





**HỌC KỲ: 02 – NĂM HỌC: 2024 - 2025**

**Giảng viên hướng dẫn:**

**Nhóm sinh viên thực hiện**:

**BÌA LÓT**

(Nội dung bìa lót giống như bìa chính)

**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

**MỤC LỤC**

[**CHƯƠNG I. GIỚI THIỆU**](#_3znysh7) **1**

[**1.1. Lý do chọn đề tài**](#_2et92p0) **1**

[**1.2. Mục tiêu đề tài**](#_mpyci9g1xdjf) **1**

[**1.3. Phương pháp nghiên cứu**](#_3dy6vkm) **1**

[**1.4. Bố cục đề tài**](#_kbq1tlwish9y) **2**

[**CHƯƠNG II. CƠ SỞ LÝ THUYẾT**](#_1t3h5sf) **2**

[**2.1. Ngôn ngữ lập trình C++ và lập trình hướng đối tượng (OOP)**](#_nq9h60ytm4eg) **2**

[**2.1.1. Lịch sử hình thành và phát triển**](#_tvidyl58oa3o) **2**

[**2.1.2. Khái niệm**](#_1w7jlw9tbmuz) **2**

[**2.1.3. Ưu điểm và nhược điểm**](#_6coxh9gih0gh) **2**

[**2.1.4. Ứng dụng trong đề tài**](#_xyeuz9s3qi57) **3**

[**2.2. Giải thuật Heuristic trong trò chơi Caro**](#_805pqq4jznvk) **3**

[**2.2.1. Lịch sử hình thành**](#_9d6feunhvk1g) **3**

[**2.2.2. Khái niệm**](#_69aptonpmrzy) **3**

[**2.2.3. Ưu điểm và nhược điểm**](#_nq08nqc301tx) **3**

[**2.2.4. Ứng dụng trong đề tài**](#_e5lgm8ksfc1k) **3**

[**2.3. Kỹ thuật vẽ đồ họa trên giao diện console**](#_4e6fp9qj3j0e) **4**

[**2.3.1. Lịch sử hình thành**](#_8tls1jr0nj3w) **4**

[**2.3.2. Khái niệm**](#_g7ydmafc6hu3) **4**

[**2.3.3. Ưu điểm và nhược điểm**](#_qyyf4zfnd4gf) **4**

[**2.3.4. Ứng dụng trong đề tài**](#_a6iqyo3lh0dv) **4**

[**CHƯƠNG III. PHÂN TÍCH VÀ XÂY DỰNG SẢN PHẨM GAME**](#_3rdcrjn) **5**

[**3.1. Tổng quan trò chơi**](#_26in1rg) **5**

[**3.1.2. Giới thiệu trò chơi**](#_uk537bdyryr8) **5**

[**3.1.3. Gameplay**](#_r9bm9z8wwsfn) **5**

[**3.1.4. Cơ chế của trò chơi**](#_fv5vkd18geng) **5**

[**3.1.5. Điều khiển**](#_dqqx9ea67xh0) **6**

[**3.1.6. Cốt truyện**](#_3ot617amfehb) **6**

[**3.1.7. Thế giới game**](#_t81vuncewp94) **6**

[**3.1.8. Nhân vật và kẻ địch**](#_i8267ngfis3h) **7**

[**3.1.9. Màn chơi**](#_lezq1gtsbbqj) **7**

[**3.1.10. UI**](#_38y5cw3jdpe1) **7**

[**3.2. Phân tích hệ thống game**](#_x55ho59o0aju) **8**

[**3.2.1. Các tác nhân của hệ thống**](#_hpizkaceh7ao) **8**

[**3.2.2. Chức năng hệ thống**](#_dcc81pa24orm) **8**

[**3.2.3. Biểu đồ use-case**](#_lurx3ivpmb7j) **9**

[**3.2.4. Biểu đồ lớp (Class Diagram)**](#_oijhgyuj85u0) **9**

[**3.2.5. Biểu đồ hoạt động (Activity Diagram)**](#_nqpzy4ntv7is) **10**

[**3.2.6. Biểu đồ tuần tự (Sequence Diagram)**](#_29noiz9hyr5d) **11**

[**3.2.7. Biểu đồ trạng thái (State Diagram)**](#_rrdmjrqub8eb) **11**

[**3.3. Xây dựng các phần thành của Game**](#_fi9glbk08edk) **12**

[**3.4. Xây dựng giao diện sản phẩm Game**](#_fi9glbk08edk) **16**

[**CHƯƠNG IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**](#_lnxbz9) **20**

**DANH MỤC HÌNH ẢNH**

[**Hình 3.1.10: Màn hình chính**](#_3c8ldura63t7) **8**

[**Hình 3.2.3: Biểu đồ use-case**](#_i8f73doth0am) **9**

[**Hình 3.2.4: Biểu đồ lớp**](#_f8ukyyv3e260) **9**

[**Hình 3.2.5: Biểu đồ hoạt động**](#_dvlb8fy7a6gr) **10**

[**Hình 3.2.6: Biểu đồ tuần tự**](#_8uvm8rh1ud4l) **11**

[**Hình 3.2.7: Biểu đồ trạng thái**](#_3oni5lzclr6i) **11**

[**Hình 3.3.1: Hình hàm \_Common.h**](#_vpivw65aw4n0) **12**

[**Hình 3.3.2: Hình hàm \_Point.h**](#_3ssapb8vt5iz) **12**

[**Hình 3.3.3: Hình hàm \_Board.h**](#_mntzbiboc1yp) **13**

[**Hình 3.3.4: Hình hàm \_Game.h**](#_vabppc7hl70o) **14**

[**Hình 3.3.4: Hình hàm \_Play.h**](#_jq2fn0mogsbj) **16**

[**Hình 3.4.1: Màn hình menu**](#_wvwh12dcl4kp) **17**

[**Hình 3.4.2: Màn hình đấu người chơi**](#_kvl0q5lfkkkr) **17**

[**Hình 3.4.3: Màn hình đấu máy**](#_irf1q5agvmkx) **18**

[**Hình 3.4.4: Màn hình hướng dẫn**](#_i7zb211644g7) **18**

[**Hình 3.4.5: Màn hình chiến thắng**](#_ueyxly7kfqg1) 19

# 

**DANH MỤC BẢNG BIỂU**

(Danh mục các bảng biểu trong Đề tài nếu có)

**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

(Danh mục các thuật ngữ và từ viết tắt trong Đề tài nếu có)

|  |  |
| --- | --- |
| **Thuật ngữ** | **Ý nghĩa** |
|  |  |

# **LỜI NÓI ĐẦU**

**CHƯƠNG I. GIỚI THIỆU**

**1.1. Lý do chọn đề tài**

**Sự đơn giản của đề tài:** game Caro hay còn được nhiều người Việt biết đến với cái tên X và O đã gắn liền với tuổi thơ, thời học sinh của bao người.

**Hoạt động trí não:** Giải trí sau giờ học tập và mang tiếp cận tính chiến thuật cao.

**Nắm chắc nền tảng:** Giúp củng cố nền tảng vững chắc C++, xử lý đồ hoạ console, cấu trúc dữ liệu và thuật toán.

**1.2. Mục tiêu đề tài**

* Hoàn thành yêu cầu của môn học.
* Thực hành bằng C++ và làm quen nhiều thư viện khác nhau.

**1.3. Phương pháp nghiên cứu**

* Game Cờ Caro được xây dựng bằng C++.
* Cung cấp chức năng chế độ chơi: người đánh với người, người đánh với máy.
* Thuật giải áp dụng :
* Vét cạn thông minh Heuristic ( tìm nước đi cho máy ).
* Các thuật giải cơ bản và kỹ thuật khác .

**1.4. Bố cục đề tài**

Game Caro là một trò chơi phổ biến, dễ tiếp cận mang tính chiến thuật cao, trong trò chơi Caro được viết bằng C++ giúp chúng ta củng cố kiến thức nền tảng lập trình và học hỏi thêm thuật toán cấu trúc dữ liệu. Game Caro có 2 chế độ chơi: đấu với người và đấu với máy, riêng đấu với máy được tạo ra thử thách thú vị được áp dụng giải thuật vét cạn thông minh Heuristic để tìm nước đi và đặc biệt game được phát triển bởi giao diện đồ hoạ console đơn giản nhưng trực quan được hiển thị rõ ràng, hỗ trợ hướng dẫn người chơi.

**CHƯƠNG II. CƠ SỞ LÝ THUYẾT**

**2.1. Ngôn ngữ lập trình C++ và lập trình hướng đối tượng (OOP)**

**2.1.1. Lịch sử hình thành và phát triển**

C++ được phát triển bởi Bjarne Stroustrup vào đầu những năm 1980 tại Bell Labs như một bản mở rộng của ngôn ngữ C. Ban đầu, C++ được gọi là “C với các lớp” (C with Classes) vì nó bổ sung các tính năng lập trình hướng đối tượng vào ngôn ngữ C. Đến năm 1985, phiên bản chính thức đầu tiên của C++ được phát hành, sau đó tiếp tục được cải tiến với các chuẩn C++98, C++11, C++14, C++17 và C++20.

**2.1.2. Khái niệm**

C++ là một ngôn ngữ lập trình đa năng hỗ trợ cả lập trình hướng thủ tục (procedural) và lập trình hướng đối tượng (OOP). Một số đặc điểm chính của C++ gồm:

* Tính đóng gói (Encapsulation): Bảo vệ dữ liệu bằng cách giới hạn quyền truy cập.
* Tính kế thừa (Inheritance): Tái sử dụng mã nguồn bằng cách kế thừa thuộc tính và phương thức từ lớp cha.
* Tính đa hình (Polymorphism): Cho phép một hàm có nhiều cách hoạt động khác nhau.

**2.1.3. Ưu điểm và nhược điểm**

Ưu điểm:

* Hiệu suất cao, phù hợp với các ứng dụng yêu cầu xử lý nhanh.
* Kiểm soát tốt bộ nhớ, giúp tối ưu chương trình.
* Hỗ trợ lập trình hướng đối tượng giúp quản lý mã nguồn tốt hơn.

Nhược điểm:

* Cú pháp phức tạp hơn so với các ngôn ngữ lập trình hiện đại như Python hoặc JavaScript.
* Quản lý bộ nhớ thủ công có thể dẫn đến lỗi rò rỉ bộ nhớ nếu không được xử lý cẩn thận.

**2.1.4. Ứng dụng trong đề tài**

Trong trò chơi Caro, C++ được sử dụng để:

* Xây dựng các lớp quản lý bàn cờ (Board), người chơi (Player) và trò chơi (Game).
* Xử lý vòng lặp trò chơi, nhập xuất dữ liệu từ người chơi.
* Triển khai thuật toán AI để đánh giá nước đi.

**2.2. Giải thuật Heuristic trong trò chơi Caro**

**2.2.1. Lịch sử hình thành**

Heuristic (thuật toán tìm kiếm có kinh nghiệm) được phát triển dựa trên nguyên tắc đánh giá và lựa chọn giải pháp tối ưu nhất trong một không gian tìm kiếm lớn. Đây là phương pháp thường được áp dụng trong AI, đặc biệt trong các trò chơi như cờ vua, cờ vây, Caro,...

**2.2.2. Khái niệm**

Thuật toán Heuristic trong game Caro đánh giá các nước đi dựa trên điểm số của từng vị trí trên bàn cờ. Mỗi ô cờ được gán một điểm số dựa trên số quân cờ X hoặc O xung quanh. Những ô có khả năng tạo ra chuỗi liên tiếp cao sẽ có điểm số lớn hơn, từ đó giúp AI chọn nước đi tối ưu.

**2.2.3. Ưu điểm và nhược điểm**

Ưu điểm:

* Tốc độ xử lý nhanh hơn so với thuật toán Minimax hoặc Alpha-Beta Pruning.
* Có thể dễ dàng tùy chỉnh mức độ khó của AI bằng cách điều chỉnh cách đánh giá điểm số.

