

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Đề tài

Game cờ caro

GVHD: Mai Đức Trung
SV: Triệu Tấn Hùng - 1812475
Nguyễn Trọng Minh - 1813093
Trần Minh Hoàng - 1812295
Cao Văn Tín - 1814334

TP. HỒ CHÍ MINH, THÁNG 11/2019



Mục lục

1	Lời mở đầu	2
2	Giới thiệu	2
3	Cơ sở lý thuyết	2
3.1	Thuật toán Min-Max	2
3.2	Cắt tỉa Alpha-Beta	2
4	Mô tả chương trình	3
4.1	Giới thiệu:	3
4.2	Tính năng	3
5	Tìm hiểu code	5
5.1	Class CaroPiece	5
5.2	Class ChessBoard	6
5.3	Class Socket Manager	7
5.4	Class Socket Data	8
5.5	Class CaroChess	8
6	Class Diagram	9
7	Kết luận	10

1 Lời mở đầu

Cờ Caro là một trong những trò chơi phổ biến, đặc biệt trong giới học sinh và sinh viên. Từ thời ban đầu con người thi đấu trên giấy, trên bề mặt viết kẻ được dần dần con người tiến tới lập trình thi đấu trên máy. Cờ Caro thể hiện tính chất đối kháng cao trong thi đấu game đòi hỏi suy nghĩ của người chơi.

Trong quá trình hoàn thành đề tài này, chúng em có tìm hiểu một số thuật toán như tìm kiếm nước đi Minmax, giải thuật amplax-beta cũng như kết hợp với kỹ năng lập trình ngôn ngữ C#. Chúng em xin cảm ơn sự hướng dẫn tận tình của thầy Mai Đức Trung, cả về chuyên môn cũng như định hướng. Vì kiến thức còn hạn hẹp nên quá trình thực hiện đề tài không tránh khỏi thiếu sót. Vì vậy rất mong sự góp ý của thầy để đề tài có thể hoàn thiện hơn nữa.

2 Giới thiệu

Trò chơi gồm có hai chế độ chơi giữa hai người chơi với nhau và giữa người và máy.

Khi chạy trò chơi thì sẽ có bàn cờ xuất hiện, trong quá trình chơi sẽ có tính năng lùi lại nước đã đi (Undo) và sẽ hiện ra thông báo thắng cuộc.

Không gian bàn cờ sẽ có kích thước là 20 hàng x 20 cột.

Khi vào vào chơi chúng ta sẽ lựa chọn 1 trong hai chế độ chơi: chơi giữa hai người và chơi giữa người với máy. Khi vào chế độ máy chúng ta có thể lựa chọn là người đi trước hay máy đi trước. Khi tham gia trò chơi thì chúng ta sẽ luân phiên đánh các quân cờ vào bàn cờ với nửa hình ảnh quy định trước.

Cuộc chơi sẽ kết thúc khi mà bên nào đánh được 5 ô liên tiếp nhau và không bị chặn hai đầu theo : hàng, cột, hai đường chéo.

3 Cơ sở lý thuyết

3.1 Thuật toán Min-Max

Trong 2 người chơi thì một người gọi là người chơi cực đại (Max) và đối thủ của họ là người chơi cực tiểu (Min). Cả 2 đấu thủ đều cố gắng đi những nước thế nào để điểm tuyệt đối của mình lớn hơn hay cao nhất có thể. Tức là người chơi Max sẽ tìm cách làm điểm của mình cao hơn và làm điểm của đối thủ bớt âm hơn (giảm về trị số). Trong khi người chơi Min thì ngược lại, sẽ cố gắng làm cho điểm của mình âm hơn và làm cho điểm của đối thủ giảm.

