ĐỒ ÁN CUỐI KỲ GAME CARO

GIÁO VIÊN HƯỚNG DẪN: TRƯƠNG TOÀN THỊNH
SINH VIÊN: NGUYỄN HOÀNG TUẤN CƯỜNG
MSSV:1712309

DECEMBER 12, 2018

ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM

K2017

Game Caro

I. GIỚI THIỆU GAME

1) Giới thiệu chung

Game caro được xây dựng cho 2 chế độ:

- Person vs Person
- Person vs Bot

2) Luật chơi

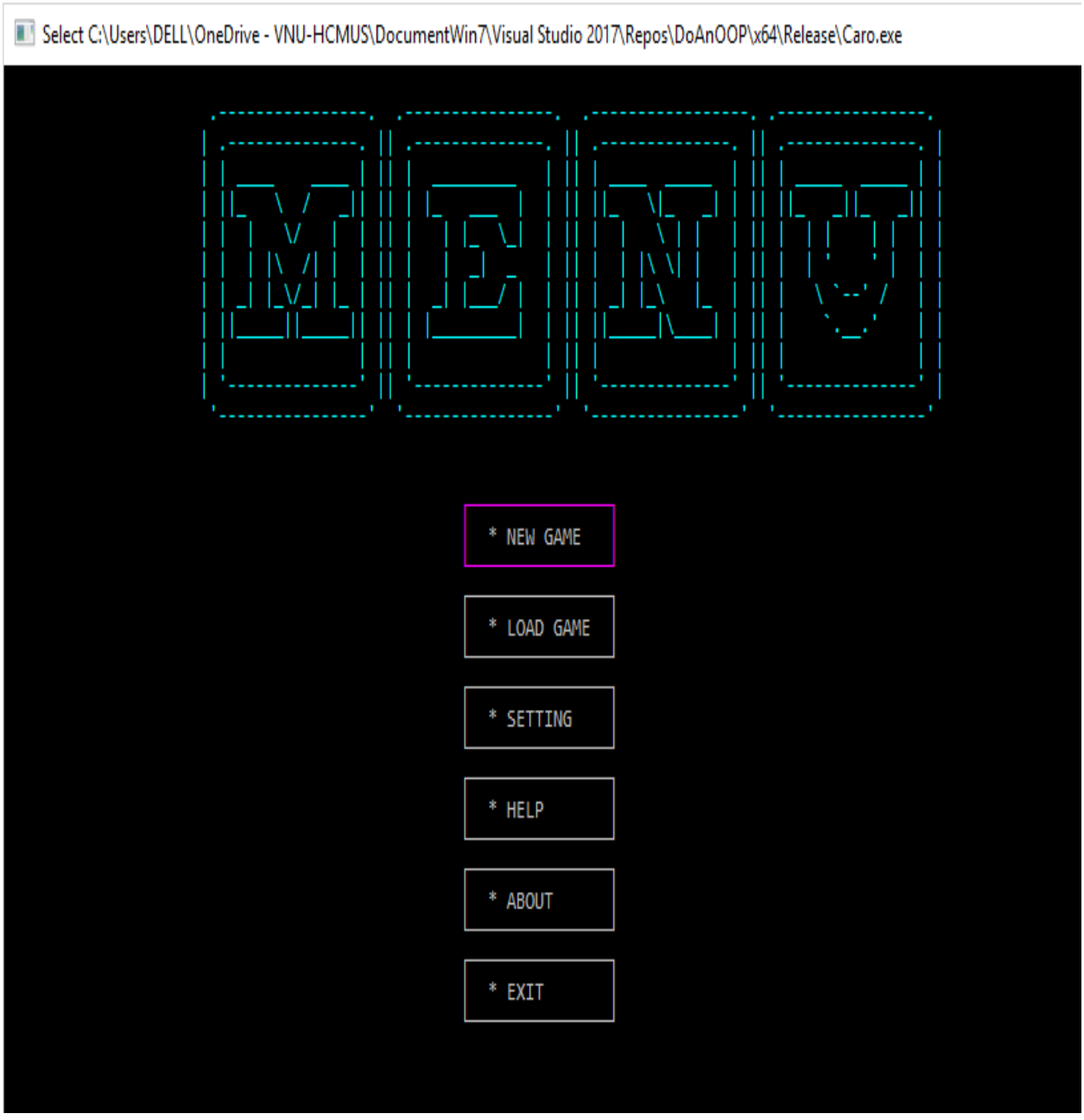
- Bàn cờ sẽ được chia thành các ô
- Có 2 quân cờ là X, O
- Quân X được mặc định đánh trước
- Nếu có 5 quân liên tiếp theo các đường: hàng ngang, hàng dọc, đường chéo chính, đường chéo phụ.
- Luật chặng 2 đầu: nếu 5 quân liên tiếp nhưng nếu 2 đầu bị bao bởi 2 quân khác loại.
  + Vd: X O O O O O X
  + Quân O có 5 quân liên tiếp nhưng bị bao bởi 2 quân X nên không được tính là thắng.
- Hai người chơi bị tính là hòa nhau khi bàn cờ đã được đánh hết ô nhưng vẫn chưa có người thắng.