Nhược điểm:

* Không đảm bảo tìm ra nước đi tối ưu nhất mà chỉ đưa ra nước đi tốt nhất theo đánh giá cục bộ.
* Dễ bị khai thác nếu đối thủ hiểu rõ chiến thuật AI.

**2.2.4. Ứng dụng trong đề tài**

* Computer sử dụng Heuristic để đánh giá các nước đi hợp lệ và chọn nước đi có điểm số cao nhất.
* Các yếu tố như số lượng quân liên tiếp, số hướng bị chặn sẽ ảnh hưởng đến điểm số.
* Có thể kết hợp với các thuật toán tìm kiếm sâu hơn để tăng độ khó.

**2.3. Kỹ thuật vẽ đồ họa trên giao diện console**

**2.3.1. Lịch sử hình thành**

Giao diện console là một dạng hiển thị văn bản trên màn hình, thường được sử dụng trong lập trình hệ thống hoặc các ứng dụng đơn giản. Trước khi các thư viện đồ họa phát triển mạnh, console là cách phổ biến để hiển thị dữ liệu và giao tiếp với người dùng.

**2.3.2. Khái niệm**

Giao diện console trong C++ sử dụng các hàm nhập xuất tiêu chuẩn như cout, cin, printf, và các thư viện như conio.h hoặc windows.h để tạo hiệu ứng đồ họa đơn giản như:

* In bàn cờ bằng ký tự ASCII.
* Di chuyển con trỏ trên màn hình mà không cần xóa toàn bộ màn hình.
* Thay đổi màu sắc chữ và nền.

**2.3.3. Ưu điểm và nhược điểm**

Ưu điểm:

* Dễ lập trình và không yêu cầu thư viện đồ họa phức tạp.
* Tương thích với hầu hết các môi trường lập trình C++.

Nhược điểm:

* Không trực quan và đẹp mắt như giao diện đồ họa hiện đại.
* Khó hiển thị các hiệu ứng động phức tạp.

**2.3.4. Ứng dụng trong đề tài**

* Bàn cờ được hiển thị bằng các ký tự ASCII (X, O, -).
* Sử dụng gotoxy() để di chuyển con trỏ giúp nhập liệu mượt hơn.
* Tạo hiệu ứng đổi màu khi đến lượt chơi.

**CHƯƠNG III. PHÂN TÍCH VÀ XÂY DỰNG SẢN PHẨM GAME**

**3.1. Tổng quan trò chơi**

**3.1.2. Giới thiệu trò chơi**

Caro là một trò chơi trí tuệ phổ biến, đặc biệt được ưa chuộng tại Việt Nam và nhiều nước châu Á. Trò chơi có cách chơi đơn giản nhưng đòi hỏi tư duy chiến thuật cao, giúp rèn luyện khả năng suy luận logic và tính toán của người chơi.

Game Caro được triển khai bằng ngôn ngữ lập trình **C++**, tận dụng các đặc điểm của lập trình hướng đối tượng (OOP) để tổ chức mã nguồn một cách rõ ràng, dễ bảo trì. Chương trình cung cấp hai chế độ chơi:

* **Chế độ chơi với người:** Hai người chơi luân phiên đánh dấu "X" và "O" trên bàn cờ.
* **Chế độ chơi với máy:** Người chơi đấu với AI, trong đó AI sử dụng thuật toán **Heuristic** để tìm nước đi tối ưu.

Giao diện trò chơi được hiển thị trên **console**, sử dụng các ký tự ASCII để thể hiện bàn cờ và các quân cờ.

**3.1.3. Gameplay**

Game Caro là một trò chơi chiến thuật theo lượt dành cho hai người chơi hoặc một người chơi đấu với máy. Trò chơi diễn ra trên một bàn cờ dạng lưới, nơi người chơi luân phiên đặt quân cờ của mình (X hoặc O) vào ô trống với mục tiêu tạo ra một hàng liên tục gồm 5 quân theo hàng ngang, dọc hoặc chéo để giành chiến thắng.

### Cách chơi tổng quan:

* Chế độ Player vs Player: Hai người chơi lần lượt đặt quân cờ trên bàn cờ. Người nào đạt được 5 quân liên tiếp trước sẽ thắng.
* Chế độ Player vs Computer: Người chơi đấu với máy tính. Computer sẽ phân tích bàn cờ và đưa ra nước đi dựa trên thuật toán Heuristic.
* Giao diện Console: Trò chơi chạy trên nền tảng console, hiển thị bàn cờ và thông báo kết quả theo từng lượt.

Tuy đơn giản nhưng game Caro yêu cầu chiến thuật tốt để kiểm soát bàn cờ và ngăn chặn đối phương chiến thắng. Trong các phần tiếp theo, chúng ta sẽ đi sâu vào các yếu tố cụ thể như cơ chế điều khiển, cốt truyện, nhân vật, UI, và các thành phần khác của trò chơi.

**3.1.4. Cơ chế của trò chơi**

Game Caro là một trò chơi chiến thuật theo lượt, trong đó hai người chơi (hoặc một người chơi với AI) thay phiên nhau đặt quân cờ trên bàn cờ. Người chơi nào đạt được 5 quân liên tiếp theo hàng ngang, dọc hoặc chéo mà không bị chặn hai đầu sẽ giành chiến thắng. Nếu bàn cờ đầy mà không có ai chiến thắng, ván đấu kết thúc với kết quả hòa.

Bàn cờ được vẽ bằng ký tự ASCII trên giao diện console, với hai loại quân cờ chính:

* "X": Quân cờ của người chơi 1.
* "O": Quân cờ của người chơi 2 (hoặc máy).

Các chế độ chơi:

* Chế độ đấu với người: Hai người chơi cùng nhau thi đấu trên một bàn cờ.
* Chế độ đấu với máy: Người chơi đấu với Computer, Computer sử dụng thuật toán Heuristic để tìm nước đi tối ưu.

**3.1.5. Điều khiển**

Người chơi nhập phím W, S, A, D để di chuyển tọa độ ô muốn đánh dấu theo định dạng hàng, cột. Cách nhập có thể là:

* Sử dụng phím W, S, A, D (ví dụ: W -> lên, S -> xuống, D -> phải, A -> trái) để di chuyển toạ độ ô cần đánh.
* Nhấn Enter để xác nhận nước đi.

Các phím chức năng:

* ESC: Thoát game quay về menu.
* Y: Tiếp tục đánh.

**3.1.6. Cốt truyện**

Game Caro là một trò chơi chiến thuật cổ điển. Tuy nhiên, có thể xây dựng một số bối cảnh để tạo sự hấp dẫn hơn như: người chơi mang tính chiến thuật đi những nước cờ kiểm soát ván đấu để tạo thử thách giới hạn bản thân.

**3.1.7. Thế giới game**

Thế giới trong game Caro được thể hiện qua bàn cờ 2D trên console. Kích thước bàn cờ mà không gian trò chơi như: tiêu chuẩn, mang tính chiến thuật cao.

Mỗi ô trên bàn cờ được xác định bằng tọa độ và có thể chứa:

* Quân X của người chơi 1.
* Quân O của người chơi 2 hoặc Computer.
* Ô trống để tiếp tục đặt quân cờ.

**3.1.8. Nhân vật và kẻ địch**

Nếu đấu với người:

* Người chơi 1 (Player 1): Điều khiển quân cờ X.
* Người chơi 2 (Player 2): Điều khiển quân cờ O.

Nếu đấu với Computer:

* Phân tích bàn cờ và dự đoán nước đi dựa trên thuật toán Heuristic.
* Tấn công và phòng thủ dựa trên thế trận hiện tại.

**3.1.9. Màn chơi**

Trò chơi Caro có màn chơi dựa trên chế độ và quy mô bàn cờ:

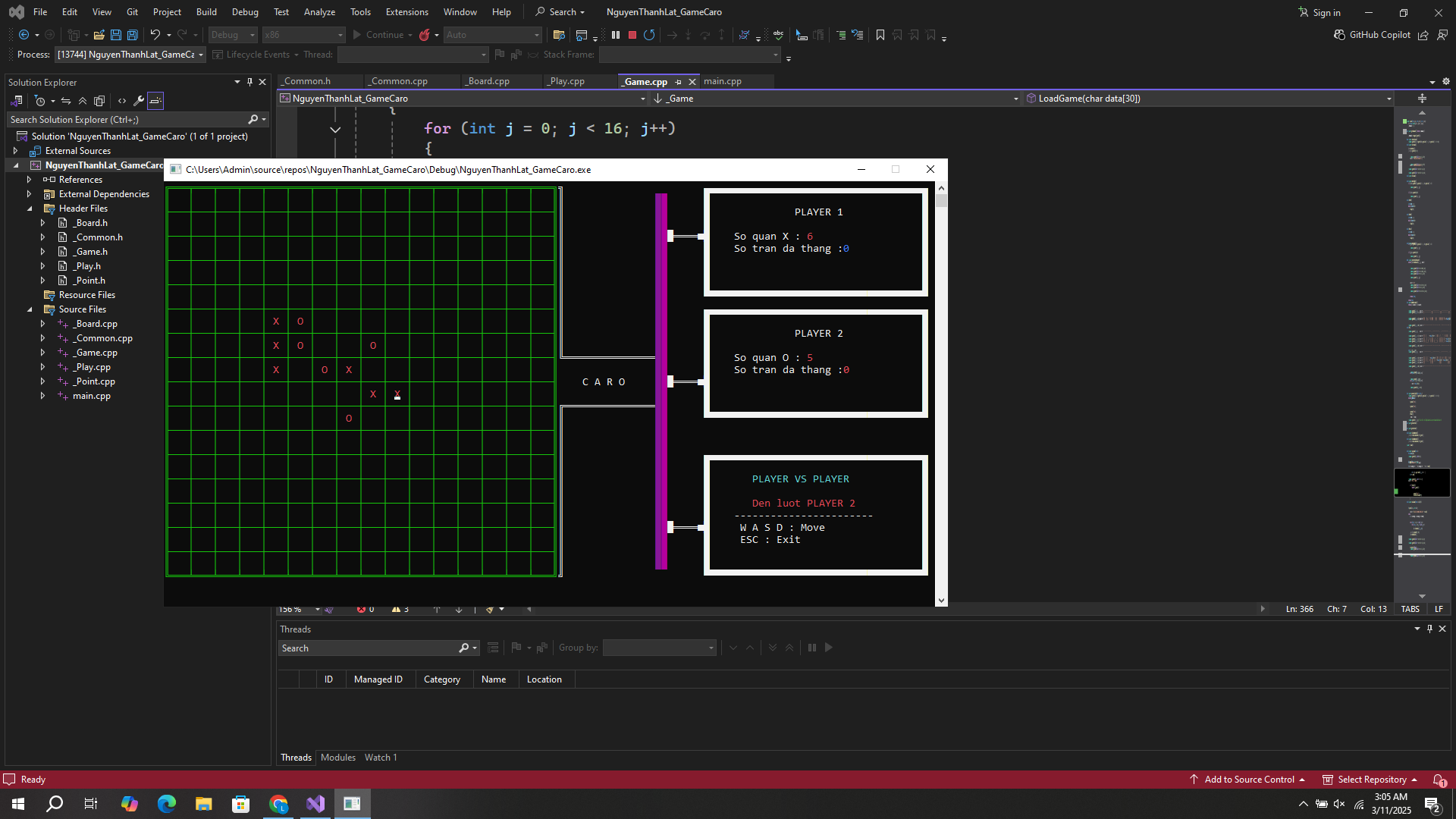
* Màn chơi 16x16: là một ván cờ với quy tắc chuẩn.
* Màn chơi thử thách (nếu mở rộng):
  + Người chơi phải thắng trong số lượt đi giới hạn.
  + Chơi với Computer có cấp độ khác nhau.
  + Bàn cờ có chướng ngại vật (một số ô không thể đi vào).