3.2 Cắt tỉa Alpha-Beta

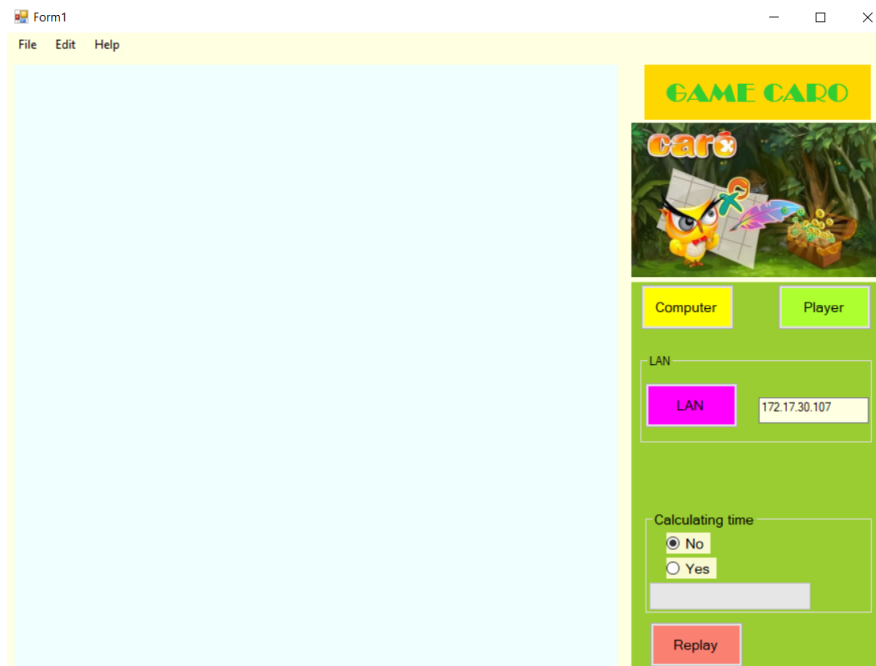
Thuật toán cắt tỉa Alpha-Beta Thuật toán cắt tỉa Alpha – Beta là cải tiến của thuật toán Min – Max với tư tưởng “Nếu đã thấy một việc làm là tệ thì không nên mất thời gian xem nó tệ đến mức nào”.

Thuật toán làm giảm số nút cần thiết của việc tìm kiếm để không lãng phí thời gian tìm kiếm những nước đi đã bất lợi rõ cho người chơi. Giải thuật Alpha – Beta cải tiến so với Min – Max bằng cách thêm vào 2 tham số là alpha và beta. Chúng cho biết các giá trị nằm ngoài khoảng [alpha, beta] là các điểm không cần xem xét nữa. Thủ tục Alpha – Beta được bắt đầu tại nút gốc với giá trị của alpha là - infinity và beta là + infinity. Thủ tục sẽ tự gọi đệ quy chính nó với khoảng cách giữa các giá trị alpha và beta ngày càng hẹp dần.

4 Mô tả chương trình

4.1 Giới thiệu:

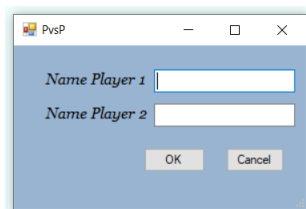
- Tên trò chơi : Game Caro
- Giao diện:



4.2 Tính năng

- Chơi giữa hai người chơi với nhau:

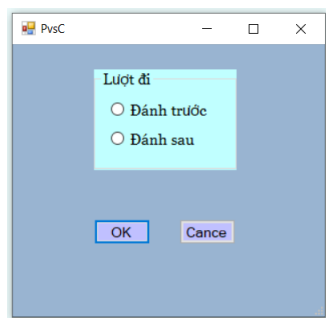
Có hai cách để khởi động tính năng này là vào File chọn New Game chọn Player vs Player hoặc là chọn vào Button Player bên phải giao diện ở dưới logo game. Thì sẽ xuất hiện một form để người chọn nhập tên của mình.



Người chơi 1 sẽ được mặc định là quân X và người chơi sẽ mặc định là quân O.

- **Chơi với máy:**

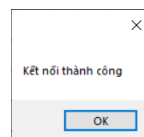
Có hai cách để khởi động tính năng này là vào File chọn New Game chọn Player vs Computer hoặc là chọn vào Button Computer bên phải giao diện ở dưới logo game. Thì sẽ xuất hiện một form để người chọn nhập tên của mình, chọn lựa đi.



Người sẽ được mặc định là quân O và máy sẽ mặc định là quân X.

- **Kết nối mạng LAN:**

Khi chọn button LAN, nếu như đã có thiết bị kết nối trước đó thì chúng ta sẽ trở thành Client, còn nếu chưa có thiết bị kết nối thì chúng ta sẽ trở thành Sever. Mặc định là Sever được quyền đánh trước còn Client đánh sau.