3) Ý tưởng

- Xây dựng bàn cờ theo tọa độ.
- Do tỷ lệ bàn cờ không đều nên tọa độ hàng ngang sẽ tính 4 ô nhỏ thành 1 ô lớn còn tọa độ hàng dọ sẽ tính 2 ô nhỏ thành 1 ô lớn.
- Xây dựng các class:
  - Board: dùng để thao tác trên bàn cờ bằng các hàm.
  - Game: dung để thực hiện các thao tác liên quan đến vận hành game.
  - Common: chứa các hàm phổ biến được sử dụng nhiều
  - Point: quản lý console bằng tọa độ x, y.

4) Demo

- Hình ảnh giao diện



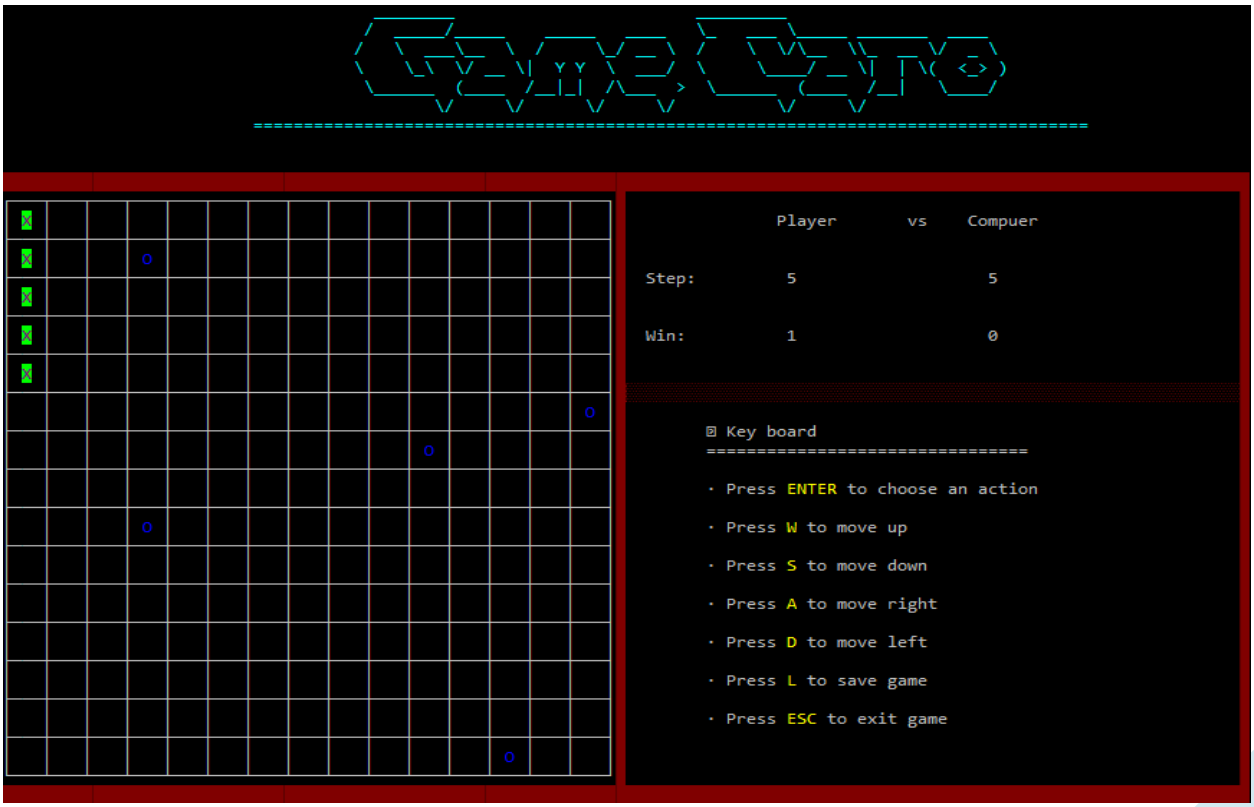
- Hình ảnh các chức năng


»» 1. Play mode: Person vs Computer
2. Sound: off
3. Back


»1. test
2. test2
3. cuong
4. test1
5. nguyenn
6. nguyeennn
7. khkhk
8. llllll
9. adsdasdasd
10. Back

- Hình ảnh vào game

- Hiệu ứng chiến thắng




Ban co muon choi lai khong? (Press Y: Yes or Diff Key to Exit Game)

II. CÀI ĐẶT GAME

1. Tổ chức game thành các class:
   - _Common
   - _Board
   - _Game
   - _Point
2. Chi tiết các class

a) _Common

- Dùng để cài đặt các hàm được sử dụng nhiều trong quá trình thao tác trong các lớp khác nên các hàm được khai báo public và theo kiểu static để các lớp khác có thể sử dụng dễ dàng.
- Các hàm
  - static void fixConsoleWindow();

⇨ Căn chỉnh console.

  - static void gotoXY(int, int);

⇨ Dùng để di chuyển con trỏ đến vị trí có tọa độ x,y trên console.

