HƯỚNG DẪN CODE GAME CARO TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C++

**PHÂN TÍCH Ý TƯỞNG:**

Hình thành ý tưởng cho việc code game:

+ Mình sẽ đưa ra 1 vài gợi ý cho bạn để hoàn thiện trò chơi caro :

B1: Sử dụng mảng 2 chiều để lưu trạng thái bàn cờ CARO.

B2: Cài đặt kiểu chuỗi kí tự trong bảng mã ASCII để người chơi đánh X và O.

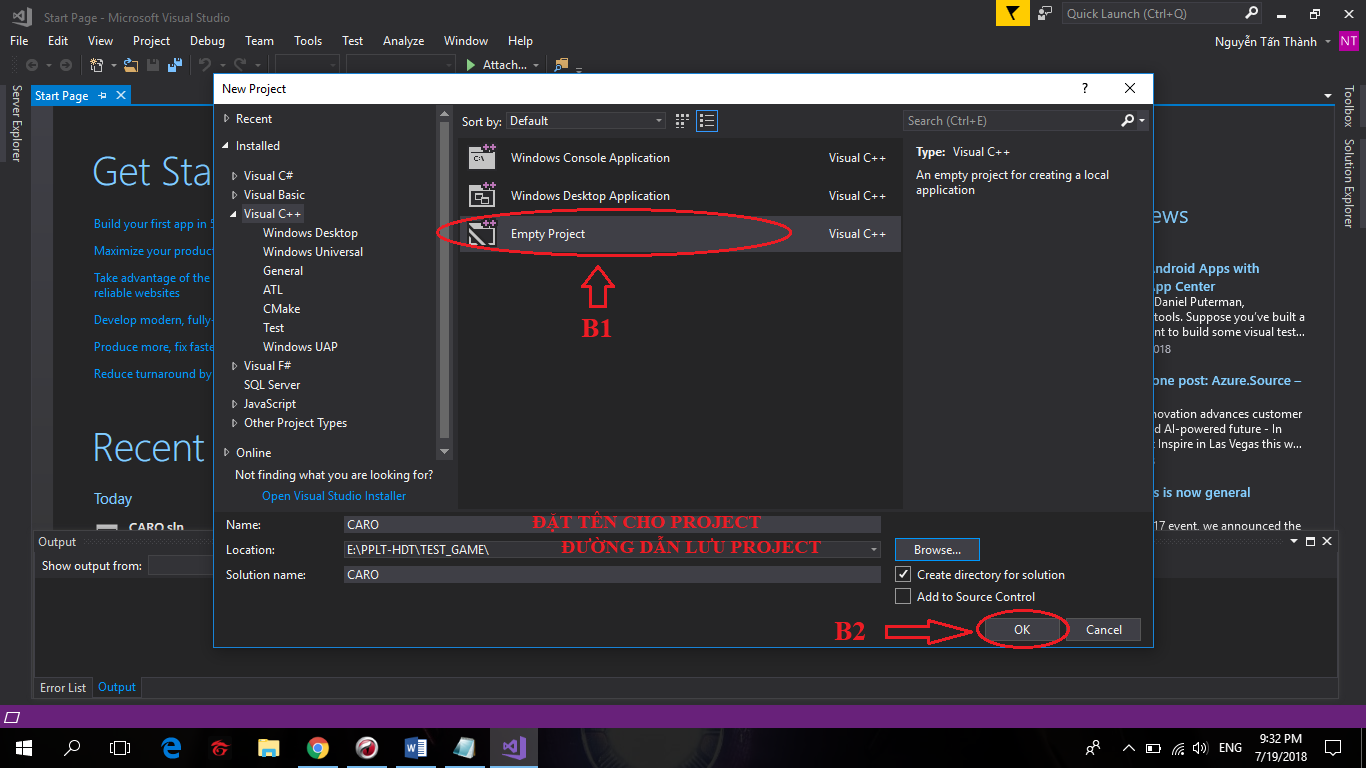
B3: Vì mỗi ô cờ giống nhau nên bạn ko cần phải cout nhiều lần để vẽ hết cả bàn cờ. Hãy dùng vòng lặp và in ra bàn cờ MxN với M, N mảng cho trước.

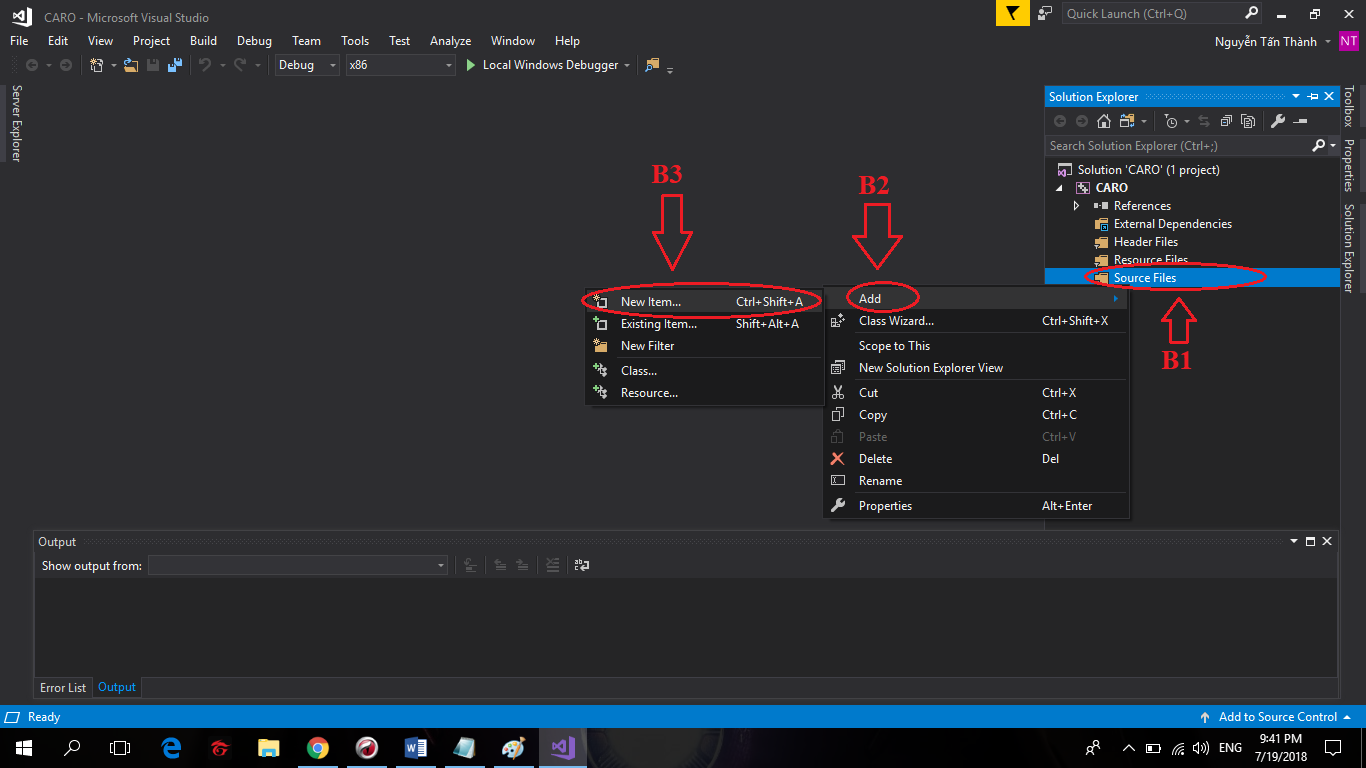
B4: Cập nhật lại bàn cờ khi người chơi đánh.

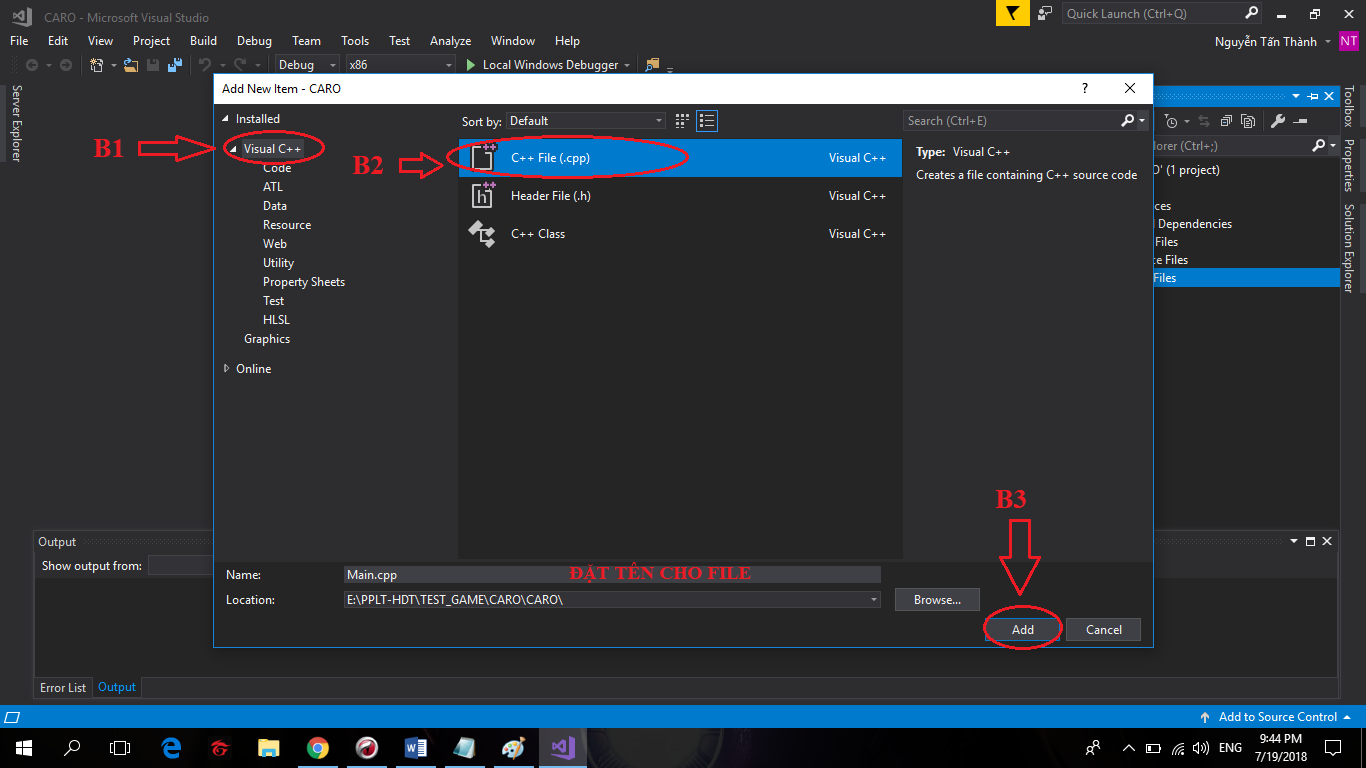
B5: Ở phần xét thắng thua của bạn, ở đây là bàn cờ 3x3. Giờ ta đã có mảng 2 chiều, bạn có thể dùng vòng lặp để duyệt từng ô, hãy nghĩ ở mỗi ô bạn xét cái gì để biết có ai thắng ko?

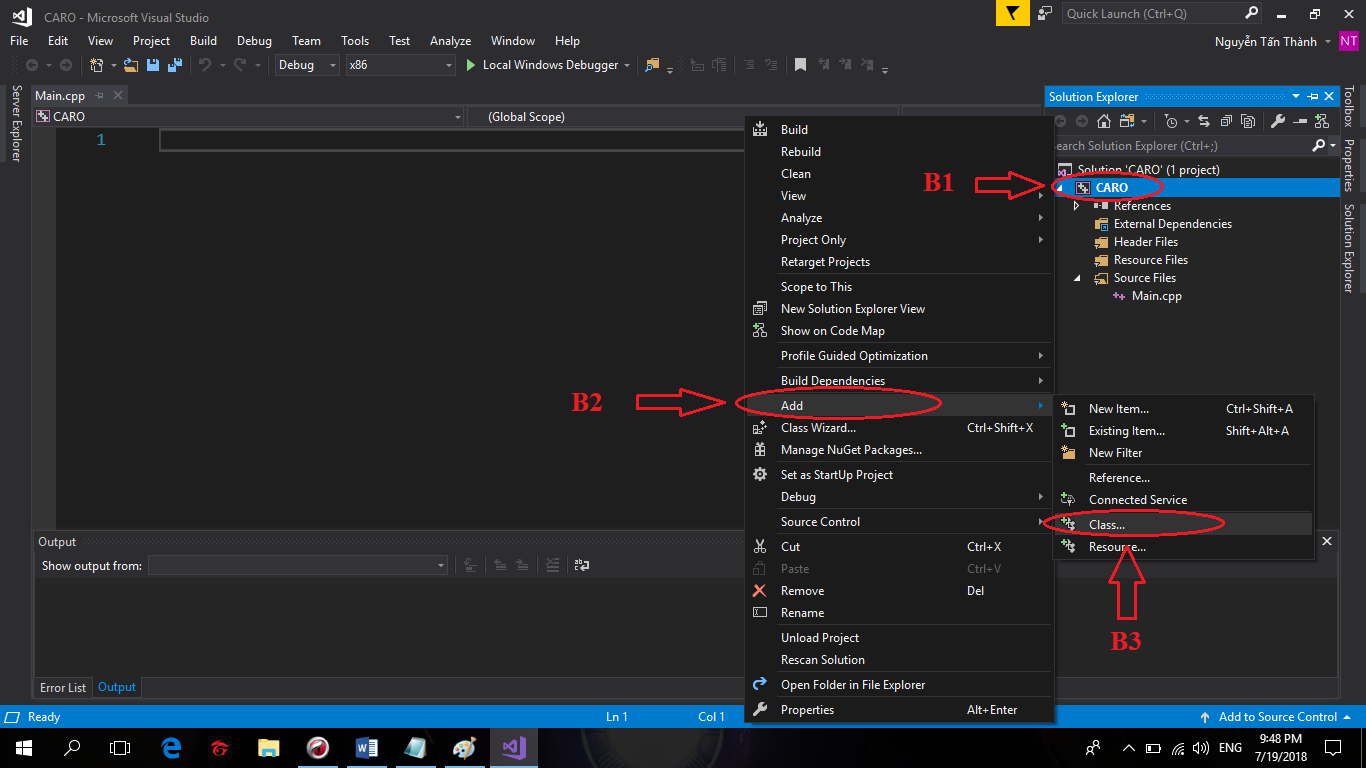
B6: Tùy chỉnh để hoàn thành game CARO.

