**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**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**GAME CARO HAI NGƯỜI CHƠI**

**(STACK)**

**SVTH1 : NGUYỄN THÀNH TÂM - 17110219**

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

**GVPT: ThS. TRẦN CÔNG TÚ**

**TP. Hồ Chí Minh, tháng 11 năm 2018**

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc531331572)

[Danh mục các hình 2](#_Toc531331573)

[Danh mục các bảng 4](#_Toc531331574)

[CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN 5](#_Toc531331575)

[1.1 Lịch sử hình thành 5](#_Toc531331576)

[1.2 Giới thiệu game caro hai người chơi 5](#_Toc531331577)

[1.3 Lý do, mục đích, mục tiêu chọn đề tài 5](#_Toc531331578)

[1.3.1 Lý do, mục đích chọn đề tài 5](#_Toc531331579)

[1.3.2 Mục tiêu chọn đề tài 6](#_Toc531331580)

[1.4 Lựa chọn ngôn ngữ cài đặt mô phỏng 6](#_Toc531331581)

[CHƯƠNG 2 : NỘI DUNG 7](#_Toc531331582)

[2.1 Quá trình và công việc thực hiện đồ án 7](#_Toc531331583)

[2.2 Luật chơi 7](#_Toc531331584)

[2.3 Khái quát thuật toán 7](#_Toc531331585)

[2.3.1 Giới thiệu ngăn xếp (stack) 7](#_Toc531331586)

[2.3.2 Ứng dụng 8](#_Toc531331587)

[2.4 Thiết kế giao diện 8](#_Toc531331588)

[2.5 Cách cài đặt và viết chương trình 9](#_Toc531331589)

[2.5.1 Một số hàm xử lý chức năng 9](#_Toc531331590)

[2.5.2 Ứng dụng của STACK 15](#_Toc531331591)

[2.5.3 Ứng dụng của QUEUE 17](#_Toc531331592)

[2.6 Sản phẩm của đồ án 19](#_Toc531331593)

[CHƯƠNG 3 : KẾT LUẬN 20](#_Toc531331594)

[3.1 Kết luận 20](#_Toc531331595)

[3.2 Ưu điểm 20](#_Toc531331596)

[3.3 Nhược điểm 20](#_Toc531331597)

[3.4 Đề xuất phương án cải thiện 20](#_Toc531331598)

[PHỤ LỤC 22](#_Toc531331599)

[TÀI LIỆU THAM KHẢO 25](#_Toc531331600)

LỜI MỞ ĐẦU

Nhóm 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.

Danh mục các hình

[Figure 1: Phân công công việc 7](#_Toc531331903)

[Figure 2: Giao diện chính 8](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331904)

[Figure 3 : Generic 9](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331905)

[Figure 4 : Push() 10](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331906)

[Figure 5 : Pop() 10](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331907)

[Figure 6 : Top() 10](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331908)

[Figure 7 : Clear() 10](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331909)

[Figure 8 : Tạo bàn cờ 11](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331910)

[Figure 9.1 : ButtonCu 12](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331911)

[Figure9.2 : ButtonCu (tt) 12](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331912)

[Figure 10 : Tạo even 13](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331913)

[Figure 11.1 : Mark() 13](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331914)

[Figure 11.2 : Mark() (tt) 13](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331915)

[Figure 12.1 : isEndGame() 14](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331916)

[Figure 12.2 : isEndGame() (tt) 14](#_Toc531331917)

[Figure 13.1 : Kiểm tra 15](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331918)

[Figure 13.2 : Kiểm tra (tt) 15](#_Toc531331919)

[Figure 14 : Layvitri() 16](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331920)

[Figure 15 : Undo() 16](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331921)

[Figure 16 : Xoabanco() 17](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331922)

[Figure 17 : LuuVanCo() 17](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331923)

[Figure 19.1 : btntieptuc\_click() 18](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331924)

[Figure 18 : MoPhong() 18](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331925)