.

**3.1.10. UI**

Trò chơi được hiển thị trên console, UI chủ yếu bao gồm:

* Bàn cờ hiển thị trực quan với ký tự ASCII.
* Thông tin lượt chơi (hiển thị người chơi nào đang đánh).
* Thông báo trạng thái trò chơi (chiến thắng, hòa, nước đi sai, v.v.).
* Tùy chọn menu: Bắt đầu lại, chế độ đấu, thoát game.



*Hình 3.1.10: Màn hình chính.*

**3.2. Phân tích hệ thống game**

**3.2.1. Các tác nhân của hệ thống**

* Người chơi 1 (Player 1): Người điều khiển quân cờ X.
* Người chơi 2 (Player 2) hoặc Computer: Đối thủ của người chơi 1, có thể là người chơi khác hoặc máy tính.
* Hệ thống trò chơi: Chịu trách nhiệm xử lý nước đi, kiểm tra thắng/thua.

**3.2.2. Chức năng hệ thống**

Hệ thống game Caro có các chức năng chính như sau:

1. Chế độ chơi:

* Chơi với người.
* Chơi với Computer.

1. Xử lý nước đi:

* Kiểm tra nước đi hợp lệ.
* Cập nhật bàn cờ sau mỗi lượt chơi.

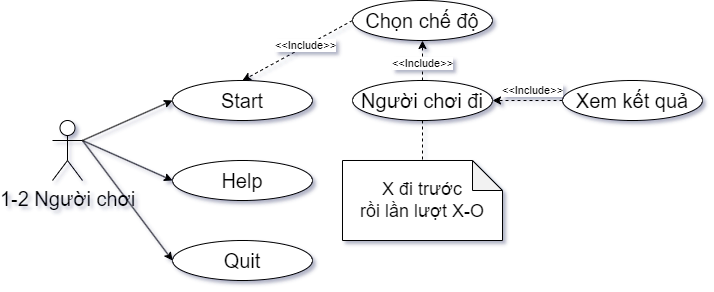
1. Kiểm tra điều kiện thắng/thua/hòa:

* Xác định người chiến thắng khi có 5 quân liên tiếp.
* Xử lý ván đấu hòa nếu bàn cờ đầy.

1. Giao diện người dùng:

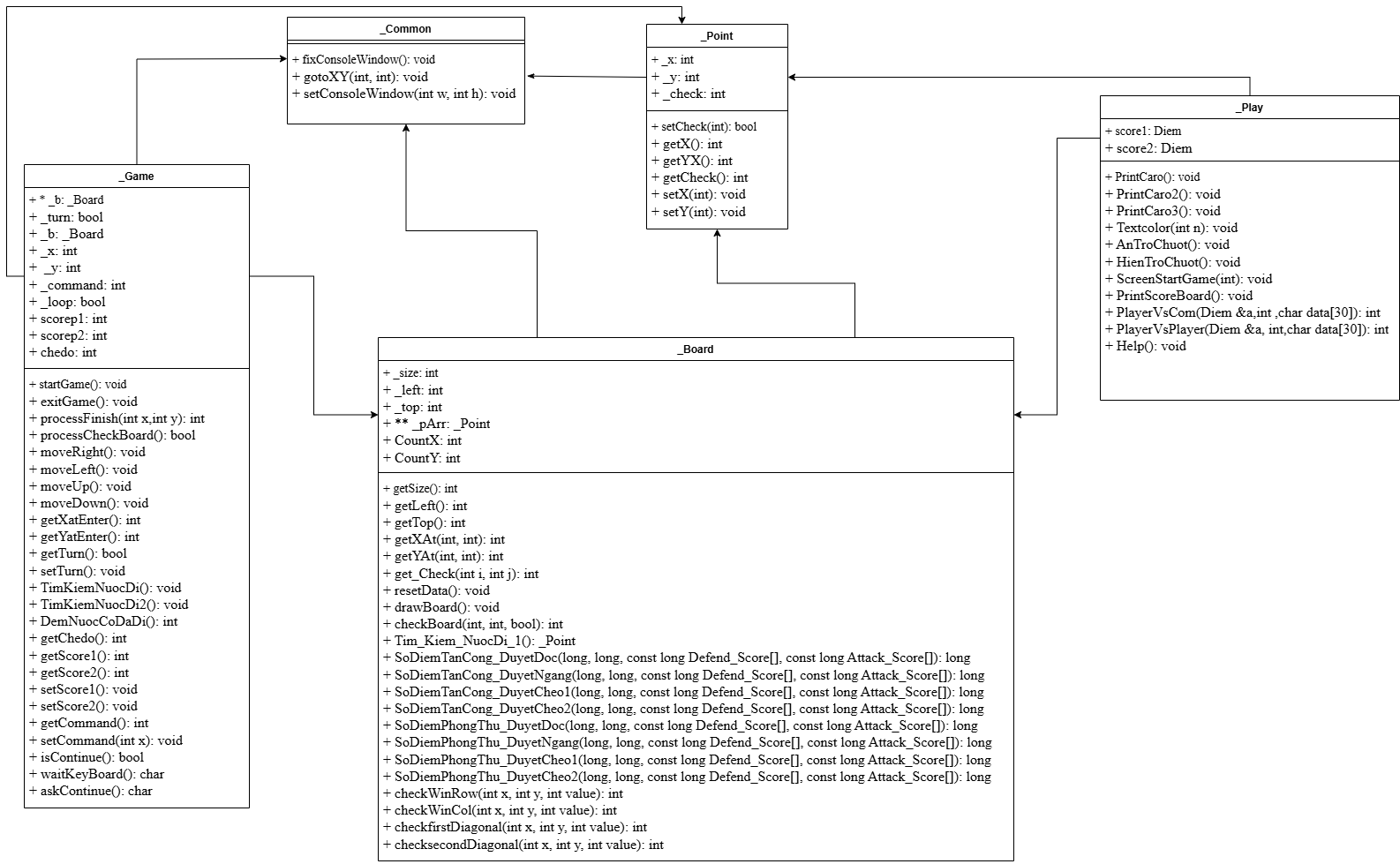
* Hiển thị bàn cờ trên console.
* Hiển thị thông tin lượt chơi, thông báo kết quả.