  - static void TextColor(int color);

⇨ Dùng để thay đổi màu chữ khi in ra console
(Tham khảo:
https://stackoverflow.com/questions/29574849/how-to-change-text-color-and-console-color-in-codeblocks)

  - static void DrawRectangle(int x, int y, int width, int height, bool draw);

⇨ Hàm dùng để vẽ các ô hình chữ nhật và có thể xóa các ô chữ nhật nếu thay đổi draw thành false.

- static void DrawCaro(int a, int b,string filename);

⇨ Hàm dùng để vẽ tiêu đề bằng các ký tự ascii được tạo sẵn

(Tham khảo: ASCII Generator)

- static void resizeConsole(int width, int height);

⇨ Hàm dùng để chỉnh lại kích thước console

(Tham khảo: https://daynhauhoc.com/t/hoi-cach-thay-doi-kich-thuoc-console-trong-c/34390)

- static void Win(int a, int b, string filename);

⇨ Hàm dùng để vẽ hiệu ứng chiến thắng

b) _Board

- Dùng để thể hiện bàn cờ và các thao tác kiểm tra trong quá trình chơi game.
- Chứa các thuộc tính:

_size: kích thước bàn cờ

_left: cách lề trái của console một khoảng là _left

_top: Cách lề trên của console một khoảng là _top

_Point **_pArr: là một mảng 2 chiều. mỗi phần tử là kiểu Point chứa địa chỉ. Dùng để quản lý bàn cờ khi người chơi đánh vào.

- **Các hàm được cài đặt.**
  - `_Board(int left, int top, int size)`

⇨ Hàm tạo bàn cờ với các thông số là: BOARD_SIZE, LEFT, TOP.