[Figure 19.2 :btntieptuc() (tt) 19](#_Toc531331926)

[Figure 19: Game Caro 19](file:///C:\Users\kieud\Documents\GitHub\CARO-2-NGUOI\BAOCAODOANHK11819.docx#_Toc531331927)

Danh mục các bảng

[Bảng 1: Bảng phân công 22](#_Toc531331998)

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 truệ 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

Trong khi học môn cấu trúc dữ liệu, 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 hàng đợi (queue) để demo game caro hai người chơi. 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, 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 caro hai người chơi để thực hiện đồ án môn học cấu trúc và dữ liệu.

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, mô phỏng lại ván cờ… 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 2017 để cài dặt mô phỏng này.

CHƯƠNG 2 : NỘI DUNG

2.1 Quá trình và công việc thực hiện đồ án

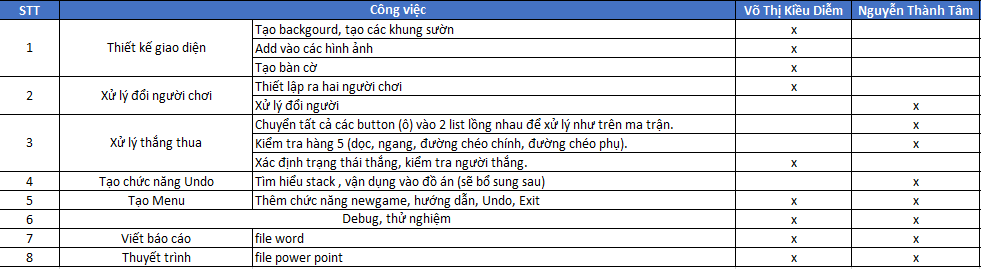


Figure 1: Phân công công việc

2.2 Luật chơi

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 : ai 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 : ai có được 5 nước dọc, ngang hoặc chéo liên tiếp mà không bị đối phương chặn hai đầu thì thắng!

2.3 Khái quát thuật toán

2.3.1 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)).

Một 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]

2.3.2 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.

2.4 Thiết kế giao diện

Figure 2: 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 chặn hai đầu và mô phỏng lại ván cờ. 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.

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 game.

Ngoài ra còn có phần mô phỏng lại ván cờ cho phép người chơi xem lại từng nước cờ mình đã đi.

2.5 Cách cài đặt và viết chương trình

2.5.1 Một số hàm xử lý chức năng

Đồ án của nhóm em gồm 5 Form và 5 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.

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

Class Stack:

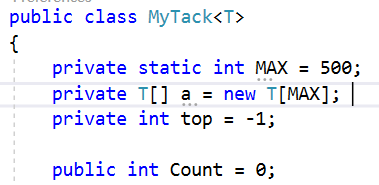
Vận dụng kiến thức đã học từ môn cấu trúc dữ liệu và giải thuật và môn lập trình hướng đối tượng, nhóm đã tạo stack từ Generic.

Figure 3 : 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.

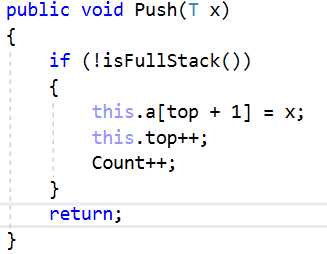
Hàm Push():

Figure 4 : Push()

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

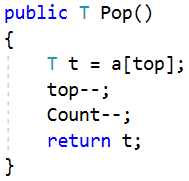
Hàm Pop():

Figure 5 : Pop()

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

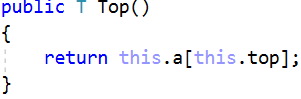
Hàm Top():

Figure 6 : Top()

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

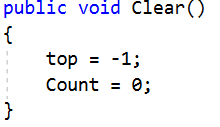
Hàm Clear():

Figure 7 : 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 propeties.

