

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**

-----□□□-----



**BÁO CÁO ĐỒ ÁN MÔN HỌC**  
**CƠ SỞ LẬP TRÌNH**

**Đề tài: Xây dựng game Caro**

**Giảng viên hướng dẫn** : TS Trương Toàn Thịnh

**Nhóm sinh viên thực hiện** : Nhóm 2 – Lớp 24CTT3

STT	Họ và tên	MSSV
1	Võ Lâm Tuấn	24120240
2	Trần Huỳnh Gia Bảo	24120267
3	Võ Nguyễn Việt Hoàng	24120249
4	Đỗ Chí Cao	24120270
5	Phạm Anh Tuấn	24120238

**TP. Hồ Chí Minh, 2024**

## LỜI MỞ ĐẦU

*Đây là bản báo cáo đồ án Cơ Sở Lập Trình của nhóm 2 chúng em, gồm 5 thành viên và đã cùng nhau đồng hành từ những buổi học đầu tiên trên con đường Đại học. Đồ án này với chúng em, ban đầu có phần hơi khó khăn song vì chúng em chưa quen biết nhau nhiều mà cũng vì lượng kiến thức yêu cầu khá nặng. Thế nhưng trong quá trình làm, chúng em đã học hỏi được nhiều điều, từ kỹ năng teamwork cho đến những kiến thức khó. Từ đó, chúng em nhận ra rằng: “Báo cáo này không chỉ là tài liệu thể hiện kết quả cuối cùng, mà còn là nhật ký của một hành trình đầy thách thức nhưng cũng thú vị”.*

## Mục lục

<b>I. Báo cáo nhóm.....</b>	<b>4</b>
1. Bảng phân công công việc tuần 1 (từ 16/09/2024 đến 22/09/2024) .....	4
2. Bảng phân công công việc tuần 2 (từ 23/09/2024 đến 29/09/2024) .....	5
3. Bảng phân công công việc tuần 3 (từ 30/09/2024 đến 06/10/2024) .....	6
4. Bảng phân công công việc tuần 4 (từ 07/10/2024 đến 13/10/2024) .....	7
5. Bảng phân công công việc tuần 5 (từ 14/10/2024 đến 20/10/2024) .....	8
6. Bảng phân công công việc tuần 6 (từ 21/10/2024 đến 27/10/2024) .....	9
7. Bảng phân công công việc tuần 7 (từ 28/10/2024 đến 03/11/2024) .....	10
8. Bảng phân công công việc tuần 8 (từ 04/11/2024 đến 10/11/2024) .....	11
9. Bảng phân công công việc tuần 9 (từ 11/11/2024 đến 17/11/2024) .....	12
10. Bảng phân công công việc tuần 10 (từ 18/11/2024 đến 24/11/2024) .....	13
11. Bảng phân công công việc tuần 11 (từ 25/11/2024 đến 01/12/2024) .....	14
12. Bảng phân công công việc tuần 12 (từ 02/11/2024 đến 08/12/2024) .....	15
<b>II. Báo cáo kỹ thuật.....</b>	<b>16</b>
1. Tictactoe.cpp (phần Gameplay) .....	16
2. Graphic.cpp .....	18
3. Draw.cpp .....	24
4. Console.cpp .....	25
5. LoadGame.cpp .....	30
6. NewGame.cpp .....	33
7. Language.cpp .....	37
8. Một số file khác: About.cpp, Help.cpp, ... ..	39

## I. Báo cáo nhóm

### 1. Bảng phân công công việc tuần 1 (từ 16/09/2024 đến 22/09/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tìm hiểu về những gì cần thiết cho một đồ án Caro	Đã tìm hiểu xong về những gì cần tìm hiểu và kiến thức cần có	
2	Trần Huỳnh Gia Bảo	Tìm hiểu về cách thức lưu trữ file.	Tham gia họp nhóm đầy đủ, tìm hiểu kiến thức cơ bản.	
3	Võ Nguyễn Việt Hoàng	Tìm hiểu, chuẩn bị các kiến thức cần thiết cho một đồ án Caro.	Tham gia họp nhóm đầy đủ và tìm hiểu những kiến thức cần thiết.	
4	Đỗ Chí Cao	Tìm hiểu những gì cơ bản cần có của 1 game Caro.	Tham gia họp nhóm đầy đủ, liệt kê được 1 game Caro cần gì.	
5	Phạm Anh Tuấn	Họp nhóm, tìm hiểu kiến thức cần chuẩn bị	Đi họp đầy đủ và tìm hiểu về những kiến thức cần có	

## 2. Bảng phân công công việc tuần 2 (từ 23/09/2024 đến 29/09/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tổ chức họp nhóm, phân công nhiệm vụ.	Phân chia nhiệm vụ cho các thành viên	Chưa phổ biến cụ thể các kiến thức mỗi bạn cần
2	Trần Huỳnh Gia Bảo	Tìm hiểu về cách thức lưu trữ file.	Đã tìm hiểu xong tạo, lưu file cơ bản.	Chưa nắm vững cách hoạt động của mảng.
3	Võ Nguyễn Việt Hoàng	Nhận và tìm hiểu về phần nhiệm vụ được giao.	Tìm hiểu xong và viết thử những hàm trong phần nhiệm vụ được giao.	Chưa hiểu rõ về những kiến thức và kỹ năng cần dùng.
4	Đỗ Chí Cao	Nhận nhiệm vụ được giao và thực hiện.	Lập trình và chạy thử các hàm cần cho nhiệm vụ.	Các hàm chưa được làm gọn.
5	Phạm Anh Tuấn	Tìm hiểu về phần và nhiệm vụ được giao	Tìm hiểu xong cần phải chuẩn bị kiến thức gì để làm	Chưa nắm vững kiến thức, kỹ năng để làm

