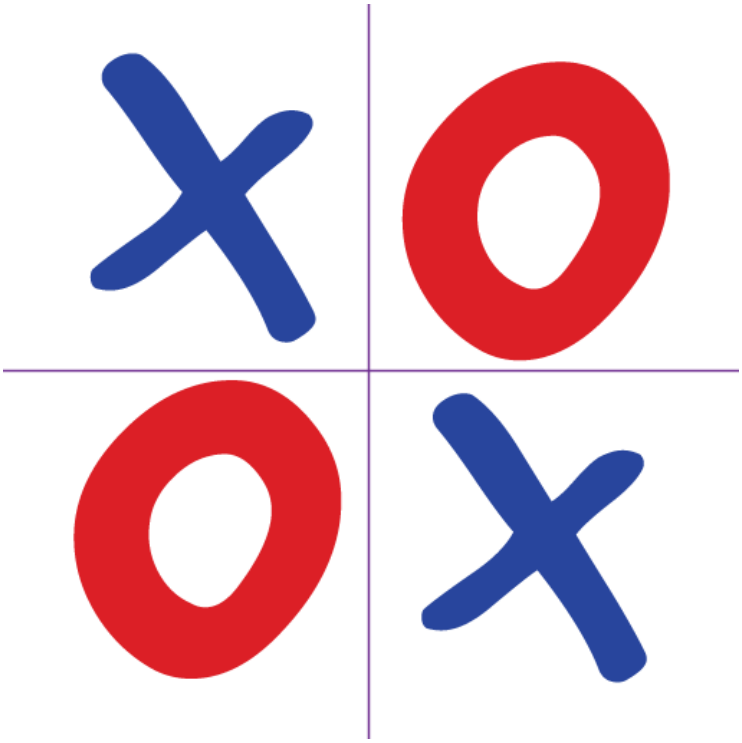
Đồ án Caro



Mục lục

I. Cách tổ chức dữ liệu:

- Các số liệu cần thiết cho trò chơi sẽ được lưu trong Data.h
- Các hàm dùng cho xử lý giao diện sẽ được viết trong file View.cpp và được tiền khai báo trong file View.h
    - Trong trường hợp mấy bạn thắc mắc là sao mình cần cái file .h để chi thì coi ví dụ này. Các hàm trong C++ sẽ được compile từ trên xuống dưới, do đó ví dụ như đoạn code dưới đây thì chương trình sẽ chạy lỗi:

```
#include <iostream>
using namespace std;

void PrintOdd(int n) {
    cout << n << ' ';
    PrintEven(n - 1);
}

void PrintEven(int n) {
    if (n == 0)
        return;
    cout << n << ' ';
    PrintOdd(n - 1);
}

int main() {
    PrintOdd(5);
    return 0;
}
```

    - Chương trình báo lỗi là do khi mình gọi lệnh PrintEven(n – 1); ở hàm PrintOdd, lúc đó hàm PrintEven chưa được khai báo nên chương trình không hiểu được là mình cần gọi hàm nào. Do đó mình cần tiền khai báo như sau:

```
#include <iostream>
using namespace std;

void PrintEven(int n);
```

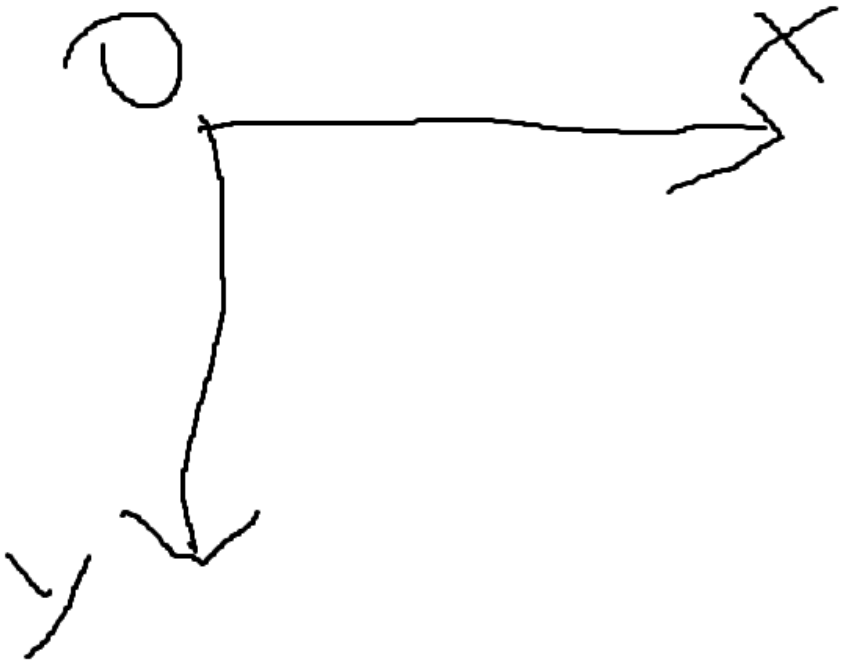
```
void PrintOdd(int n) {
    cout << n << ' ';
    PrintEven(n - 1);
}

void PrintEven(int n) {
    if (n == 0)
        return;
    cout << n << ' ';
    PrintOdd(n - 1);
}

int main() {
    PrintOdd(5);
    return 0;
}
```