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 piture mark hiện ảnh khi mới vào game và cho biết ai đi trước.

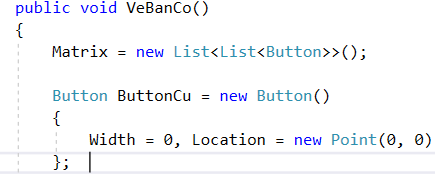
Tạo bàn cờ: đây là bước quan trọng và tương đối khó.

Figure 8 : Tạo bàn cờ

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

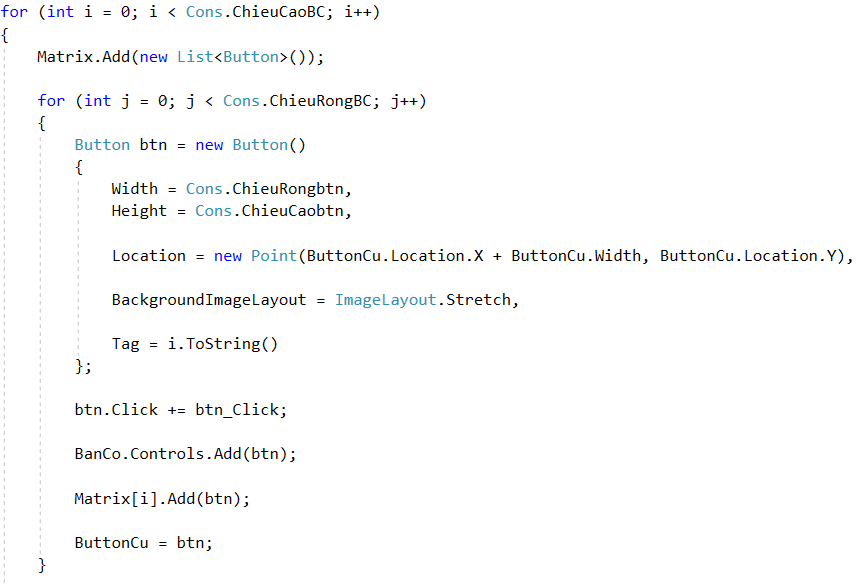
Khởi tạo ButtonCu làm mốc để tạo ra các button (ô đánh) trên panel bàn cờ.

Figure 9.1 : ButtonCu

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.

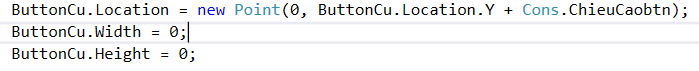
Tiếp tục làm như vậy cho đến khi hết bài cờ.

Figure9.2 : ButtonCu (tt)

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.

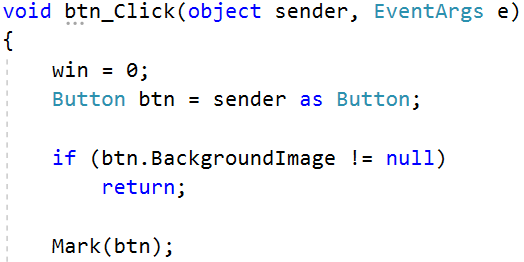
Tạo event cho mỗi button trên bàn cờ:

Figure 10 : Tạo even

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.

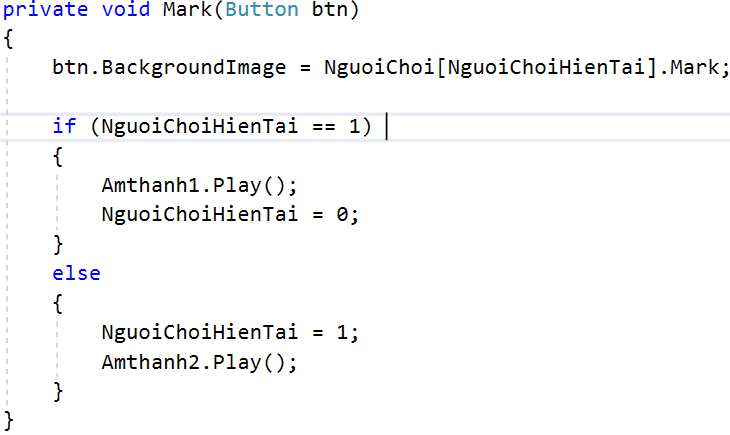
Nếu chưa, thì đánh vào ô đó bằng hàm Mark():

Figure 11.1 : 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.