### 3. Bảng phân công công việc tuần 3 (từ 30/09/2024 đến 06/10/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Bổ sung kiến thức cần để làm ra một con bot tự chơi.	Học được minimax, alpha beta pruning	Chưa trình bày lại kiến thức cho các bạn cùng hiểu
2	Trần Huỳnh Gia Bảo	Viết các file là cầu nối.	MainCaro.cpp, NewGame.cpp.	Chưa được tối ưu và trình bày gọn gàng.
3	Võ Nguyễn Việt Hoàng	Viết các hàm để in, vẽ ASCII cơ bản, vẽ pixel art có sẵn.	Viết xong phần mềm chuyển đổi từ file ảnh sang file .txt chứa dữ liệu về màu pixel của ảnh.	Chưa tìm được nguồn tìm pixel art phù hợp.
4	Đỗ Chí Cao	Tìm hiểu bảng ASCII và bảng màu.	Tạo Graphics.cpp để chứa các hàm về thiết kế.	Cửa sổ của game chưa được sửa lại cho phù hợp.
5	Phạm Anh Tuấn	Tìm hiểu về phần ngôn ngữ, học C++	Học được C++ ở mức cơ bản	Chưa hiểu về cách hoạt động của để chuyển đổi ngôn ngữ

**4. Bảng phân công công việc tuần 4 (từ 07/10/2024 đến 13/10/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tìm hiểu về phần cài đặt, tìm các projects của anh chị để tham khảo các cài đặt sao cho thống nhất với các bạn	Biết được cơ bản về cấu trúc phần cài đặt & trình bày lại thuật toán cho các thành viên trong nhóm & tổ chức họp nhóm (cả online và offline).	Chưa bắt đầu code.
2	Trần Huỳnh Gia Bảo	Viết các file là cầu nối.	Help.cpp, About.cpp.	Chưa được tối ưu và trình bày gọn gàng.
3	Võ Nguyễn Việt Hoàng	Tham khảo project của anh chị khóa trước, thống nhất tên các hàm, cấu trúc cài đặt với nhóm.	Thống nhất được tên hàm và cấu trúc phần cài đặt với nhóm, viết được hàm để cố định Console.	Chưa trình bày code của bản thân cho nhóm.
4	Đỗ Chí Cao	Tham khảo project của anh chị khóa trước, thống nhất tên và cấu trúc của các hàm với nhóm.	Thống nhất được tên và cấu trúc hàm, đã cố định được cửa sổ Game.	Chỉnh sửa trước khi push lên GitHub.
5	Phạm Anh Tuấn	Tham khảo projects của các anh chị, xác định cách làm	Học về map, bắt đầu code	Chưa áp dụng để chuyển đổi ngôn ngữ tiếng anh thành tiếng việt được

**5. Bảng phân công công việc tuần 5 (từ 14/10/2024 đến 20/10/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Quản lý tiến trình tìm hiểu của các thành viên trong nhóm & bắt đầu cài đặt “bộ não” của bot.	Nắm rõ được khó khăn của các thành viên trong nhóm & Code xong các hàm cơ bản.	
2	Trần Huỳnh Gia Bảo	Tối ưu các file.	Tối ưu được một phần của các file đã làm.	Vẫn còn những lỗi và chưa lưu được file.
3	Võ Nguyễn Việt Hoàng	Tiến hành vẽ và thiết kế bố cục cơ bản của trò chơi.	Vẽ được các khung cơ bản, bàn cờ caro cơ bản.	Chưa trang trí được nền của game.
4	Đỗ Chí Cao	Thiết kế và vẽ các khung.	Đã có bàn cờ và khung cơ bản.	Chưa trang trí quá nhiều.
5	Phạm Anh Tuấn	Xác định vấn đề và tìm cách khắc phục	Chỉnh sửa chuyển đổi ngôn ngữ từ tiếng anh sang tiếng việt không dấu	Còn chưa liên kết được với các phần khác



**6. Bảng phân công công việc tuần 6 (từ 21/10/2024 đến 27/10/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tiếp tục code các hàm khác, tổ chức hợp offline để thống nhất giữa các code.	Tổ chức hợp offline & code xong cơ bản minimax.	Hàm đánh giá nước đi vẫn còn quá lỗi, chưa fix xong.
2	Trần Huỳnh Gia Bảo	Viết file LoadGame.cpp.	Vẽ khung hiển thị, định dạng file lưu.	Chưa load được file save.
3	Võ Nguyễn Việt Hoàng	Vẽ bàn cờ caro, tìm hiểu về các phần khác.	Tham gia hợp và hỗ trợ căn chỉnh menu, sửa lại hàm cố định console.	Phần trang trí còn sơ sài.
4	Đỗ Chí Cao	Thêm các trang trí cơ bản theo chủ đề của Game.	Chỉnh sửa lại cửa sổ Game, cố định cửa sổ ở giữa màn hình.	Cần trang trí thêm giao diện Game.
5	Phạm Anh Tuấn	Đi hợp offline, thống nhất code. Tìm hiểu về các hàm của các phần khác	Hiểu về cách các hàm các phần khác và cách sử dụng những hàm đó	Chưa liên kết code vào để chuyển đổi giữa các phần