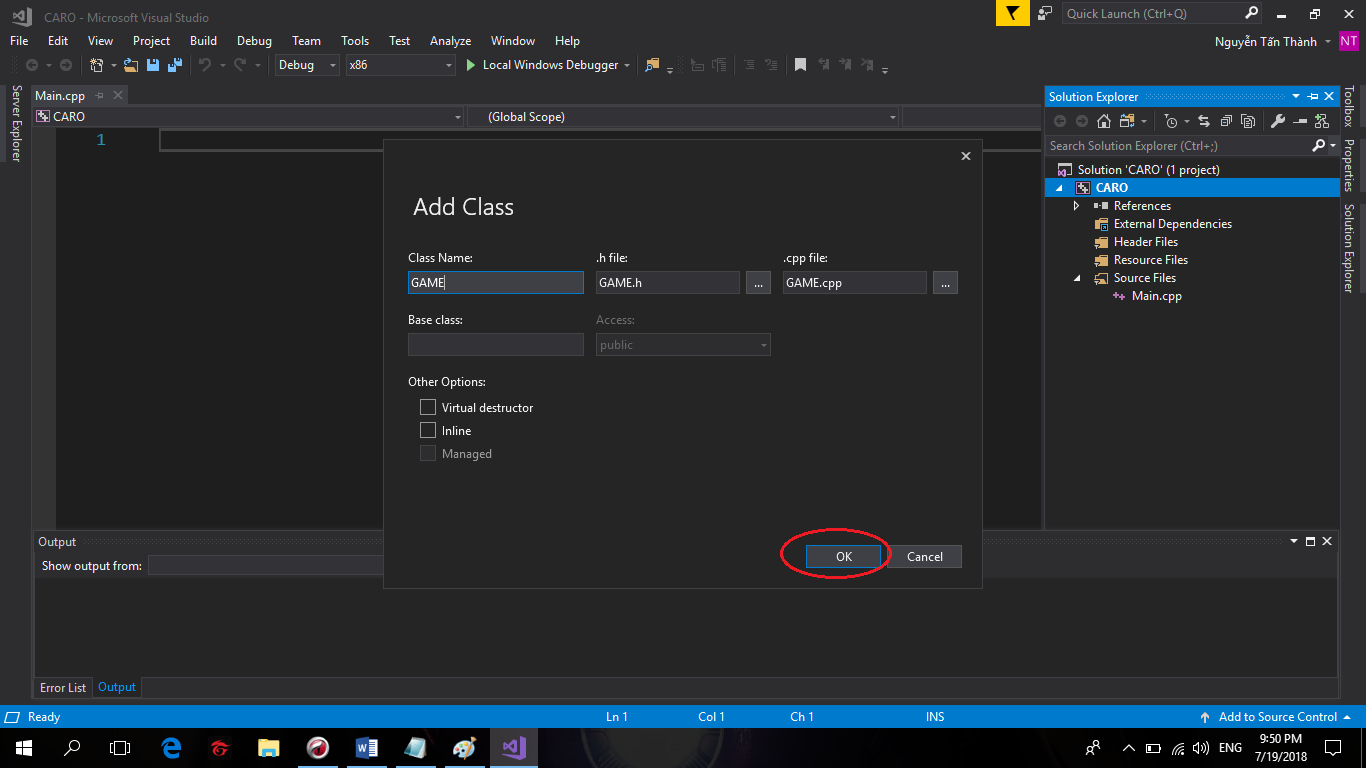
**TẠO PROJECT VỚI VISUAL STUDIO**











**TIẾP HÀNH CODE:**

Dựa vào Ý TƯỞNG ở bên trên ta có:

\*\*\* START TẠO KHUNG LƯỚI \*\*\*

Ở FILE GAME.h, ta sẽ có các dòng code:

#pragma once

class GAME // Tên Class

{

private:

char NguoiChoi; // Tạo ra biến NguoiChoi kiểu char

char Luoi[3][3]; // Tại ra 1 mảng tên Luoi[][] kiểu char

int soLuotDiChuyen;

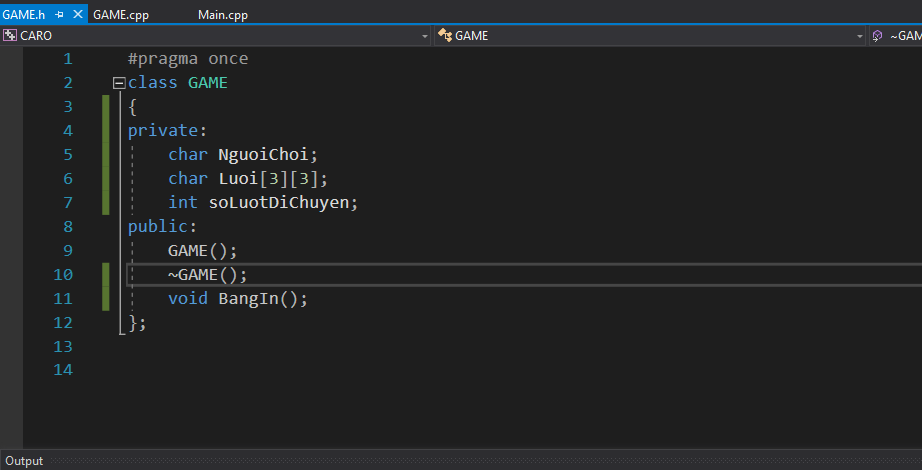
public:

GAME(); // Hàm khởi tạo mặc định

~GAME(); // Hàm hủy

void BangIn(); // Hàm tạo bảng in trong game

};



Ở File GAME.cpp, ta code:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++) // Vòng lặp trong mảng 2 chiều

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A'; // Gán giá trị ban đầu cho Lưới cờ là ‘A’

}

}

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j]; // Điều Kiện loại 3 (? ĐK đúng và : ĐK sai)

// Nếu thỏa Đk Luoi[i][j] = ‘A’ thì c = (‘0’ + b)

// Ngược lại thì c = Luoi[i][j]

cout << c;

if (j != 2) // Điều kiện để tạo khung lưới

{

cout << " | ";

}

}

if (i != 2) // Điều kiện để tạo khung lưới

{

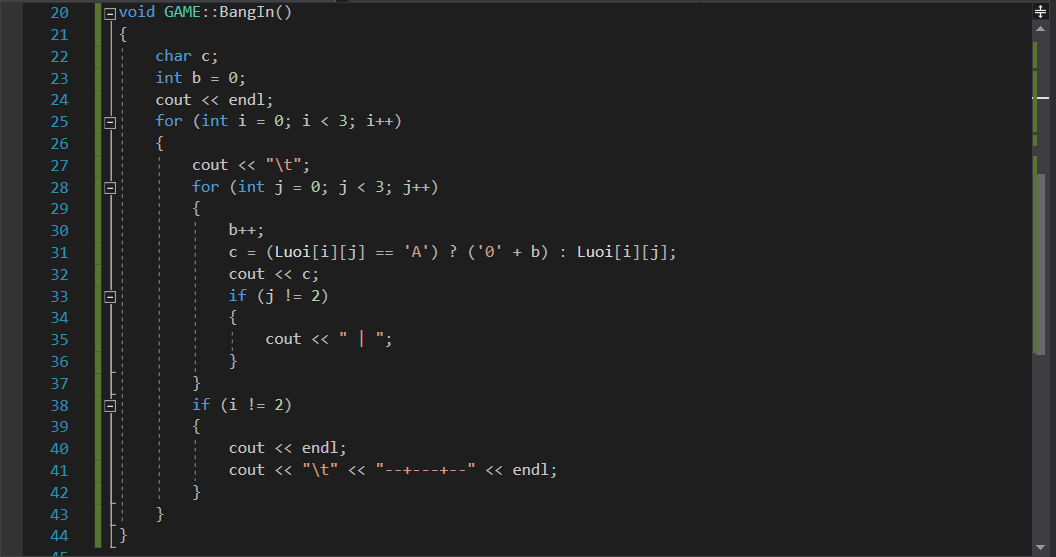
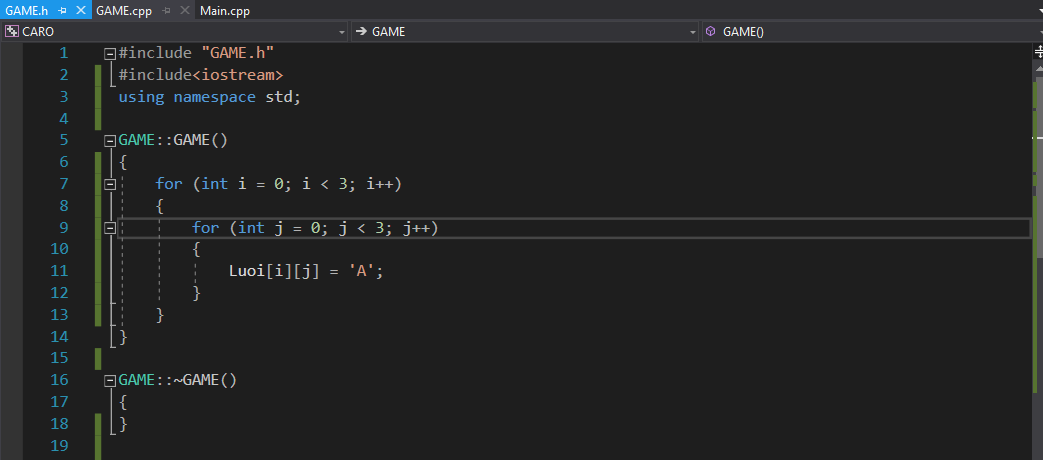
cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}



Ở file Main.cpp, ta có code:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

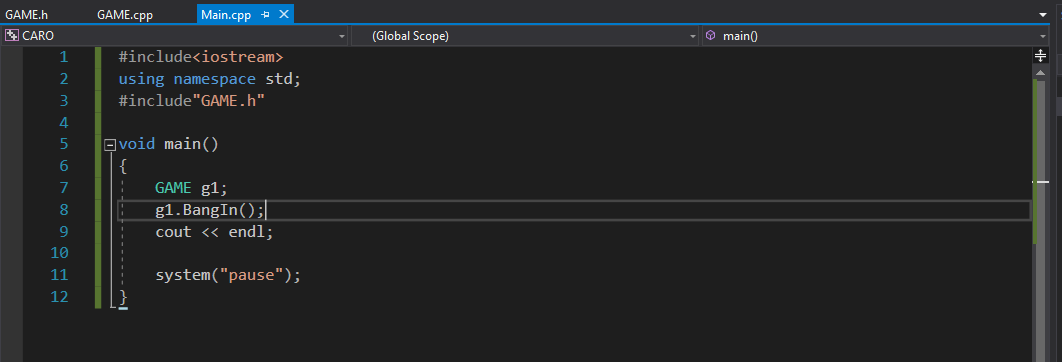
GAME g1; // Đổi tên cho Class ban đầu thành g1

g1.BangIn(); // Sử dụng (.) để gọi lại hàm Bảng in

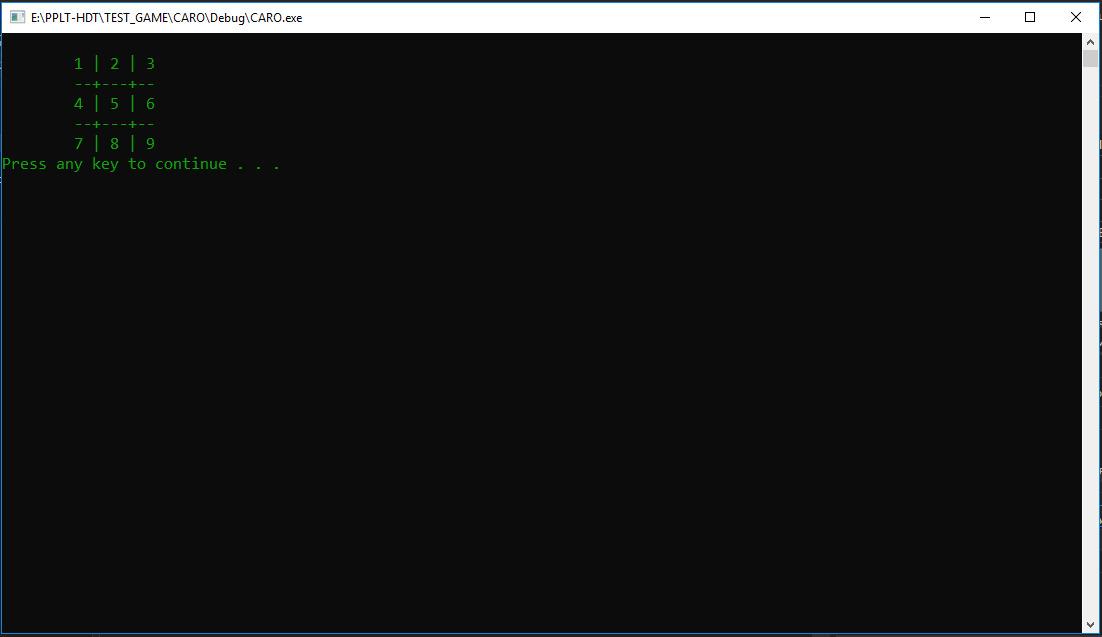
cout << endl; // Xuống dòng

system("pause"); // Dừng để xem chương trình

}



KẾT QUẢ:



Vậy là chúng ta đã vừa hoàn thành xong được khung lưới của bàn cờ caro.