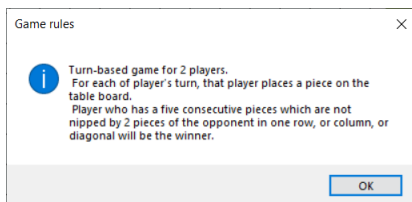


- **Undo và Redo:**

Tính năng Undo và Redo ở đây không áp dụng cho việc đánh qua mạng LAN.

- **Help:**

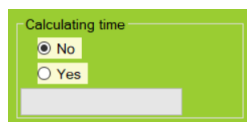
– Luật chơi.



– About: Giới thiệu



- Tính thời gian:



5 Tìm hiểu code

5.1 Class CaroPiece

- Chứa kích thước của một ô cờ:
 - Chiều rộng.
 - Chiều cao.
- Thông tin của quân cờ khi đánh vào ô cờ đó:
 - Hàng của ô cờ.
 - Cột của ô cờ.
 - Tọa độ của ô cờ.
 - Người chơi sở hữu ô cờ đó, giá trị sở hữu ban đầu là 0.
- Phương thức khởi tạo của một ô cờ.
- Code:

```
namespace Game_caro
{
    class ChessPiece
    {
    public:
        const int _Width = 30;
        const int _Height = 30;

        private int _Row;
        public int Row;

        private int _Column;
        public int Column;

        private Point _Position;
        public Point Position;

        private int _Owner;
        public int Owner;

        public ChessPiece(int row, int column, Point pos, int owner)
        public ChessPiece()
    }
}
```

5.2 Class ChessBoard

- Các thông của bàn cờ:
 - Số dòng.
 - Số cột.
- Các Phương thức:
 - Khởi tạo một bàn cờ với các giá trị dòng và cột truyền vào.
 - Vẽ bàn cờ trên một panel.
 - Vẽ quân cờ một ô cờ với tọa độ truyền vào của bàn cờ.
 - Xóa quân cờ tại một ô cờ với tọa độ truyền vào trên bàn cờ.
- Code:

```
namespace Game_caro
{
    class ChessBoard
    {
    private:
        int _NumOfLines;
        int _NumOfColumns;

    public:
        int NumOfLines;
        int NumOfColumns;

        ChessBoard();
        ChessBoard(int numOfLines, int numOfColumns);

        // Vẽ bàn cờ
        void DrawChessBoard(Graphics g);

        // Vẽ quân cờ
        void DrawChess(Graphics g, Point point, Image img);

        // Xóa quân cờ
        void RemoveChess(Graphics g, Point point, SolidBrush sb);
    }
}
```

5.3 Class Socket Manager

Dùng để kết nối hai thiết bị có chung dùng chung được PORT vào chung IP.

Ở đây chúng em tìm kiếm trên Inter là tận dụng lại source code Socket Manager từ một trang web, chúng có để đường link của trang web đã truy cập ở phần tài liệu tham khảo. Ở đây chúng em đã tận dụng nữa hàm cần thiết vào đơn giản cho mô hình của chúng em và có sửa chữa cho dễ sử dụng.

- Các hàm đã tận dụng:
 - GetLocalIPv4 : dùng để lấy ra IP V4 của card mạng đang sử dụng.
 - SerializeData : chuyển đổi một đối tượng(Object) thành một mảng byte để truyền đi. Vì máy chuyển nhận thông tin truyền đi dưới dạng byte.
 - DeserializeData: khi thông tin được truyền tới địa chỉ cần nhận thì hàm này sẽ dùng để chuyển kiểu byte thành đối tượng để máy tính dễ dàng sử dụng.
- Các hàm đã được nhóm thêm vào:
 - ConnectServer và CreateServer : khi chúng ta chọn kết nối mạng LAN, khi thiết bị sẽ thử kết với Server bằng hàm ConnectServer nếu như đã có sẵn Server để chúng ta sử dụng và khi đó chúng ta sẽ trở thành Client, ngược lại nếu như chưa có Server để tận dụng thì thiết bị chúng ta sẽ chuyển thành một Server và đợi một thiết bị kết nối với chúng ta bằng hàm CreateServer và đợi một thiết bị kết nối với chúng ta.
 - SendData và ReceiveData : ở đây chúng em sử dụng hàm SerializeData và DeserializeData để chuyển đổi dữ liệu cho việc truyền và nhận data.
 - Send và Receive : hình thức đóng gói dữ liệu.

```
namespace GameCaro
{
    class SocketManager
    {
        #region Client
        Socket client;
        public bool ConnectServer()...
        #endregion

        #region Server
        Socket server;
        public void CreateServer()...
        #endregion

        #region Both
        public string IP = "127.0.0.1";
        public int PORT = 9999;
        public const int BUFFER = 1024;
        public bool isServer = true;
        public bool Send(object data)...
        public object Receive()...
        private bool SendData(Socket target, byte[] data)...
        private bool ReceiveData(Socket target, byte[] data)...
        public byte[] SerializeData(object o)...
        public object DeserializeData(byte[] theByteArray)...
        public string GetLocalIPv4(NetworkInterfaceType _type)...
        #endregion
    }
}
```