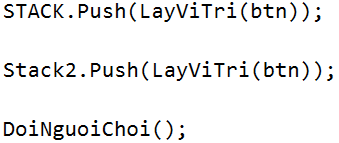


Figure 11.2 : Mark() (tt)

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

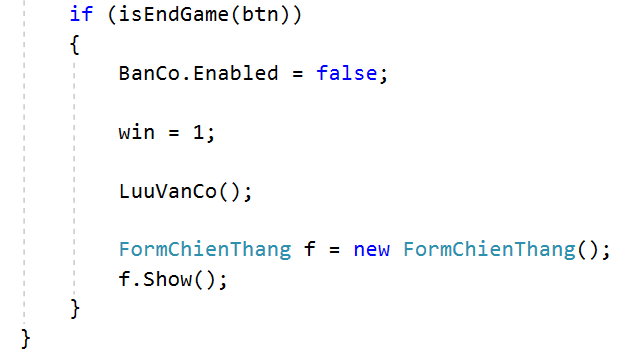


Figure 12.1 : isEndGame()

Kiểm tra nếu kết thúc game:

+ Disable bàn cờ.

+ Trạng thái thắng = 1;

+ Lưu lại ván cờ vào Queue để có thể mô phỏng lại quá trình chơi.

+ Hiện Form chiến thắng lên.

Hàm isEndGame()

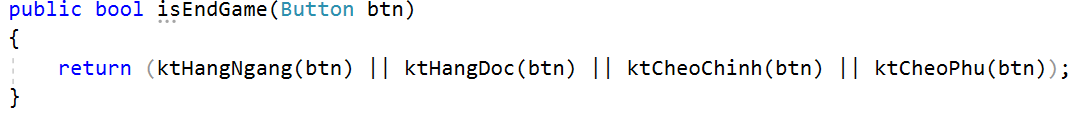


Figure 12.2 : isEndGame() (tt)

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.

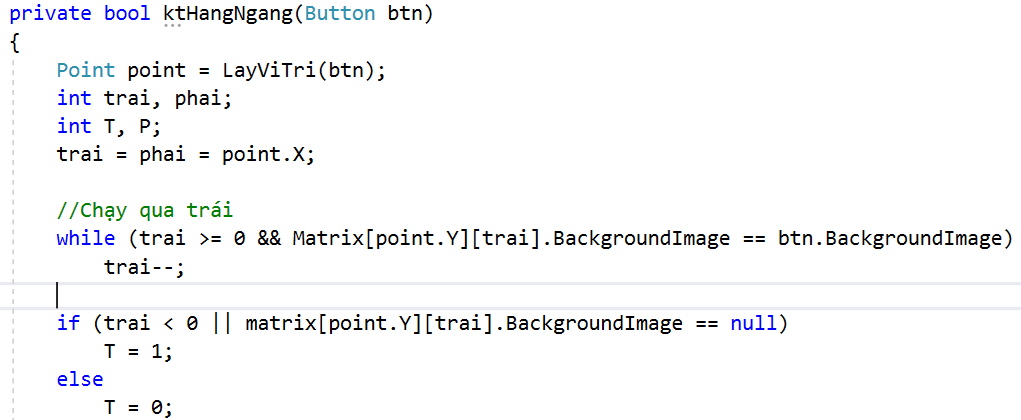
 Kiểm tra hàng ngang:

