TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



BÁO CÁO THỰC NGHIỆM/THÍ NGHIỆM

**HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**

***Đề tài:***

**XÂY DỰNG GAME AI CỜ CARO**

Sinh viên thực hiện: Nguyễn Văn Quân

Đinh Đức Tài

Lò thế Quyền

Quàng Văn Quang

Nhóm: 13

Lớp học phần: 20241IT6094004

Khóa: 17

Giảng viên hướng dẫn: Trần Thanh Huân

***Hà Nội, 12/2024***

|  |  |
| --- | --- |
| BỘ CÔNG THƯƠNG  TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – tự do – hạnh phúc  -------o0o------- |

**PHIẾU GIAO ĐỀ TÀI**

**BÁO CÁO THỰC NGHIỆM/THÍ NGHIỆM**

Nhóm thực hiện: 1) Nguyễn Văn Quân

2) Đinh Đức Tài

3) Lò Thế Quyền

4) Quàng Văn Quang

Lớp học phần: 20241IT6094004 Khóa: 17 Khoa Công nghệ thông tin

Tên đề tài: xây dựng trò chơi caro

Mục đích: Ứng dụng thuật toán Minimax, Alpha-Beta Pruning để Tạo ra một sản phẩm giải trí đơn giản, phổ biến và thú vị giúp người chơi rèn luyện tư duy chiến thuật và khả năng phân tích.

Yêu cầu:- Hoàn thành các nhiệm vụ của đề tài

-Sử dụng kỹ năng làm việc nhóm trong quá trình thực hiện báo cáo

-Trình bày báo cáo đúng yêu cầu, sử dụng kỹ năng “Viết báo cáo” khi thực hiện đề tài

-Sử dụng kỹ năng viết tài liệu kỹ thuật và phi kỹ thuật trong phần mở đầu của báo cáo

Kết quả thu được: Bản báo cáo đề tài (bản cứng và bản mềm); sản phẩm đề tài

Ngày giao đề tài: 20/11/2024

Ngày hoàn thành: 20/12/2024

Giảng viên hướng dẫn: Trần Thanh Huân

Hà nội, Ngày 06 Tháng 11 Năm 2024

GIẢNG VIÊN

Trần Thanh Huân

**PHIẾU PHÂN CÔNG NHIỆM VỤ**

Nhóm 13, gồm 4 thành viên

1. Nguyễn Văn Quân(nhóm trưởng)

2. Đinh Đức Tài

3. Lò Thế Quyền

4. Quàng Văn Quang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Người thực hiện** | **Công việc** | **Kết quả đạt được** | **Nhận xét của GV** |
| **Tuần 1** | | | | |
| 1 | Nguyễn Văn Quân | Tìm hiểu về python cơ bản và tìm hiểu một số đề tài. | Tìm được một số đề tài hay. |  |
| 2 | Đinh Đức Tài | Tìm hiểu về python cơ bản và tìm hiểu một số đề tài. | Tìm được một số đề tài. |  |
| 3 | Lò Thế Quyền | Tìm hiểu về python cơ bản và tìm hiểu một số đề tài. | Tìm được một số đề tài. |  |
| 4 | Quàng Văn Quang | Tìm hiểu về python cơ bản và tìm hiểu một số đề tài. | Tìm được một số đề tài. |  |
| **Tuần 2** | | | | |
| 1 | Nguyễn Văn Quân | 1. Chốt đề tài là xây dựng trò chơi cờ caro.  2. Tìm các tài liệu tham khảo. | 1. Tìm được các tài liệu tham khảo liên quan như thuật toán Minimax, Alpha-Beta Pruning,Board Evaluatio  2. Phân công nhiệm vụ cho các thành viên. |  |
| 2 | Đinh Đức Tài | Tìm hiểu về thuật toán Minimax, Alpha-Beta Pruning,Board Evaluation | Tổng hợp tài liệu liên quan đến thuật toán Minimax, Alpha-Beta Pruning,Board Evaluation |  |
| 3 | Lò Thế Quyền | Tìm hiểu về sơ lược lịch sử các chương trình chơi cờ | Tổng hợp tài liệu liên quan đến lịch sử các chương trình chơi cờ |  |
| 4 | Quàng Văn Quang | Tìm hiểu về cách hoạt động, Các yêu cầu của phần mềm | Đưa ra cái nhìn tổng quan, tài liệu về đề tài. |  |
| **Tuần 3** | | | | |
| 1 | Nguyễn Văn Quân | 1. Tổng hợp báo cáo  2. Hoàn thành chương trình demo | 1. Hoàn thành báo cáo  2. Hoàn thành chương trình demo |  |
| 2 | Đinh Đức Tài | Hỗ trợ làm nội dung báo cáo | Hoàn thành nội dung được phân công |  |
| 3 | Lò Thế Quyền | Hỗ trợ làm nội dung báo cáo | Hoàn thành nội dung được phân công |  |
| 4 | Quàng Văn Quang | Hỗ trợ làm nội dung báo cáo | Hoàn thành nội dung được phân công |  |

**LỜI CẢM ƠN**

Công nghệ thông tin hiện nay đã có những bước phát triển mạnh mẽ và đạt được nhiều thành tựu. Trong các lĩnh vực công nghệ thông tin, thì lập trình Game đã có một bước nhảy vọt đáng kể về cả công nghệ lẫn nội dung. Ngày nay chiếc máy tính PC hay laptop đã rất phổ biến gắn liền với học tập ,công việc, và giải trí của con người, những giờ phút căng thẳng sẽ được giải tỏa, hay cần cảm giác thú vị, chúng ta đến với Game giải trí sẽ đáp ứng các yêu cầu đó.

Hiện nay đã có nhiều công nghệ để tạo game trên máy tính ,và ở chương trình này em đã chọn pycharm môi trường viết với ngôn ngữ sử dụng là Python.Với các kiến thức đã học trong trong trường và tìm hiểu thêm một số tài liệu liên quan em xây dựng chương trình game này.

Bài toán em đặt ra không có nhu cầu khắt khe nhưng có đầy đủ những yêu cầu cơ bản nhất của một Game trên máy tính. Nhằm mục đích học hỏi, trao đổi kinh nghiệm và làm quen với thực tế.

Em xin chân thành cảm ơn thầy giáo Trần Thanh Huân đã chỉ bảo em tận tình để em có thể hoàn thành đồ án này.

# 

# **MỤC LỤC**

# 

[**MỤC LỤC 6**](#_h7nxbif5cg3)

[**DANH MỤC HÌNH ẢNH 7**](#_uthkxsmmn01m)

[**LỜI NÓI ĐẦU 8**](#_tnunlcas62xb)

[**MỞ ĐẦU 9**](#_ansd3jb6zl98)

[**Chương I. Tổng quan về trò chơi caro 10**](#_rfnajhclrewb)

[1. Sơ lược lịch sử các chương trình chơi cờ 11](#_dcvf4ph230t7)

[2. Sự phát triển của game Caro 13](#_nnm4aff886mh)

[2.1. Giới thiệu về game caro: 13](#_pkzhn6kp9d8u)

[2.2.Luật chơi 15](#_qtirqar9kzld)

[3. Tổng quan về game cờ caro 15](#_mz07sks1g6yx)

[3.1. Tổng quan về game 15](#_tyjcwt)

[3.2 Quy tắc trò chơi 15](#_1t3h5sf)

[3.3 Một số thuật toán xây dựng chess bot 15](#_4d34og8)

[3.3.1. Thuật toán Minimax 16](#_2s8eyo1)

[3.3.2. Alpha-Beta Pruning 16](#_1djdp1lh7f3l)

[3.3.3. Board Evaluation 16](#_3rdcrjn)

[**CHƯƠNG II. Phân tích và thiết kế game 17**](#_h8xl55kyl3cf)