5.4 Class Socket Data

- Chứa các thông tin cần truyền đi:
 - Lệnh thực hiện.
 - Tọa độ của mình đã nếu có.
 - Thông báo khi có tín hiệu kết nối, hoặc tín hiệu khi kết thúc game.
- Kiểu dữ liệu các lệnh truyền đi:
 - Gửi tọa độ của vị trí chúng ta đã đến.
 - Chơi game mới.
 - Thoát game.
 - Thông báo khi kết nối.
 - Kết thúc game.
- Code:

```
namespace GameCaro
{
    [Serializable]
    public class SocketData
    {
        private int command;

        public int Command...

        public Point Point...

        public string Message...

        private string message;
        private Point point;
        public SocketData(int command, string message, Point point)
        {
            this.Command = command;
            this.Point = point;
            this.Message = message;
        }
    }

    public enum SocketCommand
    {
        SEND_POINT,
        NEW_GAME,
        QUIT,
        NOTIFY,
        END_GAME,
    }
}
```

5.5 Class CaroChess

Nơi khai báo các thuộc tính, diễn ra mà thực hiện tất cả các tính năng của game tại đây.

- Các thuộc tính:
- Các phương thức:
 - Vẽ bàn cờ.
 - Tạo mảng 2 chiều ô cờ của các ô cờ trên bàn cờ.

- Phương thức đánh cờ
- Khởi động bàn cờ khi kích hoạt.
- Tính năng Undo, Redo
- Kiểm tra game đã kết thúc hay chưa và thông báo.
- Chưa thuộc toán AI của game.

```
public enum END{...}
private END _end;
public static Pen pen;
public static SolidBrush sbPnl;
private ChessPiece[,] arrChessPiece;
private ChessBoard chessBoard;
private Stack<ChessPiece> stkChessUsed;
private Stack<ChessPiece> stkChessUndo;

private int turn;
private bool ready;

public static string namePlayer1;
public static string namePlayer2;

private int mode;
// mode = 1 => PvP
// mode = 2 => PvC

private Image ImageO = new Bitmap(Game_Caro.Properties.Resources.o);
private Image ImageX = new Bitmap(Game_Caro.Properties.Resources.x);

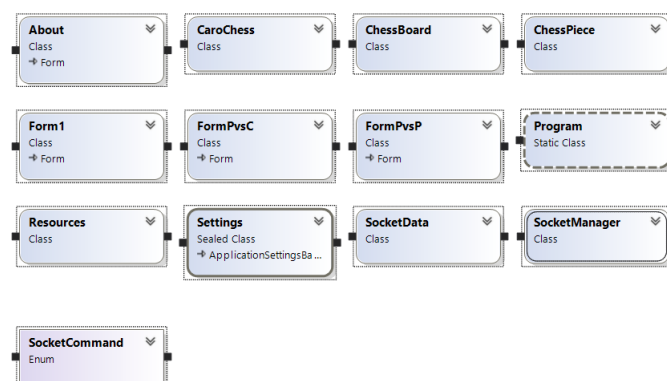
public Player Player1 = new Player(namePlayer1, new Bitmap(Game_Caro.Properties.Resources.MarkO));
public Player Player2 = new Player(namePlayer2, new Bitmap(Game_Caro.Properties.Resources.MarkX));

public bool Ready{...}

public int Mode{...}

public CaroChess(){...}
```

6 Class Diagram





7 Kết luận

Vì kiến thức và kỹ năng lập trình ứng dụng còn hạn hẹp nên còn nhiều sai sót nên mong thầy có góp ý để tui em phát triển hơn.

Tài liệu

- [1] [SocketManager.cs source code in C .NET](#)
- [2] [Thuật toán cờ caro](#)