- ~_Board()
  ⇨ Hàm hủy bàn cờ
- int **highlight
  ⇨ Mảng 2 chiều tạo ra để lưu lại vị trí các quân khi phát hiện chiến thắng
- void ResetData();
  ⇨ Dùng để khởi động dữ liệu cho bàn cờ
- void DrawBoard();
  ⇨ Vẽ bàn cờ
- int CheckBoard(int, int, bool);
  ⇨ Dùng để cập nhật thuộc tính _check tại vị trí mà người dùng gõ phím 'enter' để đánh cờ, tùy vào lượt mà _check sẽ nhận 1 hay -1. Ngoài ra ta them hàm testBoard kiểm tra thắng thua theo luật caro. Đoạn mã bên dưới chỉ là minh họa, sinh viên tự cài đặt sao cho hàm trả về -1 (người thứ 1 thắng), 0 (hòa), 1 (người thứ 2 thắng) hay 2 (chưa ai thắng).
- bool IsDraw();
  ⇨ Hàm kiểm tra xem trận đấu có hòa không bằng cách xem còn ô nào chưa đánh không
- bool TestRow(int, int);
  ⇨ Hàm kiểm tra hàng ngang có hơn 5 con liên tiếp không
- bool TestColumn(int, int);
  ⇨ Hàm kiểm tra hàng dọc có hơn 5 con liên tiếp không
- bool TestMainDiagonal(int, int);
  ⇨ Hàm kiểm tra đường chéo chính (từ trái sang phải) có hơn 5 con liên tiếp không

- `bool TestSubDiagonal(int, int);`
  ⇨ Hàm kiểm tra đường chéo phụ (từ phải sang trái) có hơn 5 con liên tiếp không.

➔ ĐỒNG THỜI TRONG MỖI HÀM TEST CÓ XÂY DỰNG PHÁT HIỆN CHẶN 2 ĐẦU. NẾU CÓ HƠN 5 QUÂN LIÊN TIẾP NHƯNG 2 ĐẦU BỊ CHẶN BỞI 2 QUÂN CỜ KHÁC LOẠI THÌ VẪN CHƯA TÍNH THẮNG

- `int TestBoard(int, int);`
  ⇨ Hàm dùng để kiểm tra tất cả trường hợp thắng, hòa.
- `int RandomCaro();`
  ⇨ Hàm dùng để máy đánh ngẫu nhiên một quân cờ trên bàn cờ bằng cách lọc hết những vị trí trống của bàn cờ rồi random trong số đó cho máy đánh.
- `void DrawGame();`
  ⇨ Hàm dùng để vẽ lại game với dữ liệu có sẵn: vd bàn cờ có sẵn các check rồi.
- `int SaveBoard(string &filename);`
  ⇨ Hàm dùng để lưu bàn cờ vào một file txt. File này sẽ lưu các check của bàn cờ và lượt hiện tại của bàn cờ để dễ cho việc tải lại game đã lưu từ trước.
- `int LoadBoard(string);`
  ⇨ Hàm dùng để tải lên các game đã lưu từ trước bằng việc đọc file txt và gán các giá trị check cho bàn cờ mới. sau đó ta dùng hàm drawgame để vẽ lại bàn cờ.

c) _Game

- Dùng để vận hành game từ lúc bắt đầu đến lúc thoát khỏi game.

- Bao gồm các thuộc tính riêng tư sau:
  - _Board *_b: một bàn cờ b;
  - bool _turn: biến thể hiện lượt đi nếu true là lượt của X còn false là lượt của O.
  - int _x, _y: hai biến thể hiện tọa độ của con trỏ.
  - int _command: biến dùng để lưu giá trị nhập từ bàn phím.
  - bool _loop: biến thể hiện có muốn tiếp tục chơi hay không, true là tiếp tục, false là kết thúc.
  - int _playmode: chế độ chơi 1 người hay 2 người.
  - bool _sound: thể hiện âm thanh tắt hay mở.
  - int _countx: dùng để đếm lượt đi của X
  - int _counto: dùng để đếm lượt đi của O

- **Cài đặt các hàm**
  - `_Game(int, int, int);`
    ⇨ Hàm tạo một game mới với các thông số là: BOARD_SIZE, LEFT, TOP.
  - `~_Game();`
    ⇨ Xóa game đã tạo.
  - `void SetPlayMode(int p)`
    ⇨ Gán giá trị của play mode theo cài đặt.
  - `int PlayMode()`
    ⇨ Lấy giá trị playmode hiện hành.
  - `int Command()`

⇨ Lấy giá trị mà _command đang giữ là ký tự nhập từ bàn phím.

