**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:ỨNG DỤNG STACK VÀO CHỨC NĂNG UNDO CHO GAME CARO 2 NGƯỜI CHƠI**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**Nhóm sinh viên thực hiên:**

**Nguyễn Văn Thắng 17110230**

**Mai Trọng Tường 17110251**

**Giảng viên hướng dẫn: ThS. Trần Công Tú**

**TP. Hồ Chí Minh, tháng 12 năm 2018**

MỤC LỤC

[I. LỜI NÓI ĐẦU 4](#_Toc532516441)

[II. GIỚI THIỆU ĐỀ TÀI 4](#_Toc532516442)

[1. Giới thiệu về game caro. 4](#_Toc532516443)

[2. Lí do chọn đề tài. 4](#_Toc532516444)

[3. Stack là gì? 4](#_Toc532516445)

[4. Ứng dụng Stack vào game Caro 2 người chơi 4](#_Toc532516446)

[III. NỘI DUNG 5](#_Toc532516447)

[1. Mục đích sử dụng và đối tượng hướng đến 5](#_Toc532516448)

[2. Nội dung thực hiện bao gồm 4 Form 5](#_Toc532516449)

[3. Quá trình thực hiện 5](#_Toc532516450)

[4. Các công cụ dùng vào thiết kế giao diện 6](#_Toc532516451)

[5. Các lớp được sử dụng xây dựng lên code 7](#_Toc532516452)

[6. Các phương thức được thiết kế trong lớp 8](#_Toc532516453)

[7. Tạo menu và hotkey 11](#_Toc532516454)

[7.1. Tạo menu: 11](#_Toc532516455)

[7.2.Tạo hotkey: 12](#_Toc532516456)

[8. Thiết kế code và ứng dụng Stack để Undo vào phần mềm 12](#_Toc532516457)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc532516458)

[1. Kết quả đạt được sau khi hoàn thành 19](#_Toc532516459)

[1.1 Về mặt đồ án 19](#_Toc532516460)

[1.2 Về mặt sinh viên 19](#_Toc532516461)

[2. Ưu điểm 20](#_Toc532516462)

[3. Nhược điểm 20](#_Toc532516463)

[4. Phương án khắc phục nhược điểm 20](#_Toc532516464)

[5. Hướng phát triển 20](#_Toc532516465)

MỤC LỤC BẢNG

[B1.Bảng công cụ sử dụng 3](#_Toc532514959)

[B2.Các lớp được thiết kế 4](#_Toc532514960)

[B3.1 Các phương thức chứa trong lớp 6](#_Toc532514961)

[B3.3 Các phương thức chứa trong lớp 7](#_Toc532514963)

[B3.4 Các phương thức chứa trong lớp 8](#_Toc532514964)

MỤC LỤC HÌNH

[H1.Kế hoạch phân công 5](#_Toc532515666)

[H2.Generic danh sách liên kết 13](#_Toc532515667)

[H3.Generic Stack 14](#_Toc532515668)

[H4.Khai báo biến sử dụng cấu trúc Stack 15](#_Toc532515669)

[H5.Vẽ bàn cờ 15](#_Toc532515670)

[H6.Thao tác click 16](#_Toc532515671)

[H7.Hàm quay lại 17](#_Toc532515672)

[H8.1 Cài đặt lượt Undo cho người chơi 18](#_Toc532515673)

[H8.2 Cài đặt lượt Undo cho người chơi 18](#_Toc532515674)

[H8.3 Cài đặt lượt Undo cho người chơi 18](#_Toc532515675)

[H8.4 Cài đặt lượt Undo cho người chơi 19](#_Toc532515676)

1. LỜI NÓI ĐẦU

* Game caro là một game rất phổ biến đối với thời học sinh của chúng ta.khi chúng ta chơi game caro trên giấy thì sẻ tốn 1 lượng giấy khá lớn và có khi k đủ giấy để chơi. Để giải quyết vấn đề đó nhóm tác giả đã quyết định tạo ra 1 game caro chơi trên máy tính vừa tiết kiệm giấy vừa thoải mái chơi và củng để ứng dụng những gì nhóm tác giả đã được học. Trong quá trình thực hiện để tài của mình nhóm tác giả còn có nhiều hạn chế mong được thầy và các bạn bổ sung đóng góp để game ngày càng hoàn thiện hơn. Nhóm tác giả xin cảm ơn!!