\*\*\* END TẠO KHUNG LƯỚI \*\*\*

\*\*\* START NGƯỜI CHƠI \*\*\*

Tiếp theo, chúng ta sẽ xây dựng người chơi để có thể đánh được trên bàn cờ:

Ở file GAME.h, chúng ta sẽ thêm như sau:

#pragma once

class GAME

{

private:

char NguoiChoi;

char Luoi[3][3];

int soLuotDiChuyen;

public:

GAME();

~GAME();

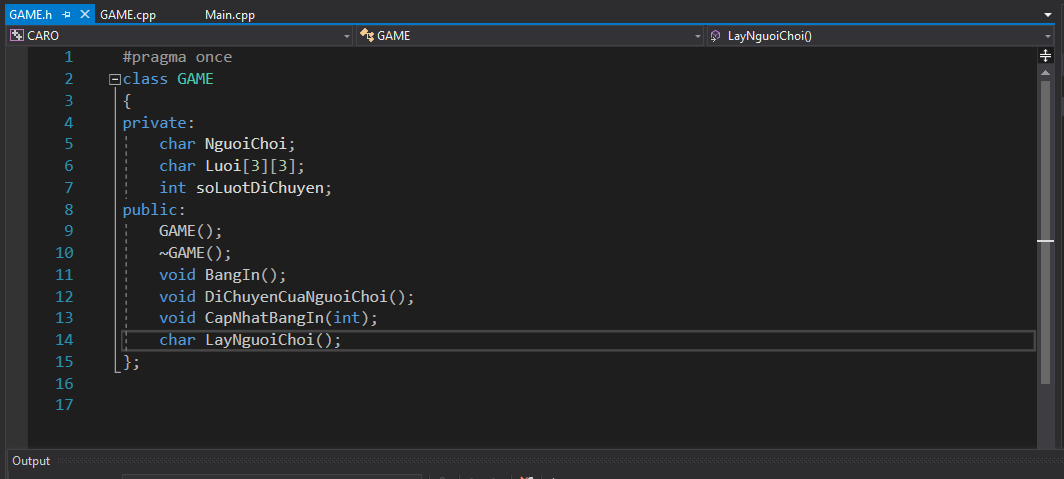
void BangIn();

void DiChuyenCuaNguoiChoi(); // Tạo hàm để xác định di chuyển của người chơi

void CapNhatBangIn(int); // Tạo hàm Cập nhật lại bảng in với tham số truyền vào thuộc kiểu int

char LayNguoiChoi(); // Tạo hàm lấy người chơi thuộc kiểu char

};



Ở file GAME.cpp, ta thêm:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A';

}

}

NguoiChoi = 'O'; // Setup người chơi mặc định là O

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j];

cout << c;

if (j != 2)

{

cout << " | ";

}

}

if (i != 2)

{

cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}

char GAME::LayNguoiChoi()

{

return NguoiChoi; // Trả về giá trị hiện hành của người chơi

}

void GAME::DiChuyenCuaNguoiChoi()

{

int dichuyen;

BangIn();

cout << "\nChon Vi Tri: ";

cin >> dichuyen;

CapNhatBangIn(dichuyen);

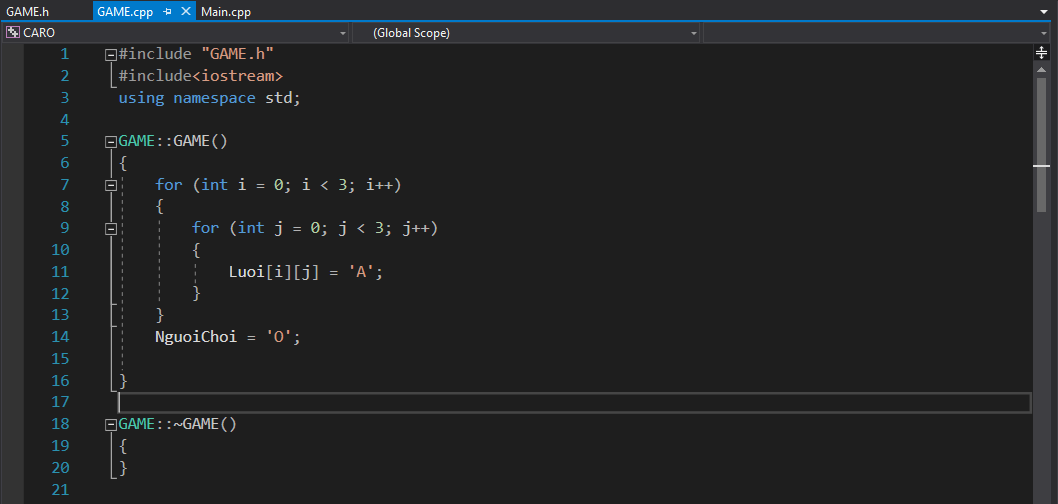
}

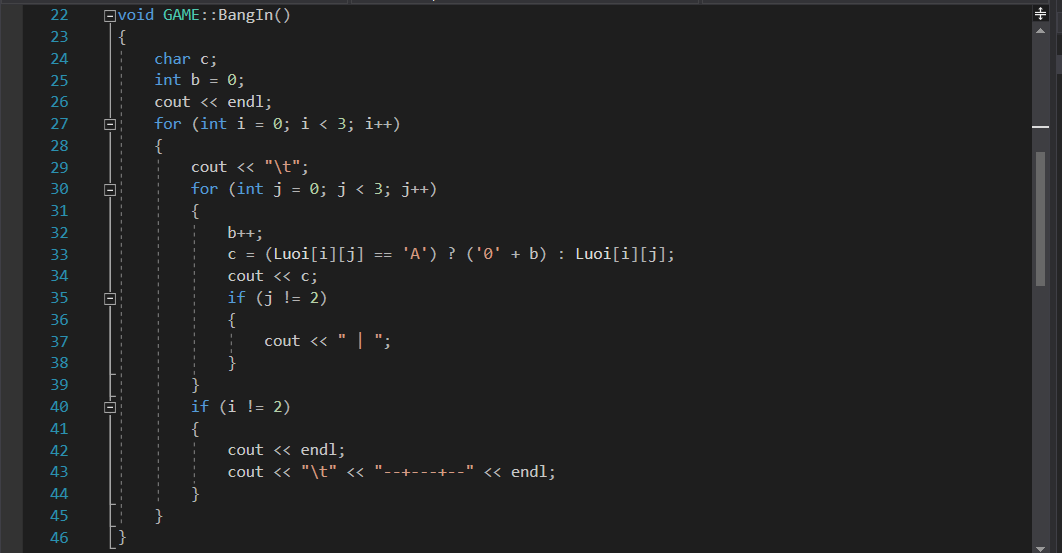
void GAME::CapNhatBangIn(int myMove)

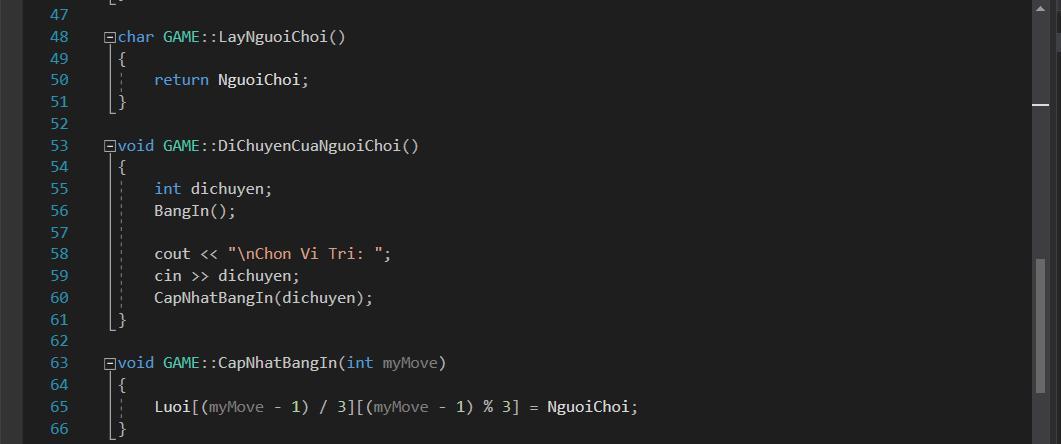
{

Luoi[(myMove - 1) / 3][(myMove - 1) % 3] = NguoiChoi;

}







Ở file Main.cpp, ta thêm các dòng code:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

GAME g1;

//g1.BangIn();

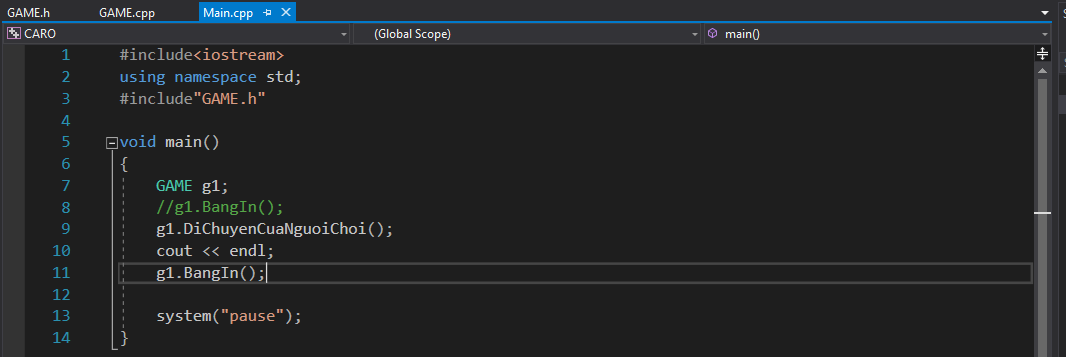
g1.DiChuyenCuaNguoiChoi(); // Dùng (.) để gọi lại hàm DiChuyenCuaNguoiChoi

cout << endl;

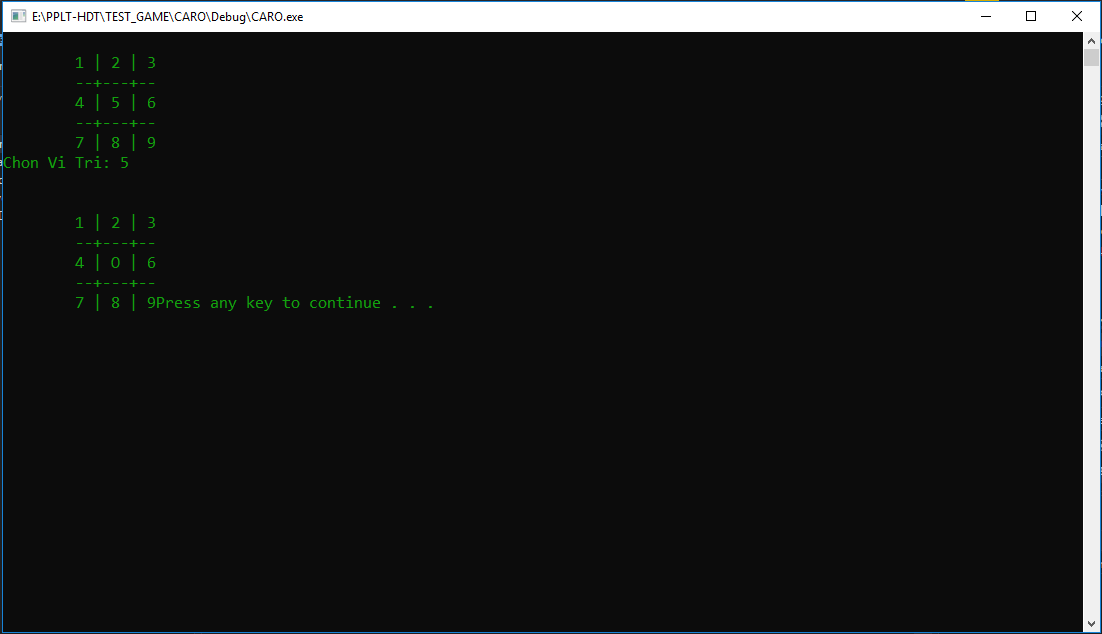
g1.BangIn();

system("pause");

}



KẾT QUẢ:



Tiếp theo, chúng ta sẽ cài đặt code cho 2 người chơi có thể đánh qua lại với nhau:

Ở file GAME.h, ta thêm các dòng code:

#pragma once

class GAME

{

private:

char NguoiChoi;

char Luoi[3][3];

int soLuotDiChuyen;

public:

GAME();

~GAME();

void BangIn();

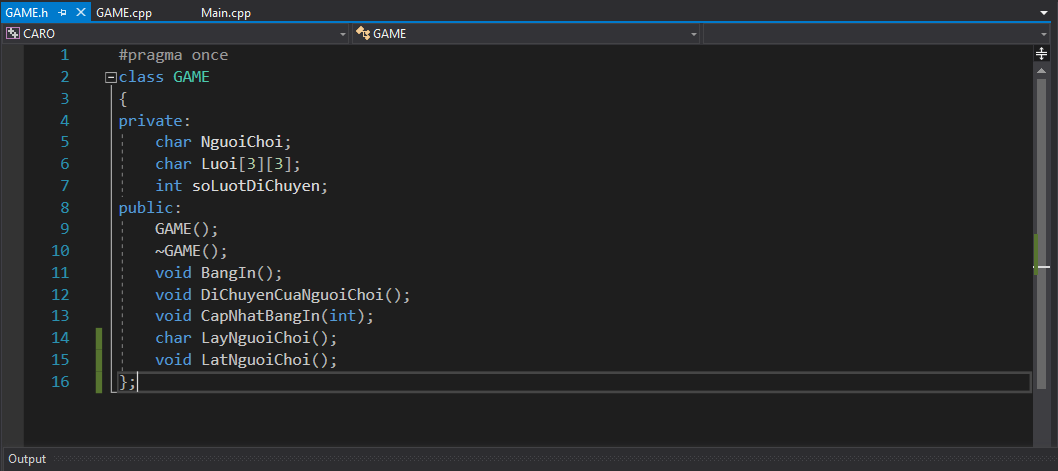
void DiChuyenCuaNguoiChoi();

void CapNhatBangIn(int);

char LayNguoiChoi();

void LatNguoiChoi(); // Tạo hàm lật người chơi để thấy được X và O

};



Ở file GAME.cpp, ta có các dòng code:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A';

}

}

NguoiChoi = 'O';

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j];

cout << c;

if (j != 2)

{

cout << " | ";

}

}

if (i != 2)

{

cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}

char GAME::LayNguoiChoi()

{

return NguoiChoi;

}

void GAME::DiChuyenCuaNguoiChoi()

{

int dichuyen;

BangIn();

LatNguoiChoi();

cout << "\nChon Vi Tri: ";

cin >> dichuyen;

CapNhatBangIn(dichuyen);

}

void GAME::CapNhatBangIn(int myMove)

{

Luoi[(myMove - 1) / 3][(myMove - 1) % 3] = NguoiChoi;

}

void GAME::LatNguoiChoi()

{

if (NguoiChoi == 'O') // Điều kiện xác định người chơi là X hay là O

{

NguoiChoi = 'X';

