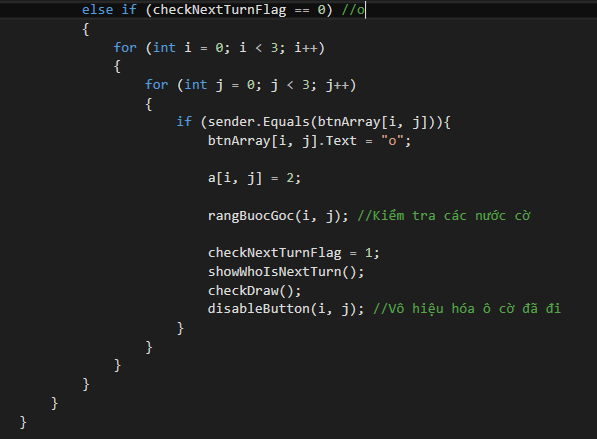
Hàm sẽ duyệt các ô cờ, kiểm tra nếu tất cả các ô cờ đều khác rỗng (tức là a[i, j] khác 0) thì tăng countDraw lên 1 đơn vị. Đến khi countDraw = 9 và biến cờ checkDrawFlag = 1 thì chương trình hiển thị thông báo “Draw”



*Hàm kiểm tra nước cờ có hòa hay không*

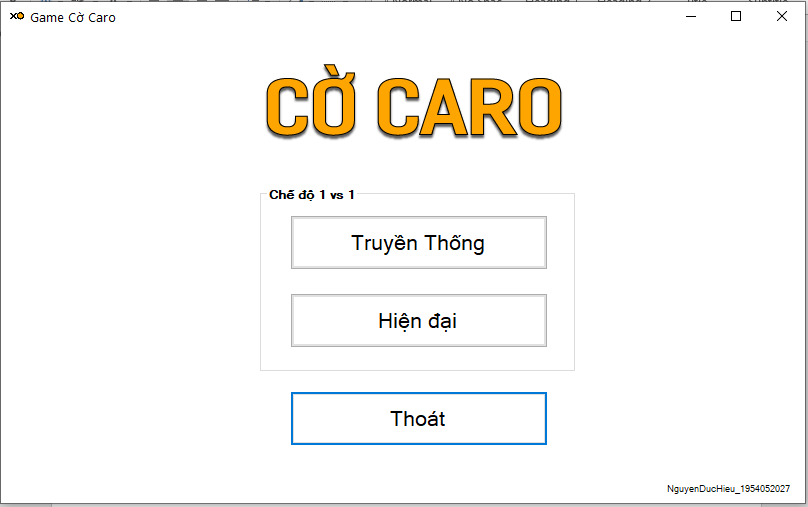
Vị trí đặt hàm *checkDrawFlag()* là bên trong hàm xử lý sự kiện khi người chơi đánh một nước cờ *button\_Click(object sender, MouseEventArgs e)*





*Hàm xử lý sự kiện khi người chơi đánh nước cờ vào một ô cờ bất kỳ*

1. **Phần kết quả**
2. **Kết quả chạy chương trình**



*Màn hình giao diện Menu trò chơi*

*Kết quả khi người chơi chọn button Truyền thống, màn hình giao diện chế độ chơi Truyền thống được hiển thị*

*Kết quả chương trinh khi một trong hai người chơi chiến thắng*

*Kết quả khi người chơi chọn button Hiện đại, màn hình giao diện chế độ chơi Hiện đại được hiển thị*

*Kết quả khi một trong hai người chơi chiến thắng*

*Kết quả khi hai người chơi hòa nhau*

*Kết quả khi người chơi chọn button Thoát*

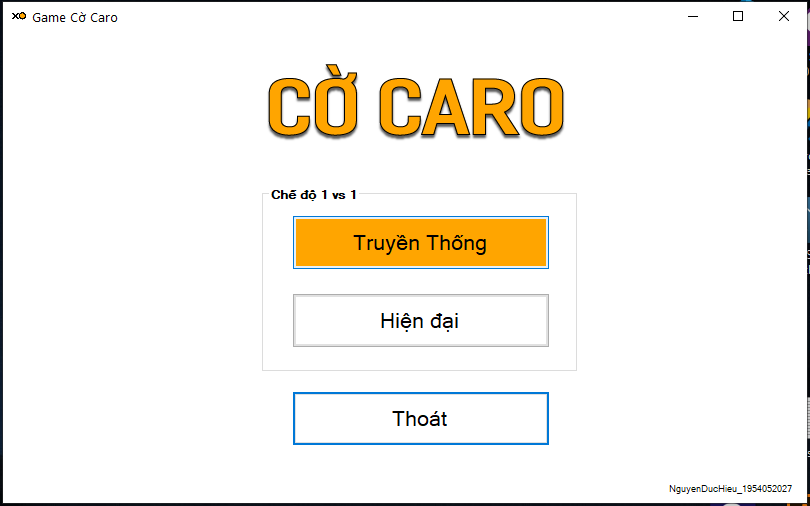
1. **Hướng dẫn sử dụng chương trình**

Để bắt đầu trò chơi, người chơi click đúp chuột vào icon Game cờ caro trên màn hình