**3.2.3. Biểu đồ use-case**

****

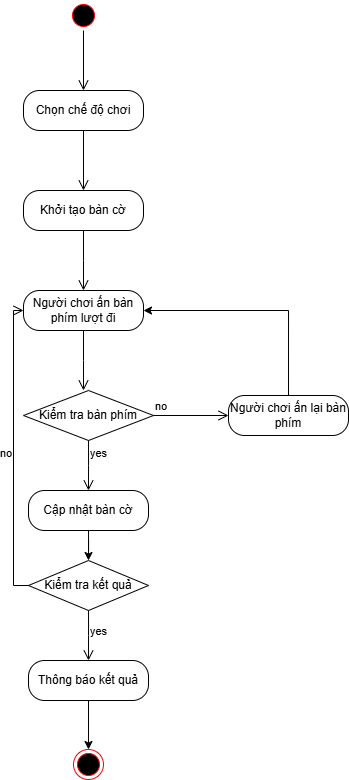
*Hình 3.2.3: Biểu đồ use-case*

**3.2.4. Biểu đồ lớp (Class Diagram)**

****

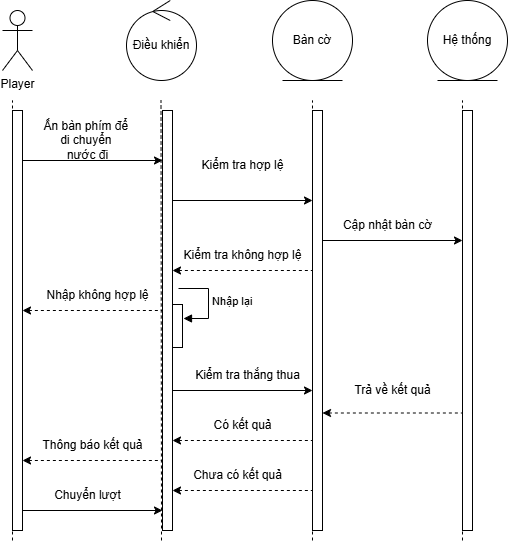
*Hình 3.2.4: Biểu đồ lớp.*

**3.2.5. Biểu đồ hoạt động (Activity Diagram)**

****

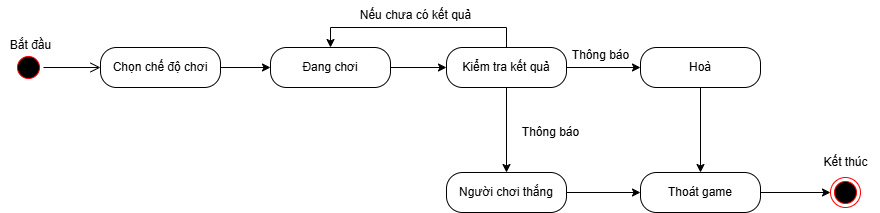
*Hình 3.2.5: Biểu đồ hoạt động.*

**3.2.6. Biểu đồ tuần tự (Sequence Diagram)**

****

*Hình 3.2.6: Biểu đồ tuần tự.*