}

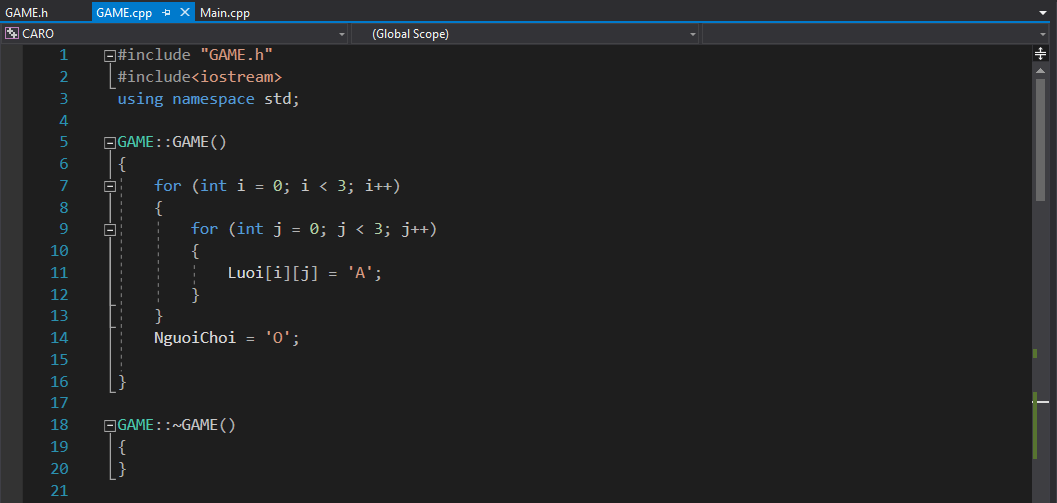
else

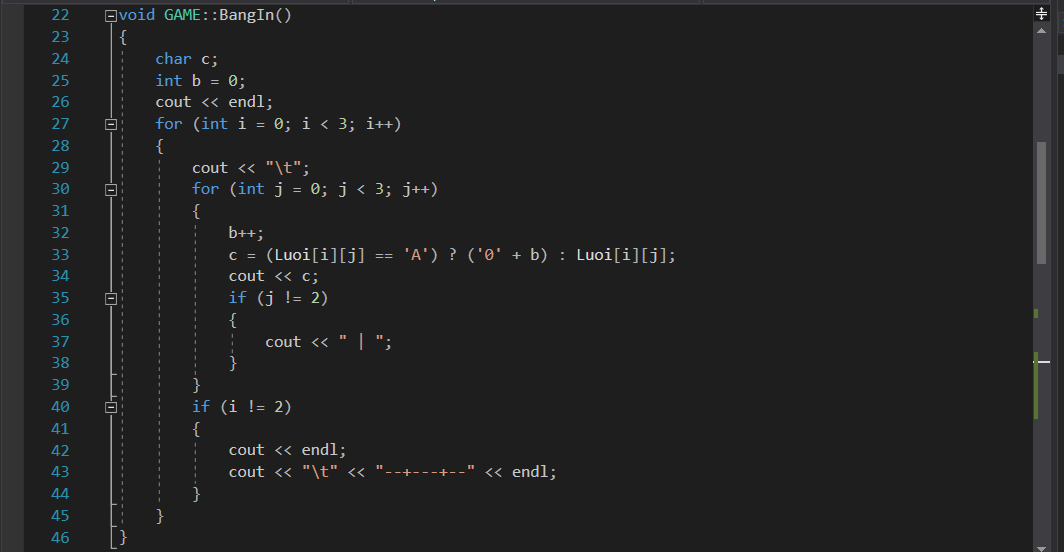
{

NguoiChoi = 'O';

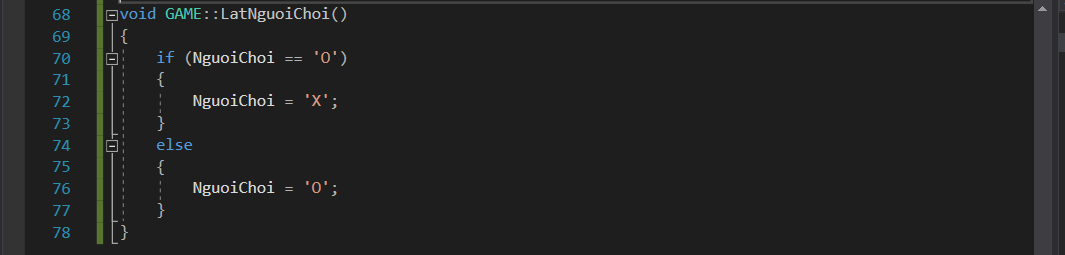
}

}









Ở file Main.cpp, ta có các dòng code:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

GAME g1;

//g1.BangIn();

while (1) // Cho vòng lặp bằng 1 để nó lặp vô tận

{

g1.DiChuyenCuaNguoiChoi(); // Dùng (.) để gọi lại hàm DCCNC thực thi

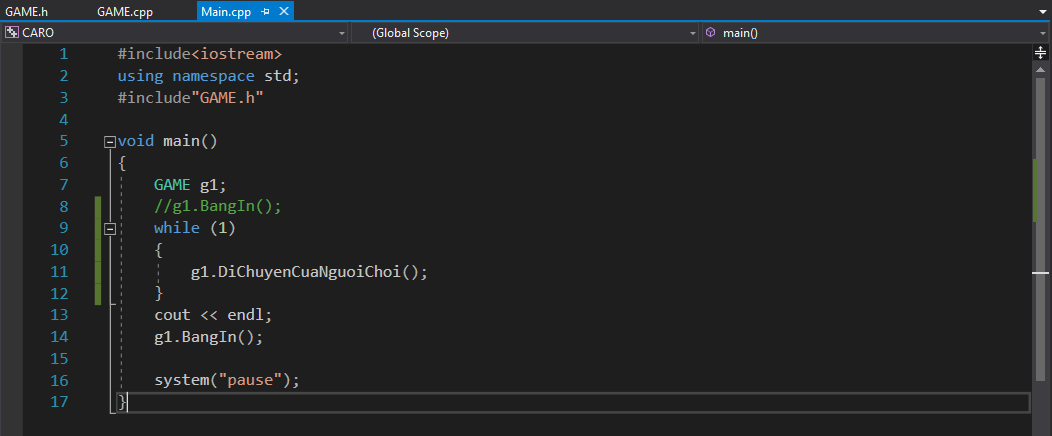
}

cout << endl;

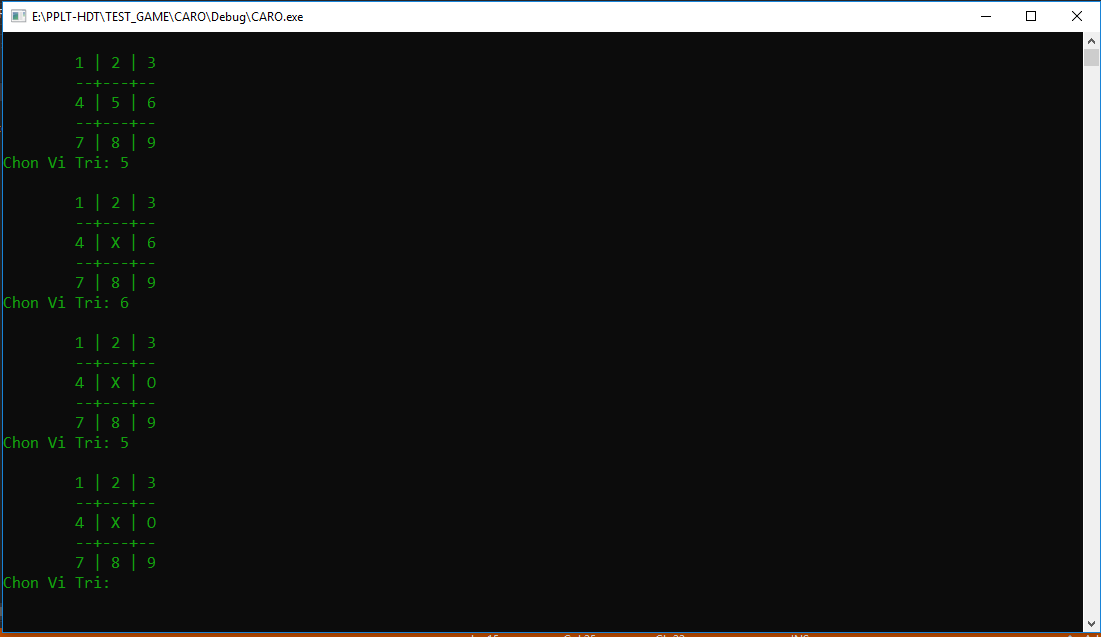
g1.BangIn();

system("pause");

}



KẾT QUẢ:



Vậy là chúng ta đã đánh được lên bàn cờ caro. Nhưng khuyết điểm ở đây là khi bạn lập lại một vị trí đã đánh thì nó sẽ ghi đè lên và chuyển sang người chơi đối diện (Như hình bên trên). Như vậy, là không thỏa đáng. Vì vậy, chúng ta sẽ có bước tiếp theo…

\*\*\* END NGƯỜI CHƠI \*\*\*

\*\*\* START VỊ TRÍ \*\*\*

Và Bước tiếp theo, chúng ta sẽ kiểm tra các vị trí để có thể đánh vào một cách hợp lệ.

Ở file GAME.h, ta có các dòng code:

#pragma once

class GAME

{

private:

char NguoiChoi;

char Luoi[3][3];

int soLuotDiChuyen;

public:

GAME();

~GAME();

void BangIn();

void DiChuyenCuaNguoiChoi();

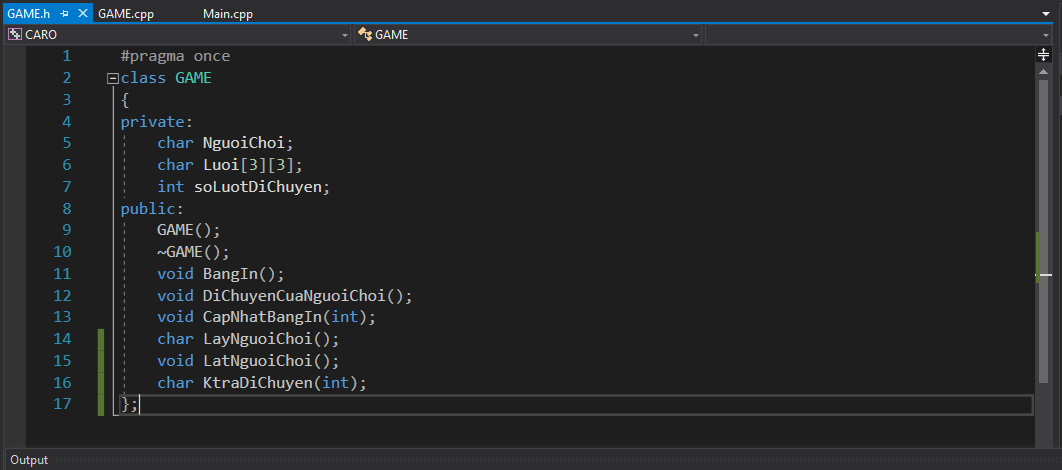
void CapNhatBangIn(int);

char LayNguoiChoi();

void LatNguoiChoi();

char KtraDiChuyen(int); // Tạo hàm ktra thuộc kiểu char với tham số truyền vào thuộc kiểu int

};



Ở file GAME.cpp, ta sẽ có các dòng code:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A';

}

}

NguoiChoi = 'O';

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j];

cout << c;

if (j != 2)

{

cout << " | ";

}

}

if (i != 2)

{

cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}

char GAME::LayNguoiChoi()

{

return NguoiChoi;

}

void GAME::DiChuyenCuaNguoiChoi()

{

int dichuyen;

BangIn();

LatNguoiChoi();

cout << endl;

cout << "\nNguoi Choi Hien Tai: " << NguoiChoi << endl;

cout << "\nChon Vi Tri: ";

cin >> dichuyen;

char valid = KtraDiChuyen(dichuyen);

while (valid != 'Y')

{

cout << "Di chuyen khong hop le !!!";

cout << "\nChon vi tri khac: ";

cin >> dichuyen;

valid = KtraDiChuyen(dichuyen);

}

CapNhatBangIn(dichuyen);

}

void GAME::CapNhatBangIn(int myMove)

{

Luoi[(myMove - 1) / 3][(myMove - 1) % 3] = NguoiChoi;

}

void GAME::LatNguoiChoi()

{

if (NguoiChoi == 'O')

{

NguoiChoi = 'X';

}

else

{

NguoiChoi = 'O';

}

}

char GAME::KtraDiChuyen(int myMove)

{

char gridVal = Luoi[(myMove - 1) / 3][(myMove - 1) % 3];

if (myMove > 0 && myMove < 10 && gridVal == 'A')

{

return 'Y';

}

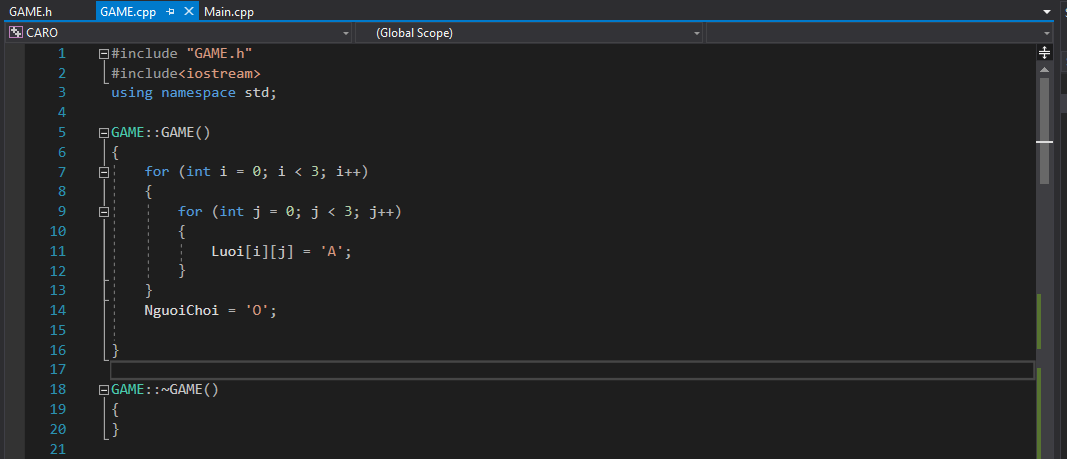
else

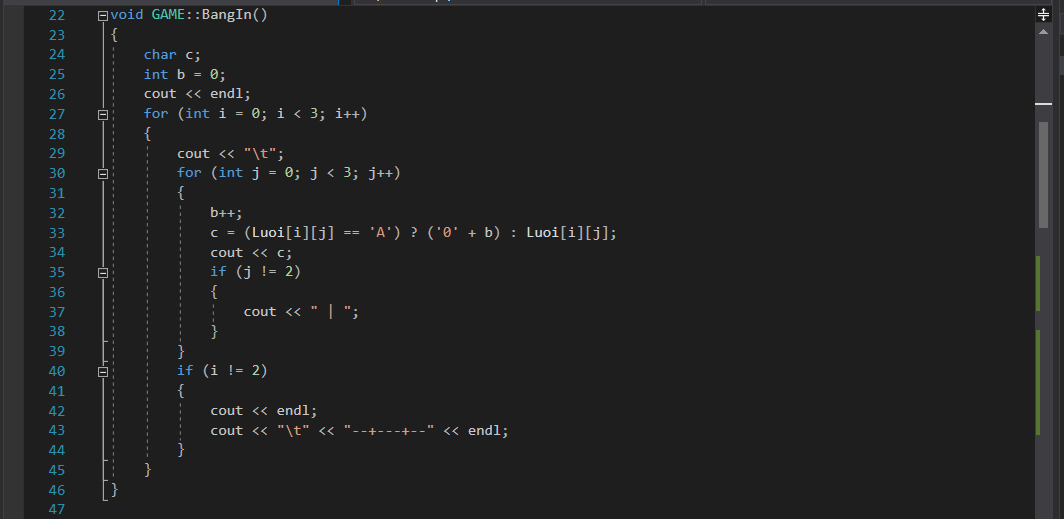
{

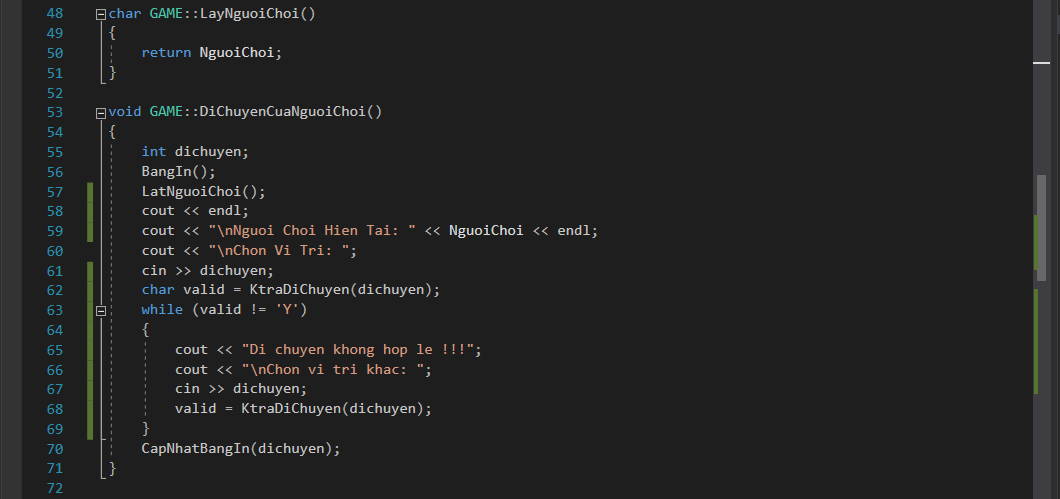
return 'N';