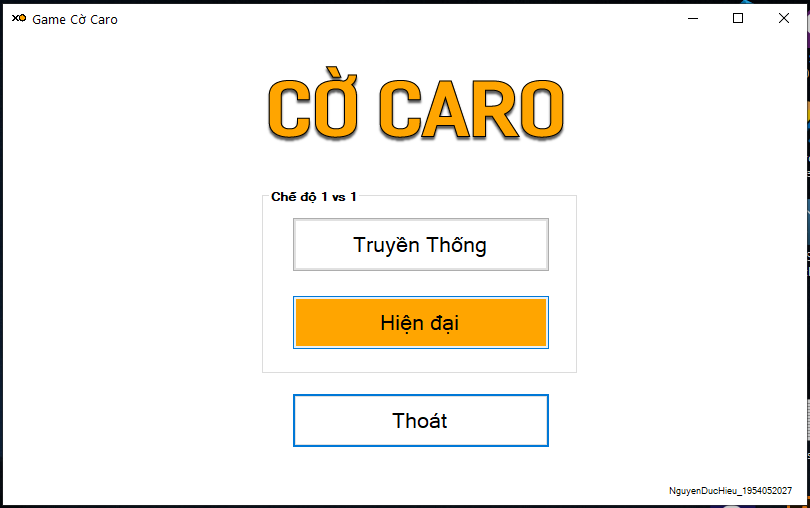


*Icon Game cờ caro trên màn hình Desktop*

Giao diện Menu trò chơi sẽ hiển lên, người chơi chọn một trong hai chế độ chơi là Truyền thống hoặc Hiện đại



*Giao diện khi chọn chế độ chơi Truyền thống*

**

*Giao diện khi chọn chế độ chơi Hiện đại*

Nếu người chọn chế độ Truyền thống, màn hình giao diện chế độ chơi Truyền thống sẽ hiện lên đồng thời ẩn đi màn hình Menu

*Giao diện màn hình chế độ chơi Truyền thống*

Hai người chơi sẽ thảo luận với nhau để chọn nước cờ mình thích, người chơi chọn nước X sẽ đi trước theo sau là lượt chơi của người chọn nước cờ O

Khi một trong hai người chơi dành chiến thắng ở lượt chơi hiện tại, chương trình sẽ hiển thị thông báo “Kết thúc”, người chơi nhấn OK để tiếp tục

*Một trong hai người chơi dành chiến thắng ở lượt chơi hiện tại*

*Sau khi người chơi nhấn OK*

Lúc này, người chơi có ba lựa chọn là chơi tiếp, chơi lại từ đầu và trở về . Nếu chọn Tiêp tục, bàn cờ sẽ được tạo mới đồng thời điểm số vẫn được giữ nguyên. Nếu chọn Chơi lại, bàn cờ cũng sẽ được tạo mới nhưng điểm số sẽ được thiết lập về 0 cho cả hai người chơi. Trong trường hợp người chơi chọn Trở về, màn hình Menu ban đầu sẽ hiện lên đồng thời màn hình giao diện chế độ chơi Truyền thống sẽ được tắt đi và toàn bộ điểm số, các lượt chơi hiện tại sẽ bị xóa đi.

*Kết quả khi người chơi chọn Tiếp tục*

*Kết quả khi người chơi chọn Chơi lại*

*Kết quả khi người chơi chọn Trở về*

Ngoài chế độ chơi Truyền thống, người chơi có thể chọn chế độ chơi Hiện đại bằng cách nhấn vào button Hiện đại trên màn hình Menu. Giao diện màn hình chế độ Hiện đại sẽ hiện lên, các bước sử dụng tiếp theo tương tự với chế độ chơi Truyền thống

*Màn hình giao diện chế độ chơi Hiện đại*

Để thoát khỏi chương trình, người chơi chọn button Thoát ở màn hình Menu. Sau khi chọn, bảng thông báo sẽ hiện ra với hai lựa chọn “Yes” và “No”. Nếu người chơi chọn “Yes” chương trình sẽ được đóng, nếu chọn “No” người dùng sẽ quay lại giao diện Menu hiện tại.

*Bảng thông báo hiện lên khi chọn Thoát cho người dùng lựa chọn*

1. **Phần kết luận**
2. **Kết quả đạt được**

- Áp dụng được kiến thức đã học về lập trình giao diện bằng ngôn ngữ C#, sử dụng được các Control trong Winform để lập trình Game

- Hiểu hơn về cách người dùng thao tác với một chương trình ứng dụng. Từ đó, xử lý tốt hơn các sự kiện phát sinh khi người dùng tương tác với chương trình.

1. **Hạn chế của chương trình**

**-** Kích thước bàn cờ ở chế độ chơi Truyền thống bị giới hạn, chưa thể tự động mở rộng khi người chơi lấp đầy hết các ô cờ

**-** Xuất hiện ngẫu nhiên viền xanh trên ô cờ (button) bất kỳ sau khi đánh một nước cờ

**-** Hai nước cờ X và O khi đánh chưa hiển thị được màu sắc riêng biệt

1. **Hướng phát triển tiếp theo**

- Giảm thiểu các hạn chế gặp phải trong chương trình hiện tại

- Cải thiện thêm về giao diện chương trình

- Phát triển thêm chế độ chơi với máy

- Phát triển thêm chế độ thi đấu Online hoặc thi đấu qua mạng LAN giữa hai người chơi trong mạng nội bộ