**3.2.7. Biểu đồ trạng thái (State Diagram**)

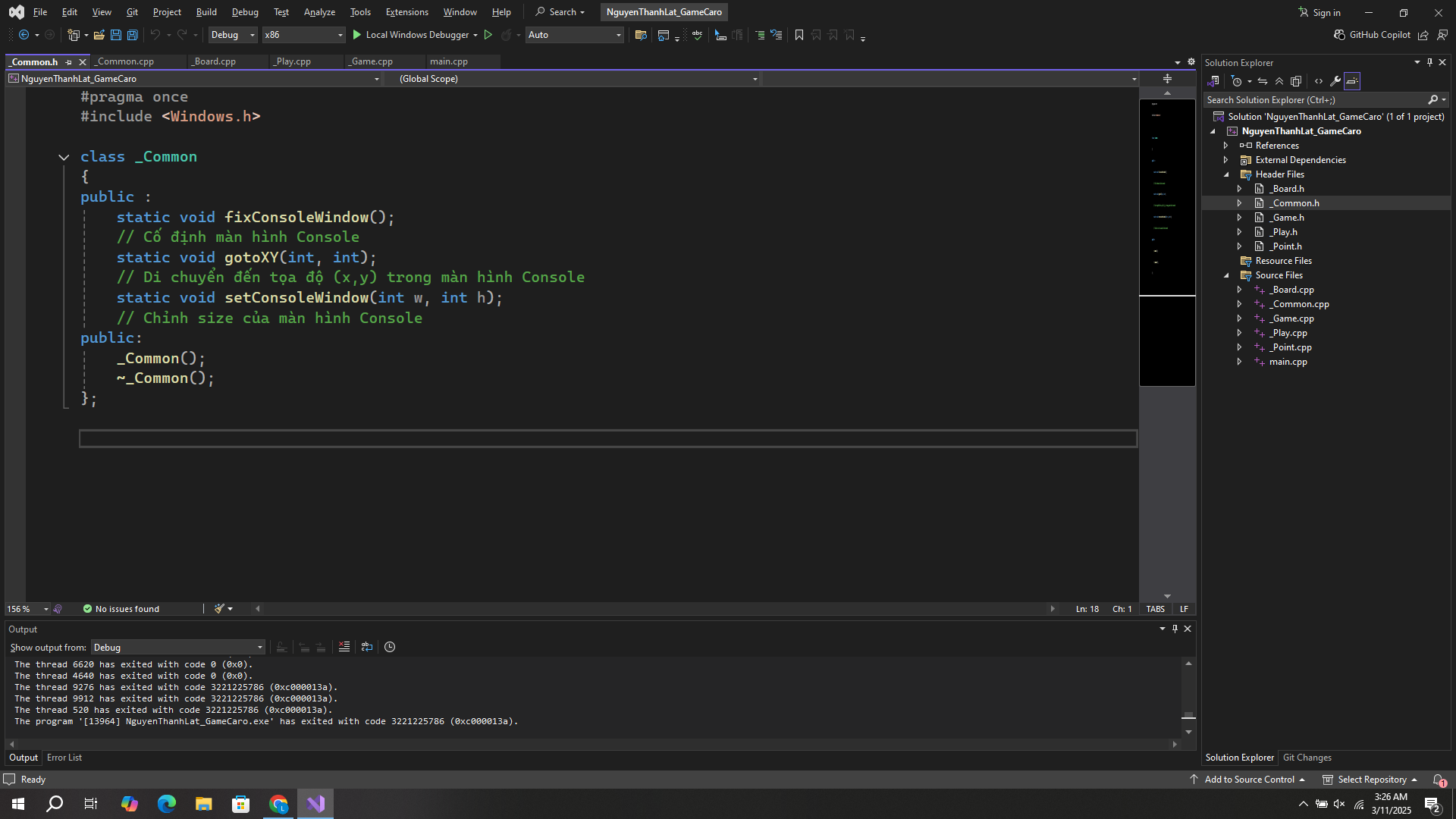
****

*Hình 3.2.7: Biểu đồ trạng thái.*

**3.3. Xây dựng các phần thành của Game**

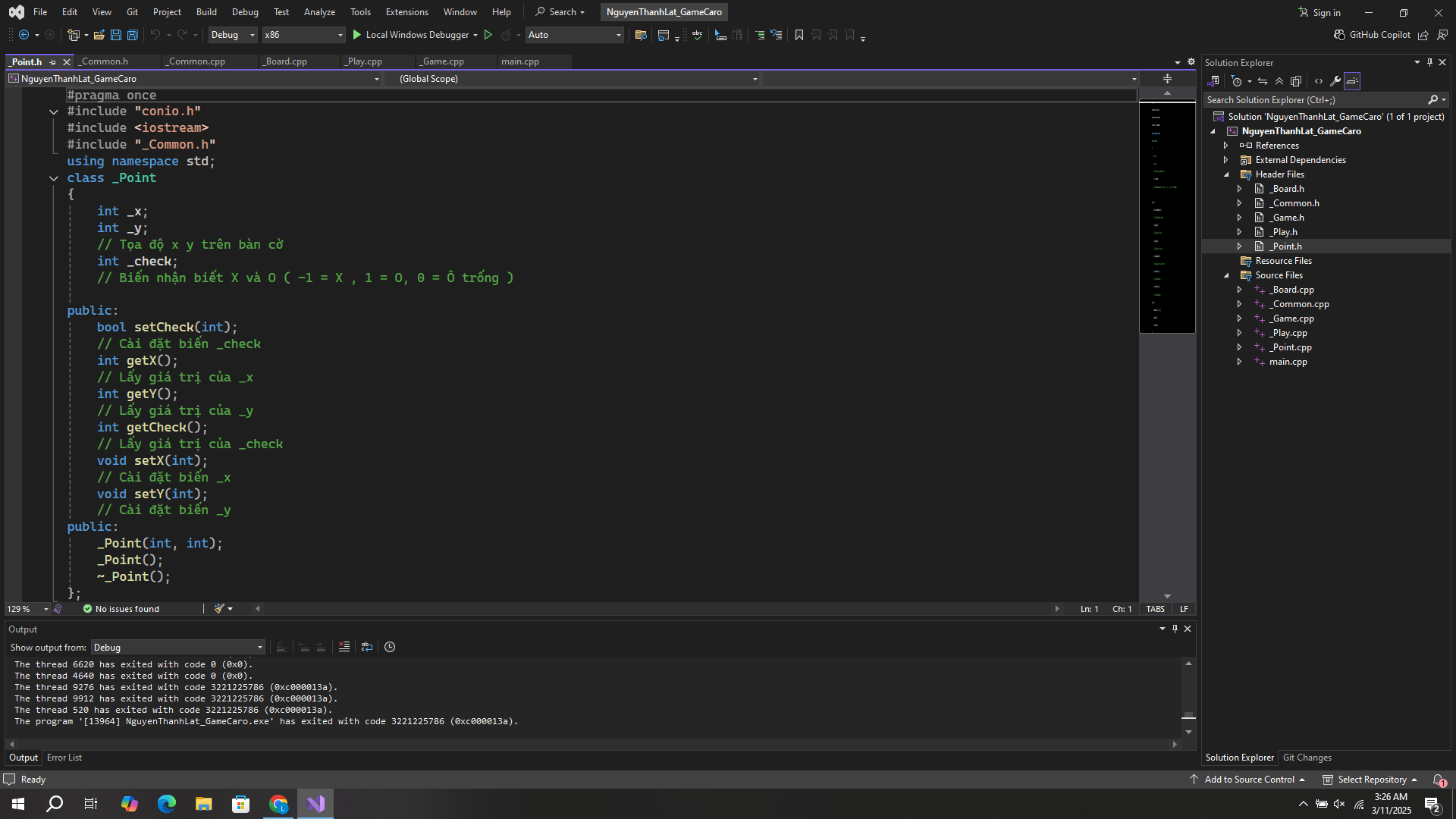
**Chức năng các hàm trong file .h :**

**\_Common.h : Xử lí các hàm thông dụng**



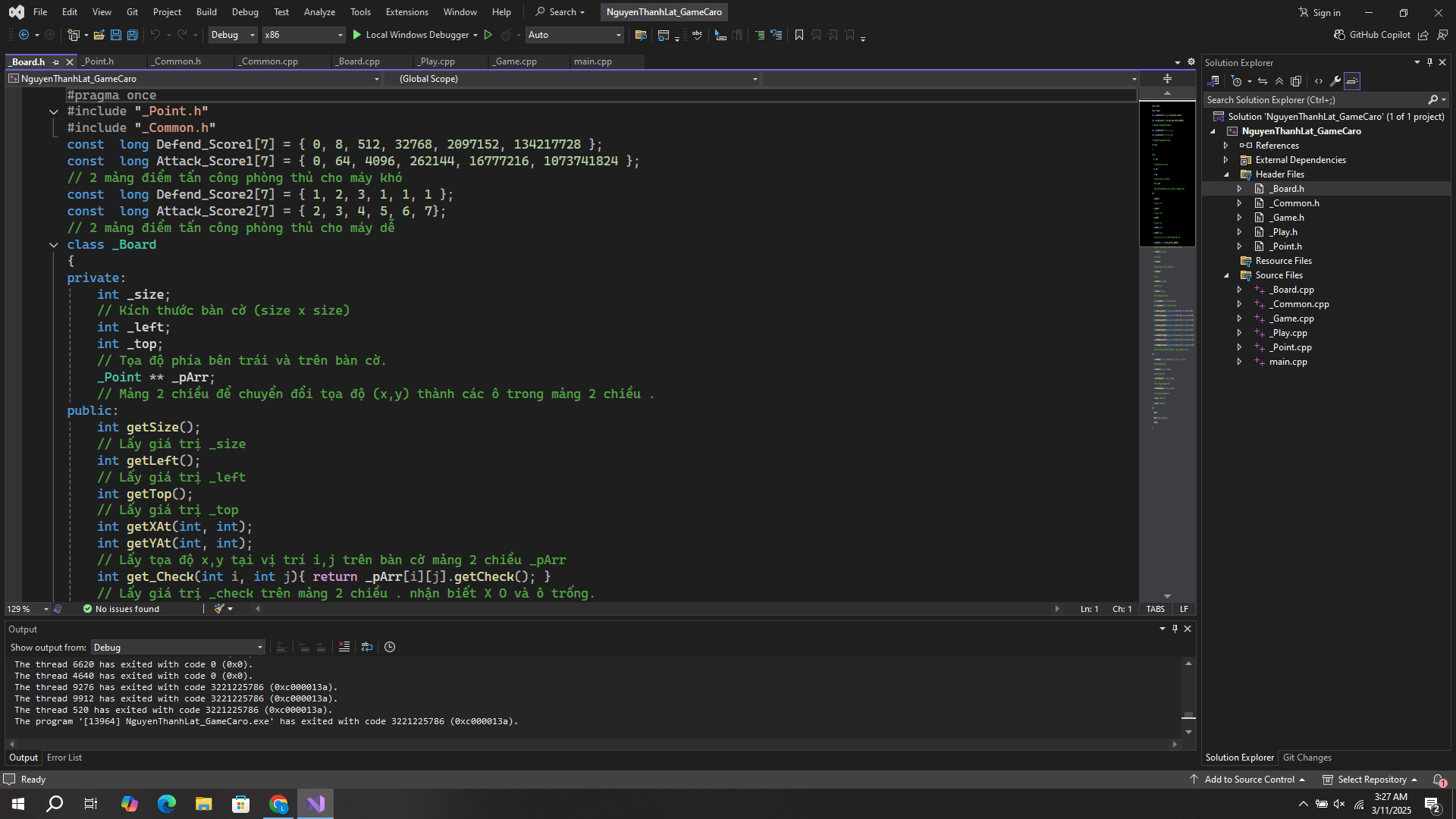
*Hình 3.3.1: Hình hàm \_Common.h.*