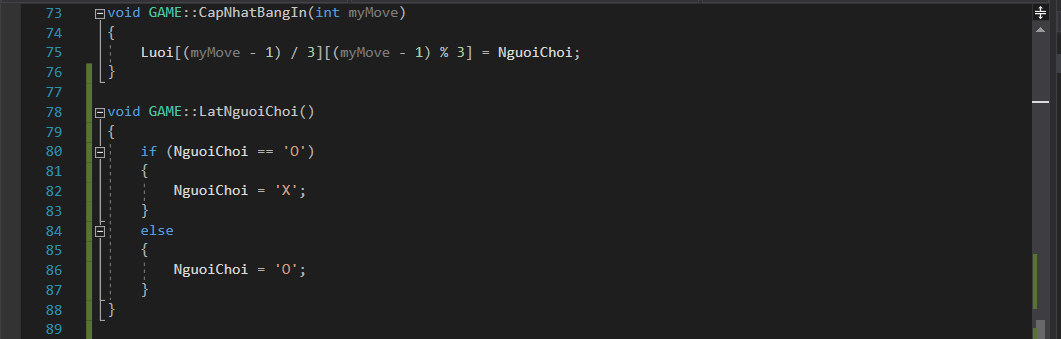
}

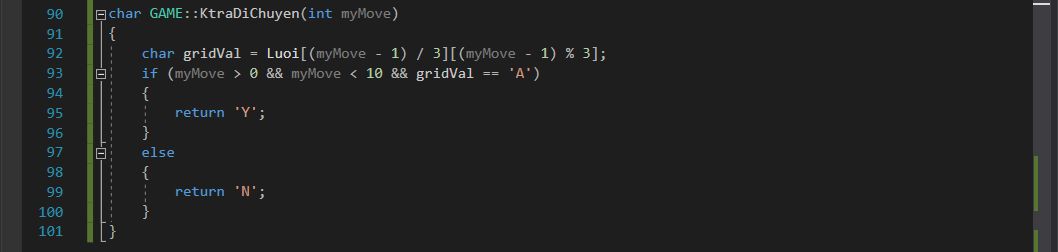
}











Ở file Main.cpp, ta có:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

GAME g1;

//g1.BangIn();

while (1)

{

g1.DiChuyenCuaNguoiChoi();

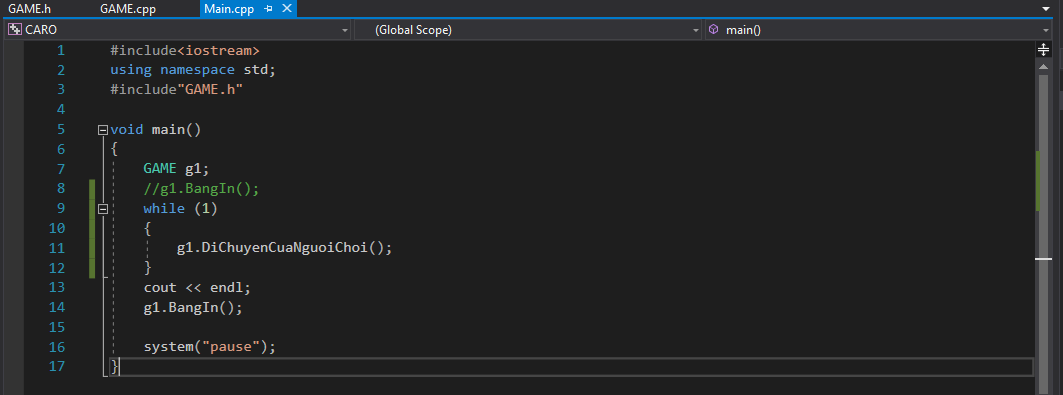
}

cout << endl;

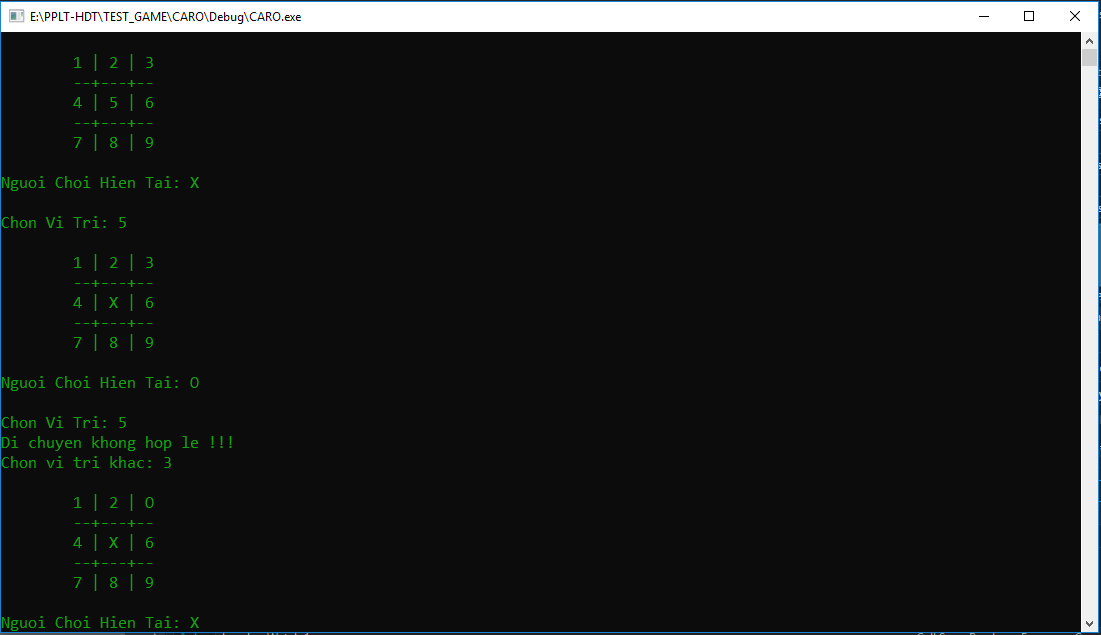
g1.BangIn();

system("pause");

}



KẾT QUẢ:



Ok, như vậy là chúng ta đã hoàn thành việc kiểm tra vị trí đánh hiện hành của người chơi.

\*\*\* END VỊ TRÍ \*\*\*

\*\*\* START KẾT QUẢ \*\*\*

Và cuối cùng, điều không thể thiếu trong quá trình chơi caro hay bất kì một game nào khác đó là phân định thắng thua. Chúng ta sẽ:

Ở file GAME.h, ta có các dòng code:

#pragma once

class GAME

{

private:

char NguoiChoi;

char Luoi[3][3];

int soLuotDiChuyen;

public:

GAME();

~GAME();

void BangIn();

void DiChuyenCuaNguoiChoi();

void CapNhatBangIn(int);

char LayNguoiChoi();

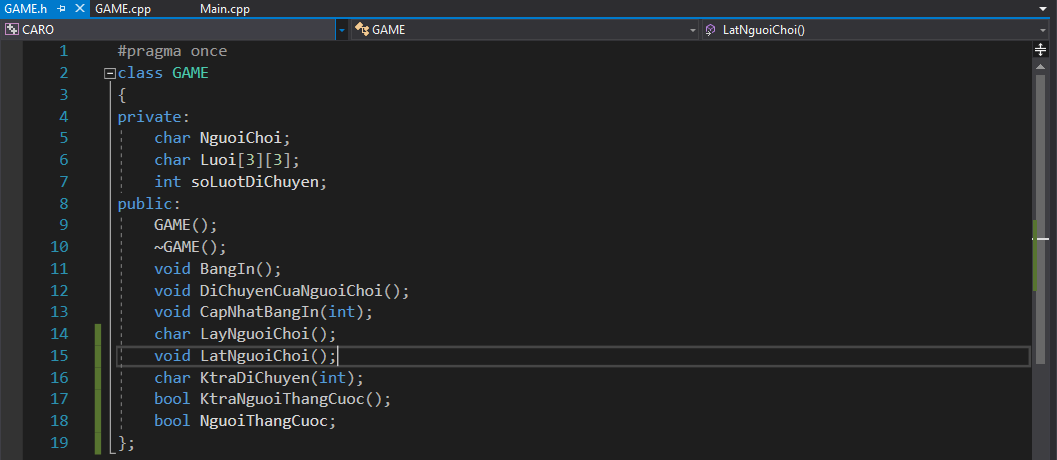
void LatNguoiChoi();

char KtraDiChuyen(int);

bool KtraNguoiThangCuoc(); // Tạo hàm Ktra người thắng thuộc kiểu bool

bool NguoiThangCuoc; // Tạo biến Người thắng cuộc kiểu bool

};



Ở file GAME.cpp, ta có các dòng code:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A';

}

}

NguoiChoi = 'O';

NguoiThangCuoc = false; // Setup không có người thắng mặc định

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j];

cout << c;

if (j != 2)

{

cout << " | ";

}

}

if (i != 2)

{

cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}

char GAME::LayNguoiChoi()

{

return NguoiChoi;

}

void GAME::DiChuyenCuaNguoiChoi()

{

int dichuyen;

BangIn();

LatNguoiChoi();

cout << endl;

cout << "\nNguoi Choi Hien Tai: " << NguoiChoi << endl;

cout << "\nChon Vi Tri: ";

cin >> dichuyen;

char valid = KtraDiChuyen(dichuyen);

while (valid != 'Y')

{

cout << "Di chuyen khong hop le !!!";

cout << "\nChon vi tri khac: ";

cin >> dichuyen;

valid = KtraDiChuyen(dichuyen);

}

CapNhatBangIn(dichuyen);

NguoiThangCuoc = KtraNguoiThangCuoc();

}

void GAME::CapNhatBangIn(int myMove)

{

Luoi[(myMove - 1) / 3][(myMove - 1) % 3] = NguoiChoi;

}

void GAME::LatNguoiChoi()

{

if (NguoiChoi == 'O')

{

NguoiChoi = 'X';

}

else

{

NguoiChoi = 'O';

}

}

char GAME::KtraDiChuyen(int myMove)

{

char gridVal = Luoi[(myMove - 1) / 3][(myMove - 1) % 3];

if (myMove > 0 && myMove < 10 && gridVal == 'A')

{

return 'Y';

}

else

{

return 'N';

}

}

bool GAME::KtraNguoiThangCuoc()

{

int LastSq;

int LastSq1;

// Ktra dong

for (int i = 0; i < 3; i++)

{

LastSq = 0; // Cho số kí hiệu ban đầu là 0

for (int j = 0; j < 3; j++)

{

if (Luoi[i][j] == NguoiChoi) // Lưới[i][j] 🡪 dòng

LastSq++; // Số kí hiệu tăng lên sau mỗi lần lặp

}

if (LastSq == 3) // Nếu đủ số kí hiệu = 3 thì trả về true

return true;

}

// Ktra cot

for (int i = 0; i < 3; i++) // TƯƠNG TỰ KTRA DÒNG

{

LastSq = 0;

for (int j = 0; j < 3; j++)

{

if (Luoi[j][i] == NguoiChoi) // Lưới[j][i] 🡪 cột

LastSq++;

}

if (LastSq == 3)

return true;

}

// Ktra duong cheo

int y = 0;

int c = 2;

LastSq = 0;

LastSq1 = 0;

for (int i = 0; i < 3; i++, y++, c--) // Vòng lặp nhiều biến i , y , c

{

if (Luoi[i][y] == NguoiChoi) // Điều kiện ktra đường chéo chính (đcc)

LastSq++; // Tăng biến đếm sq lên 1

if (Luoi[i][c] == NguoiChoi) // Điều kiện ktra đường chéo phụ (đcp)

LastSq1++; // Tăng biến đếm sq1 lên 1

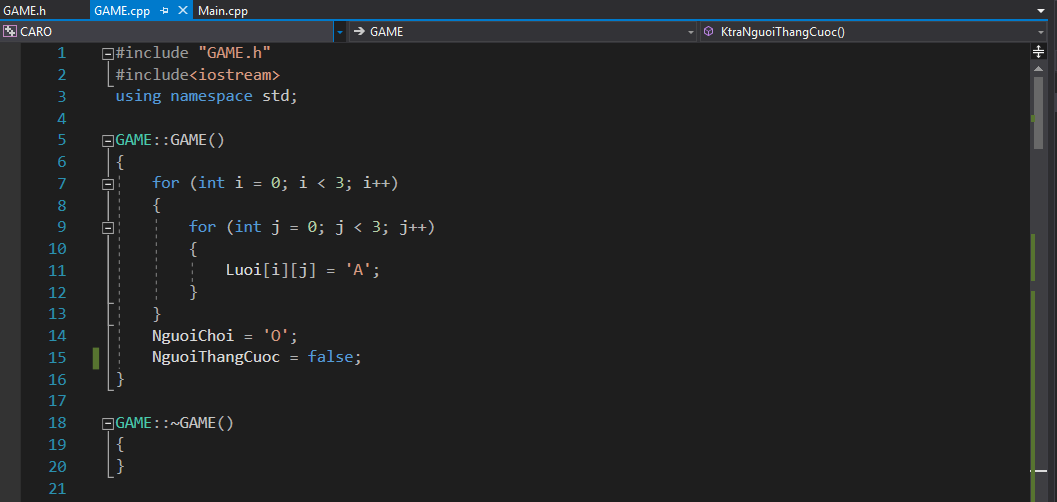
if (LastSq == 3 || LastSq1 == 3) // Điều kiện nếu đcc hoặc đcp bằng 3