1. GIỚI THIỆU ĐỀ TÀI
2. Giới thiệu về game caro.

* Cờ caro không chỉ được chơi ở Việt Nam mà còn ở rất nhiều nước trên thế giới.  Ở mỗi nước nó lại có cái tên khác nhau.
* Ở Nhật Bản, người ta gọi cờ caro là “**Gomoku”.**
* Ở Anh, Cờ Caro được gọi là “**Connect5”.**
* Người Trung Quốc gọi cờ caro là “**Guziqi”**.
* Người vùng Đông Âu có cách gọi cờ caro bằng **“Five in a row”** nghĩa là Một Hàng Năm.
* Người Hàn Quốc gọi cờ Caro là “**Omok**.”

1. Lí do chọn đề tài.

* Tạo ra một game lành mạnh phục vụ việc vui chơi giải trí sau những giờ học làm việc cho mọi người bên cạnh đó là ứng dụng những kiến thức đã học được trong môn cấu trúc dữ liệu và giải thuật vào thực tiễn.

1. Stack là gì?

* Ngăn xếp (Stack) là một dạng danh sách được cài đặt nhằm sử dụng cho các ứng dụng cần xử lý theo thứ tự đảo ngược. Có thể hình dung ngăn xếp thông qua hình ảnh một chồng đĩa đặt trên bàn. Nếu muốn thêm vào 1 đĩa, người ta phải đặt nó lên trên đỉnh. Nếu lấy ra 1 đĩa, người ta phải lấy ra đĩa ở trên đỉnh. Ngăn xếp là một cấu trúc dữ liệu trừu tượng có tính chất “vào sau ra trước”(LIFO) hay “vào trước ra sau”(FILO)

1. Ứng dụng Stack vào game Caro 2 người chơi

* Người chơi muốn bỏ nước đi mình vừa đi và muốn đi nước mới do nhận thấy đánh sai vị trí hay một chút sai sót tạo sơ hở cho đối thủ, là lí do chính chúng ta cần tới chức năng này. Vì vậy nhóm tác giả quyết định dùng stack để làm chức năng undo cho game theo cơ chế LIFO.

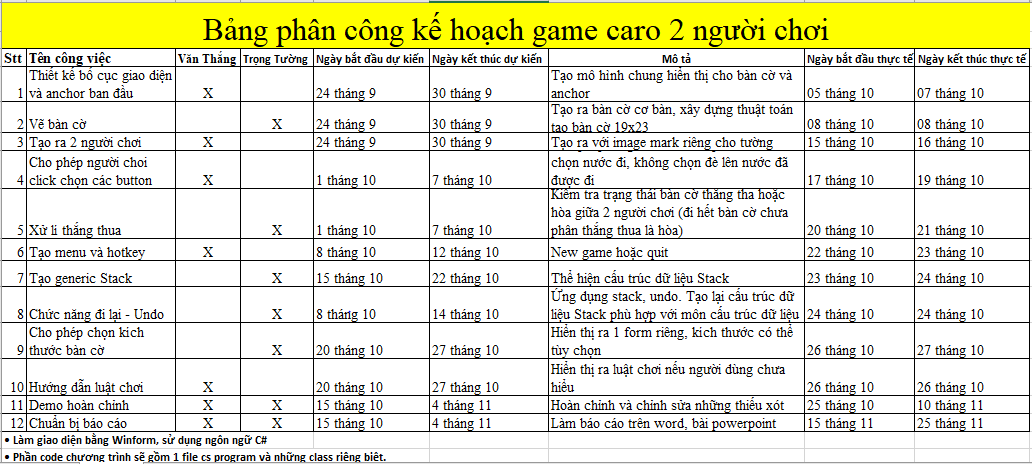
1. Ngôn ngữ C#.NET Framework 4.7.2 sử dụng Windows Forms App(.Net Framework) để cài đặt
2. NỘI DUNG
3. Mục đích sử dụng và đối tượng hướng đến

* Cờ Caro 2 người chơi được sử dụng để 2 người chơi so tài trí lược, giảm bớt căng thẳng. Cờ Caro có đặc tính dễ chơi, dễ thắng, dễ thua, nhanh, phù hợp cho những người muốn xả stress, so kè cùng bạn bè mà không muốn mất quá nhiều thời gian.
* Cờ caro là trò chơi thư giãn tốt cũng như không kém phần trí tuệ dành cho nhiều lứa tuổi, giúp tắng khả năng tư duy, khả năng kiên nhẫn để có thể dành được kết quả tốt nhất là thắng lấy đối thủ của mình.