[1. Yêu cầu của phần mềm 17](#_9cznblv8j1cj)

[1.1. Yêu cầu chơi 17](#_u09j0z8k0ak)

[1.2. Giao diện: 18](#_1f3azviziv22)

[1.3. Hướng dẫn chơi : 18](#_7bt7otxuxb9c)

[1.4. Tương tác nhanh : 18](#_3asoonpevwq)

[2. Phân tích và thiết kế 19](#_ho9lqu5bfhng)

[2.1. Xây dựng các Module: 19](#_4mil31dmzuer)

[2.1.1. Sơ đồ chức năng trò chơi: 19](#_7bb0f5okkc2z)

[2.1.2. Luồng xử lí của trò chơi: 20](#_s8jgiw8sjtlr)

[2.1.3. Sơ đồ luồng xử lí trò chơi: 20](#_cjiaa132hg5y)

[2.1.3.1.Sơ đồ luồng xử lí trò chơi: 20](#_1jckklxl9nxy)

[2.1.3.2.Các sự kiện xử lí chức năng trong chương trình: 21](#_va5ggdi5ax3b)

[2.2. Thuật toán xây dựng trò chơi Caro 23](#_iqx0c1sdqpgz)

[2.2.1. Tham khảo một số thuật toán 23](#_v22g92aa2wb1)

[2.2.1.1. Dạng trò chơi 23](#_5iq0dzm5zljl)

[2.2.1.2. Cây trò chơi : 24](#_dprxytsjbb7i)

[2.2.1.3. Vét cạn : 25](#_8euwdkl7clre)

[2.2.1.4. Chiến lược tìm kiếm trong trò chơi : 26](#_xua4585h47h7)

[2.2.1.5. Thủ tục Minimax[1] : 26](#_ahge6lhb7rqs)

[2.2.1.6. Thủ tục AlphaBeta pruning: 32](#_huwsljd55l7)

[2.2.1.7. Hướng cải thiện việc tỉa nhánh của thuật toán AlphaBeta : 38](#_7lz9t5dnzknv)

[2.2.1.8.Tổng kết : 38](#_7seljyu1n94p)

[2.3. Thuật toán xây dựng trò chơi 38](#_d7d2ygnuse5z)

[2.3.1. Tìm nước đi cho máy: 38](#_jpkrjeb3kr0x)

[2.4. Thiết kế giao diện: 43](#_5xeeiamk477v)

[**CHƯƠNG III. Cài đặt và kiểm thử 45**](#_6en72jw5e3zd)

[1. Môi trường cài đặt 45](#_4zsol8imz7m9)

[1.1. Cấu hình máy: 45](#_32cpwis75o37)

[1.2. Hệ điều hành: 45](#_75w16uk6p4k9)

[1.3. Các phần mềm hệ thống: 45](#_j8bh5922vz8z)

[2. Cách cài đặt và kiểm thử 45](#_jo0kx06it598)

[2.1. Cài đặt môi trường Pycharm 45](#_orq4ns6pc26m)

[**KẾT LUẬN 50**](#_edfbqh6dmzzb)

[**TÀI LIỆU THAM KHẢO 52**](#_ar823mu5xhq0)

# **DANH MỤC HÌNH ẢNH**

Hình 1.1: Game trên PC…………………………………………………………....13

Hình 1.2: Game caro Online……………………………………………………….14

Hình 1.3: Game Caro trên điện thoại di động……………………………………...14

Hình 2.1. sơ đồ chức năng trò chơi………………………………………………...19

Hình 2.2. luồng xử lí trò chơi……………………………………………………....20

Hình 2.3*.*Sơ Đồ luồng xử lý của Chương trình……………………………………21

Hình 2.4. Các loại trò chơi cờ……………………………………………………...24

Hình 2.5.Cây trò chơi……………………………………………………………....25

Hình 2.6. Minh họa chiến lược cờ của lẫn máy.…………………………………...27

Hình 2.7. Thuật toán AlphaBeta…….……………………………………………..33

Hình 2.8. Cây trò chơi……………….……………………………………………..36

Hình 2.9. *S*ơ đồ thuật toán lượng giá cho thế cờ…………………………………...41

Hình 2.10. Sơ đồ thuật toán tính nước cờ của máy………………………………...42

Hình 2.11. Giao diện chính………….……………………………………………..43

Hình 2.12. Giao diện bàn cờ………….……………………………………………44

Hình 3.1. Download PyCharm.……...……………………………………………..46

Hình 3.2. Các phiên bản PyCharm…..……………………………………………..46

Hình 3.3 Setup PyCharm…………………………………………………………..47

Hình 3.4. Install Location PyCharm….……………………………………………48

Hình 3.5 Install PyCharm…………...……………………………………………..48

Hình 3.6. Hoàn tất cài đặt PyCharm...……………………………………………..49

Hình 3.7 Chạy chương trình cờ caro...……………………………………………..49

# 

# **LỜI NÓI ĐẦU**

Ngày nay Trí tuệ nhân tạo (AI) đang có những tác động sâu sắc đến mọi khía cạnh của cuộc sống, từ cách chúng ta tương tác với thế giới xung quanh đến cách chúng ta làm việc và học tập. Trò chơi cờ caro là một trong những trò chơi trí tuệ đơn giản nhưng đầy thử thách, gắn liền với tuổi thơ của nhiều người. Với luật chơi dễ hiểu, cờ caro không chỉ là một công cụ giải trí mà còn là một cách để phát triển tư duy chiến lược và kỹ năng phản xạ nhanh nhạy. Trong bối cảnh hiện nay, khi công nghệ thông tin và ứng dụng phần mềm ngày càng phát triển, việc xây dựng một trò chơi cờ caro trên máy tính hoặc thiết bị di động mang lại nhiều ý nghĩa thực tiễn, cả trong học tập và giải trí.

Báo cáo này trình bày quá trình nghiên cứu, thiết kế và xây dựng trò chơi cờ caro, bao gồm việc áp dụng các thuật toán trí tuệ nhân tạo như Minimax để phát triển chế độ chơi với máy, đồng thời đảm bảo giao diện thân thiện, dễ sử dụng cho người dùng.

Cấu trúc của báo cáo

Báo cáo gồm 3 chương:

+ Chương 1: Tổng quan về trò chơi caro

+ Chương 2: Phân tích và thiết kế game

+ Chương 3: Cài đặt và kiểm thử

- Kết thúc báo cáo là phần kết luận, tổng kết lại những nội dung chính của báo cáo.

# **MỞ ĐẦU**

**1. Lý do chọn chủ đề/đề tài**

Trò chơi caro là một trong những trò chơi trí tuệ phổ biến và thú vị nhất trên thế giới. Trò chơi này đòi hỏi người chơi phải có khả năng suy nghĩ chiến thuật, phân tích tình huống và đưa ra quyết định tối ưu. Tuy nhiên, không phải ai cũng có thời gian, đối thủ hoặc điều kiện để chơi caro thường xuyên. Vì vậy, việc xây dựng một chương trình máy tính có thể chơi caro với con người hoặc với chính nó là một thách thức và một cơ hội lớn cho các nhà nghiên cứu trong lĩnh vực trí tuệ nhân tạo.

**2. Mục tiêu nghiên cứu.**

Xây dựng một chương trình máy tính có thể chơi caro ở một mức độ khá với thang elo từ 1000 - 1500 theo chuẩn thế giới, gọi là chess bot. Chess bot sẽ sử dụng các thuật toán tìm kiếm và đánh giá để sinh ra các nước đi tốt nhất trong mỗi trạng thái của bàn cờ. Chess bot cũng sẽ có khả năng điều chỉnh độ khó của trò chơi để phù hợp với trình độ của người chơi.