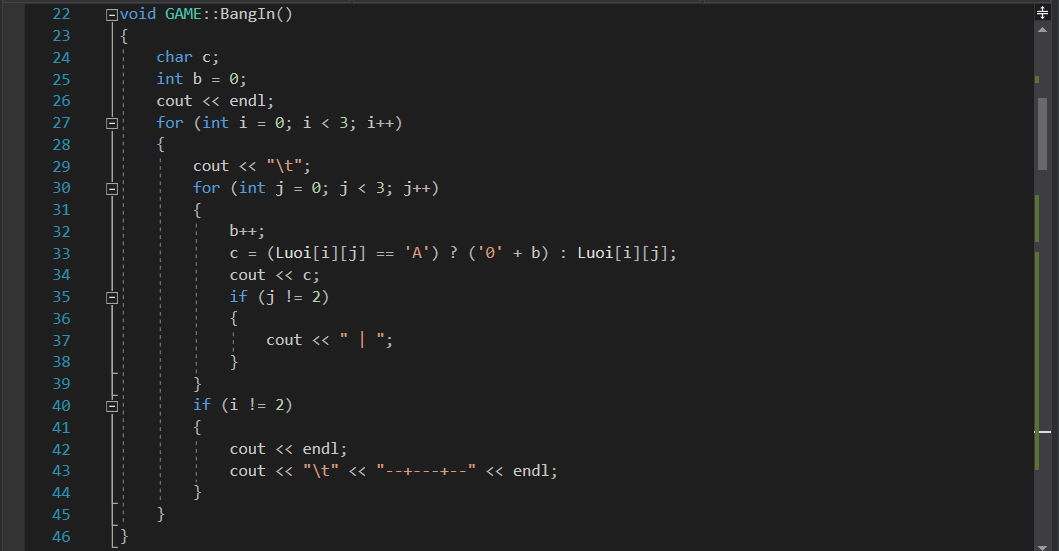
return true; // trả về giá trị true cho người chơi

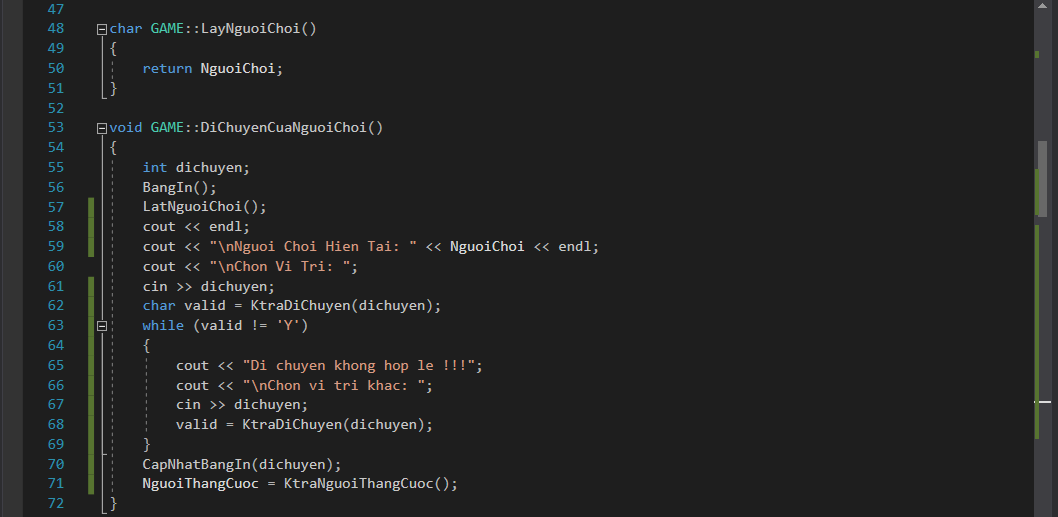
}

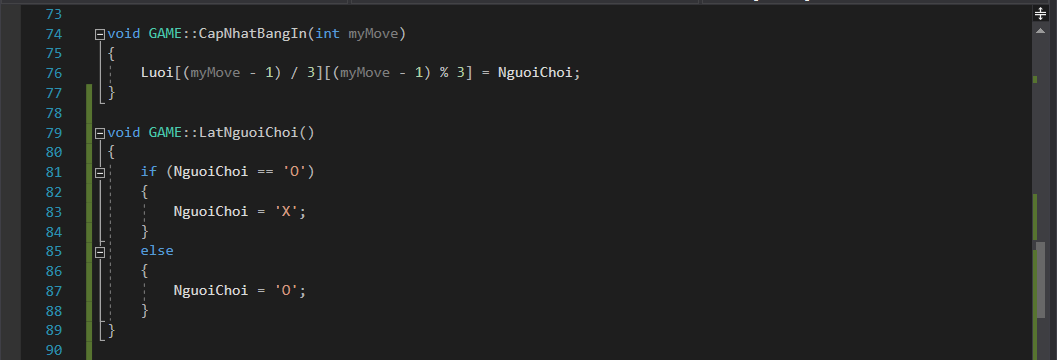
return false;

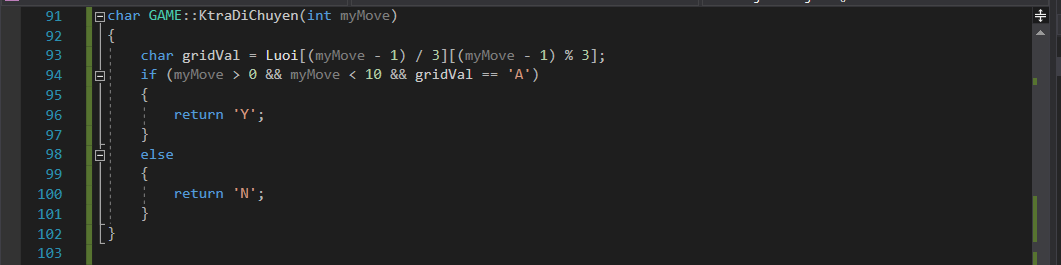
}

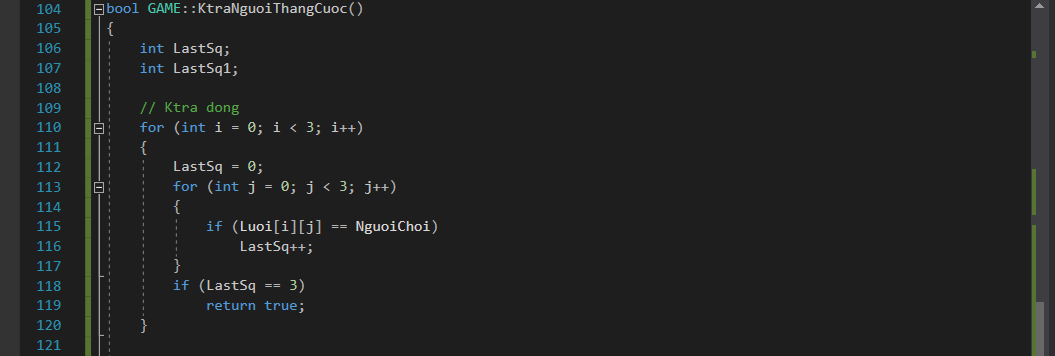


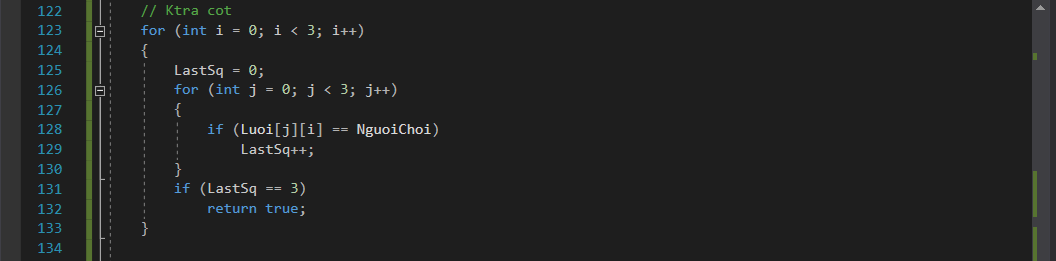


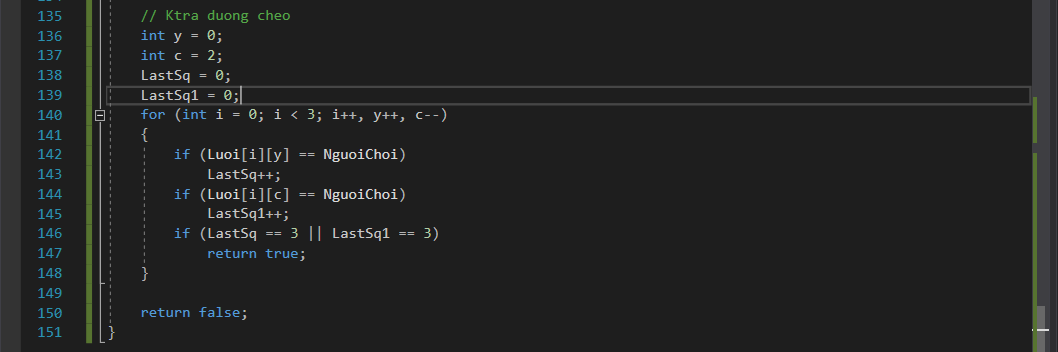












Ở file Main.cpp, ta có:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

GAME g1;

//g1.BangIn();

while (!g1.NguoiThangCuoc) // Nếu không có người thắng thì cho người chơi đánh tiếp

{

g1.DiChuyenCuaNguoiChoi(); // Dùng (.) để gọi lại hàm Di chuyển của người chơi

}

char player = g1.LayNguoiChoi();

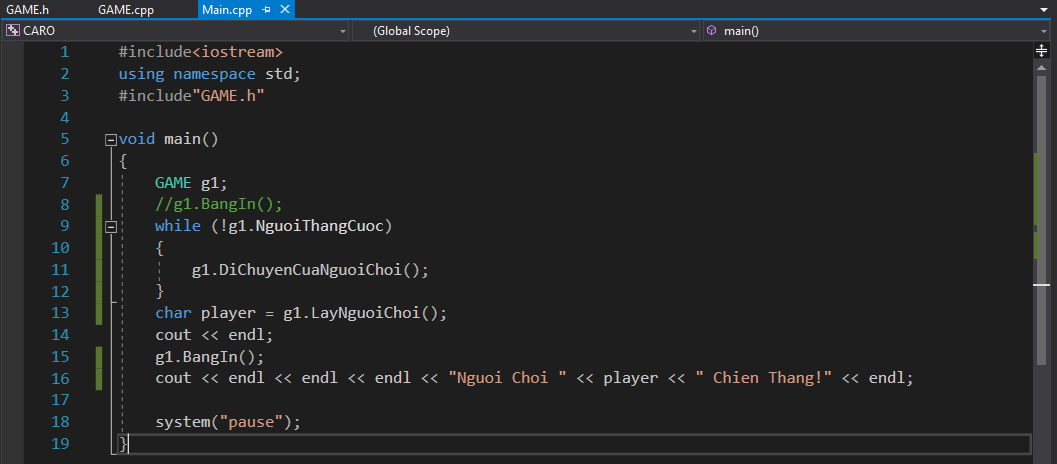
cout << endl;

g1.BangIn();

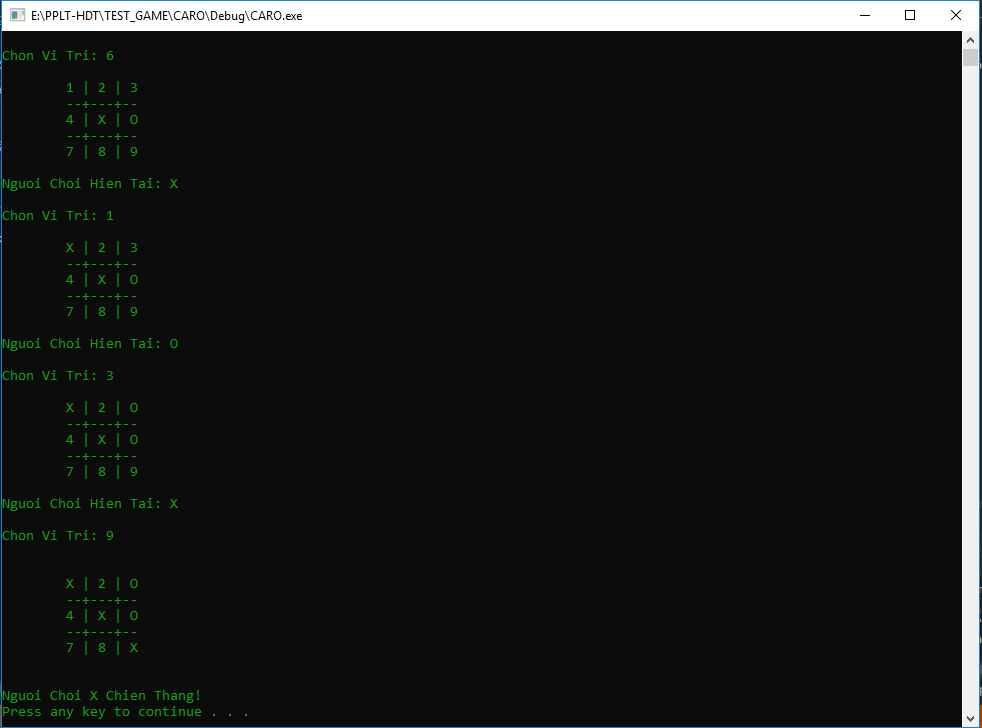
cout << endl << endl << endl << "Nguoi Choi " << player << " Chien Thang!" << endl;

system("pause");

}



KẾT QUẢ:



Cuối cùng để cho nó màu mè xíu, thì mình sẽ thêm các dòng code trong file Main.cpp là:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

int newGame;

cout << "\n============ CHAO MUNG CAC BAN DEN VOI GAME CARO 3X3 ============";

cout << "\nNhan phim 1 de bat dau tro choi" << endl;

cout << "\nLua Chon Cua Ban: ";

cin >> newGame;

while (newGame == 1)

{

GAME g1;

//g1.BangIn();

while (!g1.NguoiThangCuoc)

{

g1.DiChuyenCuaNguoiChoi();

}

char player = g1.LayNguoiChoi();

cout << endl;

g1.BangIn();

cout << endl << endl << endl << "Nguoi Choi " << player << " Chien Thang!" << endl;