1. Nội dung thực hiện bao gồm 4 Form

* FormBackground: đây là form giao diện ban đầu cho phần mềm, như là bộ mặt ban đầu giao tiếp với người dùng.
* Size: là form tương tác với người dùng, cho phép người dùng thiết đặt kích thước bàn cờ mà mình muốn chơi tùy ý(3x3,5x5,…)
* Form1: là form chính của phần mềm, bao gồm các kí hiệu, chức năng của ứng dụng để người dùng có khả năng sử dụng tối ưu phần mềm nhất, thực hiện được yêu cầu là tương tác với người dùng bằng những khả năng sẵn có
* Rule: Giới thiệu về cách chơi dành cho những người chơi lần đầu hoặc có thể quên.

1. Quá trình thực hiện



#### H1.Kế hoạch phân công

1. Các công cụ dùng vào thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Công cụ được sử dụng  chính | Công dụng được sử dụng | ứng dụng vào đề tài |
| 1 | Text box | Trả ra dữ liệu | Hiển thị ra tên người chơi. |
| 2 | Picture box | Cho phép truyền ảnh vào | Nơi hiển thị hình ảnh mặc định sẵn hoặc trả ra hình ảnh giúp phần mềm tang tính hấp dẫn. |
| 3 | Button | Click | Tương tác chính với người dùng như cấu tạo lên bàn cờ, click chọn để thực hiện các chức năng. |
| 4 | MenuStrip | Hiển thị list và click chọn | Hiển thị dưới dạng danh sách cho người dùng sử dụng chức năng thay vì click chọn button. |
| 5 | ToolTip | Hiển thị thông tin phím tắt | Khi để chuột gần button sẽ hiển thị phím tắt nhanh để sử dụng chức nắng đó. |
| 6 | RichtextBox | Hiển thị dữ liệu dạng văn bản | Hiển thị luật chơi. |

##### B1.Bảng công cụ sử dụng

1. Các lớp được sử dụng xây dựng lên code

* Ngoài những lớp chính của form, nhóm thực hiện còn xây dựng thêm những lớp riêng biệt để có thể dễ dàng phát triển thêm về sau.

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Tên các sinh viên phụ trách viết | Mục đích chính của lớp trong chương trình |
| 1 | Manager | Văn Thắng  Trọng Tường | Dùng để quản lí những phương thức chính sẽ sử dụng vào lớp của Form1, cũng như phần mềm này |
| 2 | StackCaro | Trọng Tường | Tạo ra 1 Generic mới về cấu trúc dữ liệu Stack để áp dụng vào xây dựng chức năng Undo cho phần mềm |
| 3 | LinkedListCaro | Trọng Tường | Tạo ra 1 Generic danh sách liên kết để chứa dữ liệu, xây dựng lên Generic Stack |
| 4 | InfoPlayer | Trọng Tường | Tại 1 ô đã được tích trên bàn cờ, đối tượng được tạo ra từ lớp này có thể dùng để lưu xem người chơi nào đã tích vào ô đó và tọa độ của ô đó |
| 5 | Player | Văn Thắng | Tạo ra đối tượng như một người chơi với tên và hình ảnh kí hiệu nước đi tượng trưng cho người chơi |

##### B2.Các lớp được thiết kế

1. Các phương thức được thiết kế trong lớp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên lớp chứa | Tên phương thức | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình | Input | Output | Tên lớp có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | Manager | DrawPanelChessBoard() | Trọng Tường | Vẽ bàn cờ bằng các nút | Kích thước bàn cờ | Vẽ bàn cờ | Lớp Form1,  Đòng số 31 |
| 2 |  | Buttonclick  () | Văn Thắng | Xử lí sự kiện click chọn | Object, EventArgs | Đánh dấu lên ô cờ | Được kích hoạt nếu click bàn cờ |
| 3 | Manager() | Văn Thắng | Truyền ký hiệu quân cờ vào cho người chơi |  | Quân cờ của 2 người chơi | Lớp manager  Dòng 43,50 |
| 4 | EnableUndo() | Trọng Tường | Gửi dưới dạng event sang Form 1 |  | EventArgs | Lớp Manager,  Dòng 203 |
| 5 | Undo() | Trọng Tường | Các thao tác để rút lại nước vừa đi và hiệu ứng |  | Thành công là true,còn lại là false | Lớp Form1, dòng 71 |