**3. Đối tượng và phạm vi nghiên cứu.**

Đối tượng nghiên cứu bao gồm: Game caro bao gồm chương trình máy tính và game online; Pygame; Các tài liệu liên quan.

Phạm vi nghiên cứu: Chess bot AI, các thuật toán tìm kiếm và đánh giá được sử dụng trong chess bot.

*\*Đề tài này không bao gồm các vấn đề liên quan đến giao diện người dùng, âm thanh, hình ảnh hay các tính năng khác của trò chơi cờ vua.*

**4. Kết quả mong muốn đạt được của đề tài**

* Tạo ra một chess bot có mức độ elo từ 1000 - 1500 theo thang tiêu chuẩn, hoạt động ổn định, hiệu quả và thân thiện với người dùng.
* Chess bot sẽ được kiểm tra và đánh giá bằng cách chơi với con người.
* Chess bot cũng sẽ được so sánh với các chương trình máy tính chơi caro khác về độ mạnh, độ nhanh và độ thông minh

# 

# **Chương I. Tổng quan về trò chơi caro**

## **1. Sơ lược lịch sử các chương trình chơi cờ**

Vào năm 1950, Alan Turing - một nhà nghiên cứu người Anh đi tiên phong trong lĩnh vực máy tính số, đã viết chương trình chơi cờ đầu tiên. Vào lúc đó, Turing phải viết và chạy chương trình của ông bằng... bút chì và giấy. Chương trình đó, cũng như chủ nhân của nó, chơi cờ rất tồi, nhưng đạt được mục đích: cho thấy máy tính có thể chơi được cờ. Cũng vào năm đó, Claude Shannon đã vạch ra một chiến lược cho máy tính chơi cờ tốt. Nhưng vào những năm 1950 tốc độ máy tính rất chậm nên không ai dám tiên đoán liệu máy tính có thể thắng con người được không, dù trong các trò chơi đơn giản như trò Checker.

Năm 1958, một chương trình chơi cờ đã lần đầu tiên hạ được đối phương là con người. Người thua là một cô thư kí của chính đội lập trình ra nó, cô chưa bao giờ chơi cờ trước đó và được dạy chơi cờ chỉ một giờ trước cuộc đấu. Đối với ngày nay chiến công này thật nhỏ nhoi, nhưng nó cho thấy tri thức có thể được đưa vào trong một chương trình chơi cờ. Lượng tri thức này được đo chính xác bằng một giờ học chơi.

Sau chiến thắng đó, một số người trong nhóm lập trình cờ đầu tiên đã tiên đoán rằng vào những năm 60 sẽ có chương trình chơi cờ được liệt vào hàng ngũ kiện tướng thế giới. Vào những năm cuối của thập kỷ 60, Spassky đã trở thành kiện tướng cờ thế giới và các chương trình chơi cờ đã chiếm được những thứ hạng cao trong hàng ngũ những người chơi cao cấp. Nhưng nhiều người cho rằng máy tính sẽ không bao giờ có thể giải quyết được những nhiệm vụ thông minh, không thể đạt được chức Vô địch cờ thế giới.

Lời tiên đoán này được nhắc lại một lần nữa vào những năm 70, liên quan đến một cuộc đánh cược giữa David Levy, một kiện tướng quốc tế người Anh (theo phân loại của Liên đoàn cờ quốc tế các đẳng cấp cao bao gồm: Kiện tướng quốc tế, Đại kiện tướng và Vô địch thế giới) và John McCarthy, một nhà nghiên cứu trong lĩnh vực trí tuệ nhân tạo. Lời thách đấu được đưa ra vào năm 1978. Trận đấu đã được diễn ra và chương trình cờ tốt nhất thời đó, CHESS 4.7 đã bị Levy hạ trong trận đấu có năm ván tại Toronto với thành tích ba ván người thắng, một hoà và một máy thắng. Levy không chỉ chiến thắng mà còn đút túi số tiền đánh cược 1000 bảng.

Nếu như mục đích của cuộc đánh cược là làm cho những nhà nghiên cứu phải nghĩ kĩ trước khi tiên đoán đến ngày thắng lợi, thì lần đánh cược này cho thấy: mặc dù tiên đoán sai trong những năm 1958-1968 và 1968-1978, các chuyên gia chương trình cờ lại tiếp tục tiên đoán tiếp rằng máy tính sẽ đạt đến vô địch cờ thế giới trong thập kỉ tiếp theo.

Nhưng một lần nữa, vào năm 1988, Vô địch cờ thế giới vẫn là con người. Trong năm tiếp theo, Deep Thought, một chương trình cờ mạnh nhất từ xưa đến nay đã chiến thắng một cách dễ dàng Kiện tướng Quốc tế Levy. Bộ não của Deep Thought có 250 chip và hai bộ xử lí trong một bảng mạch đơn, nó có khả năng xét 750.000 thế cờ trong một giây và tìm trước được đến 10 nước. Cũng trong năm đó, nó là máy tính đầu tiên hạ được một Đại kiện tướng (Bent Larsen). Deep Thought đã trở thành một trong một trăm người chơi cờ mạnh nhất thế giới. Nhưng trong trận đấu diễn ra vào năm 1989 giữa nhà Vô địch thế giới Garry Kasparov và Deep Thought thì nó đã bị nhà vô địch đè bẹp.

Các lời tiên đoán lại đến như các lần trước. Đã ba lần các nhà nghiên cứu tiên đoán: 'trong thập kỷ tới'. Nhưng lần này họ lại sửa lại là: 'trong 3 năm tới'... Trong năm 1993, Deep Thought đã hạ Judit Polgar - lúc đó là Đại kiện tướng trẻ nhất trong lịch sử và là người phụ nữ chơi hay nhất thế giới, trong trận đấu 2 ván.

Trong năm 1996, Deep Blue (tên mới của Deep Thought và lúc này nó thuộc hãng IBM) là một máy tính song song có 32 bộ xử lí với 256 mạch tích hợp cỡ lớn, khả năng xét từ 2 đến 400 triệu nước đi mỗi giây) đã thắng Gary Kasparov trong ván đầu tiên của trận đấu 6 ván, nhưng lại thua trong toàn trận (với tỉ số máy thắng 1, hoà 2 và thua 3).

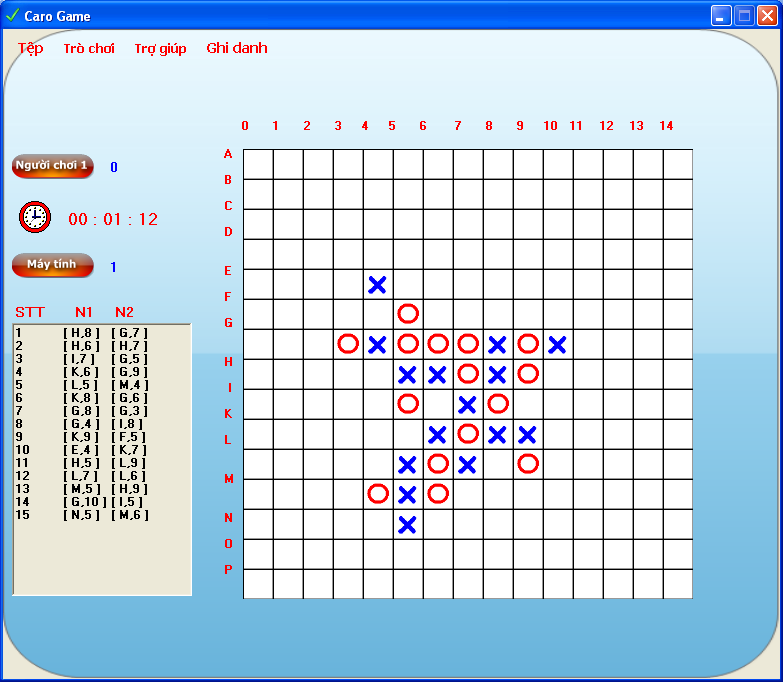
Cuối cùng đích mà mọi người chờ đợi đã tới, nhưng sau 9 năm từ lời tiên đoán cuối và 39 năm từ lúc có chương trình chơi cờ đầu tiên, Deep Blue đã chiến thắng nhà đương kim Vô địch thế giới Garry Kasparov vào tháng 5/1997 trong một cuộc chiến dài đầy khó khăn, với tỷ số sát nút 2 thắng, 1 thua và 3 hoà.