**\_Point.h : Các hàm xử lí tọa độ trên console và bàn cờ.**

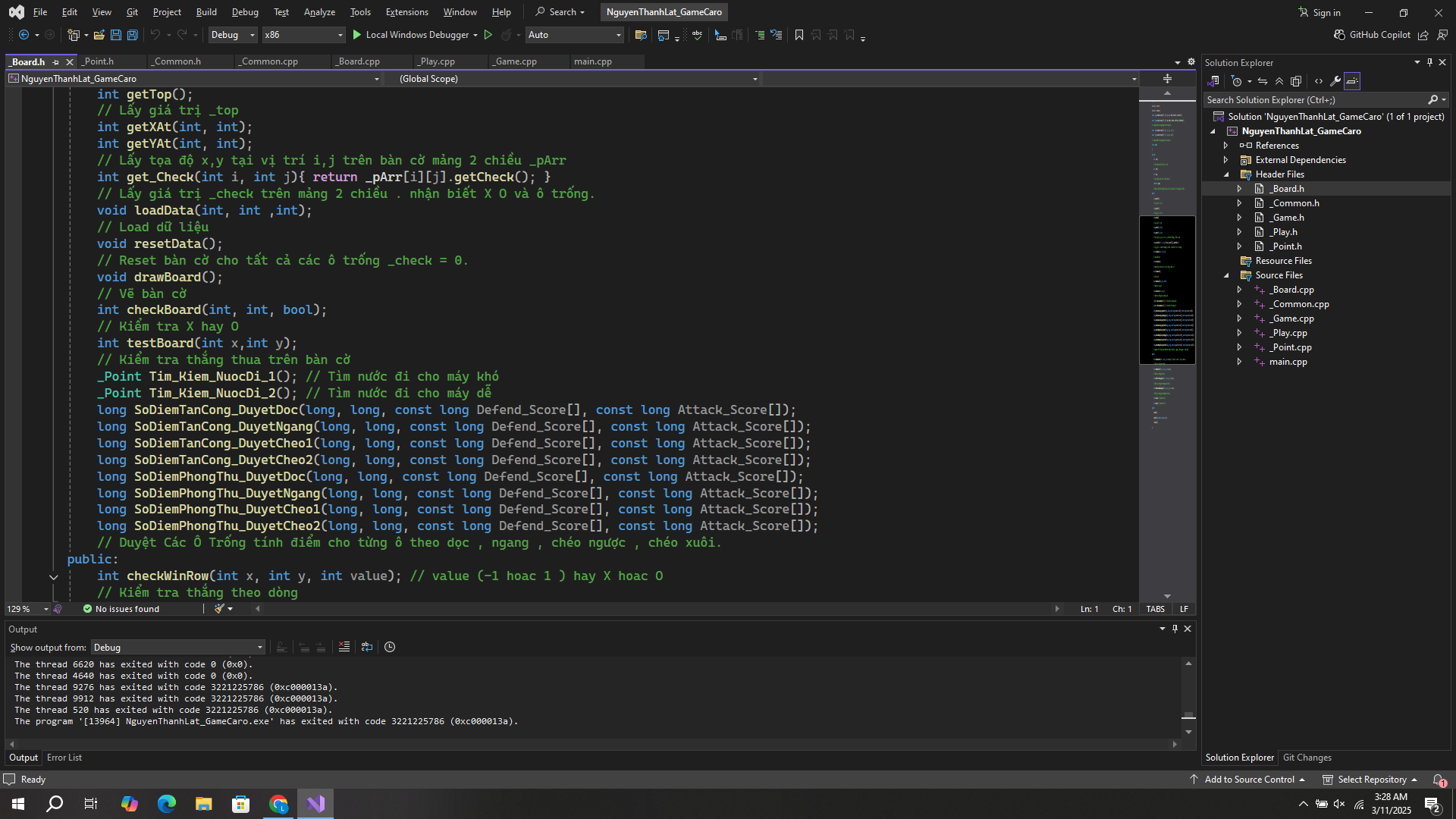


*Hình 3.3.2: Hình hàm \_Point.h.*

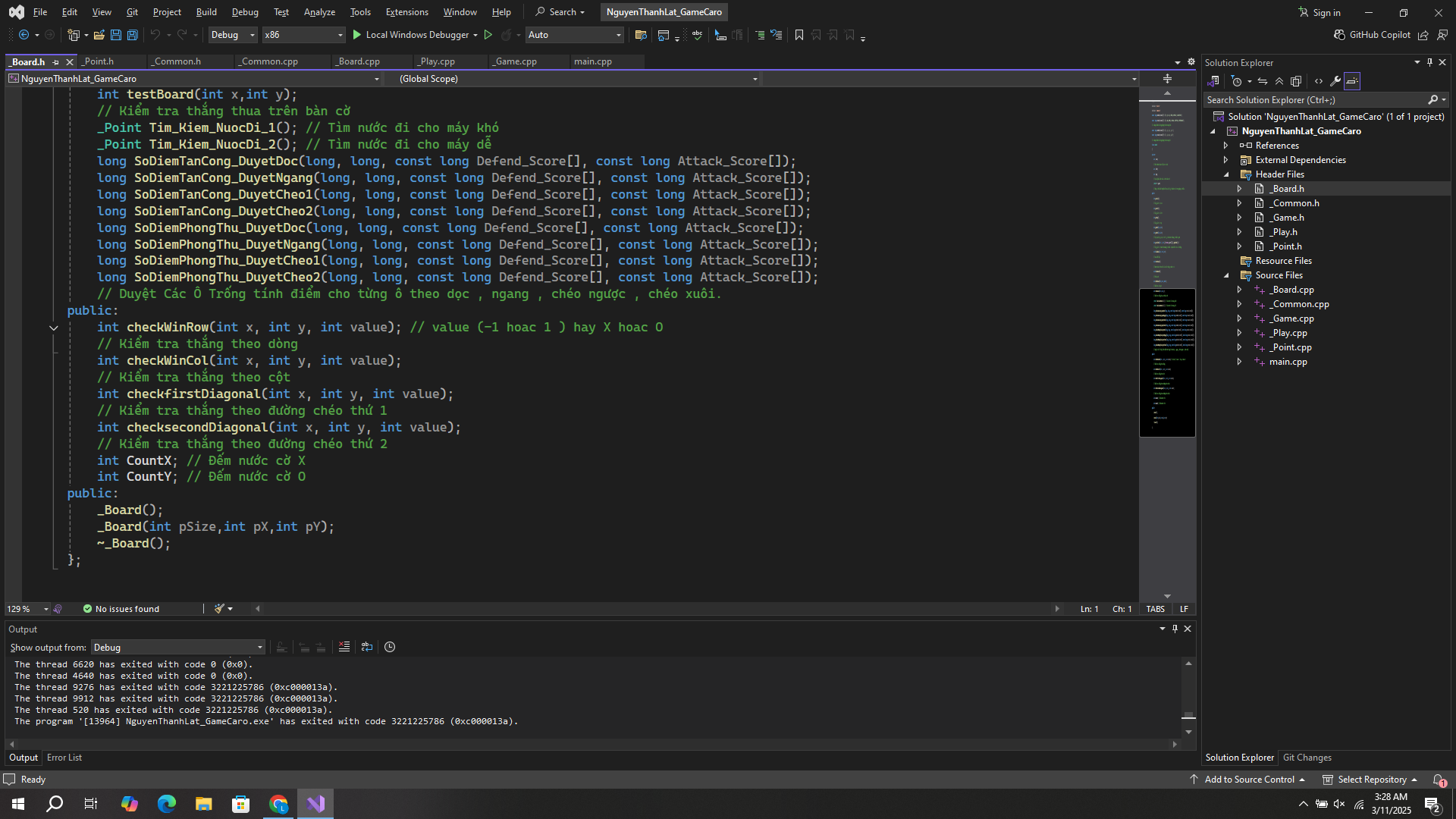
**\_Board.h : Xử lí trên bàn cờ .**



*Hình 3.3.3: Hình hàm \_Board.h.*

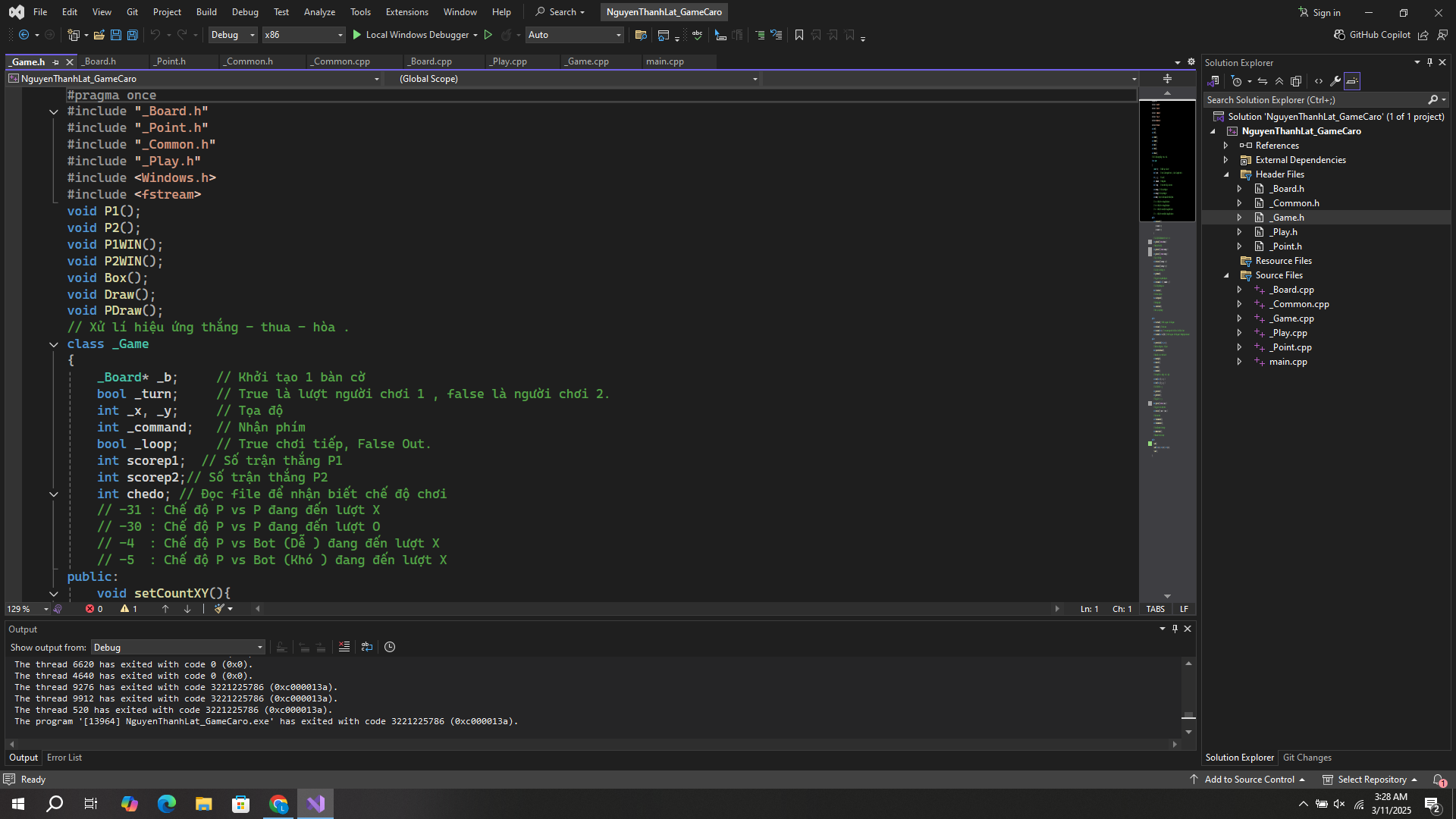


*Hình 3.3.3: Hình hàm \_Board.h.*

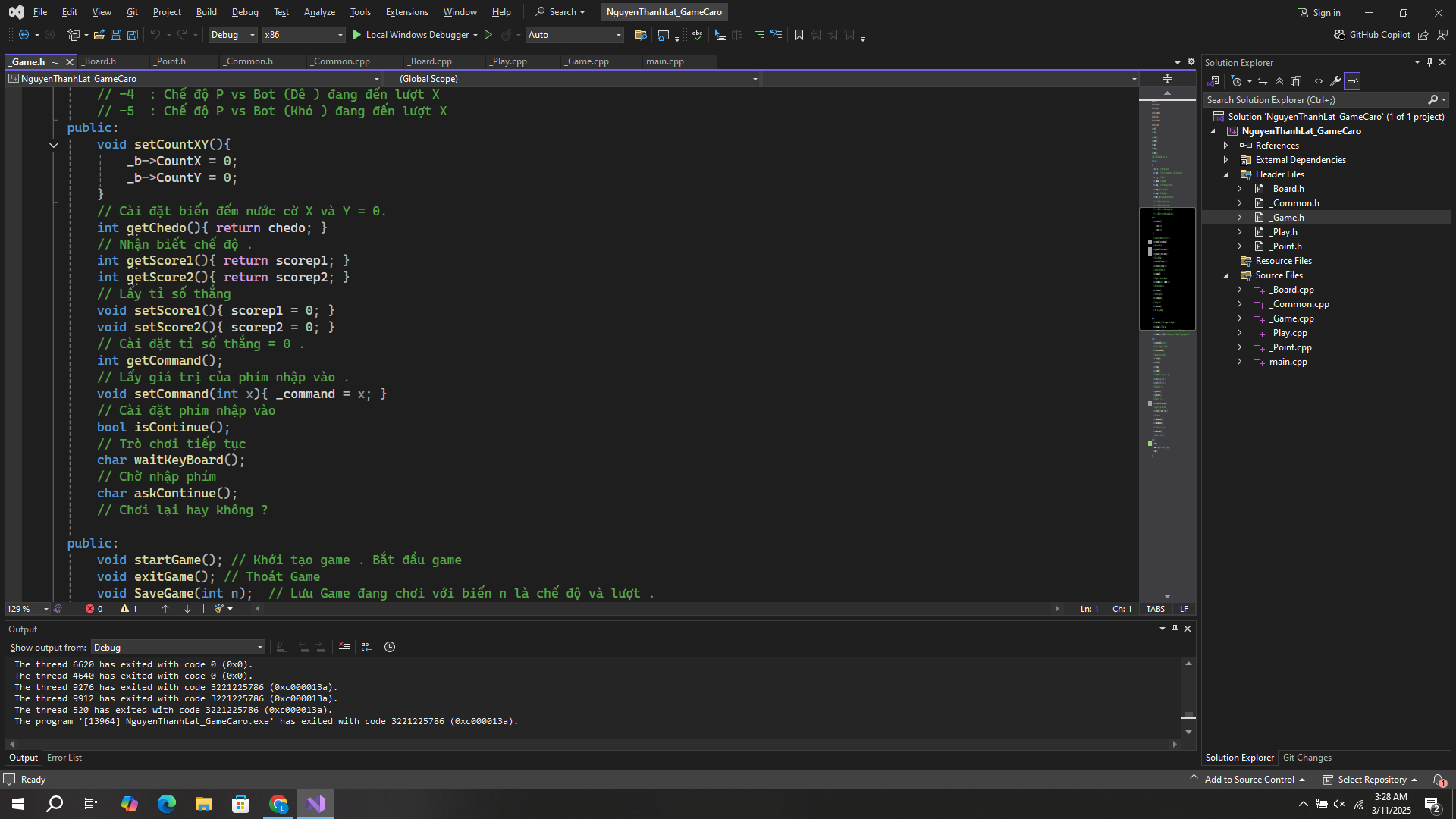


*Hình 3.3.3: Hình hàm \_Board.h.*

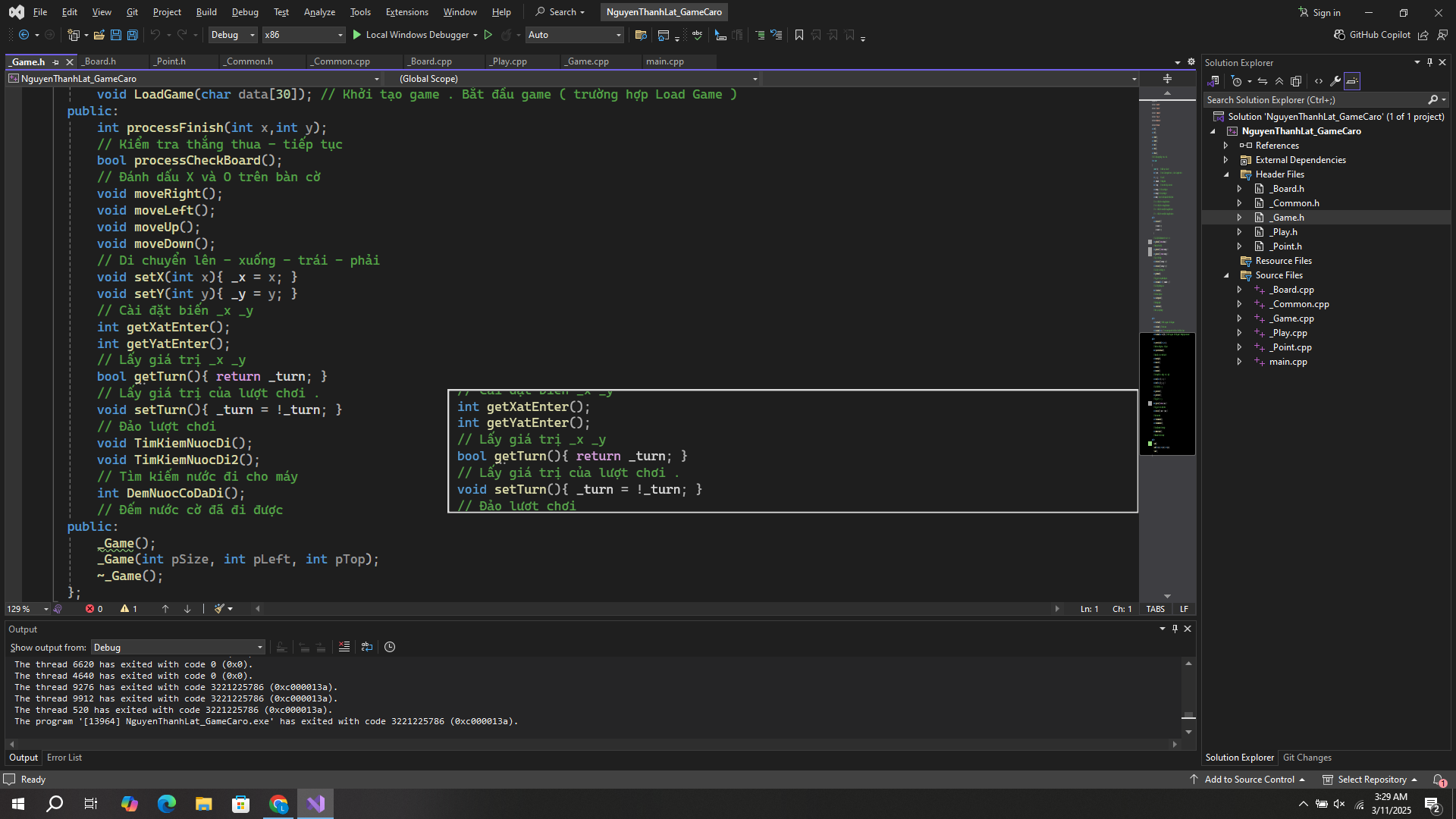