##### B3.1 Các phương thức chứa trong lớp

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 |  | SizePanel() | Trọng Tường | | Thiết lập lại kích thước của Panel chứa bàn cờ theo kích thước bàn cờ trên Form1 |  |  | Lớp Form1, dòng 29 |
| 7 |  | LocationPanel() | | Trọng Tường | Thiết lập lại tọa độ bàn cờ cho phù hợp với Form1 | Bàn cờ |  | Lớp Form1, dòng 30 |
| 8 |  | EndGame() | | Trọng Tường | Thực hiện 1 vài thao tác khi kết thúc trò chơi, gửi dưới dạng Event sang Form1 |  | EventArgs | Lớp Manager  ,dòng 191 |
| 9 |  | IsEndGame() | | Trọng Tường | Nhận ô cờ của người chơi vừa đánh và gọi những phương thức khác kiểm tra kết thúc game | button | Kết thúc là true,  còn lại là false | Lớp Manager  , dòng 189 |
| 10 |  | GetChess() | | Trọng Tường | Chuyển tọa độ ô cờ thành các điểm có thể kiểm tra trên ma trận | button | point | Lớp Manager  , dòng 185 |
| 11 |  | IsEndRow() | | Trọng Tường | Kiểm tra thắng thua theo hàng ngang | button | Thành công là true,còn lại là false | Lớp Manager  ,dòng 277 |
| 12 |  | IsEndCol() | | Trọng Tường | Kiểm tra thắng thua theo hàng dọc | button | Thành công là true,còn lại là false | Lớp Manager  ,dòng 277 |

##### B3.2 Các phương thức chứa trong lớp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 13 |  | IsEndPrimary() | Trọng Tường | Kiểm tra thắng thua theo đường chéo chính | button | Thành công là true,còn lại là false | Lớp Manager  ,dòng 277 |
| 14 | IsEndSub() | Trọng Tường  Tham khảo thêm tại: | Kiểm tra thắng thua theo đường chéo phụ | button | Thành công là true,còn lại là false | Lớp Manager  ,dòng 277 |
| 15 | Mark() | Văn Thắng | Tự chuyển đổi ký hiểu của người chơi khi người chơi thay đổi |  | Ký hiểu của người đang đánh | Lớp manager  Dòng 34 |
| 16 | ChangePlayer() | Văn Thắng | Đổi tên và ký hiểu của người chơi khi 2 người chơi thay đổi |  | Tên và ký hiệu | Lớp manager dòng 34 |
| 17 | StackCaro | Push() | Trọng Tường | Thêm dữ liệu vào StackCaro | InfoPlayer |  | Lớp Manager  , dòng 185 |
| 18 | Pop() | Trọng Tường | Lấy phần tử trên cùng re khỏi StackCaro |  | InfoPlayer | Lớp Manager  , dòng 220 |
| 19 | Top() | Trọng Tường | Trả về thông tin phần tử trên cùng |  | InfoPlayer | Lớp Manager  ,dòng 239 |

##### B3.3 Các phương thức chứa trong lớp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20 |  | Thuộc tính Count | Trọng Tường | Nhận lấy giá trị số phần tử StackCaro |  | Số nguyên | Lớp Manager  ,dòng 194, dòng 214, dòng 233. |
| 21 | LinkedListCaro | Add() | Trọng Tường | Thêm phần tử vào đầu danh sách | InfoPlayer | Trả về 1 nếu thành công, còn lại là 0 | Lớp StackCaro, dòng 23 |
| 22 |  | Remove() | Trọng Tường | Xóa phần tử đầu danh sách |  |  | Lớp StackCaro, dòng 33 |

##### B3.4 Các phương thức chứa trong lớp

1. Tạo menu và hotkey

7.1. Tạo menu:

+ vào toolbox chọn menustrip để tạo menu.Trong menu tạo các control newgame,quit,undo.

Tạo event newgame,quit và xử lý các event đó.

+ Newgame.

void NewGame()

{

ChessBoard.DrawPanelChessBoard();

}

Khi nhóm tác giả làm như v thì có bug xảy ra là bàn cờ không được tạo lại mới hướng giải quyết là theo nhóm tác giả thấy là khi muốn newgame nhưng các quân cờ k có xóa nên bàn cờ mới không thể tạo để giải quyết vấn đề đó thì nhóm tác giả đã thực hiện.