## **2. Sự phát triển của game Caro**

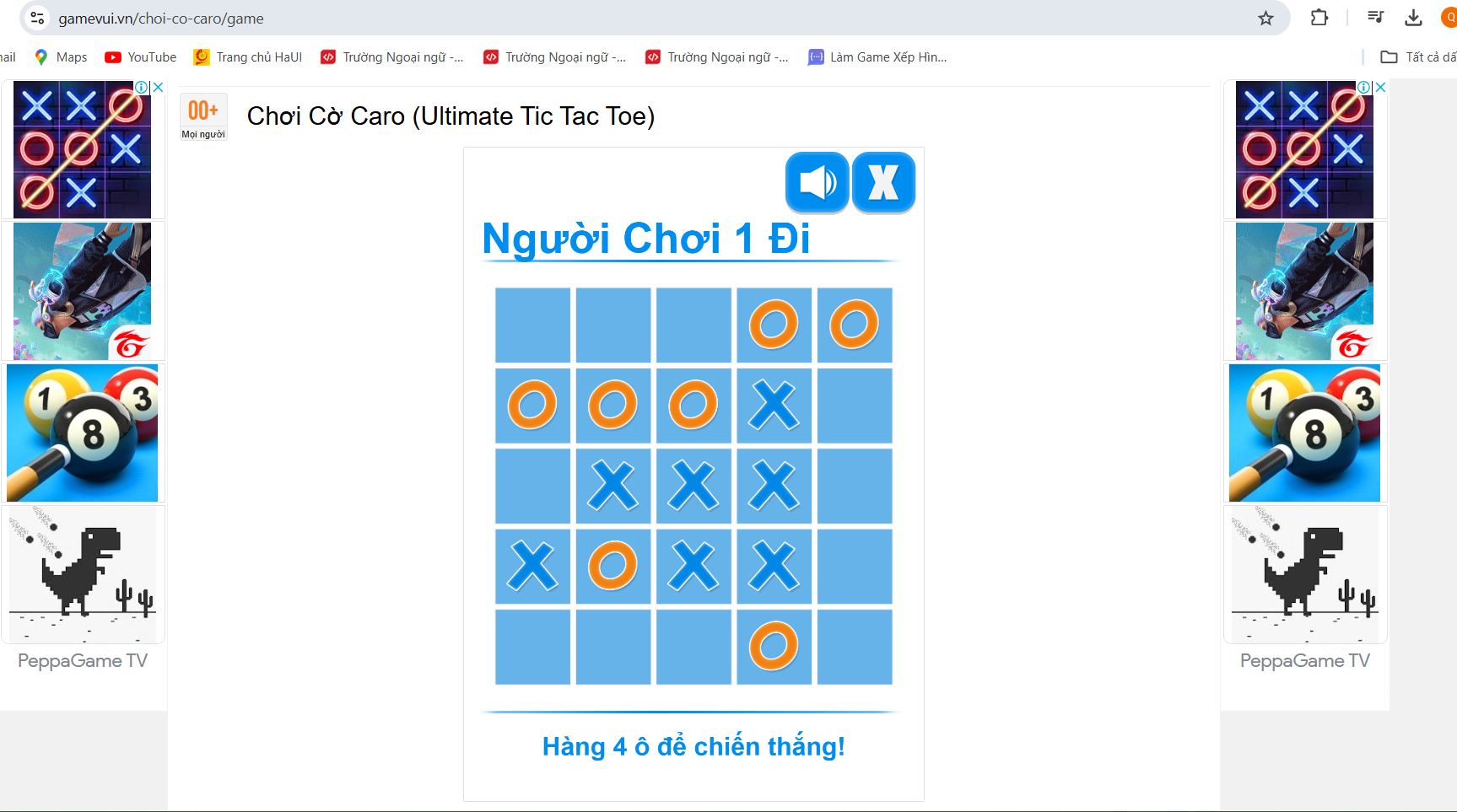
### **2.1. Giới thiệu về game caro:**

Đã trải qua bao thập kỷ và cho đến nay Caro vẫn được ưa chuộng trên toàn thế giới, với cách chơi đơn giản, nhưng càng chơi thì sự biến hóa của các nước cờ là vô cùng kì diệu. Sự hấp dẫn game Caro đã lôi cuốn các bạn trẻ Việt Nam rất mạnh, nhất là đối với các học sinh, sinh viên. Sau giờ nghỉ giải lao đều có thể chơi với 1 mảnh giấy nhỏ kẻ ô vuông và 2 cây bút. Và đây chính là 1 game giải trí lành mạnh, phát triển trí tuệ cho các bạn.

Giờ đây vơi sự phát triển của công nghệ ,Game Caro đã được đưa vào PC với rất nhiều các phiên bản cùng nhiều cấp độ cho người chơi với máy, hay hai người chơi vơi nhau. Cùng với đó Caro cũng chiếm 1 phần trong làng game giải trí online, chủ yếu để 2 người chơi thi đấu và thư giãn khi đánh với máy. Thật tiện ích khi game caro được đưa vào các điện thoại di động có thể chơi được mọi lúc.



*Hình 1.1:Game trên PC*



*Hinh 1.2:Game caro Online*



*Hình 1.3: Game Caro trên điện thoại di động*

### **2.2.Luật chơi**

a ) Tic tac toe:

* Bàn cờ trước là 3x3(3 hàng , 3 cột), có 9 ô vuông ,có 2 bên quân trắng và đen.
* Quân đen đi trước, quân trắng đi sau, khi nào 1 trong 2 quân có 3 quân cùng màu liền nhau thì quân đó thắng.

b) Caro

* Bàn cờ 15x15 (15 hàng, 15 cột), 225 ô vuông, cũng có bên quân trắng, quân đen. Quân đen đi trước, trắng đi sau liên tiếp. Khi nào 1 trong 2 quân có 5 quân cùng màu liên tiếp cạnh nhau (xét theo cả 2 đường chéo, đường dọc và đường ngang) thì quân đó chiến thắng. Ván cờ hòa khi hết ô vuông trống mà chưa phân thắng bại
* Bàn cờ 25 x 25 thì luật chơi tương tự trên.

## **3. Tổng quan về game cờ caro**

### **3.1. Tổng quan về game**

Game caro một người chơi, đối thủ là bot chơi caro tự động (sau đây gọi là Chess bot), có thể tự đưa ra các nước đi hợp lý tương đương với một người chơi cờ vua có chỉ số elo 1000 – 1500 theo mức tiêu chuẩn thế giới (thang đo chỉ mang tính tương đối).

### **3.2 Quy tắc trò chơi**

caro là một trò chơi dạng board game dành cho 2 người. Một trận đấu chỉ có 3 kết quả có thể xảy ra: A thắng B thua, A thua B thắng, A và B hòa nhau.caro có thể được giải ngay từ nước đi đầu tiên một cách tuyệt đối (xác định được A thắng hay B thắng) (định lý Zermelo), không may là ước tính số thế cờ hợp lệ trong caro là 4x1044, với độ phức tạp vào khoảng 10123, quá lâu để có thể tính toán kể cả với những công nghệ hiện đại ngày nay. Tuy nhiên, có thể dùng định lý trên có thể sử dụng các thuật toán để tìm ra được nước đi đủ tốt ở một độ sâu không quá phức tạp.