**7. Bảng phân công công việc tuần 7 (từ 28/10/2024 đến 03/11/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Fix hàm đánh giá nước đi & fix các lỗi khi thống nhất các code của các thành viên.	Bot đã có thể chơi được.	Nhiều lỗi, chưa xử lý các corner case, bot đánh còn chậm. sau khi hợp code với các bạn thì có những bug nhỏ xảy ra chưa fix được.
2	Trần Huỳnh Gia Bảo	Tiếp tục làm file LoadGame.cpp.	Đã lưu được file	Chưa chọn, load được file save.
3	Võ Nguyễn Việt Hoàng	Tiếp tục vẽ, trang trí game, chơi thử con bot.	Hoàn thành bàn caro, chơi thử con bot và cho phản hồi về các bug và độ thông minh của con bot.	Code thiếu tổ chức.
4	Đỗ Chí Cao	Tiếp tục trang trí, làm thêm các hàm hỗ trợ các hàm trước.	Bàn caro đã được hoàn thành, làm tester.	Các hàm còn khá cồng kềnh.
5	Phạm Anh Tuấn	Thêm một phần language trong menu và chọn ngôn ngữ	Áp dụng các hàm ở phần khác để thêm cho phần language	Còn chưa liên kết được với các phần khác

**8. Bảng phân công công việc tuần 8 (từ 04/11/2024 đến 10/11/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Fix lỗi liên quan đến logic game. Tester, tìm kiếm lỗi	Tìm được nhiều lỗi.	Chưa fix hết các lỗi.
2	Trần Huỳnh Gia Bảo	Tiếp tục làm file LoadGame.cpp.	Đã vẽ được 9 ô chọn file save.	Chưa chọn, load được file save.
3	Võ Nguyễn Việt Hoàng	Tiếp tục trang trí màn hình game, căn chỉnh vị trí bàn cờ, test bot.	Sửa được một vài bug, căn chỉnh được bàn cờ.	
4	Đỗ Chí Cao	Trang trí thêm các họa tiết.	Phát hiện, sửa lỗi cửa sổ vừa được phát hiện.	Vẫn còn 1 số lỗi chưa được sửa hoàn toàn.
5	Phạm Anh Tuấn	Liên kết ngôn ngữ vào các phần khác	Chính sửa để code liên kết với các phần khác	Code còn chưa hoàn chỉnh

**9. Bảng phân công công việc tuần 9 (từ 11/11/2024 đến 17/11/2024)**

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tối ưu bot	Bot đánh nhanh hơn, nước đi có độ chính xác cao hơn, kết hợp cả chiến thuật tấn công và phòng thủ.	
2	Trần Huỳnh Gia Bảo	Tiếp tục làm file LoadGame.cpp.	Đã di chuyển được trong menu load.	Chưa load được file save.
3	Võ Nguyễn Việt Hoàng	Chỉnh sửa lại bàn cờ theo ý kiến của nhóm.	Vẽ lại bàn cờ mới, thêm các thông tin người chơi.	Bàn cờ mới còn bug.
4	Đỗ Chí Cao	Sửa các lỗi từ tuần trước.	Hiện tên cùng với các họa tiết của Game.	
5	Phạm Anh Tuấn	Hoàn chỉnh phần liên kết với các phần khác	Hoàn chỉnh, bổ sung code liên kết với các phần khác	Fix cho từng ngôn ngữ để hiển thị một cách chính chu.

### 10. Bảng phân công công việc tuần 10 (từ 18/11/2024 đến 24/11/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tìm hiểu kiến thức cần để train một con AI có thể chơi ở mức khó.	Học được nhiều kiến thức mới.	Chưa biết cách cài đặt.
2	Trần Huỳnh Gia Bảo	Tiếp tục làm file LoadGame.cpp.	Fix một số lỗi về file save.	Chưa load được file save.
3	Võ Nguyễn Việt Hoàng	Fix bug, làm màn hình loading.	Tìm được ảnh ưng ý và hoàn thành màn hình loading.	
4	Đỗ Chí Cao	Hoàn thiện thêm cửa sổ của Game.	Canh chỉnh sao cho phù hợp với nội dung của nhóm.	
5	Phạm Anh Tuấn	Fix cho từng ngôn ngữ để hiển thị một cách hoàn chỉnh	Fix cho ngôn ngữ tiếng Anh và tiếng Việt hiển thị hoàn chỉnh	Tìm hiểu về các phần khác, bổ sung nếu cần

### 11. Bảng phân công công việc tuần 11 (từ 25/11/2024 đến 01/12/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Train AI chơi bot.	Đã train xong và tích hợp vào code.	Chưa biết cách tích hợp bot vào thặng C++ mà không qua Python.
2	Trần Huỳnh Gia Bảo	Tiếp tục làm file LoadGame.cpp.	Đã load được file save.	Chưa tối ưu code.
3	Võ Nguyễn Việt Hoàng	Hoàn thành những hiệu ứng còn thiếu khi kết thúc ván chơi.	Đã hoàn hình Popup khi kết thúc ván đấu.	Tỉ lệ bị lệch
4	Đỗ Chí Cao	Chỉnh sửa thêm các họa tiết vừa thêm.	Đã hoàn thiện và chỉnh sửa được các họa tiết vừa thêm.	Canh lề còn lỗi.
5	Phạm Anh Tuấn	Tìm hiểu các phần khác và chỉnh sửa các chi tiết nhỏ	Tìm hiểu các phần khác và chỉnh sửa các chi tiết nhỏ	Chuẩn bị các phần cần thiết để báo cáo, thuyết trình

## 12. Bảng phân công công việc tuần 12 (từ 02/11/2024 đến 08/12/2024)

STT	Thành viên	Nhiệm vụ	Đã làm xong	Chưa làm xong
1	Võ Lâm Tuấn	Tiếp tục tối ưu lại bot.	Bot thông minh hơn (độ sâu duyệt từ 4 lên 6), thêm một vài tối ưu nhỏ như là đầu trận thì giảm độ sâu tìm kiếm. Ngoài ra, hủy tích hợp với AI được trained vì con bot đó dùng thư viện ngoài.	
2	Trần Huỳnh Gia Bảo	Hoàn thành file LoadGame.cpp.	Thêm một vài chú thích cho load game như: ghi đè file, load file save, rename, delete.	
3	Võ Nguyễn Việt Hoàng	Tiếp tục trang trí thêm.	Hoàn thành phần trang trí của game.	
4	Đỗ Chí Cao	Tiếp tục trang trí và chỉnh sửa giao diện game.	Hoàn thành gần hết phần trang trí của Game.	
5	Phạm Anh Tuấn	Chuẩn bị những gì cần cho buổi báo cáo, thuyết trình đồ án	Tham khảo các đồ án của các anh chị và chuẩn bị cho buổi thuyết trình	