ChessBoard.Controls.Clear();

Khi thêm dòng này vào trong hàm vẽ bàn cờ thì bàn cờ đã được vẽ lại mới.tuy nhiên bây giờ nhóm tác giả đặt ra 1 câu hỏi lỡ k may ấn lộn nút newgame thì phải chơi lại từ đầu như v sẻ không hay nên nhóm tác giả đã đưa vào một thông báo hỏi xem người chơi có muốn tạo mới không nếu tạo mới thì ấn ok không thì cancel.

private void newGameToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (undoToolStripMenuItem.Enabled == false || (MessageBox.Show("Bạn có muốn tạo bàn cờ mới?", "Thông báo!",

MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK) )

{

NewGame();

}

}

+ Quit:Tương tự như khi newgame thì khi quit game nhóm tác giả muốn show ra 1 dòng thông báo để người chơi lựa chọn. Có một vấn đề xảy ra đó là khi chọn quit show ra bình thường nhưng khi chọn event thoát thì k có show thông báo mà thoát luôn. Nhóm tác giả đã xử lý bằng cách tạo event thoát và show thông báo.

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (MessageBox.Show("Bạn có chắc chắn muốn thoát!!", "Thông báo",

MessageBoxButtons.OKCancel) != System.Windows.Forms.DialogResult.OK)

e.Cancel = true;

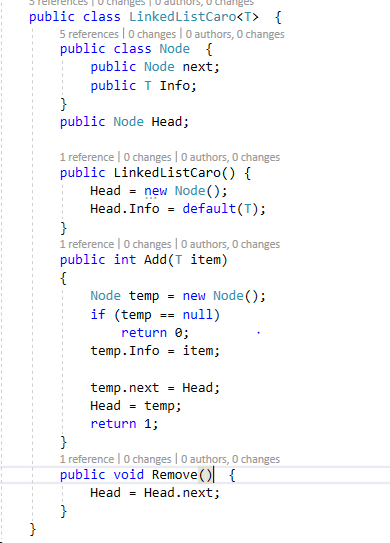
}

7.2.Tạo hotkey:

Để tạo hotkey nhóm tác giả sử dụng shoutcutkey trong properties chọn ctrol, shift, alt + một chữ cái bất kỳ để tạo hotkey.

1. Thiết kế code và ứng dụng Stack để Undo vào phần mềm

* Đầu tiên, nhận thấy quá trình thiết kế chức năng Undo cần phải thỏa mãn ràng buộc. Đó là mỗi một lượt đi, người chơi chỉ được phép sử dụng chức năng này một lần để bỏ đi nước đi của mình vừa đi, không được sử dụng 2 lần liên tiếp do có thể ảnh hưởng tới nước cờ của đối thủ. Người chơi sẽ không thể sử dụng chức năng này ngay khi đối thủ đi nước cờ tiếp theo
* Cấu trúc dữ liệu rất phù hợp để giải quyết vấn đề này. Việc thêm nước đi vào và lấy ra gần như ngay tức khắc do đặc trưng của kiêu dữ liệu này.Do đó, đầu tiên chúng ta cần có cấu trúc dữ liệu này.
* Ở đây, chúng ta vừa xây dựng chức năng cũng như yêu cầu cần làm rõ cấu trúc dữ liệu Stack nên chúng ta cần xây dựng lại kiểu cấu trúc này. Một Generic Stack sẽ giải quyết được vấn đề.
* Khi xây dựng, chúng ta tạo thêm một cấu trúc dữ liệu danh sách liên kết đơn để chứa dữ liệu.
* Danh sách liên kết là một cấu trúc dữ liệu bao gồm một nhóm các nút (node) tạo thành một chuỗi. Mỗi nút gồm dữ liệu ở nút đó và tham chiếu đến nút kế tiếp trong chuỗi.