- bool IsContinue()
  ⇨ Hàm kiểm tra xem có tiếp tục chơi game hay không
- char WaitKeyBoard();
  ⇨ Hàm đợi giá trị từ bàn phím.
- char AskContinue();
  ⇨ Hàm hỏi xem có muốn tiếp tục chơi không.
- bool SaveGame()
  ⇨ Hàm lưu game đang chơi bằng cách cho người dùng nhập vào tên file muốn lưu game. Game sẽ được lưu vào fie dạng txt chứa các check của bàn cờ đang chơi.
- void LoadGameCurrent()
  ⇨ Hàm dùng để tải trò chơi mà dữ liệu còn chưa bị xóa. Trong hàm này ta sẽ dùng hàm Drawgame để vẽ lại game với dữ liệu có sẵn.
- void LoadGame();
  ⇨ Hàm dùng để tải game từ một file đã được lưu từ trước.
- void StartGame();
  ⇨ Hàm bắt đầu game mới.
- void Play();
  ⇨ Hàm này dùng để quản lý toàn bộ quá trình chơi, từ chọn chế độ hay thoát game.
- void DrawInfo();

⇨ Hàm dùng để điền info của game như số lượt, tỉ số thắng thua.vv..

- `void Help();`

⇨ Hàm để hướng dẫn trước khi chơi.

- `void About();`

⇨ Hàm xem thông tin của game.

- `void NewGame();`

⇨ Hàm bắt đầu game mới theo playmode.(mặc định là 1 người chơi với máy).

- `void NewGame1P();`

⇨ Game mới 1 người chơi với máy.

- `void NewGame2P();`

⇨ Game mới chơi 2 người.

- `void Setting();`

⇨ Hàm cài đặt game: chế độ chơi game và bật tắt âm thanh.

- `int Menu();`

⇨ Hàm hiện chức năng và chọn chức năng của game.

- `void ExitGame();`

⇨ Hàm thoát game không chơi nữa.

- `void Exit();`

⇨ Hàm này dùng cho nút ESC, khi đang chơi ấn ESC thì sẽ chạy hàm này. Chức năng là hỏi xem có lưu game trước khi thoát hay không. Khác với hàm savegame() ở chỗ savegame() sẽ chơi tiếp còn Exit() thì sau khi save sẽ thoát khỏi game.

- `int ProcessFinish1P();`

⇨ Hàm check xem có chiến thắng chưa của một người chơi với máy.

- `int ProcessFinish2P();`
  ⇨ Hàm check xem có chiến thắng chưa của hai người kèm theo hiệu ứng thắng.
- `bool ProcessCheckBoard1P();`
  ⇨ Hàm dùng để kiểm tra ô cờ được chọn xem đã đánh chưa, nếu chưa thì đánh vào X vì người chơi luôn đánh trước.
- `bool ProcessCheckBoard2P();`
  ⇨ Hàm dùng để kiểm tra ô cờ được chọn xem đã đánh chưa, nếu chưa thì đánh vào X, O theo lượt hiện hành.
- `void MoveRight();`
- `void MoveLeft();`
- `void MoveUp();`
- `void MoveDown();`

} Dùng di chuyển con trỏ.

d) _Point

- Dùng để quản lý các ô bằng tọa độ x và y.
- Bao gồm các thuộc tính riêng tư:
  - _x: hoành độ
  - _y: tung độ
  - _check: dùng để đánh dấu các ô trên bàn cờ:
    - _check == 1: là O
    - _check == 2: là X
    - _check == 0: là ô đó chưa đánh.

III. TỔNG KẾT

Game được xây dựng với kỹ thuật hướng đối tượng cơ bản song vẫn còn nhiều thiếu sót như chế độ 1 người chơi với máy, máy vẫn chưa có thuật toán nảy sinh đường đi rõ ràng cụ thể. Những thao tác và hiệu ứng còn chưa được phong phú trong game. Tuy còn nhiều hạn chế nhưng các chức năng hoạt động cũng tạm ổn định tuy còn thô sơ.

Cám ơn thầy Trương Toàn Thịnh đã hướng dẫn và giúp đỡ trong suốt quá trình làm đồ án.

(*chú thích: các phần có nền tô vàng đó là các hàm tìm hiểu và viết thêm để hoàn thiện game).