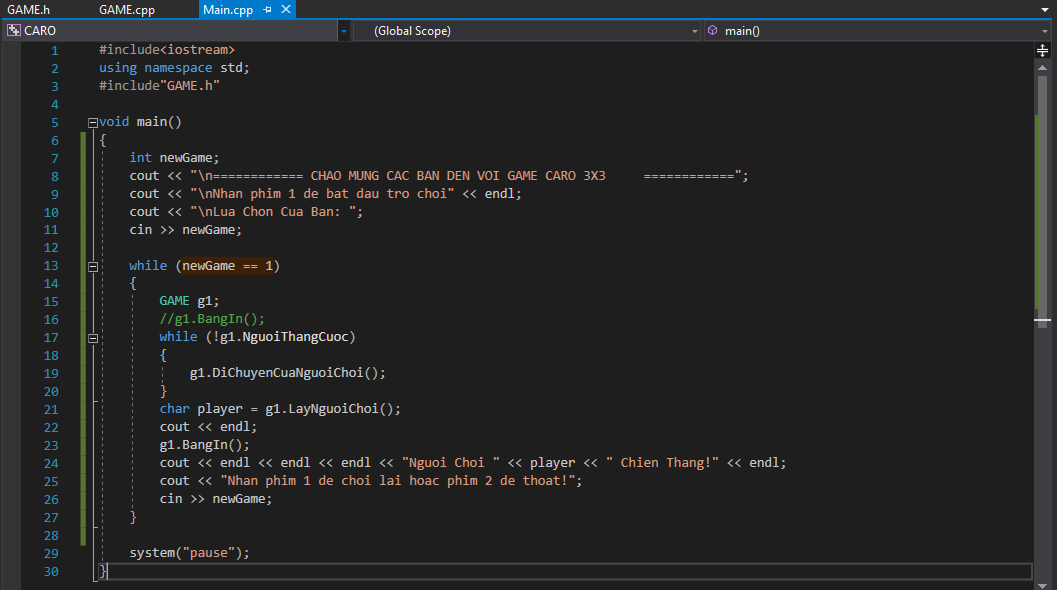
cout << "Nhan phim 1 de choi lai hoac phim 2 de thoat!";

cin >> newGame;

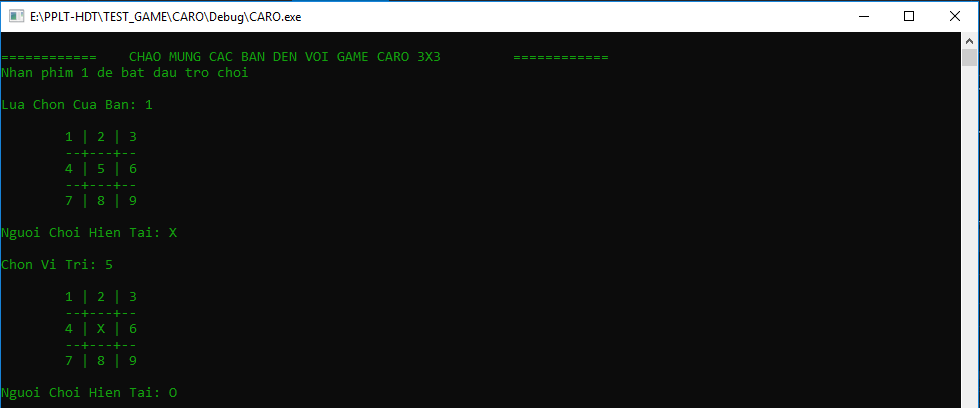
}

system("pause");

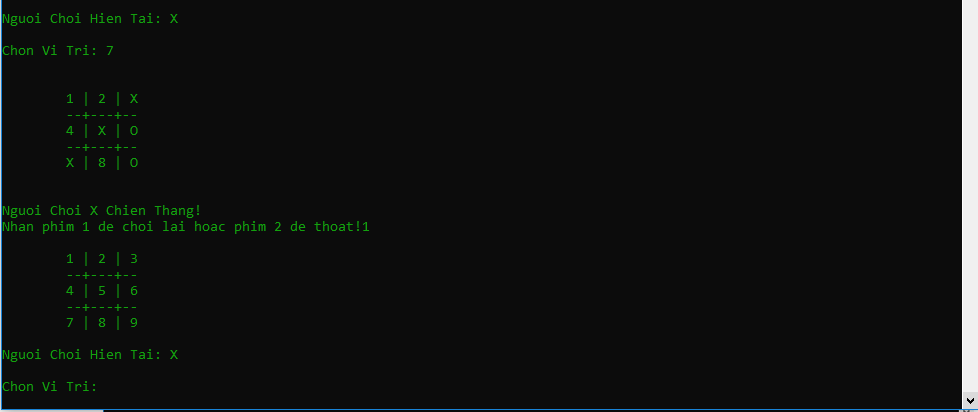
}



KẾT QUẢ:







Cuối cùng, Chúng ta sẽ cài đặt việc kiểm tra kết quả 2 người hòa nhau.

Ở file GAME.h, ta có các dòng code:

#pragma once

class GAME

{

private:

char NguoiChoi;

char Luoi[3][3];

int soLuotDiChuyen; // Biến để đếm số lượt di chuyển của người chơi

public:

GAME();

~GAME();

void BangIn();

void DiChuyenCuaNguoiChoi();

void CapNhatBangIn(int);

char LayNguoiChoi();

void LatNguoiChoi();

char KtraDiChuyen(int);

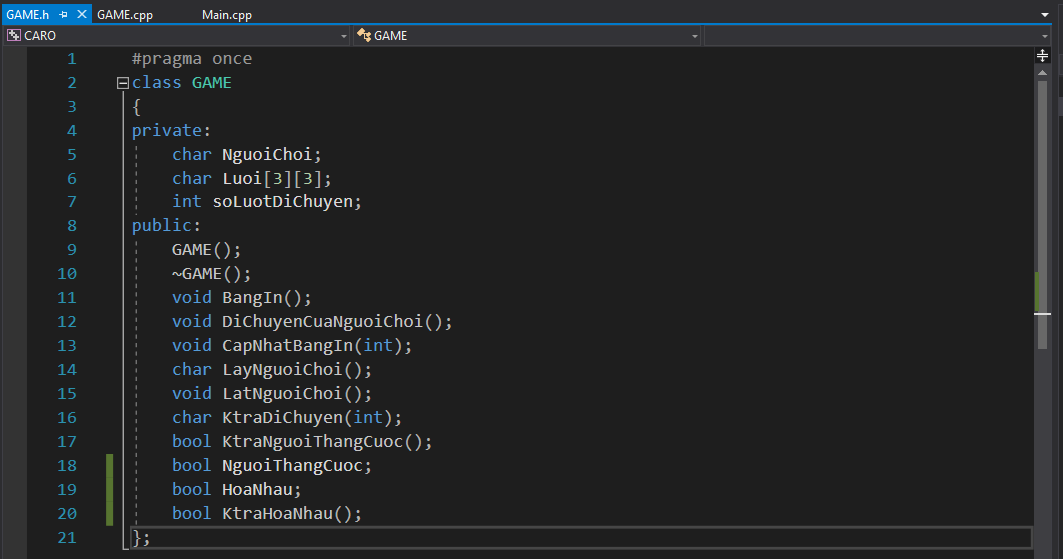
bool KtraNguoiThangCuoc();

bool NguoiThangCuoc;

bool HoaNhau; // Tạo biến HoaNhau kiểu bool

bool KtraHoaNhau(); // Tạo 1 hàm Ktra hòa nhau kiểu bool

};



Ở file GAME.cpp, ta có các dòng code:

#include "GAME.h"

#include<iostream>

using namespace std;

GAME::GAME()

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

Luoi[i][j] = 'A';

}

}

NguoiChoi = 'O';

NguoiThangCuoc = false;

HoaNhau = false; // Gán giá trị cho HoaNhau là false

}

GAME::~GAME()

{

}

void GAME::BangIn()

{

char c;

int b = 0;

cout << endl;

for (int i = 0; i < 3; i++)

{

cout << "\t";

for (int j = 0; j < 3; j++)

{

b++;

c = (Luoi[i][j] == 'A') ? ('0' + b) : Luoi[i][j];

cout << c;

if (j != 2)

{

cout << " | ";

}

}

if (i != 2)

{

cout << endl;

cout << "\t" << "--+---+--" << endl;

}

}

}

char GAME::LayNguoiChoi()

{

return NguoiChoi;

}

void GAME::DiChuyenCuaNguoiChoi()

{

int dichuyen;

BangIn();

LatNguoiChoi();

cout << endl;

cout << "\nNguoi Choi Hien Tai: " << NguoiChoi << endl;

cout << "\nChon Vi Tri: ";

cin >> dichuyen;

char valid = KtraDiChuyen(dichuyen);

while (valid != 'Y')

{

cout << "Di chuyen khong hop le !!!";

cout << "\nChon vi tri khac: ";

cin >> dichuyen;

valid = KtraDiChuyen(dichuyen);

}

CapNhatBangIn(dichuyen);

NguoiThangCuoc = KtraNguoiThangCuoc();

if (!NguoiThangCuoc) // Nếu không ai thắng thì sẽ vào ktra hòa

{

if (KtraHoaNhau()) // Ktra 2 người hòa nhau

{

HoaNhau = true;

cout << "2 nguoi hoa nhau." << endl;

return;

}

}

}

void GAME::CapNhatBangIn(int myMove)

{

Luoi[(myMove - 1) / 3][(myMove - 1) % 3] = NguoiChoi;

}

void GAME::LatNguoiChoi()

{

if (NguoiChoi == 'O')

{

NguoiChoi = 'X';

}

else

{

NguoiChoi = 'O';

}

}

char GAME::KtraDiChuyen(int myMove)

{

char gridVal = Luoi[(myMove - 1) / 3][(myMove - 1) % 3];

if (myMove > 0 && myMove < 10 && gridVal == 'A')

{

return 'Y';

}

else

{

return 'N';

}

}

bool GAME::KtraNguoiThangCuoc()

{

int LastSq;

int LastSq1;

// Ktra dong

for (int i = 0; i < 3; i++)

{

LastSq = 0;

for (int j = 0; j < 3; j++)

{

if (Luoi[i][j] == NguoiChoi)

LastSq++;

}

if (LastSq == 3)

return true;

}

// Ktra cot

for (int i = 0; i < 3; i++)

{

LastSq = 0;

for (int j = 0; j < 3; j++)

{

if (Luoi[j][i] == NguoiChoi)

LastSq++;

}

if (LastSq == 3)

return true;

}

// Ktra duong cheo

int y = 0;

int c = 2;

LastSq = 0;

LastSq1 = 0;

for (int i = 0; i < 3; i++, y++, c--)

{

if (Luoi[i][y] == NguoiChoi)

LastSq++;

if (Luoi[i][c] == NguoiChoi)

LastSq1++;

if (LastSq == 3 || LastSq1 == 3)

return true;

}

return false;

}

bool GAME::KtraHoaNhau()

{

int count = 0; // Khởi tạo biến đếm bằng 0

for (int i = 0; i<3; i++)

{

for (int j = 0; j<3; j++)

{

if (Luoi[i][j] == 'X' || Luoi[i][j] == 'O')

count++; // Tăng biến đếm lên 1 sau mỗi lần lặp

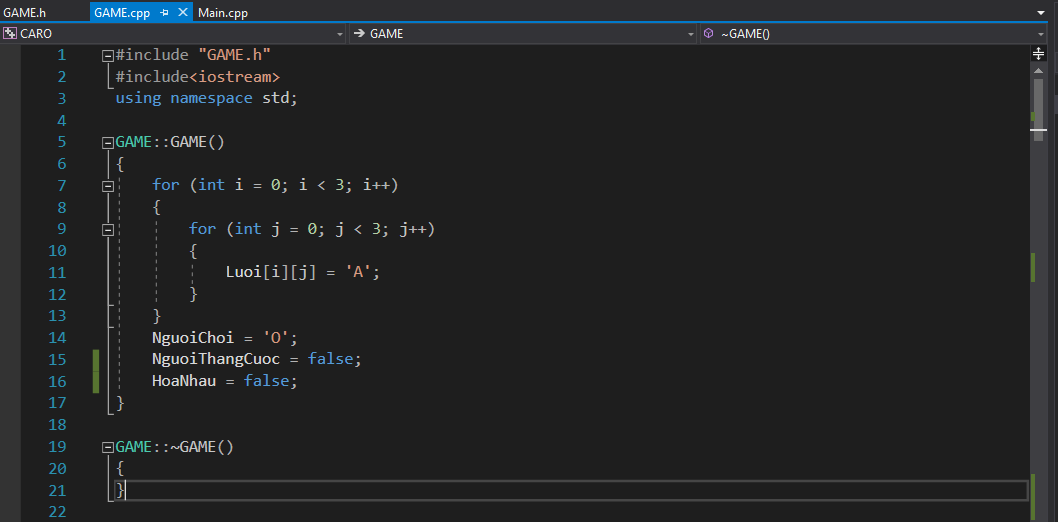
}

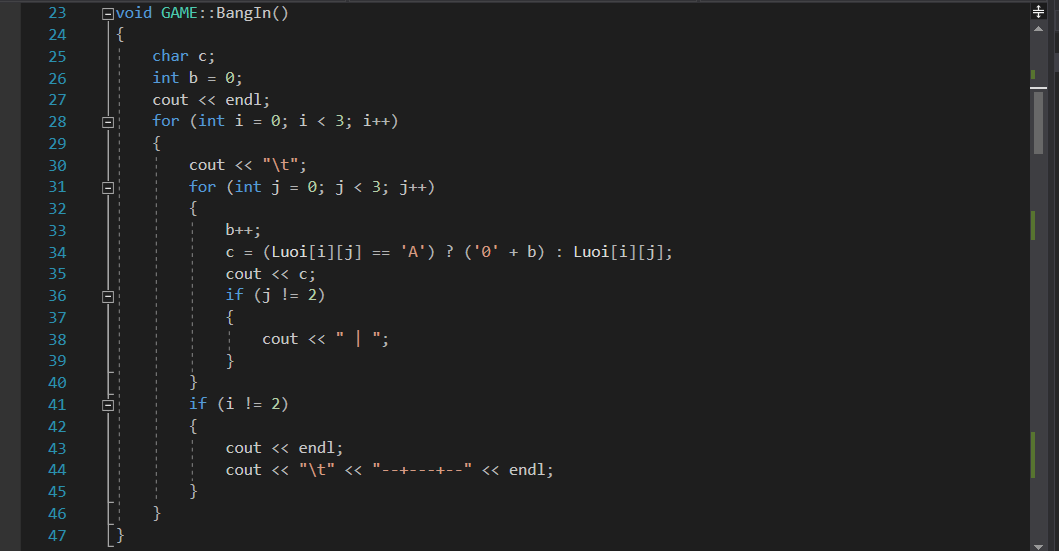
}

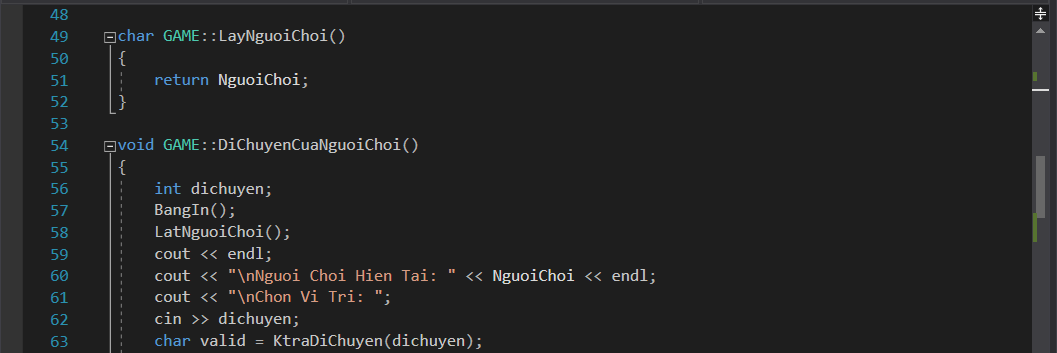
return (count == 9); // Gán kết quả trả về cho biến đếm bằng 9