### **3.3 Một số thuật toán xây dựng chess bot**

Có một số thuật toán để tạo ra các nước đi hợp lý, đánh giá bàn cờ, và tối ưu hóa quá trình tìm kiếm. Một số thuật toán phổ biến và hiệu quả cho mục đích này là:

#### **3.3.1. Thuật toán Minimax**

**Minimax** là một thuật toán đệ quy lựa chọn bước đi kế tiếp trong một trò chơi có hai người bằng cách định giá trị cho các node trên cây trò chơi, sau đó tìm node có giá trị phù hợp để đi bước tiếp theo. Thuật toán này giả định rằng cả hai người chơi đều chơi tối ưu, và sử dụng hai lớp MAX và MIN để biểu diễn người chơi luôn muốn chiến thắng và người chơi luôn muốn làm cho người kia thua.

#### **3.3.2. Alpha-Beta Pruning**

**Alpha-Beta Pruning** là một kỹ thuật cải thiện thuật toán Minimax bằng cách cắt bỏ những nhánh không cần thiết trên cây trò chơi, giúp giảm thiểu số lượng node cần duyệt và tăng tốc độ tìm kiếm. Kỹ thuật này sử dụng hai giá trị alpha và beta để xác định giới hạn trên và dưới của giá trị tốt nhất có thể đạt được ở mỗi node.

#### 3.3.3. Board Evaluation

**Board Evaluation** là một chức năng dự đoán giá trị của bàn cờ ở một trạng thái nhất định, dựa trên sức mạnh tương đối của các quân cờ trên bàn. Một cách đơn giản để đạt được điều này là sử dụng một bảng điểm để gán giá trị cho mỗi quân cờ theo vị trí của nó trên bàn.

# **CHƯƠNG II. Phân tích và thiết kế game**

## **1. Yêu cầu của phần mềm**

### **1.1. Yêu cầu chơi**

Game Caro đa phổ biến với rất nhiều người,với nhiều phiên bản game được đưa ra ,ở đây em xây dựng game Caro này phù hợp với nhiều đối tượng Người chơi.

Người chơi có thể chọn 1 trong 2 chế độ chơi:

* Hai Người chơi.
* Chơi với máy.

Ở chế độ “Hai Người chơi”, hai bên lần lượt đi từng nước cờ, chọn loại quân cờ phù hợp, chọn có hoặc không có âm thanh khi chơi.

Ở chế độ “Chơi với máy”,Người chơi và máy lần lượt đi từng nước cờ, NC chọn loại quân cờ phù hợp, chọn âm thanh khi chơi. Chọn kinh nghiệm cho máy như:

* + - Tấn công.
    - Phòng thủ.
    - Tấn công , phòng thủ.

Người chơi có thể chọn các chức năng của Game như: chọn xem bảng “Hướng dẫn chơi”, chọn xem thông tin bảng “Ghi Danh Người chơi”. Trong quá trình chơi, Người chơi có thể lưu lại ván cờ đang chơi,mở ván cờ đã được lưu từ trước,

, và có thể tạm dừng ván cờ hay chơi tiếp ván cờ theo ý muốn......

* Tạo ván chơi mới:trong lúc đang chơi mà không muốn tiếp tục ván chơi đó có thể tạo ván chơi mới khởi động lại từ đầu.
* Nước đi: nhanh, có âm thanh đi kèm (khi Người chơi chọn âm thanh). Sau mỗi nước đi sẽ hiện ra vị trí nước đi đó cho từng bên.
* Thời gian: hiển thị thời gian chơi được tính bắt đầu khi Người chơi chọn ván chơi mới.
* Tạm dừng:có thể dừng trò chơi lại,hay có thể tiếp tục chơi ván chơi.
* Đưa ra 3 kiểu quân cờ mà Người chơi có thể chọn: “kinh điển”, “nút”, “mạng nhện” .
* Đưa ra bảng “Ghi danh Người chơi”.
* Lưu lại ván chơi:có thể lưu lại ván chơi ra 1 file XML .
* Mở lại ván chơi:có thể mở lại ván chơi đã lưu trước đó để chơi tiếp.
* Đưa ra “Hướng dẫn cho Người chơi” để hướng dẫn luật chơi cờ.

### **1.2. Giao diện**:

1. Thuận tiện: các nút chức năng phải dễ nhìn ,dễ sử dụng.
2. hình ảnh: ảnh nền,ảnh bàn cờ ,ảnh quân cờ,ảnh nút,….đơn giản, dễ nhìn.
3. Sinh động:có âm thanh nước đi trong khi chơi.
4. Giao diện:
   1. Giao diện chính:

* Hiện bàn cờ :bàn cờ gồm 225 ô chia làm 15 hàng dọc và 15 hàng ngang( mỗi hàng gồm 15 ô vuông liên tiếp nhau, thẳng hàng 1 ô vuông = 30 px. )
* Bên cạnh giao diện chính là bàn cờ, cần hiển thị các chức năng để xử lí chương trình (ví dụ: 1 thanh Menu công cụ bao gồm tất cả các chức năng xử lí chương trình), hiển thị về thời gian chơi, hiển thị tỷ số ván thắng của cả 2 bên, hiển thị tọa độ các nước đã đi của hai bên.
  1. Giao diện về bảng “Chức năng”: hiển thị các chữ viết và hình ảnh: hỏi xem người chơi có muốn chọn âm thanh, người chơi muốn đi trước hay để máy đi trước, người chơi muốn chọn loại quân cờ nào và chọn kinh nghiệm máy khi chơi.
  2. Giao diện bảng “Hướng dẫn chơi”: đưa ra luật chơi bằng cả chữ viết và hình ảnh hướng dẫn người chơi.
  3. Giao diện bảng “Ghi danh”: đưa ra thông tin về người chơi chiến thắng máy: tên người chơi, số nước đi người chơi và thời gian chơi, kinh nghiệm chơi của máy.

### **1.3. Hướng dẫn chơi :**

Hướng dẫn Người chi tiết bằng văn bản và hình ảnh về luật chơi, cách đi cờ hợp lệ.

### **1.4. Tương tác nhanh :**

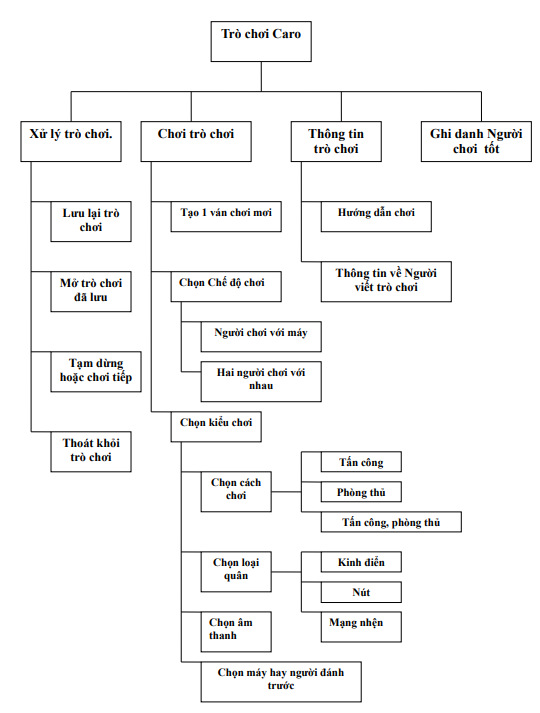
Người chơi có thể tương tác nhanh với chương trình bằng phím hoặc chuột.

* Phím hoặc chuột dùng để điều khiển chương trình.
* Chuột dùng để đi nước cờ.