## II. Báo cáo kỹ thuật.

### 1. Tictactoe.cpp (phần Gameplay)

#### Thuật toán chính:

##### \* *Minimax*:

- Giới thiệu thuật toán:
- + Minimax là một thuật toán được phát triển dựa trên lý thuyết trò chơi, được sử dụng để tìm chiến lược tối ưu trong các trò chơi dạng đối kháng hai người (Zero-sum game). Nguyên tắc cơ bản của thuật toán: Max: Người chơi tối ưu hóa chiến lược để đạt được điểm số cao nhất. Min: Đối thủ cố gắng giảm tối đa lợi ích của Max bằng cách chọn nước đi tệ nhất cho Max. Thuật toán được triển khai thông qua một cây trạng thái (Game Tree), trong đó mỗi đỉnh (node) là một trạng thái của trò chơi và các cạnh (edge) là các hành động có thể thực hiện.
- Nguyên lý hoạt động:
- + Mỗi đỉnh của cây biểu thị trạng thái của trò chơi (ví dụ: bố cục bàn cờ). Mỗi cạnh mô tả một nước đi có thể thực hiện. Lá của cây trạng thái là các trạng thái cuối cùng (khi trò chơi kết thúc) và được đánh giá bằng một hàm đánh giá (Board Evaluation).
- + Nếu trạng thái là trạng thái cuối (End state), giá trị trạng thái được xác định theo kết quả của trò chơi (thắng/thua/hòa). Nếu trạng thái chưa kết thúc, giá trị được tính trên cơ sở giả định cả hai bên đều chơi tối ưu.
- + Từ các trạng thái cuối cùng (lá), giá trị được tính ngược về phía gốc cây: Max chọn giá trị lớn nhất trong các nước đi khả thi. Min chọn giá trị nhỏ nhất để làm giảm lợi thế của Max.
- Tại nút gốc (trạng thái ban đầu), thuật toán chọn hành động với giá trị tối ưu nhất.

=> Nhưng như vậy sẽ không tối ưu vì tình huống tệ nhất có thể khám phá toàn bộ cây! Do đó ta có các thuật toán tối ưu sau:

##### \* *Alpha-beta pruning*:

- + Là một kỹ thuật cắt tỉa các nhánh không cần thiết của cây trạng thái. Cho phép Minimax bỏ qua (không tính toán) những nhánh không ảnh hưởng đến kết quả cuối cùng, giảm tải lượng công việc.



+ Thay vì duyệt đến trạng thái cuối cùng, Minimax dừng duyệt ở độ sâu xác định và sử dụng hàm Board Evaluation cải tiến để đánh giá trạng thái. Hàm lượng giá cần phản ánh mức độ lợi thế của mỗi trạng thái (ví dụ: đếm số lượng quân trên cờ, xem lợi thế đang ở phe nào).

**\* *Board Evaluation:***

+ Board Evaluation là một dạng Heuristic Evaluation, trong đó trạng thái bàn cờ được đánh giá dựa trên các tiêu chí chủ quan đặt ra. Ví dụ: đối thủ có đang chiếm ưu thế không? Việc tấn công tại vị trí này có mang lại lợi thế không? Thông qua các đánh giá như vậy, ta có thể ước lượng "giá trị" của một trạng thái mà không cần phải duyệt toàn bộ cây xuống các node lá, giúp giảm độ phức tạp tính toán trong quá trình ra quyết định.

**\* *Move Prioritization:***

+ Move Prioritization là một dạng Heuristic Evaluation nhằm tối ưu hóa việc chọn lựa nước đi trong các trò chơi đối kháng. Kỹ thuật này tập trung đánh giá các nước đi tiềm năng dựa trên các tiêu chí chiến lược: liệu nước đi có tạo thế tấn công mạnh như chuỗi 3 hoặc 4 quân liên tiếp không, hay có khả năng chặn thế tấn công nguy hiểm từ đối thủ không. Ngoài ra, Move Prioritization còn ưu tiên các nước đi mang lại lợi thế kép, như thiết lập thế thắng đồng thời từ nhiều hướng. Kỹ thuật này giúp giảm không gian tìm kiếm, cải thiện chất lượng nước đi và tăng hiệu quả tính toán, nhất là khi kết hợp với thuật toán như Minimax hoặc Alpha-Beta Pruning.

**\* *Depth-limit search:***

+ Depth Limit trong Minimax là một kỹ thuật quan trọng giúp kiểm soát độ sâu của cây trạng thái mà thuật toán Minimax sẽ duyệt. Trong các trò chơi có không gian trạng thái lớn (như cờ vua, cờ vây), việc duyệt toàn bộ cây trạng thái đến các node lá là không khả thi vì số lượng trạng thái tăng theo cấp số nhân với mỗi bước đi. Do đó, depth limit được đặt ra để giới hạn độ sâu tìm kiếm của Minimax ở một mức cố định, từ đó giảm thiểu thời gian và tài nguyên tính toán. Khi đạt đến độ sâu giới hạn, thay vì tiếp tục duyệt cây, thuật toán sẽ dừng lại và sử dụng một hàm lượng giá (Evaluation Function) để ước lượng giá trị của trạng thái hiện tại của trò chơi. Giá trị này sẽ được sử dụng để giả định kết quả của nhánh đó, đóng vai trò tương tự như giá trị của các node lá trong trường hợp duyệt toàn bộ cây.