Figure 13.1 : Kiểm tra

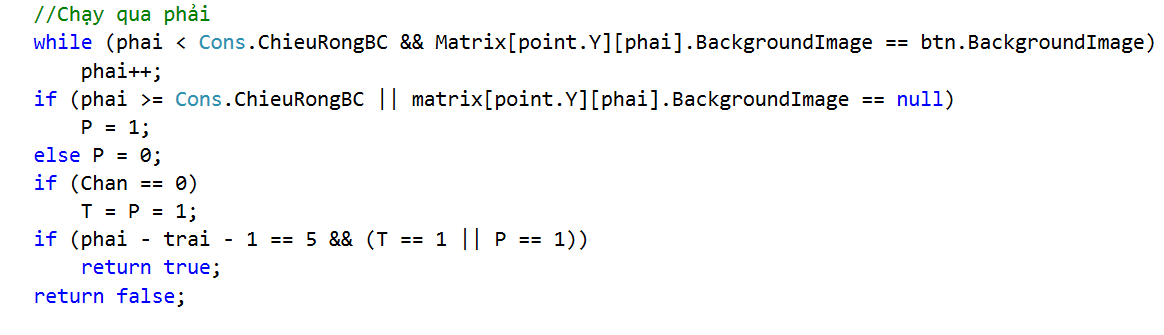


Figure 13.2 : Kiểm tra (tt)

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.

2.5.2 Ứng dụng của STACK

A. 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:

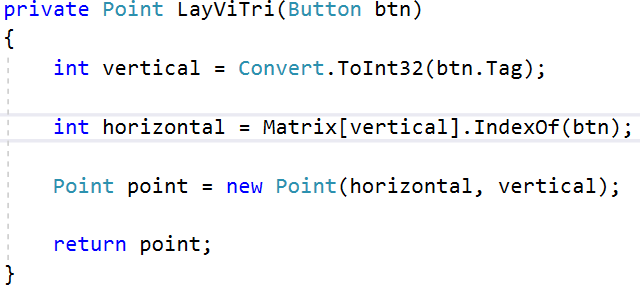
Hàm lấy vị trí:

Figure 14 : Layvitri()

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).

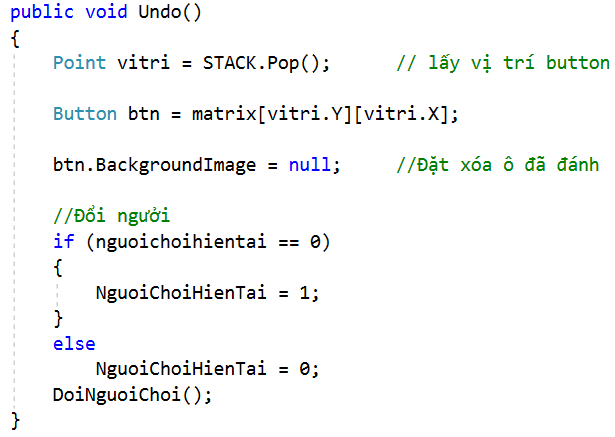
Hàm Undo():

Figure 15 : Undo()

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

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

Hàm Xoabanco():

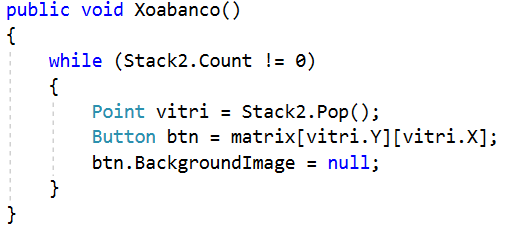
2.5.3 Ứng dụng của QUEUE

Figure 16 : Xoabanco()

A. Ứng dụng của của queue vào đồ án chính là mô phỏng ván cờ mới chơi, nghĩa là ta có thể xem lại từng nước cờ mà hai người chơi đánh, đây chính là một phần sáng tạo mà nhóm đã đạt được.

Ý tưởng: đổ các phần tử của STACK vào một Stack khác, rồi mới đưa các phần tử vào trong Queue.