- Lúc này chương trình sẽ không báo lỗi và chạy được. Dù là trong project tụi mình làm thì (cho tới giờ :D) không có 2 hàm gọi nhau như thế nhưng mà khi lúc mình làm thì include trực tiếp các file .cpp trong file source thì sẽ báo lỗi nên mình sẽ tiền khai báo trong file .h. Nôm na là vậy thôi, mấy bạn muốn tìm hiểu thêm thì đọc ở đây.

- Tương tự như vậy, những gì liên quan đến điều khiển game của mình thì ở trong 2 file Control.cpp và Control.h, những phần liên quan đến xử lý các dữ liệu của trò chơi thì ở Model.cpp và Model.h.

II. Đồ họa trong game:

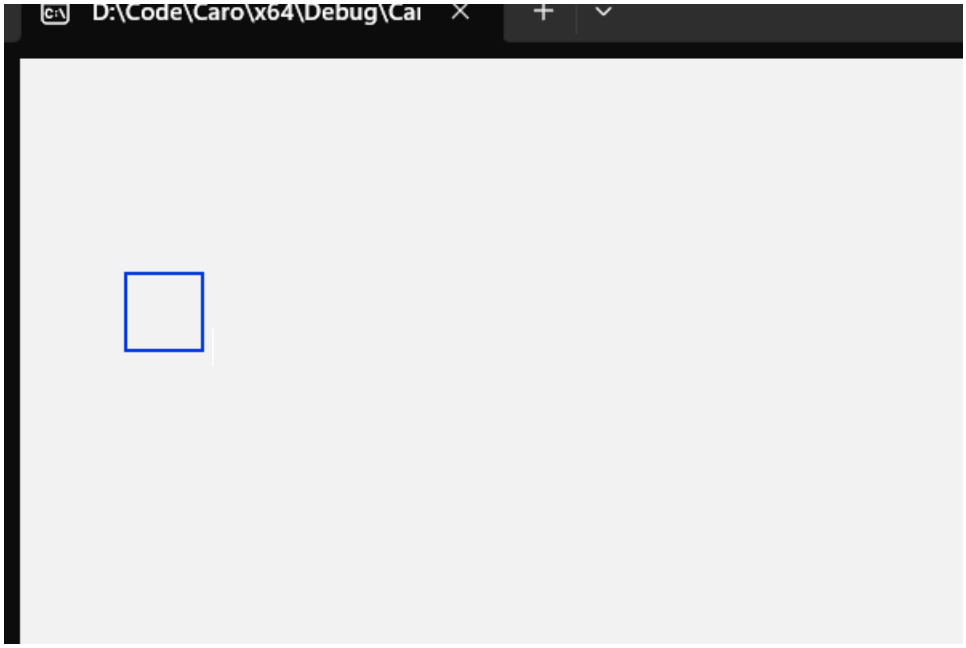
- Đầu tiên thì trục tọa độ của màn hình console sẽ gần giống với tọa độ Oxy nhưng mà trong console thì trục Oy nó ngược lại.



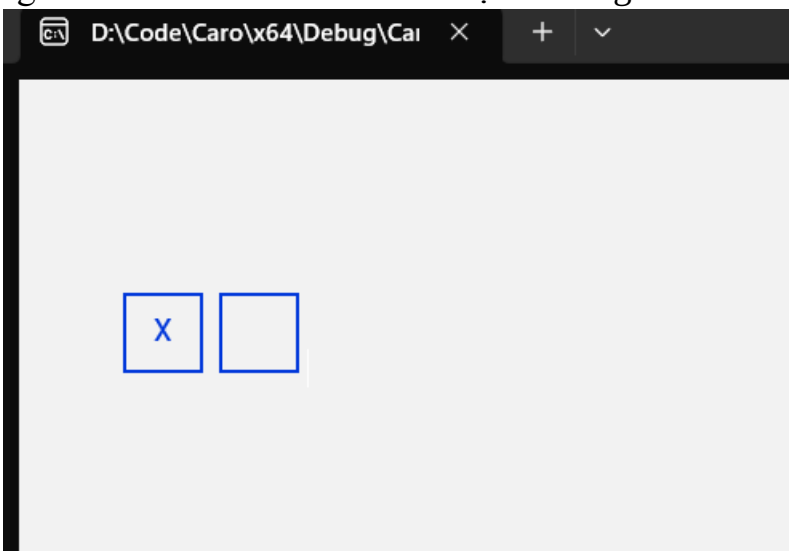
- Tiếp theo thì hình như là trong màn hình console thì 1 đơn vị theo chiều ngang thì sẽ bé hơn 1 đơn vị theo chiều cao (kiểu thay vì trục Ox thì giờ là trục Ox', x' = x / 2 chẳng hạn), nên việc vẽ cái bàn cờ cho nó đẹp hơi khó. Nên mình có "tham khảo" code của mấy anh chị năm trước thì có học được cách để vẽ như sau. Một ô đầu tiên trong bàn cờ của mình (mình sẽ lấy là ô trái trên) sẽ có kích thước 5x3, giả sử mình vẽ 1 ô có góc trái của ô ở vị trí (5, 5) thì:

```
GotoXY(5, 5);
cout << TOP_LEFT << H_LINE << H_LINE << H_LINE << TOP_RIGHT;
GotoXY(5, 6);
cout << V_LINE << SPACE << SPACE << SPACE << V_LINE;
GotoXY(5, 7);
cout << BOTTOM_LEFT << H_LINE << H_LINE << H_LINE << BOTTOM_RIGHT;
```