## **2. Phân tích và thiết kế**

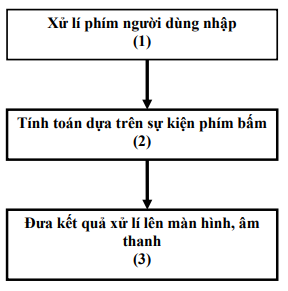
### **2.1. Xây dựng các Module:**

#### **2.1.1. Sơ đồ chức năng trò chơi**:



Hình 2.1. sơ đồ chức năng trò chơi

#### **2.1.2. Luồng xử lí của trò chơi**:

****

**Hình 2.2. luồng sử lí trò chơi**

1. nhận phím bấm, xử lí sự kiện bấm phím đó

( kích chuột để đi quân, và lựa chọn một chức năng trong menu…)

**(2):** với mỗi đầu vào **(1),** game sẽ thực hiện một chưc năng tương ứng( tìm nước đi quân hợp lệ, kiểm tra ván cờ kết thúc …)

**(3):** đảm nhiệm chức năng vẽ hình và hiển thị:vẽ quân cờ, hiển thị thông báo,…

#### **2.1.3. Sơ đồ luồng xử lí trò chơi**:

##### **2.1.3.1.Sơ đồ luồng xử lí trò chơ**i:

Chú thích:

**+ vị trí hợp lệ:** là vị trí phải nằm trong tọa độ bàn cờ, và vị trí đó phải là ô vuông trống.

**+ chế độ:**

Người chơi với máy: là chế độ chơi có hai bên đối thủ : 1 là người chơi, 2 là máy.

Hai người chơi: là chế độ chơi có người chơi là đối thủ của nhau



##### **2.1.3.2.Các sự kiện xử lí chức năng trong chương trình:**

* Tạo ván chơi mới:
  + Khởi tạo bàn cờ.
  + Đưa bàn cờ về trạng thái chưa kết thúc.(trạng thái kết thúc là khi ván chơi có 1 trong 2 bên đối thủ giành chiến thắng)
  + Thông báo về chiến thắng là rỗng.
  + Tỷ số trận thắng của 2 bên được cập nhật.
  + Đếm nước đi khởi tạo giá trị 0.
  + Thời gian chơi khởi tạo về giá trị 0.
  + Đưa có hoặc không có âm thanh khi chơi( do người chơi chọn, mặc định là có âm thanh ).
  + Đưa ra kinh nghiệm máy khi chơi (do người chơi chọn kinh nghiệm máy là tấn công, phòng thủ hay tấn công và phòng thủ, mặc định là kinh nghiệm máy tấn công) sau đó sẽ gán các giá trị cho kinh nghiệm máy.
  + Đưa đường dẫn ảnh quân cờ do người chơi chọn loại quân cờ từ trước (mặc định quân cờ kiểu kinh điển).
* Chơi với máy:
  + Đưa ra máy tính
  + Đưa bàn cờ về trạng thái chưa kết thúc
  + Hiện lại phần chọn kinh nghiệm cho máy tính, chọn máy đi trước hay không.
  + Khởi tạo giá trị cho biến về số ván thắng của người chơi và máy=0.
  + Thông báo chiến thắng khởi tạo về rỗng.
  + Đường dẫn ảnh để hiển thị ảnh máy tính
  + Gọi lại chức năng “tạo ván chơi mới”.
* Hai người chơi:
  + Đưa ra máy tính
  + Đưa bàn cờ về trạng thái chưa kết thúc
  + Ẩn phần chọn kinh nghiệm cho máy tính, chọn máy đi trước hay không.
  + Khởi tạo giá trị cho biến về số ván thắng của người chơi và máy=0.
  + Thông báo chiến thắng khởi tạo về rỗng.
  + Đường dẫn ảnh để hiển thị ảnh máy tính
  + Gọi lại chức năng “tạo ván chơi mới”.
* Chọn kiểu chơi: hiển thị form :
* Người chơi chọn âm thanh,
* Chọn máy đi trước hay không,
* Chọn kinh nghiệm cho máy,
* Chọn loại quân

### **2.2. Thuật toán xây dựng trò chơi Caro**

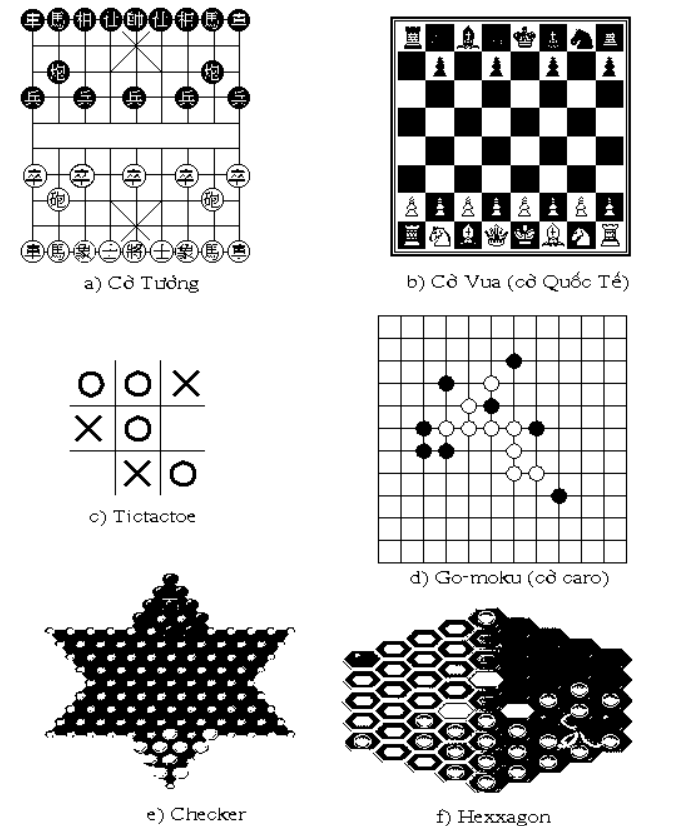
#### **2.2.1. Tham khảo một số thuật toán**

##### **2.2.1.1. Dạng trò chơi**

Trong phần này, ta sẽ xem cách một chương trình máy tính có thể chơi được các trò chơi đấu trí như các trò chơi cờ Vua, cờ Tướng, cờ vây, cờ caro (go-moku), go, checker... như thế nào. Các trò này còn gọi là các trò chơi đối kháng, diễn ra giữa hai đấu thủ. Nói chung, các trò chơi đó đều có thể chuyển về một dạng bài toán tìm kiếm đặc biệt: tìm đường đi đến các điểm cao nhất giữa hai đấu thủ. Đặc điểm của các trò chơi trên như sau:

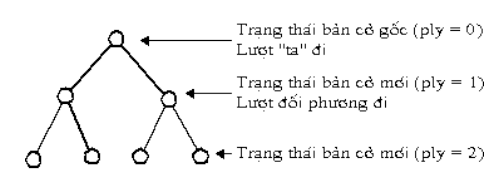
* Có hai đấu thủ, mỗi người chỉ đi một nước khi tới lượt.
* Các đấu thủ đều biết mọi thông tin về tình trạng trận đấu.
* Trận đấu không kéo dài vô tận, phải diễn ra hòa, hoặc một bên thắng và bên kia thua.

Thông thường ta hay gọi các trò chơi này là các loại cờ. Đôi khi ta gọi đây là các trò chơi Minimax (dựa trên tên của thuật toán tìm kiếm cơ bản áp dụng cho chúng). Hình 6 là ví dụ về một số trò chơi nói trên. Các trò chơi như chơi bài, dò mìn, xúc xắc... không thuộc lớp trò chơi này.