+ Ví dụ, trong trò chơi Caro, độ sâu giới hạn được đặt ở mức 4 hoặc 6 tầng. Điều này có nghĩa là Minimax chỉ tính toán trước khoảng 4-6 bước đi, sau đó sử dụng hàm lượng giá để ước tính giá trị của

trạng thái. Mặc dù không thể phân tích hết kịch bản dài, nhưng độ sâu này thường đủ để đưa ra một nước đi hợp lý mà vẫn đảm bảo hiệu suất tính toán. Kết hợp cùng các cải tiến như Alpha-Beta Pruning, Depth Limit khiến Minimax trở thành lựa chọn hiệu quả trong nhiều trò chơi đối kháng.

## 2. Graphic.cpp

### 2.1. Hiển thị giao diện:

#### a) *IntroAnimation:*

- Tạo hiệu ứng "Press ENTER to start the game" nhấp nháy trên màn hình chờ.

```
while (!stopBlinking)
{
    setColor(color, 0);

    setPos(55, 26);

    cout << "                                ";

    setPos(55, 28);

    cout << "                                ";

    setPos(55, 27);

    cout << "    Press ENTER to start the game...    " << flush;

    sleep_for(milliseconds(500));

    color = (color == 11) ? 15 : 11;

    if (_kbhit())
    {
        char key = _getch();

        if (key == 13)
        {
            // ...
        }
    }
}
```

```
        stopBlinking = true;
    }
    else
    {
        while (_kbhit()) _getch();
    }
}
}
```

- Sử dụng đồng bộ hóa.
- a (atomic<bool>) để quản lý hiệu ứng tắt/mở.

**b) *printCenteredAt:***

- In văn bản được căn giữa tại một tọa độ cụ thể.

```
int startX = x - (str.length() / 2);
setPos(startX, y);
cout << str;
```

**2.2. Vẽ giao diện bảng và menu:**

**a) *drawBox, drawMenu1, drawMenu2, drawMenu3:***

- Vẽ hộp và các menu khác nhau với định dạng và kiểu hiển thị riêng.
- Kết hợp các ký tự đồ họa để tạo viền, nền, và nội dung.
- Ví dụ: code drawMenu1

```
setColor(gameTheme);
setPos(x, y);
cout << char(symbol.topLeft);
for (int i = 0; i < size - 2; ++i) cout << char(symbol.horiBar);
```

```
cout << char(symbol.topRight);

cout << char(symbol.blank);

setPos(x, y + 1);

cout << char(symbol.vertiBar);

setPos(x + size - 1, y + 1);

cout << char(symbol.vertiBar);

setPos(x + size, y + 1);

cout << char(symbol.blank);

setPos(x + size, y + 2);

cout << char(symbol.blank);

setPos(x, y + 2);

cout << char(symbol.botLeft);

for (int i = 0; i < size - 2; ++i) cout << char(symbol.horiBar);

cout << char(symbol.botRight);

setPos(x + 1, y + 1);

setColor(gameTheme);

cout << text;

setColor(gameTheme);
```

**b) *drawBoardSize* và *drawBoard*:**

- Vẽ bảng cờ chính với các đường lưới, khung bao quanh, và các vị trí ô cờ.

```
setColor(gameTheme);

setPos(xx, yy); cout << char(218);

for (int i = 0; i < width - 2; ++i) cout << char(196);
```

```
cout << char(191);

for (int i = 1; i < height - 1; ++i)

{

    setPos(xx, yy + i); cout << char(179);

    setPos(xx + width - 1, yy + i); cout << char(179);

}

setPos(xx, yy + height - 1); cout << char(192);

for (int i = 0; i < width - 2; ++i) cout << char(196);

cout << char(218);
```

- Sử dụng các vòng lặp để định vị và hiển thị các ký tự đồ họa.

**c) *updateBoard:***

- Cập nhật trạng thái bảng chơi (ví dụ: hiển thị các nước cờ mới).

```
setColor(gameTheme);

int newX = x - 20, newY = y - 2;

for (int y = 0; y <= boardSize * 2; ++y)

{

    for (int x = 0; x <= boardSize * 4; ++x)

    {

        setColor(x, y);

    }

}
```

### **2.3. Hiển thị thông tin và các hiệu ứng đồ họa:**

**a) *drawSavedFile, drawCaroMain, newGameText:***

- Hiển thị các chữ lớn trên giao diện (như "NEW GAME", "CARO").
- Sử dụng các hàm hỗ trợ vẽ ký tự.

```
void drawSavedFile(int XX, int YY)

{

    drawS(XX - 7, YY, gameTheme);

    drawA(XX + 2, YY, gameTheme);

    drawV(XX + 10, YY, gameTheme);

    drawE(XX + 19, YY, gameTheme);

    drawD(XX + 27, YY, gameTheme);

    drawF(XX + 42, YY, gameTheme);

    drawI(XX + 50, YY, gameTheme);

    drawL(XX + 54, YY, gameTheme);

    drawE(XX + 62, YY, gameTheme);

    drawS(XX + 70, YY, gameTheme);

    setColor(gameTheme);

}
```

***b) printTimeAtPosition:***

- Hiển thị thời gian ở vị trí cụ thể trên màn hình.

```
void printTimeAtPosition(int x, int y, int minutes, int seconds)

{

    COORD currentPos;

    saveCurPos(currentPos);

    ShowCur(0);

}
```

```
setPos(x, y);

setColor(gameTheme);

cout << minutes << ":" << (seconds < 10 ? "0" : "") << seconds << "  ";

setPos(currentPos.X, currentPos.Y);

ShowCur(1);

}
```