Không đưa trực tiếp từ STACK vào Queue vì như vậy sẽ bị ngược, và cũng không đưa vào Queue trong lúc đánh, vì khi Undo, ván cờ được lưu sẽ không còn chính xác.

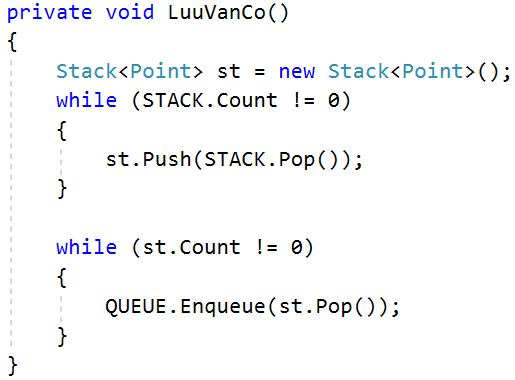
 Hàm LuuVanCo()

Figure 17 : LuuVanCo()

Hàm MoPhong():

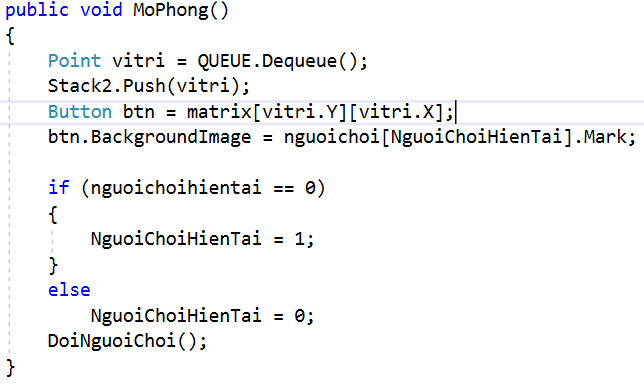
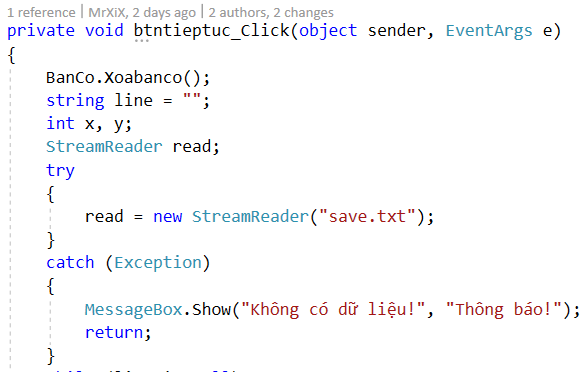
B. Ứng dụng thứ hai của queue vào đồ án là lưu lại dữ liệu ( các tọa độ ) đã được lưu trên file, khi mở game đã lưu thì sẽ gọi hàm mô phỏng ván cờ, và ván cờ đã lưu sẽ được mở ra và có thể tiếp tục chơi.

Figure 19.1 : btntieptuc\_click()

Figure 18 : MoPhong()

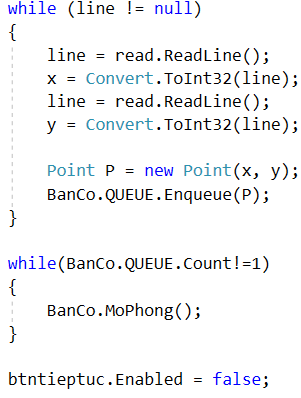


Figure 19.2 :btntieptuc() (tt)

Đoạn code trên thực hiện việc mở file và đưa các tọa độ đã lưu trên file vào queue.

2.6 Sản phẩm của đồ án

Một game caro hoàn chỉnh dành cho hai người chơi.



Figure 19: Game Caro

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. Bên cạnh đó nhóm vận dụng kiến thức đã học về Queue để mô phỏng lại bàn cờ vừachơi và lưu lại ván cờ đang chơi dở trên file để có thể tiếp tục chơi vào lúc khác đây là một nét nổi bật trong game của nhóm.