Hình 2.4. Các loại trò chơi cờ

##### **2.2.1.2. Cây trò chơi**:

Các trạng thái bàn cờ khác nhau (hay còn gọi là một thế cờ, tình huống cờ) trong quá trình chơi có thể biểu diễn thành một cây tìm kiếm (được gọi là cây trò chơi - hình 2.5) và ta sẽ tiến hành tìm kiếm trên cây để tìm được nước đi tốt nhất. Cây trò chơi có các nút của cây là các tình huống khác nhau của bàn cờ, các nhánh nối giữa các nút sẽ cho biết từ một tình huống bàn cờ chuyển sang tình huống khác thông qua chỉ một nước đi đơn nào đó. Dĩ nhiên, các nước đi này diễn ra theo cặp do hai đấu thủ lần lượt tiến hành. Độ sâu của cây trò chơi ply là số tầng của cây (chính là độ sâu d của cây). Thuật ngữ “nước đi” trong sách được thống nhất chỉ bao gồm một lần đi của một đấu thủ hoặc một lần đi phản ứng lại của đối thủ bên kia. Chú ý nó khác với thói quen dùng trong thực tế một nước đi bao gồm lần đi của ta và một lần đi của đối thủ. Nói cách khác, nước đi ở đây thực chất chỉ là "nửa nước" theo cách hiểu của làng cờ.   
  
  
 Hình 2.5.Cây trò chơi

##### **2.2.1.3. Vét cạn :**

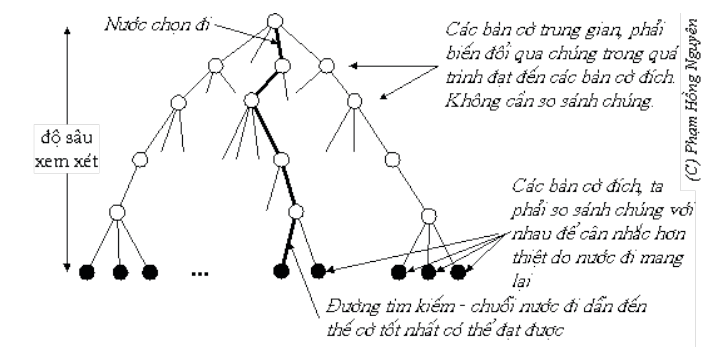
Dùng một thuật toán vét cạn để tìm kiếm trên cây trò chơi dường như là một ý tưởng đơn giản. Ta chỉ cần chọn nhánh cây sẽ dẫn tới nước thắng để đi quân là đảm bảo thắng lợi. Nếu đúng vậy, các loại cờ sẽ trở thành các trò chơi buồn tẻ, sẽ chẳng còn đâu những bí quyết huyền ảo thần kì và bàn cờ sẽ chẳng khác gì bàn... tính. Rất tiếc (hoặc rất may) rằng, cách làm này lại không thể thực hiện nổi do cái gọi là bùng nổ tổ hợp. Ví dụ, nếu từ một thế cờ, trung bình có khả năng đi được 16 nước đi khác nhau (ta gọi đó là hệ số nhánh con tại mỗi nút là b = 16). Như vậy, sau một tầng ta sẽ có 16 nút con, mỗi nút này lại có thể có 16 con nữa. Tổng số nút con ở độ sâu thứ hai là 16x16 = b^2. Cứ như vậy ở độ sâu d sẽ có b^d nút.   
  
Nếu giả sử độ sâu của cây là 100 (hệ số nhánh 16 và độ sâu 100 đều là những con số còn nhỏ hơn con số thường gặp trong trò chơi cờ), thì số nhánh phải duyệt lên đến 16^100 hay xấp xỉ 10^120 - một con số lớn khủng khiếp. Để hình dung số đó lớn thế nào, ta giả sử tất cả các nguyên tử trong vũ trụ đều trở thành máy tính để tính nước đi với tốc độ một giây tính được cỡ 10^10 (10 tỷ) nước đi, và nếu chúng hoạt động cật lực từ thời vụ nổ lớn đến nay (theo một số lý thuyết, thì thế giới này hình thành sau một vụ nổ gọi là vụ nổ lớn bigbang, trước đây cỡ 15 tỷ năm) thì đến bây giờ mới có thể đi được nước đi đầu tiên.

##### **2.2.1.4. Chiến lược tìm kiếm trong trò chơi :**

Một chiến lược thường được cả người lẫn máy dùng là phân tích thế cờ chỉ sau một số nước đi nào đó của cả hai bên. Sau khi "nhìn xa" xem bàn cờ có những khả năng biến đổi như thế nào sau một số nước, ta sẽ đánh giá độ xấu tốt của các thế cờ nhận được. Tiếp theo, ta sẽ chọn nước đi sẽ dẫn tới một thế cờ tốt nhất trong số đó có cân nhắc đến cách đi của cả hai bên. Với máy, thế cờ này được đánh giá là tốt hơn thế cờ kia nhờ so sánh điểm của thế đó do bộ lượng giá trả lại. Chúng ta chỉ có khả năng xét trước một số hữu hạn các nước (ví dụ đại kiện tướng chơi cờ vua có thể xét trước 8-10 nước đi, người thường chỉ 2-4 nước đi). Rõ ràng là nếu xét càng sâu thì chơi càng giỏi. Nhưng không thể thực hiện điều này với độ sâu quá lớn được do số nút ở độ sâu đó có thể trở nên lớn khủng khiếp và không đủ thời gian để phân tích. Nếu dừng ở một độ sâu hợp lý thì bộ phân tích có thể hoàn thành việc tính toán trong một thời gian hạn định. 

##### **2.2.1.5. Thủ tục Minimax[1] :**

Giả sử chúng ta có một bộ phân tích thế cờ có thể áp dụng tất cả các luật, các phương pháp đánh cờ khác nhau vào từng thế cờ và chuyển đổi chúng thành một con số đại diện (cho điểm thế cờ). Mặt khác, ta giả sử con số đó là dương khi áp dụng cho thế cờ của một đấu thủ (được gọi là người chơi cực đại - maximizer), và là âm khi áp dụng cho đấu thủ bên kia (được gọi là người chơi cực tiểu - minimizer). Quá trình tính toán cho điểm thế cờ được gọi là lượng giá tĩnh (static evaluation). Hàm thực hiện việc tính toán được gọi là một bộ lượng giá tĩnh, và giá trị nhận được gọi là điểm lượng giá tĩnh. Cả hai đấu thủ đều cố gắng đi như thế nào đó để đạt được điểm tuyệt đối lớn nhất. Người chơi cực đại sẽ tìm những nước đi dẫn đến điểm của mình trở nên lớn hơn (hay cao nhất có thể được) hay điểm của đối thủ bớt âm hơn (nhỏ hơn về giá trị tuyệt đối). Còn đấu thủ của anh ta, người chơi cực tiểu, lại ra sức phản kháng lại, để dẫn tới điểm âm của anh ta âm hơn hay điểm dương của đối thủ nhỏ đi (hình 2.6).



Hình 2.6. Minh họa chiến lược cờ của lẫn máy. Các nút trắng là các thế cờ trung gian phải trải qua để đến được thế cờ đích.Không cần phải xét độ xấu tốt của các thế cờ trung gian. Các nút đen là các thế cờ đích và phải xem xét độ tốt xấu để so sánh chúng với nhau. Từ đó sẽ tìm được đường đi để đến thế cờ tốt nhất có thể được.