**c) *drawWord\_X\_WIN, drawWord\_O\_WIN, drawEndMatch:***

- Hiển thị thông báo thắng cuộc cho người chơi X hoặc O.
- Vẽ bảng popup thông báo khi kết thúc trận đấu.

```
void drawWord_X_WIN(int startX, int startY, int bground, int fground)

{

    drawX(startX, startY, bground, fground);

    drawW(startX + 11, startY, bground, fground);

    drawI(startX + 22, startY, bground, fground);

    drawN(startX + 26, startY, bground, fground);

}
```

## **2.4. Các hàm tiện ích:**

**a) *drawOutBoard* và *drawName\_LoadBoard*:**

- Hiển thị các thông báo phụ, chẳng hạn lưu trò chơi, thoát, hoặc ghi tên bảng.

```
void drawName_LoadBoard(int XX, int YY, string fileName)

{

    setPos(XX, YY + 2);

    setColor(15, 0); cout << (*selectedLanguage)["Board's Name: "];
```

```
if ((int)fileName.size() == 0) fileName = "[NULL]";

while ((int)fileName.size() <= 10) fileName.push_back(' ');

cout << fileName;

setColor(15, 0);

}
```

### **3. Draw.cpp**

- Vẽ các chữ cái trong bảng chữ cái theo kiểu ASCII:

```
void drawA(int XX, int YY, int bground, int fground) { // 6 * 8

    setPos(XX + 1, YY);

    setColor(fground, fground); cout << "  ";

    setColor(bground, fground); cout << char(187);

    setPos(XX, YY + 1);

    setColor(fground, fground); cout << " ";

    setColor(bground, fground); cout << char(201) << char(205) << char(205);

    setColor(fground, fground); cout << " ";

    setColor(bground, fground); cout << char(187);

    setPos(XX, YY + 2);

    setColor(fground, fground); cout << "  ";

    setColor(bground, fground); cout << char(186);

    setPos(XX, YY + 3);

    setColor(fground, fground); cout << " ";

    setColor(bground, fground); cout << char(201) << char(205) << char(205);
```



```
setColor(fground, fground); cout << " ";  
  
setColor(bground, fground); cout << char(186);  
  
setPos(XX, YY + 4);  
  
setColor(fground, fground); cout << " ";  
  
setColor(bground, fground); cout << char(186) << " ";  
  
setColor(fground, fground); cout << " ";  
  
setColor(bground, fground); cout << char(186);  
  
setPos(XX, YY + 5);  
  
cout << char(200) << char(205) << char(188) << " " << char(200) << char(205) <<  
char(188);  
  
}
```

#### **4. Console.cpp**

##### ***a) Cấu hình cửa sổ Console:***

```
void fixConsoleWindow(int WIDTH, int HEIGHT)  
{  
  
    system("COLOR f0");  
  
    HWND consoleWindow = GetConsoleWindow();  
  
    LONG style = GetWindowLong(consoleWindow, GWL_STYLE);  
  
    style = style & ~(WS_MAXIMIZEBOX | WS_THICKFRAME);  
  
    SetWindowLong(consoleWindow, GWL_STYLE, style);  
  
    SMALL_RECT r{};
```

```
r.Top = r.Left = 0;

r.Right = WIDTH - 1;

r.Bottom = HEIGHT - 1;

COORD NewSize{ };

NewSize.X = WIDTH;

NewSize.Y = HEIGHT;

HANDLE Handle = GetStdHandle(STD_OUTPUT_HANDLE);

MoveWindow(consoleWindow, 248, 10, NewSize.X * 16, NewSize.Y * 8, TRUE);

SetConsoleScreenBufferSize(Handle, NewSize);

SetConsoleWindowInfo(Handle, TRUE, &r);

ShowScrollBar(consoleWindow, SB_VERT, FALSE);

ShowScrollBar(consoleWindow, SB_HORZ, FALSE);

}
```

- Thiết lập kích thước và vị trí của cửa sổ console.
- Vô hiệu hóa thanh cuộn và các chức năng như phóng to cửa sổ.

**b) Quản Lý Con Trỏ**

- setPos(x, y): Đặt vị trí con trỏ tại tọa độ (x, y).

```
COORD coord = { };
```

```
coord.X = x;
```

```
coord.Y = y;
```

```
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
```

- saveCurPos(currentPos): Lưu vị trí hiện tại của con trỏ.

```
CONSOLE_SCREEN_BUFFER_INFO csp;
```

```
GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &csp);
```

```
currentPos = csp.dwCursorPosition;
```

- ShowCur(CursorVisibility): Hiển thị hoặc ẩn con trỏ.

```
HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
CONSOLE_CURSOR_INFO cursor = { 1, CursorVisibility };
```

```
SetConsoleCursorInfo(handle, &cursor);
```

#### **c) Tùy Chỉnh Màu Sắc**

- setColor(backgroundColor, foregroundColor): Đặt màu nền và màu chữ trong console.

```
HANDLE consoleHandler = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
int color = 16 * backgroundColor + foregroundColor;
```

```
SetConsoleTextAttribute(consoleHandler, color);
```

#### **d) Xử Lý Nhập Liệu**

- isNextMove(): Phân tích các phím bấm của người dùng:

```
while (true)
```

```
{
```

```
    char ch = _getch();
```

```
    if (ch == 13 || ch == ' ') return 0;
```

```
    if (ch == 27) return 5;
```

```
    if (ch == 'U' || ch == 'u') return 6;
```

```
if (ch == 'T' || ch == 'i') return 7;

if (ch == -32)

{

    ch = _getch();

    if (ch == 72) return 1;

    if (ch == 75) return 2;

    if (ch == 80) return 3;

    if (ch == 77) return 4;

}

if (ch == 'W' || ch == 'w') return 1;

if (ch == 'A' || ch == 'a') return 2;

if (ch == 'S' || ch == 's') return 3;

if (ch == 'D' || ch == 'd') return 4;

}
```

- Hỗ trợ phím mũi tên, WASD, ENTER, ESC, U, I.