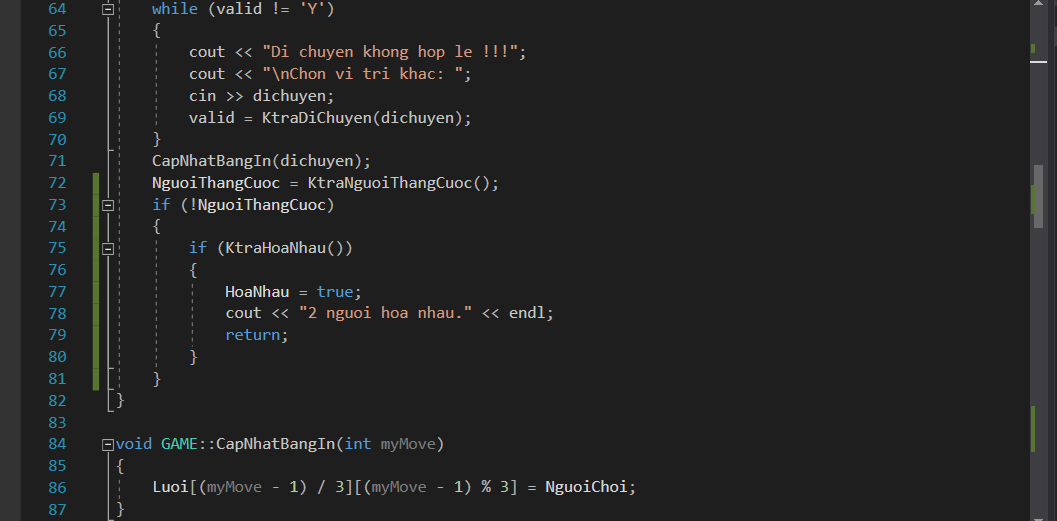
(Vì bàn cờ 3x3 thì có 9 vị trí để đánh)

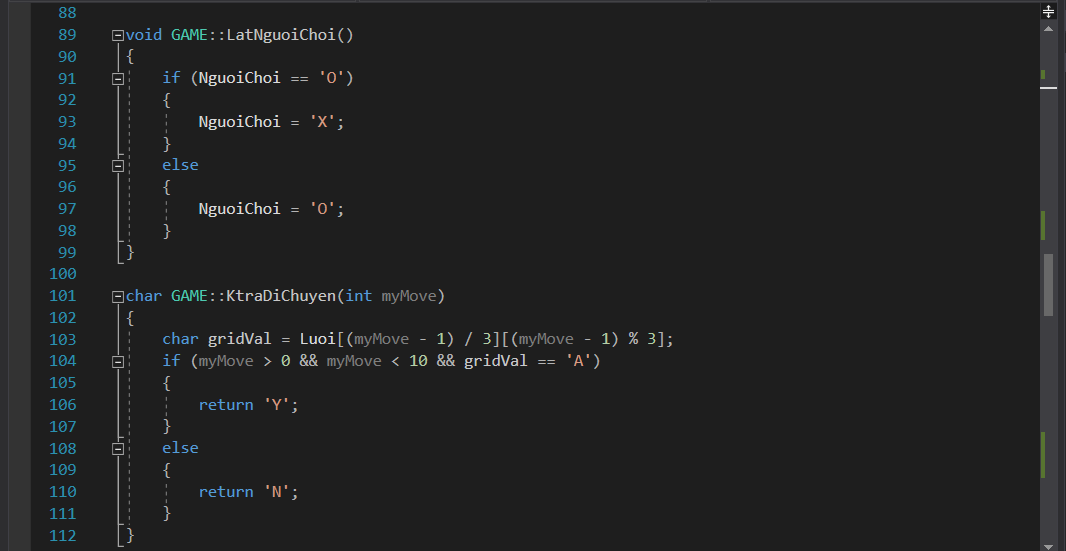
}

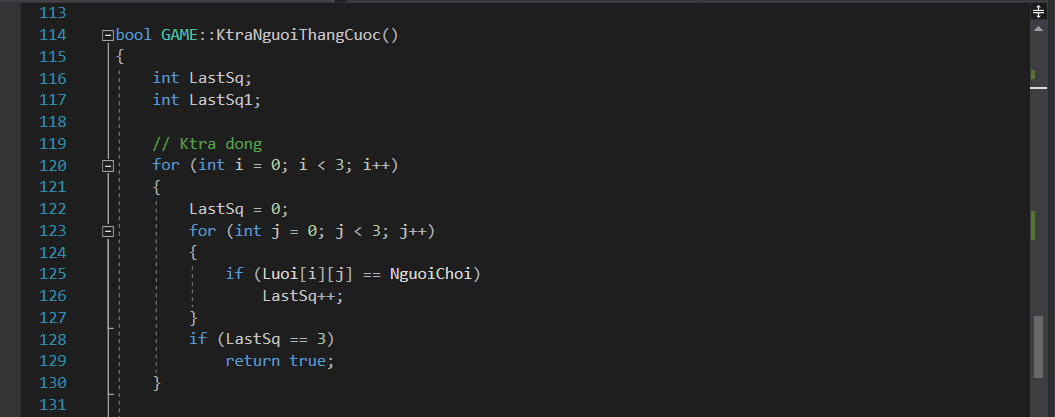


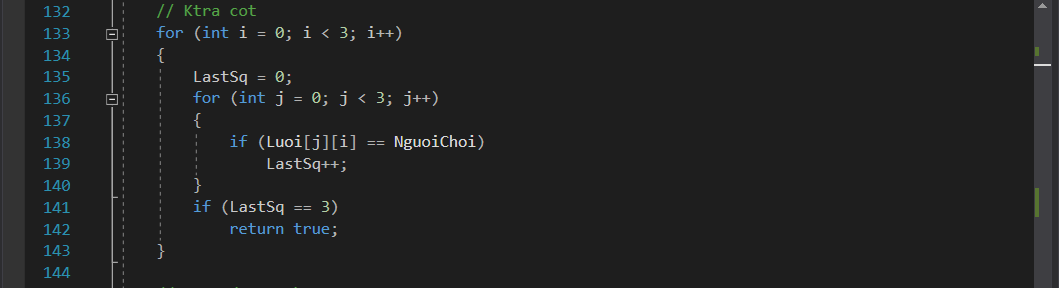


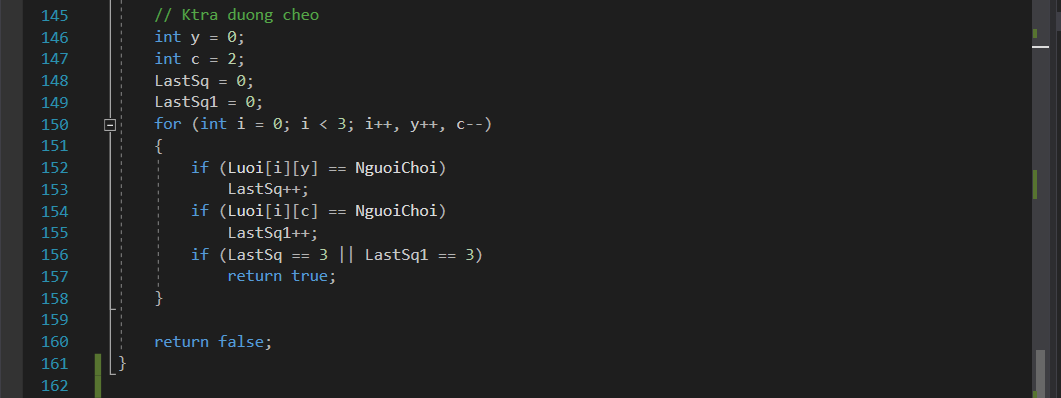


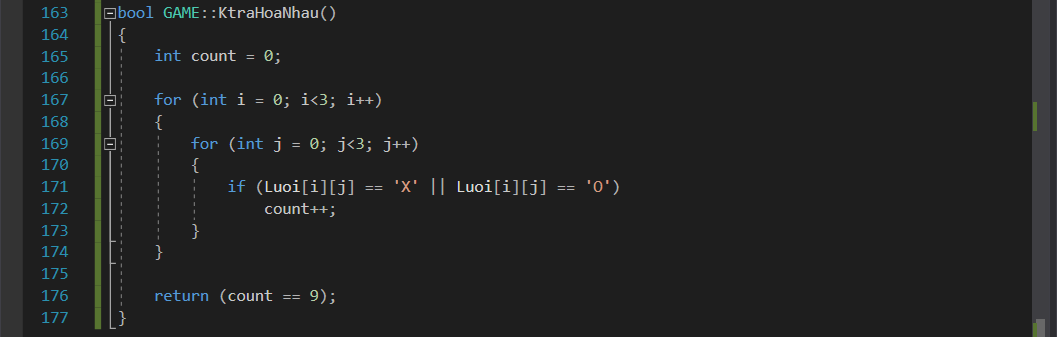












Ở file Main.cpp, ta có các dòng code:

#include<iostream>

using namespace std;

#include"GAME.h"

void main()

{

int newGame;

cout << "\n============ CHAO MUNG CAC BAN DEN VOI GAME CARO 3X3 ============";

cout << "\nNhan phim 1 de bat dau tro choi" << endl;

cout << "\nLua Chon Cua Ban: ";

cin >> newGame;

while (newGame == 1)

{

GAME g1;

//g1.BangIn();

while (!g1.NguoiThangCuoc && !g1.HoaNhau) // Thêm ĐK hòa nhau vào vòng lặp

{

g1.DiChuyenCuaNguoiChoi();

}

char player = g1.LayNguoiChoi();

cout << endl;

g1.BangIn();

if (g1.NguoiThangCuoc) // Nếu có người thắng thì thực hiện cậu lệnh bên trong

{

cout << endl << endl << endl << "Nguoi Choi " << player << " Chien Thang!" << endl;

}

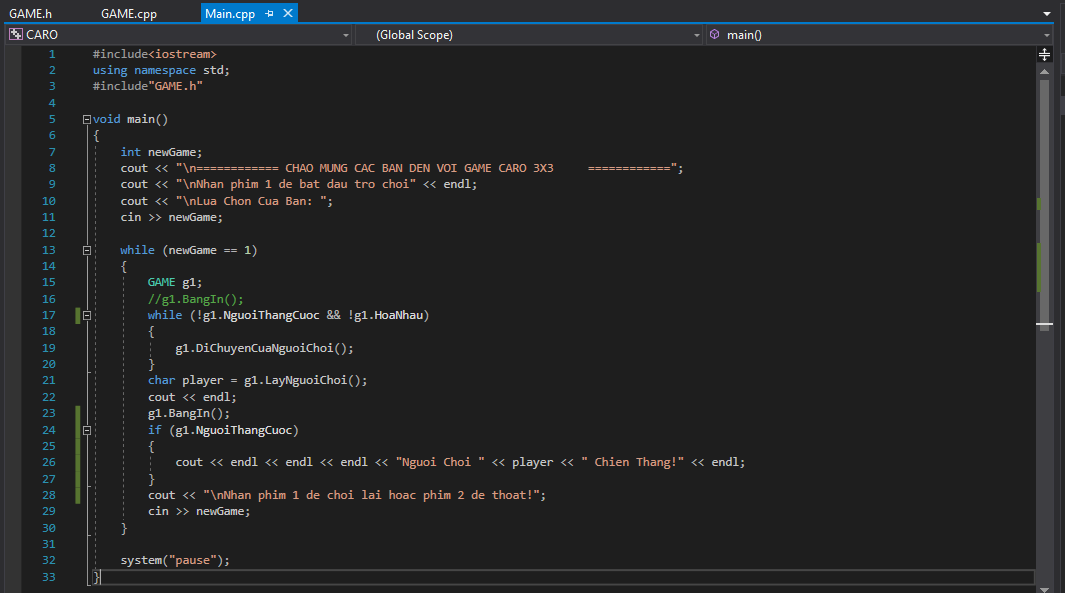
cout << "\nNhan phim 1 de choi lai hoac phim 2 de thoat!";

cin >> newGame;

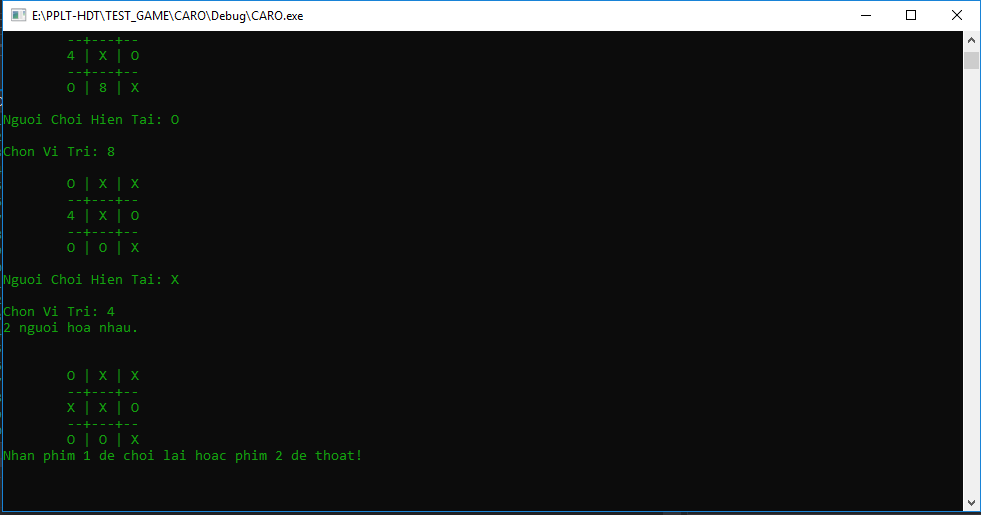
}

system("pause");

}



KẾT QUẢ:



\*\*\* END KẾT QUẢ \*\*\*

Ok. Chúng ta vừa hoàn thành game Caro dựa trên lập trình hướng đối tượng c++

Bây giờ thì cùng nhau trải nghiệm thôi nào !



**LINK FULL CODE:**

<https://drive.google.com/open?id=1EXBRnLRfyq5ajcWzwecR4MU7p88_0D8R>