**\_Game.h : Xử lí ván game .**



*Hình 3.3.4: Hình hàm \_Game.h.*

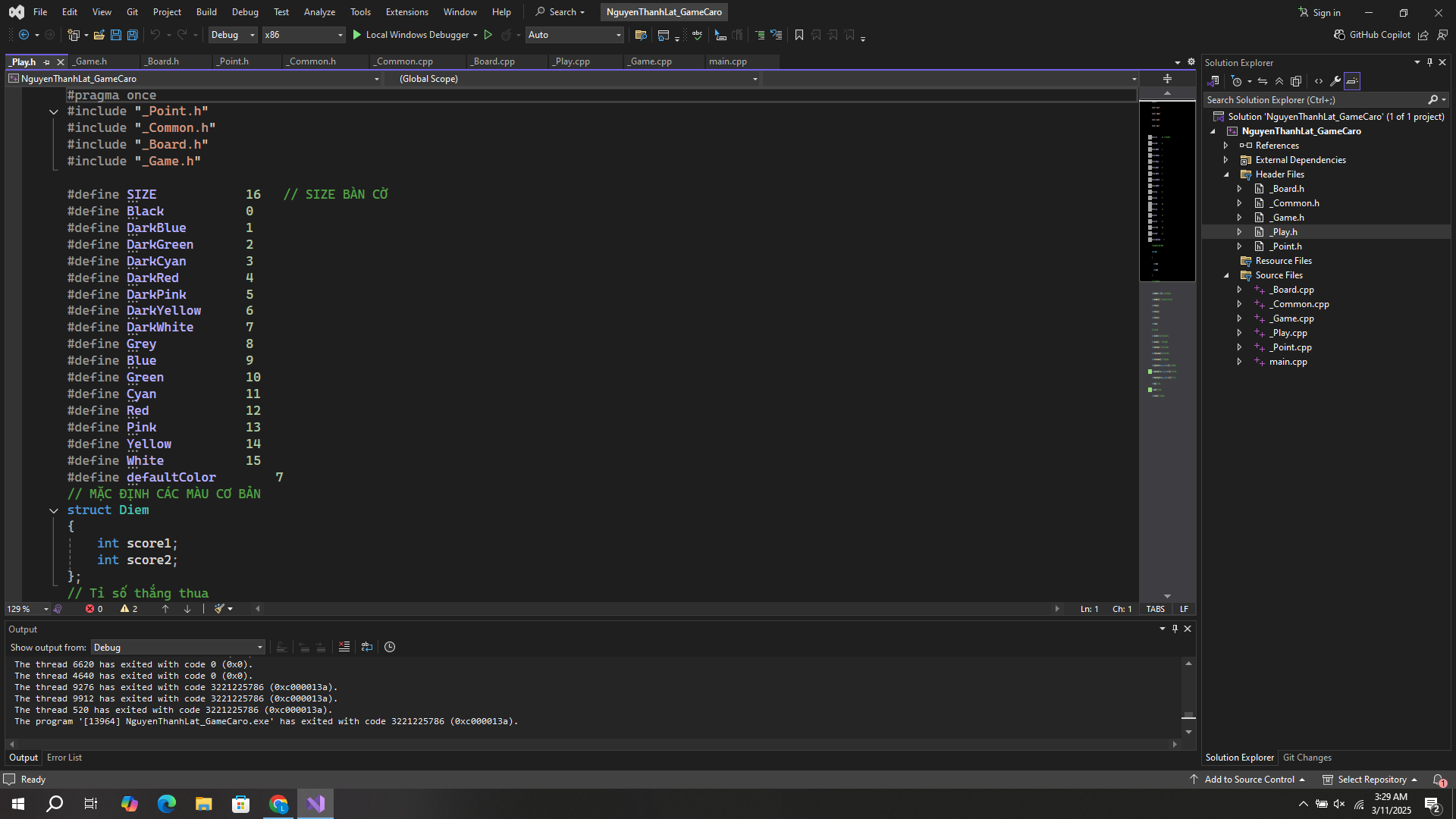


*Hình 3.3.4: Hình hàm \_Game.h.*

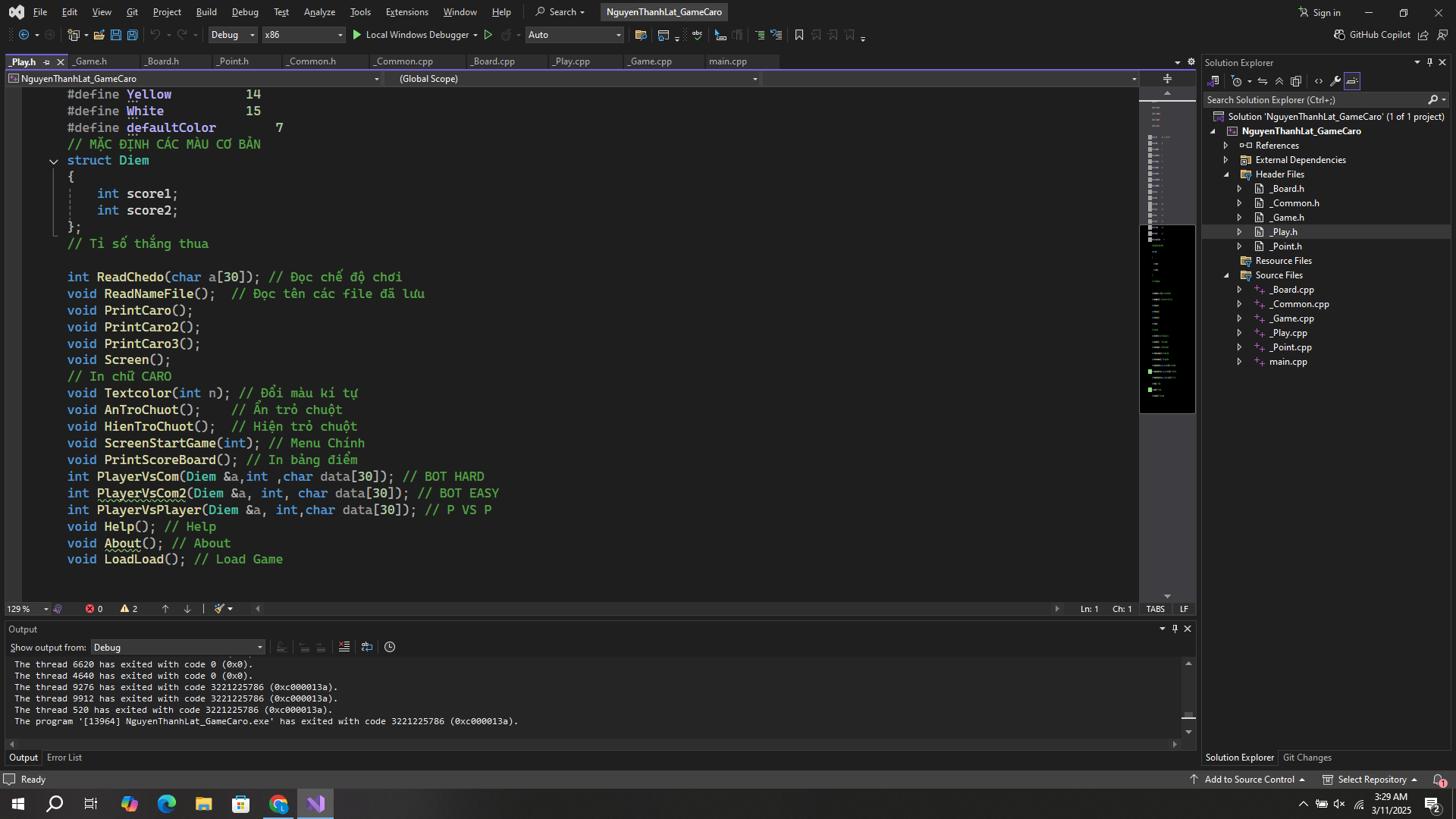


*Hình 3.3.4: Hình hàm \_Game.h.*

**\_Play.h : Các hàm xử lí Menu**



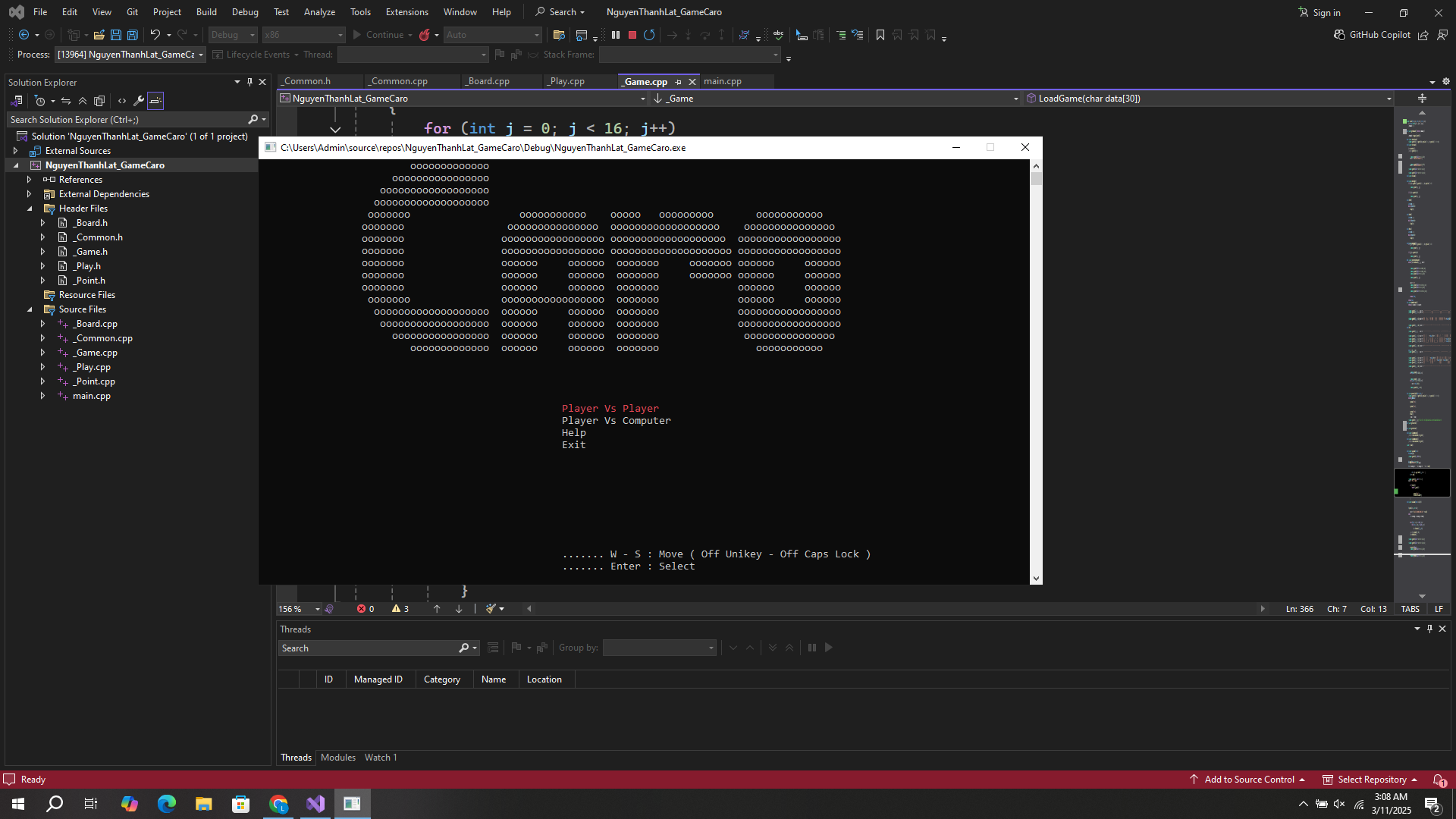
*Hình 3.3.4: Hình hàm \_Play.h.*



*Hình 3.3.4: Hình hàm \_Play.h.*

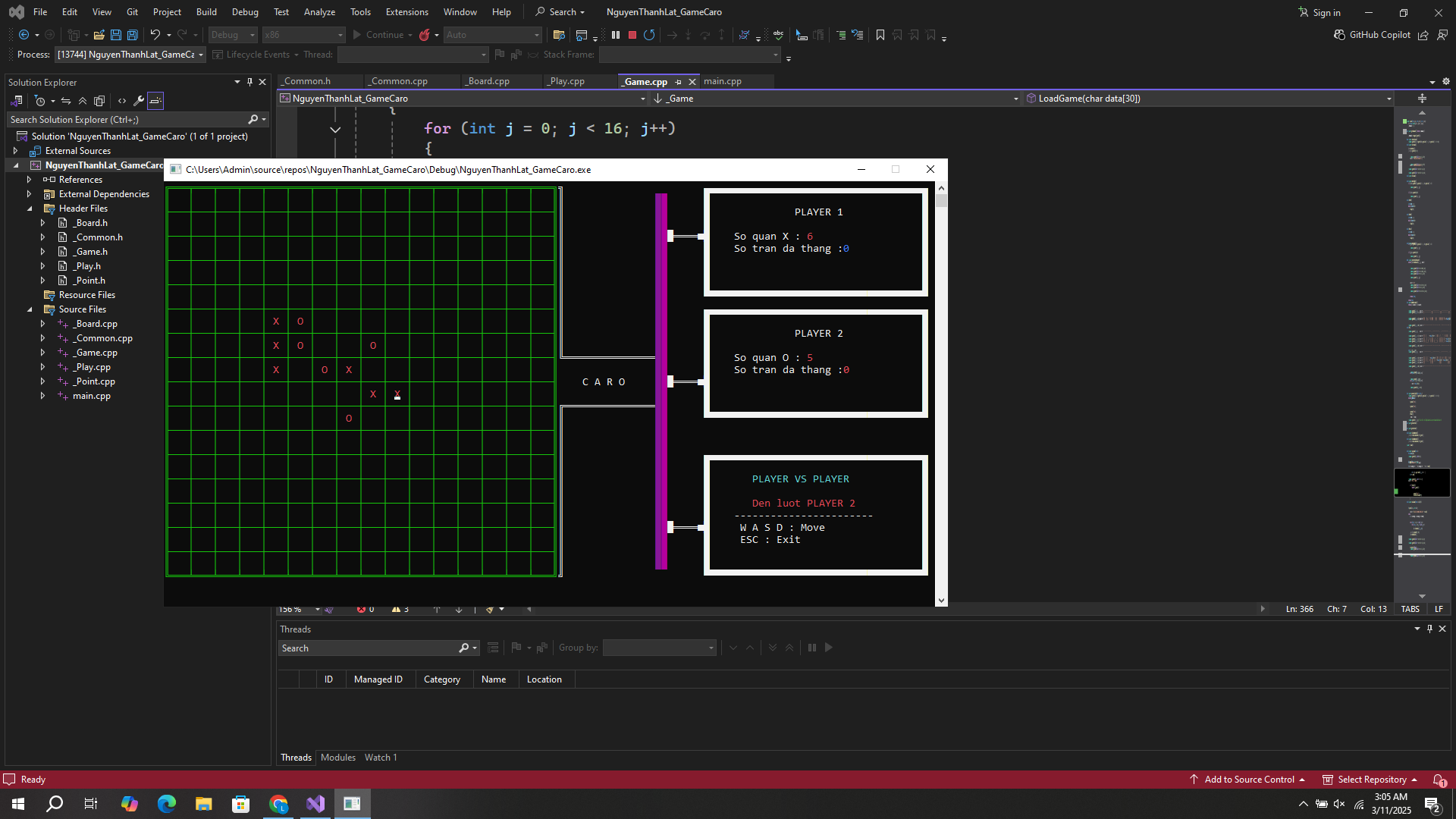
**3.4. Xây dựng giao diện sản phẩm Game**