***e) Vẽ Hình Khối và Trang Trí***

- drawSquare(i, j, x, y, color): Vẽ một ô vuông tại tọa độ (x, y).
- txtDraw(x, y, nameFile): Vẽ từ dữ liệu trong tệp tin (mã màu).
- decorate\_draw\_overlap(...): Vẽ trang trí với tính năng chồng hình.

```
ifstream file(nameFile.c_str());
```

```
string s;
```

```
int i = 0;

while (getline(file, s))

{

    i++;

    stringstream ss(s);

    int j = 0;

    while (!ss.eof())

    {

        j++;

        int color;

        ss >> color;

        if (color != 15) draw_square(x, y, j, i, color);

    }

}

if (isReversed == 0)

{

    draw_square(x, y, 8, 4, 15);

    draw_square(x, y, 9, 4, 15);

}

else
```

```
{  
  
    draw_square(x, y, 4, 4, 15);  
  
    draw_square(x, y, 5, 4, 15);  
  
}
```

## 5. LoadGame.cpp

### 1. LoadFile:

- Sau khi người dùng chọn LoadGame thì sẽ hiện ra một bảng chọn file lưu 3x3.

```
int size = min((int)fileSave.size() - 1, 8);  
  
loadMenu s1[10], s2[10];  
  
for (int i = 0; i < 9; ++i) {  
  
    s1[i] = { XX - 30 + 30 * (i / 3), YY + 5 + (i % 3) * 4, 20, 11,  
(*selectedLanguage)["  EMPTY  " ] };  
  
    s2[i] = { XX - 30 + 30 * (i / 3), YY + 5 + (i % 3) * 4, 20, 8, "" };  
  
  
    if (i <= size) {  
  
        string tmp = fileSave[i];  
  
        int siz = 14 - (int)tmp.size();  
  
  
        for (int j = 1; 2 * j <= siz; ++j)  
  
            tmp = ' ' + tmp + ' ';  
  
        if (siz & 1) tmp += ' ';
```

```
s1[i].str = " " + tmp + " ";  
s2[i].str = ">>" + tmp + "<<";  
s1[i].b_color = 15;  
}  
}
```

- Số file có sẵn dựa trên việc đọc một file ListNameFiles.txt, trong file này có chứa tên của những file lưu khác.
- Sử dụng mũi tên hoặc WASD để di chuyển đến file cần load.

```
int mouSe = -1;  
  
if (inputt == 1) mouSe = S - 1;  
  
else if (inputt == 2) mouSe = S - 3;  
  
else if (inputt == 3) mouSe = S + 1;  
  
else if (inputt == 4) mouSe = S + 3;  
  
if (mouSe <= size && mouSe >= 0) S = mouSe;
```

- File được trỏ đến sẽ có giao diện khác với những file còn lại để người dùng dễ sử dụng.
- Enter để chọn load file đó.
- Xuất hiện 3 lựa chọn PLAY, RENAME, DELETE.

## 2. SaveFile:

- Khi người chơi bấm “I” sẽ xuất hiện thông báo đặt tên file lưu.
- Nếu tên đã tồn tại thì người chơi có thể ghi đè hoặc đặt tên khác.
- Sau khi lưu xong thì file tự động được tạo và ghi vào trạng thái bàn cờ.

## 3. Một số tính năng khác:

- Dùng remove() để xóa file.

```
void deleteFile(string nameFile) {  
  
    nameFile += ".txt";
```

```
remove(nameFile.c_str());  
  
}
```

- Dùng hàm rename() để đổi tên file.

renameFile:

```
string tmp;  
  
setPos(XX - 24, YY + 17); cout << (*selectedLanguage)[" New Name:  
"];  
  
if (!insertName(XX - 13, YY + 17, tmp = fileSave[S])) continue;  
  
string old_name = fileSave[S] + ".txt";  
  
string new_name = tmp + ".txt";  
  
rename(old_name.c_str(), new_name.c_str());  
  
fileSave[S] = tmp;  
  
int siz = 14 - (int)tmp.size();  
  
for (int t = 1; 2 * t <= siz; ++t) tmp = ' ' + tmp + ' ';  
  
if (siz & 1) tmp += ' ';  
  
s1[S].str = " " + tmp + " ";  
  
s2[S].str = ">>" + tmp + "<<";  
  
}  
  
pushList();
```

- pullList() là hàm để cập nhật lại ListNameFiles.txt sau khi lưu.

```
void pullList() {  
  
    string s; fileSave.clear();
```



```
        ifstream file("ListNameFiles.txt");  
  
        while (file >> s) fileSave.emplace_back(s);  
  
        file.close();  
  
    }
```

- pushList() là hàm để đẩy tên file đã lưu từ ListNameFiles.txt.

```
void pushList() {  
  
    ofstream File("ListNameFiles.txt");  
  
    for (string s : fileSave) File << s << '\n';  
  
    File.close();  
  
}
```