#### H2.Generic danh sách liên kết

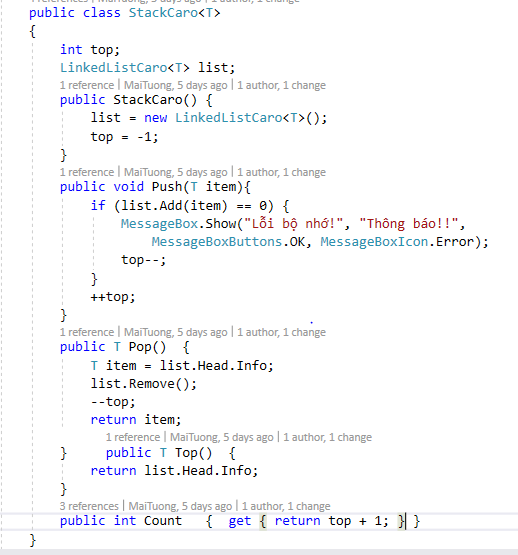
* Do thiết kế để phù hợp với phần mềm này, nên danh sách liên kết này chứa các Node và một Node Head là Node đầu của danh sách liên kết.

Với phương thức Add() giúp thêm dư liệu kiểu T vào đầu danh sách

và phương thức Remove() để xóa 1 phần tử dữ liệu ở đầu Head ra khỏi

danh sách, hỗ trợ cho cấu trúc dữ liệu StackCaro.

* Tiếp đến là Generic StackCaro

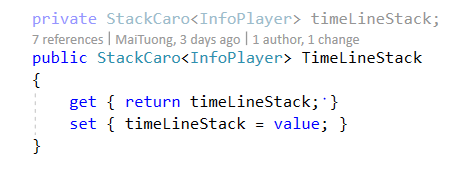


#### H3.Generic Stack

* Cấu trúc dữ liệu StackCaro là kiểu cấu trúc dữ liệu Stack. Ở trên, cấu trúc dữ liệu này được tạo ra bao gồm hàm dựng chứa 1 danh sách liên kết bên trong và 1 biến top được khởi tạo với giá trị ban đầu mang giá trị -1. Khác với hàm dựng là các phương thức đặc trưng, cơ bản của kiểu cấu trúc dữ liệu này. Không quy định số lượng phần tử có thể chứa của StackCaro nên không cần quan tâm đến ngăn xếp đầy.
* Phương thức Push(), dữ liệu sẽ được đẩy vào 1 đầu danh sách liên kết(đã trình bày ở trên), đáp ứng được yêu cầu của kiểu dữ liệu cấu trúc này là vào sau thì xếp sau, sự sắp xếp không quan tâm đến việc sắp xếp dữ liệu bên trong từng thành phần(Last In). Khi StackCaro không có giới hạn số lượng phần tử, điều cần quan tâm đến là danh sách chứa là danh sách liên kết động, mỗi lượt thêm vào chúng ta cấp phát cho danh sách thêm một vị trí nhất định trong bộ nhớ thiết bị để chứa dữ liệu, nếu chương trình chạy rất lâu mà không dừng đến 1 lúc nào đó thì sẽ không cấp phát được nữa thì sẽ xảy ra lỗi. Do đó cần kiểm tra xem có cấp phát được không, nếu không được thì sẽ không thêm dữ liệu vào và thông báo cho người dùng.
* Phương thức Pop(), lấy dữ liệu ra khỏi StackCaro tại vị trí thêm vào, thể hiện đặc trưng của kiểu cấu trúc dữ liệu (First Out).

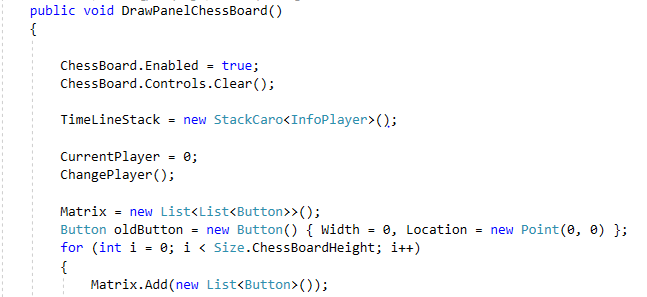
Khi lấy ra thì giảm số lượng phần tử xuống.