**Màn hình menu:**

****

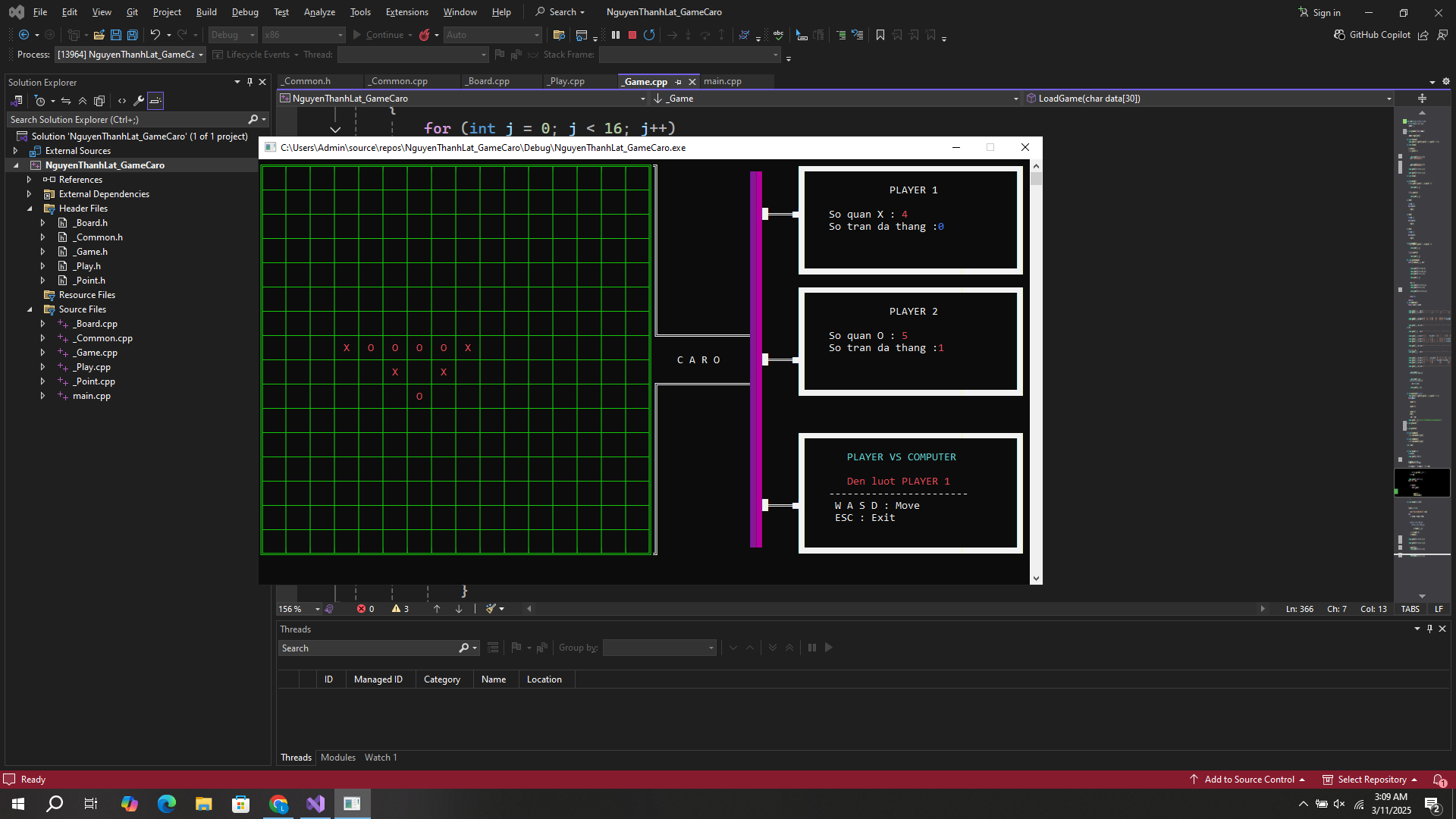
*Hình 3.4.1: Màn hình menu.*

**Màn hình đấu với người:**



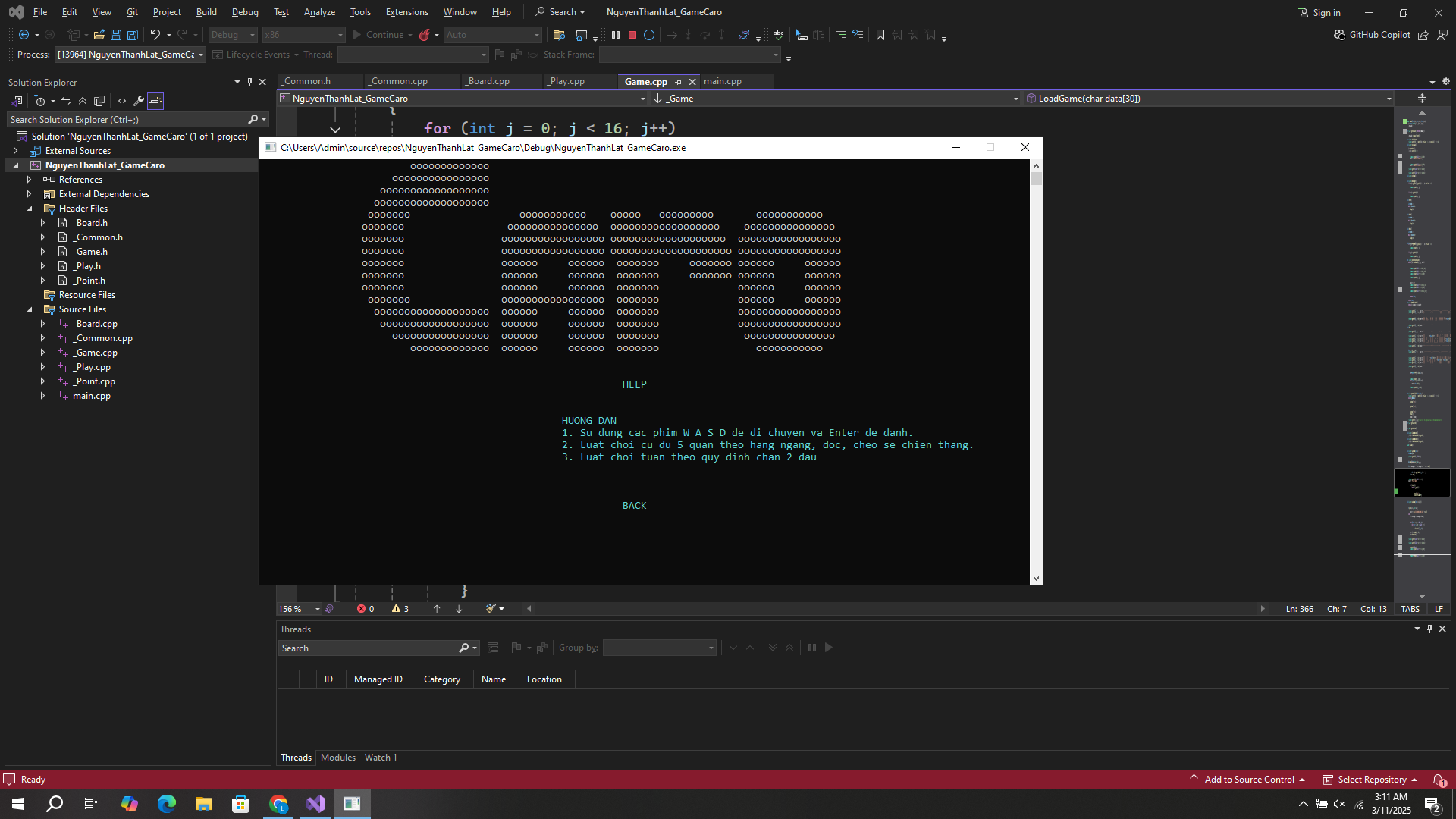
*Hình 3.4.2: Màn hình đấu người chơi.*

**Màn hình đấu với máy:**



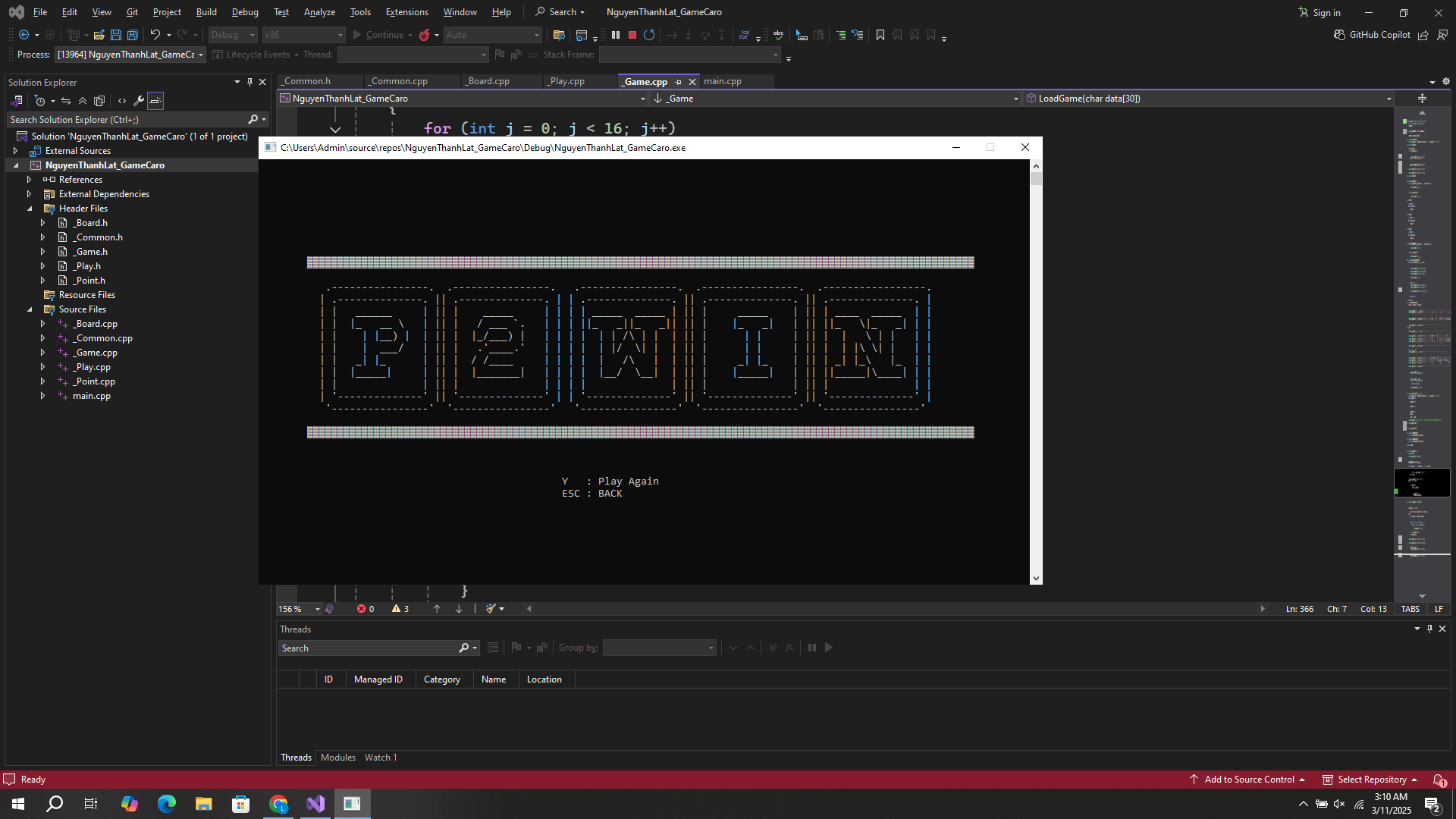
*Hình 3.4.3: Màn hình đấu máy.*

**Màn hình chức năng hướng dẫn chơi:**



*Hình 3.4.4: Màn hình hướng dẫn.*

**Màn hình chiến thắng:**



*Hình 3.4.5: Màn hình chiến thắng.*

**CHƯƠNG IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**4.1. Kết luận**

Sau quá trình nghiên cứu và phát triển, game Caro đã được xây dựng thành công bằng ngôn ngữ lập trình C++ trên nền tảng console. Trò chơi đáp ứng khá đầy đủ các chức năng cơ bản của một game Caro, bao gồm:

* Chế độ chơi 2 người giúp người chơi trải nghiệm những trận đấu đối kháng trực tiếp.
* Chế độ chơi với máy sử dụng thuật toán Heuristic, giúp Computer có thể đưa ra những nước đi thông minh.
* Giao diện console đơn giản, dễ sử dụng, phù hợp với những người mới học lập trình..
* Kiểm tra điều kiện thắng/thua/hòa một cách chính xác và xử lý lỗi nhập liệu từ người chơi.

Thông qua quá trình triển khai, đề tài không chỉ giúp sinh viên nâng cao kỹ năng lập trình C++, mà còn giúp hiểu sâu hơn về cấu trúc dữ liệu, thuật toán tìm nước đi, và cách tổ chức một phần mềm theo hướng đối tượng.

Tuy nhiên, do giới hạn về thời gian và tài nguyên, trò chơi vẫn còn một số hạn chế như chưa có giao diện đồ họa chuyên nghiệp, Computer chưa đạt mức độ mạnh mẽ như Minimax hay Alpha-Beta Pruning, và chưa hỗ trợ chơi online.

**4.2. Hướng phát triển**

Để nâng cao chất lượng game, trong tương lai có thể mở rộng và cải tiến các tính năng sau:

* Cải thiện Computer.
* Phát triển giao diện đồ họa (GUI).
* Hỗ trợ chơi online.
* Tối ưu hóa hiệu suất.
* Thêm chế độ chơi mới và chức năng mới.

**TÀI LIỆU THAM KHẢO**

1. Vẽ bàn cờ caro : https://www.youtube.com/watch?v=AOvXmLpucXk

2. Khóa học lập trình C# - Game caro – myclass.vn : ( bao gồm AI ) https://www.youtube.com/playlist?list=PLJbBHp6iPUiFHWtjeDmUpadn5lRzKFs8

3. Tài liệu hướng dẫn giảng viên : DoAnCaro\_OOP.pdf

4. Các lệnh chèn âm thanh , tô màu kí tự, ẩn trỏ chuột , thay đổi kích thước console trên các diễn đàn học lập trình .