## 6. NewGame.cpp

### 1. Các chế độ chơi:

- mode1: Người với Bot.
- mode2: Người với Người.

```
void mode1(int XX, int YY) {  
  
    system("cls");  
  
    string name;  
  
    newGameText(XX + 2, YY);  
  
    drawMenu1(XX, YY + 10, 27, 14, (*selectedLanguage)["PLAYER NAME: "]);  
  
    if (!insertName(XX + 16, YY + 11, name =  
    (*selectedLanguage)["[PLAYER]"])) return;  
  
    startGame(true, true, XX - 3, YY, name, "Bot", {}, "", 0, 0);  
  
}
```

```
void mode2(int XX, int YY) {  
  
    system("cls");  
  
    string name1, name2;  
  
    newGameText(XX + 2, YY);  
  
  
    drawMenu1(XX, YY + 10, 29, 14, (*selectedLanguage)["PLAYER NAME 1:  
"]);  
  
    if (!insertName(XX + 18, YY + 11, name1 =  
(*selectedLanguage)["player1"])) return;  
  
  
    drawMenu1(XX, YY + 13, 29, 14, (*selectedLanguage)["PLAYER NAME 2:  
"]);  
  
    if (!insertName(XX + 18, YY + 14, name2 =  
(*selectedLanguage)["player2"])) return;  
  
  
    startGame(false, true, XX - 3, YY, name1, name2, { }, "", 0, 0);  
  
}
```

## **2. Hàm newGame():**

- Vẽ giao diện, cho người chơi chọn chế độ bằng bàn phím.

```
while (true) {  
  
    newGameText(XX, YY);  
  
    for (int i = 0; i < 2; ++i)  
        a1[i].draw();  
  
    a2[S].draw();  
  
}
```

```
int inputNewGame = isNextMove();

if (inputNewGame == 1 || inputNewGame == 3) S ^= 1;

else if (inputNewGame == 0) {

    if (S == 0) mode1(XX - 2, YY);

    else if (S == 1) mode2(XX - 3, YY);

    else assert(false);

    break;

}

else if (inputNewGame == 5) break;

}
```

### **3. Hàm insertName():**

- Cho người chơi nhập tên của họ.
- Giao diện đẹp mắt.

```
while (true) {

    char a = _getch();

    if ((int)a < 0) {

        a = _getch();

        continue;

    }

    if (a == 13) break;

    if (a == 27) return 0;

    if (a == '\b' && b) {

        str.pop_back(); --b; --XX;
```

```
        setPos(XX, YY);

        cout << " ";

        setPos(XX, YY);

        if (b == 0) {

            setColor(15, 8);

            setPos(origin_XX, origin_YY);

            cout << str;

        }

    }

    else if (((('a' <= a && a <= 'z') || ('A' <= a && a <= 'Z') || isdigit(a)) && b
< 10) {

        if (b == 0) {

            str = "";

            setColor(15, 0);

            setPos(origin_XX, origin_YY);

            cout << "      ";

        }

        str.push_back(a); ++b;

        setPos(XX, YY);

        cout << a; ++XX;

    }

}
```

## 7. Language.cpp

### 1. Hàm dict():

- Hàm này dùng để dịch các câu chữ được hiển thị trên màn hình console từ tiếng anh sang tiếng việt cũng như tiếng việt sang tiếng anh bằng cách sử dụng cấu trúc dữ liệu map.

```

english[" TRO CHOI MOI "] = " NEW GAME ";
english[" TAI TRO CHOI "] = " LOAD GAME ";
english[" THONG TIN "] = " ABOUT ";
english[" TRO GIUP "] = " HELP ";
english[" NGON NGU "] = " LANGUAGE ";
english[" THOAT "] = " EXIT ";

vietnamese[" NEW GAME "] = " TRO CHOI MOI ";
vietnamese[" LOAD GAME "] = " TAI TRO CHOI ";
vietnamese[" ABOUT "] = " THONG TIN ";
vietnamese[" HELP "] = " TRO GIUP ";
vietnamese[" LANGUAGE "] = " NGON NGU ";
vietnamese[" EXIT "] = " THOAT ";

```

### 2. Hàm language():

- Hàm language() có vai trò xử lý việc lựa chọn ngôn ngữ cho người dùng. Chức năng chính của hàm này là hiển thị menu ngôn ngữ và quản lý việc người dùng chọn ngôn ngữ mong muốn từ hàm dict().

```

dict();

system("cls");

int XX = Xi - 6, YY = Yi + 1;

string language;

```

```
languageMenu a1[2], a2[2];

a1[0] = { XX + 3, YY + 10, 25, 15, (*selectedLanguage)["    ENGLISH    "] };
a1[1] = { XX + 3, YY + 15, 25, 15, (*selectedLanguage)["    VIETNAMESE   "] };
a2[0] = { XX + 4, YY + 11, 25, 15, (*selectedLanguage)[">>    ENGLISH    <<"] };
a2[1] = { XX + 4, YY + 16, 25, 15, (*selectedLanguage)[">>    VIETNAMESE   <<"] };

system("cls");

int S = 0;

while (true) {

    drawWord("Language", XX - 20, YY - 5, 15, 0);

    for (int i = 0; i < 2; ++i)

        a1[i].draw();

    a2[S].draw2();

    int inputNewGame = isNextMove();

    if (inputNewGame == 1 || inputNewGame == 3) S ^= 1;

    else if (inputNewGame == 0) {

        selectedLanguage = (S == 0) ? &english : &vietnamese;

        break;

    }

    else if (inputNewGame == 5) break;

}
```

+ drawWord hiển thị tiêu đề "Language".

+ Vòng lặp liên tục vẽ menu và chờ nhận đầu vào từ người dùng thông qua isNextMove().

- + Nếu người dùng nhấn phím điều hướng (1 hoặc 3), chỉ số S sẽ thay đổi, làm cho lựa chọn hiện tại thay đổi.
- + Khi người dùng nhấn phím "OK" (0), ngôn ngữ sẽ được cập nhật dựa trên chỉ số S và hàm sẽ thoát.
- + Nếu người dùng nhấn phím thoát (5), hàm cũng sẽ thoát mà không thay đổi ngôn ngữ.

## **8. Một số file khác: About.cpp, Help.cpp, ...**

- Sử dụng cout, setPos và setColor để vẽ một console thân thiện, đẹp mắt.