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 queue (hàng đợi) để mô phỏng lại ván cờ của người chơi giúp người chơi có thể xem lại quá trình chơi cờ.Nhóm đã tạo ra nhiều chứ năng trong game caro giúp người chơi dễ dàng sử dụng và thích thú với game hơn.

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

3.3 Nhược điểm

Chức năng lưu ván cờ trên file chưa được hoàn chỉnh : có thể lưu được ván cờ của hai người chơi nhưng vẫn có trường hợp chỉ lưu được vị trí các nước cờ đã đi mà chưa xác định được X hay O.

Form nhập thông tin sao khi nhập thông tin hai người chơi thì không tự động tắt mà vẫn còn chạy song song với form game chính nên khi nhập lại tên trong form đó thì một giao diện game nữa sẽ hiện ra đồng thời với giao diện trước.

Vẫn chưa kết nối được mọi người với nhau như qua LAN, internet…

3.4 Đề xuất phương án cải thiện

- Tìm hiểu về kết nối người chơi qua mạng LAN.

- Phát triển chức năng lưu ván cờ trên file để lưu cả vị trí lẫn xác định chính xác nước cờ.

- Tìm cách để cho form nhập thông tin sẽ tự tắt khi người chơi nhập xong thông tin hoặc làm cho form đó không chạy ẩn khi tắt đi.

PHỤ LỤC

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Công việc | | Võ Thị Kiều Diễm | Nguyễn Thành Tâm | Ngày bắt đầu dự kiến | Ngày kết thúc dự kiến | Ngày bắt đầu thực tế | Ngày kết thúc thực tế | Ghi chú |
| Thiết kế giao | Tạo background, tạo các khung sườn | X | X | 11/9 | 12/9 | 11/9 | 12/9 |  |
| Add vào các hình ảnh | X |  | 13/9 | 17/9 | 13/9 | 14/9 |  |
| Tạo bàn cờ | X |  | 13/9 | 17/9 | 18/9 | 20/9 |  |
| Xử lý đổi người chơi | Thiết lập ra hai người chơi | X |  | 13/9 | 17/9 | 13/9 | 17/9 |  |
| Xử lý đổi người |  | X | 22/9 | 25/9 | 24/9 | 25/9 |  |
| Xử lý thắng thua | Chuyển tất cả các button vào 2 list lồng nhau để xử lý như trên ma trận. |  | X | 26/9 | 2/10 | 24/9 | 28/9 |  |
| Kiểm tra hàng 5 (ngang, dọc, chéo chính, chéo phụ) |  | X | 3/10 | 9/10 | 1/10 | 5/10 |  |
| Xác định trạng thái thắng, kiểm tra người thắng. | X |  | 10/10 | 15/10 | 6/10 | 10/10 |  |
| Tạo chức năng Undo | Tìm hiểu stack |  | X | 16/10 | 24/10 | 11/10 | 15/10 |  |
| Tạo các phím chức năng | New game, hướng dẫn, mô phỏng, undo, exit, music, chế độ chơi | X | X | 25/10 | 31/10 | 16/10 | 25/10 |  |
| Mô phỏng và lưu ván cờ | Lưu ván cờ vàoQueue dể mô phỏng và lưu trên file để xem lại hoặc chơi tiếp |  | X | 27/10 | 2/11 | 26/10 | 2/11 |  |
| Debug và thử nghiệm | | X | X | 1/11 | 5/11 | 3/11 | 4/11 |  |
| Viết báo cáo | File Word | X | X | 2/11 | 7/11 | 2/11 | 29/11 |  |
| Thuyết trình | File Power Point | X | X | 5/11 | 10/11 | 26/11 |  |  |

Bảng 1: Bảng phân công

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.howkteam.vn/course/lap-trinh-game-caro-voi-c-winform-14