* Phương thức Top(), lấy ra thông tin phần tử tại vị trí thêm vào mà không hủy bỏ phần tử đó ra khỏi StackCaro.
* Thuộc tính Count, trả về số lượng phần tử của StackCaro.
* Áp dụng vào phần mềm:



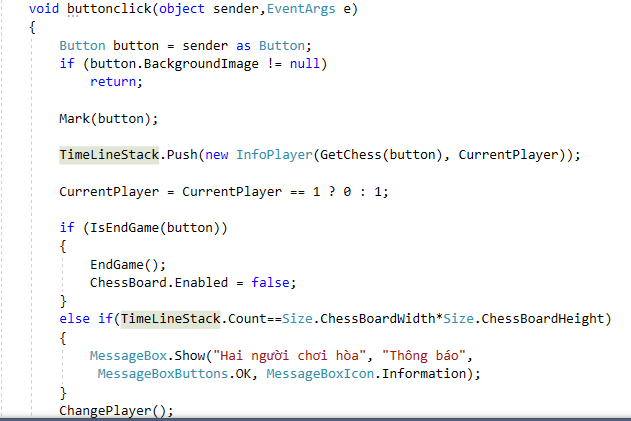
#### H4.Khai báo biến sử dụng cấu trúc Stack

* Khai báo TimeLineStack có kiểu cấu trúc dữ liệu StackCaro có thể thay đổi và trả ra giá trị.



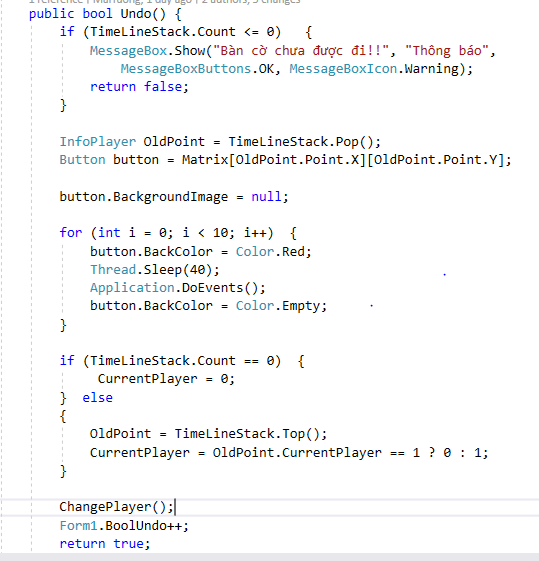
#### H5.Vẽ bàn cờ

* Khai báo khởi tạo đối tượng này trong phương thức vẽ bàn cờ là để mỗi lần vẽ ra 1 bàn cờ mới thì sẽ tạo ra đối tượng để sẵn sàng hoạt động khi người chơi bắt đầu đi nước cờ. Như vậy TimeLineStack đã là một ngăn xếp. Đồng thời sẽ khai báo một Matrix dạng có kiểu List chứa List để lưu lại vị trí các button được thêm vào bàn cờ, để khi muốn xác định vị trí button trên bàn cờ ta có thể xác định trên ma trận này.



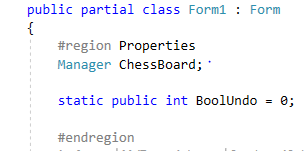
#### H6.Thao tác click

* Khi người chơi click đi trên bàn cờ, InfoPlayer sẽ được tạo ra chứa thông tin vị trí button click chọn và người chơi hiện tại là người chơi nào. InfoPlayer được thêm vào để có thể dễ dàng lấy ra.
* Tại đây sau khi kiểm tra thắng thua, có thể dùng thuộc tính Count để xét hòa nhau trường hợp 2 người chơi đã đi hết bàn cờ mà vẫn chua phân thắng thua, vì bàn cờ là cố định mà 2 người chơi đã hết đường đi.
* Khi có ngăn xếp đã có dữ liệu, tiếp đến là phần cài đặt chức năng chính là Undo. Để làm được điều này, cần xây dựng sẵn phương thức Undo.



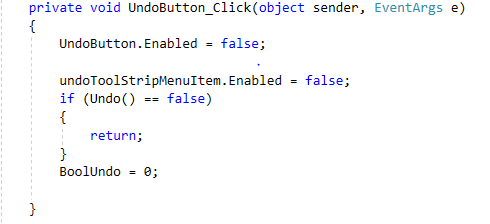
#### H7.Hàm quay lại

* Những thao tác chính của phương thức này là kiểm tra xem ngăn xếp có rỗng không, nếu không rỗng thì lấy InfoPlayer ra khỏi ngăn xếp, vì InfoPlayer có chứa thông tin vị trí nên ta có thể xác định trên Matrix đó là button tại vị trí nào và xóa nước đi tại đó. Đồng thời thay đổi người chơi do lượt qua rồi và cần hồi lại.
* Như trên là cơ bản hoàn tất chức năng Undo nhưng theo thiết kế code thì ta cần thiết lập mỗi người chơi chỉ được phép Undo 1 lần trên 1 lượt đi. Để làm được điều này có 1 hướng đi hiệu quả đó là vô hiệu hóa người chơi thao tác chức năng Undo tại nước cờ vừa sử dụng. Việc vô hiệu hóa này thao tác được ở class Form1 mà phương thức Undo và phương thức chính duy trì hoạt động của bàn cờ đặt tại class Manager.
* Xây dựng 1 phương thức chung giữa 2 class là giải pháp được đề xuất.