mình sẽ được



ví dụ mình đánh con X ở ô này thì sẽ đi tới tọa độ (7, 6) rồi cout << “X”; (này mấy bạn tự code thử nha). Vậy còn mấy ô bên cạnh thì sao, nếu mấy ô bên cạnh cũng để kích thước 5x3 thì nó sẽ bị hở ra ở giữa



nên chỉ có 1 ô đầu là kích thước 5x3 thôi, các ô bên cạnh mình sẽ ghép nó chung cạnh với ô trước đó. Ý tưởng là thế thôi nhưng mà mình chưa code :D.

III. Về các thứ có trong file (nhưng mà chi tiết hơn):

- Do trong project này tụi mình code chung nên là mình sẽ ra mấy cái chuẩn trước để dựa vô đó thì tụi mình code từng phần rồi ghép vào nó sẽ (mong là) không bị lỗi. Phần này thì tất nhiên là chưa đầy đủ rồi nên mình sẽ cố cập nhật thêm.

1. Data:
   a. Data.h:
   - Board size là kích thước bảng, hiện tại thì mình để mặc định theo cái console của mình lúc chạy chương trình luôn là 120 cột, 30 hàng, nên là máy bạn nào khác thì nhớ chỉnh lại nha.
   - Color code là mã màu để chỉnh màu của chữ, cụ thể hơn mình sẽ nói trong hàm TextColor của phần View.
   - ASCII code là mã ASCII của mấy ký tự trên bàn phím, trong đó có mấy cái mình đã ép kiểu về char để vẽ cái bảng.

   b. Data.cpp:
   - Hầu như chưa có gì.

2. Control
   a. Control.h và Control.cpp:
   - Hầu như chưa có gì.

3. View:
   a. Control.h:
   - Lưu tiền khai báo các hàm.

   b. Control.cpp:
   - void FixConsoleWindow();
     - Hàm này thì dùng để cố định size của console, hàm này trong file Project thầy đã đưa, mấy bạn đọc trong đó. Nhưng mà không biết là do máy mình hay do mình hiểu sai chức năng của hàm nhưng mà dù gọi hàm rồi nhưng mà mình vẫn kéo thả cái size của console được. Này mấy bạn thử lại nha.
   - void GotoXY(int x, int y);
     - Hàm này cũng là trong file thầy gửi, đơn giản là đưa con trỏ tới vị trí (x, y) trên console.
   - void SetConsoleColor();

- Hàm này là mình dùng để vẽ nền trắng cho Console.

- void DrawBoard(int row, int col, int width, int height, int x, int y);
  - Hàm để vẽ bàn cờ có row hàng, col cột, width, height của mấy anh chị truyền vào là 3 và 1, nếu mình hiểu đúng thì đó là khoảng cách của 2 biên trái phải và trên dưới -> do đó kích thước 1 ô là 5x3 (+2 là tính 2 biên), vẽ bắt đầu từ ô (x, y). Hàm này là mình lấy từ mấy anh chị khóa trước đã làm project này, hàm mình lấy tạm để nghiên cứu cách vẽ thôi nên sẽ cố code lại chứ không copy.
- void DrawTableLine(int numOfCol, char mainSym, char subSym, int width);
  - Hàm này đơn giản là vẽ cái đường ngang trên bàn cờ, trên 1 đường ngang đó sẽ có 2 loại ký tự chính ví dụ đường ngang đầu tiên thì sẽ có – và | để vẽ cái khung của bàn cờ, hoặc là | và ‘ ‘ để vẽ các gạch dọc trên 1 hàng và khoảng trắng giữa chúng.
- void TextColor(int x);
  - Hàm để chỉnh màu chữ, như mình đã nói ở trên thì ví dụ khi mình gọi *TextColor(BLUE);* thì *BLUE* ở đây mang giá trị ((15 << 4) | 1) = 241, sau đó sẽ được chuyển vào hàm *SetConsoleTextAttribute()* là một hàm để chỉnh các thuộc tính về text của console.