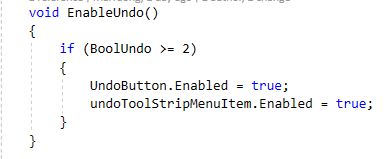
#### H8.1 Cài đặt lượt Undo cho người chơi

* Khai báo biến BoolUndo với giá trị bằng 0. Giá tị của biến sẽ tăng lên 1 sau khi người dùng tích chọn trên bàn cờ(H6)



#### H8.2 Cài đặt lượt Undo cho người chơi

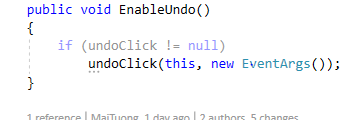
* Thao tác sử dụng Undo sẽ bị vô hiệu ngay sau khi người dùng sử dụng, đồng thời biến BoolUndo sẽ được gán trở lại bằng 0.
* Để mở lại chức năng này cho người chơi tiếp theo sử dụng, thì biến BoolUndo cần tăng lên bằng 2, được thực hiện tại Form1.



#### H8.3 Cài đặt lượt Undo cho người chơi

* Phương thức này được gọi để kiểm tra mỗi lần người chơi click chọn(H ). Tiếp theo là xây dựng phương thức bên class Manager và tạo liên kết giữa 2 phương thức là hoàn tất.



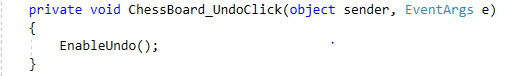


#### H8.4 Cài đặt lượt Undo cho người chơi

* Class Manager tryền event này sang Form1



* Phương thức mới tự được tạo, cuối cùng là truyền phương thức chung tại class này vào.



* Quá trình xây dựng cấu trúc dữ liệu, cài đặt và ràng buộc hoàn thành.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được sau khi hoàn thành

### Về mặt đồ án

* Phần mềm đáp ứng những chức năng cơ bản cho một phần mềm game caro hai người chơi.
* Ứng dụng được kiến thức môn Cấu Trúc Dữ Liệu và Giải Thuật, trọng tâm là ngăn xếp Stack vào phần mềm
  1. Về mặt sinh viên
* Nắm được những kiến thức về bộ môn Cấu Trúc Dữ Liệu và Giải Thuật, cụ thể là việc áp dụng một cấu trúc dữ liệu ngăn xếp StackCaro vào giải quyết bài toán thực tế , áp dụng vào tạo ra một sản phẩm
* Học sử dụng những công cụ, phần mềm mới như là GitHub,

WinForm

* Nâng cao khả năng lập trình, khả năng tư duy.

1. Ưu điểm

Đáp ứng như một trò chơi hoàn chỉnh để 2 người có thể giải trí

1. Nhược điểm

* Khi ở chế độ toàn màn hình thì các bố cục trên form sẽ bị phá vỡ
* Khi người chơi kết thúc thắng ở nước cờ nào thì chưa có một hiệu ứng hợp lí để hai người chơi có thể nhìn, khó nhìn hơn khi người dùng chọn kích thước bàn cờ là tối đa
* Giới hạn 2 người chơi trên cùng 1 máy, khi người dùng sử dụng chức năng Undo, ta không thể xác định là người chơi nào đang sử dụng chức năng này.

1. Phương án khắc phục nhược điểm

* Phát triển thêm 2 người chơi có thể chơi trên 2 máy riêng, kết nối mạng lan hoặc chơi online.

1. Hướng phát triển

* Phát triển 2 người chơi có thể chơi với nhau qua 2 thiết bị riêng bằng mạng LAN hoặc online.
* Qua đó, 2 người chơi có thể trò chuyện với nhau bằng tin nhắn hoặc âm thanh.