Người chơi cực đại hi vọng chọn nước đi bên phải để đạt được điểm 8. Thế nhưng nếu đi như vậy thì khi đến lượt đi của người chơi cực tiểu, anh ta sẽ cố gắng không cho người chơi cực đại đạt được điểm này bằng cách chọn nước đi nhánh bên trái và như vậy, người chơi cực đại chỉ được có 1 điểm thay vì 8. Ngược lại, nếu người chơi cực đại chọn nước đi bên trái, thì trong tình huống xấu nhất anh ta vẫn còn được 2 điểm, lớn hơn là chọn nước đi bên phải. Nói chung, người chơi cực đại sẽ phải tìm cách nhận ra các nước đi của đối phương tiếp theo làm cho điểm giảm xuống. Và tương tự như vậy, người chơi cực tiểu phải nhận biết được nước đi của người chơi cực đại cố gắng làm tăng điểm lên. Thủ tục tìm nước đi tốt nhất trên cây trò chơi như trên được gọi là thủ tục Minimax do điểm ở mỗi nút có thể là điểm cực đại hoặc có thể là điểm cực tiểu và có thuật toán như sau:   
  
  
**Thuật toán Minimax**   
- Nếu như đạt đến giới hạn tìm kiếm (đến tầng dưới cùng của cây tìm kiếm), tính giá trị tĩnh của thế cờ hiện tại ứng với người chơi ở đó. Ghi nhớ kết quả   
- Nếu như mức đang xét là của người chơi cực tiểu, áp dụng thủ tục Minimax này cho các con của nó. Ghi nhớ kết quả nhỏ nhất   
- Nếu như mức đang xét là của người chơi cực đại, áp dụng thủ tục Minimax này cho các con của nó.Ghi nhớ kết quả lớn nhất.   
  
  
  
**Viết chương trình cho thuật toán Minimax**

Bây giờ, ta thử dựa vào phát biểu trên để viết chương trình cho thuật toán này bằng ngôn ngữ tựa Pascal. Đây là một hàm có tên là Minimax và sẽ là loại đệ qui. Trước hết, để hàm này biết đã đạt đến giới hạn tìm kiếm chưa, ta cần cung cấp cho nó một tham số về độ sâu tìm kiếm depth (để biết phải tìm đến đâu), đồng thời ta cũng phải cho biết thế cờ hiện tại pos để nó từ đó nó biết cách tính tiếp. Giá trị trả về của hàm chính là điểm của thế cờ (bàn cờ) pos. Vậy hàm sẽ có khai báo dạng:   
  
**function** Minimax (pos, depth): integer;   
  
Mỗi khi Minimax được gọi, nó sẽ càng gần đến giới hạn tìm kiếm, do đó ta sẽ gọi hàm này với độ sâu bằng độ sâu cũ trừ đi một. Đạt đến độ sâu giới hạn chính là khi depth = 0. Khi đạt độ sâu này ta sẽ gọi hàm lượng giá Eval để đánh giá chất lượng của thế cờ pos hiện tại (thực hiện điều một của thuật toán). Như vậy bước đầu hàm này có dạng sau: 

function Minimax (pos, depth): integer;

begin

if depth = 0 then { Đã đạt đến giới hạn }

Minimax := Eval (pos) { Tính giá trị thế cờ pos }

else begin

...

Minimax (pos, depth - 1); { Gọi đệ qui với độ sâu giản dần}

...

end;

end;

Ở trên, Minimax được gọi với độ sâu giảm đi một. Đó là độ sâu của các thế cờ là con. Các thế cờ con pos' đó là các thế cờ được tạo ra từ pos bằng cách đi một nước đi hợp lệ m nào đó. Do đó ta phải có các lệnh thực hiện đi quân để đến các thế cờ mới. Để biết từ thế cờ pos có thể đi được những nước nào, ta dùng một thủ tục Gen có tham số là thế cờ cha pos. Thủ tục này sẽ cất các thế cờ con pos' đó vào bộ nhớ (dạng danh sách). Việc tiếp theo là ta lấy từng thế cờ đó ra và áp dụng tiếp thủ tục Minimax cho nó để tính điểm value của nó.   
  
Vậy hàm Minimax bây giờ có dạng: 

function Minimax (pos, depth): integer;

begin

if depth = 0 then

Minimax := Eval (pos) { Tính giá trị thế cờ pos }

else begin

Gen (pos); { Sinh ra mọi nước đi từ thế cờ pos }

while còn lấy được một nước đi m do

begin

pos := Tính thế cờ mới nhờ đi m;

value := Minimax (pos, depth-1); { Tính điểm của pos }

...

end;

...

end;

end;

Theo phát biểu của thuật toán, ta thấy các điều 2 và 3 chỉ khác nhau ở cách chọn kết quả tốt nhất best phụ thuộc vào người chơi đang là người chơi cực đại hay cực tiểu. Cuối cùng thuật toán sẽ trả về điểm tốt nhất đạt được. Vậy hàm này được phát triển tiếp thành: 

function Minimax (pos, depth): integer;

begin

if depth = 0 then

Minimax := Eval (pos) { Tính giá trị thế cờ pos }

else begin

Gen (pos); { Sinh ra mọi nước đi từ thế cờ pos }

while còn lấy được một nước đi m do

begin

pos := Tính thế cờ mới nhờ đi m;

value := Minimax (pos, depth-1); { Tính điểm của pos }

{ Chọn điểm tốt nhất tuỳ thuộc theo người chơi }

if người chơi là người cực đại then

begin

if best < value then best := value;

end

else begin

if best > value then best := value;

end

end;

Minimax := best; { Trả về giá trị tốt nhất }

end;

end;

Thông thường để cho tiện (và cũng rất gần sự thực) ta coi cả hai người chơi (hai bên) có cùng cách đánh giá về một thế cờ. Có điều thế cờ này là tốt với một người thì phải được đánh giá là tồi với người kia và ngược lại. Trong máy tính cách thể hiện tốt nhất là ta cho điểm một thế cờ có thêm dấu âm dương: dấu âm dành cho người chơi cực đại và dấu âm cho người chơi cực tiểu. Với người chơi cực đại sẽ mong muốn điểm này càng dương càng tốt, còn người chơi cực tiểu lại mong muốn điểm này càng âm càng tốt. Do đó để dễ xử lí ta sẽ tuỳ theo mức người chơi mà đổi dấu giá trị đánh giá thế cờ pos. Chú ý rằng, thay đổi độ sâu là chuyển sang đối phương nên phải đổi dấu. Chương trình thực hiện đổi dấu như sau:   
  
value := -Minimax (pos, depth-1); { Tính điểm của pos }   
  
Cũng do dùng cùng hàm lượng giá nên khi đến lượt người chơi cực đại và cực tiểu có cùng cái nhìn như nhau về một thế cờ. Điều này dẫn đến có thể dùng cùng cách chọn nước đi tốt nhất cho họ (gộp được điều 2 và 3 lại với nhau được). Giá trị best cần được khởi đầu rất nhỏ để đảm bảo không vượt mọi giá trị value, tốt nhất là giá trị -vô cùng: 

function Minimax (pos, depth): integer;

begin

if depth = 0 then

Minimax := Eval (pos) { Tính giá trị thế cờ pos }

else begin

best := -INFINITY;

Gen (pos); { Sinh ra mọi nước đi từ thế cờ pos }

while còn lấy được một nước đi m do

begin

pos := Tính thế cờ mới nhờ đi m;

value := -Minimax (pos, depth - 1);

if value > best then best := value;

end;

Minimax := best;

end;

end;

Thông thường, bàn cờ được biểu diễn bằng các biến toàn cục. Do đó thay cho truyền tham số là một bàn cờ mới pos vào thủ thục Minimax thì người ta biến đổi luôn biến toàn cục này nhờ thực hiện nước đi "thử" (nước đi dẫn đến bàn cờ mới pos). Sau khi Minimax thực hiện việc tính toán dựa vào bàn cờ lưu ở biến toàn cục thì thuật toán sẽ dùng một số thủ tục để loại bỏ nước đi này. Như vậy Minimax bỏ các tham số pos như sau: 

function Minimax (depth): integer;

begin

if depth = 0 then Minimax := Eval { Tính thế cờ pos trong biến toàn cục }

else begin

best := -INFINITY;

Gen; { Sinh ra mọi nước đi từ thế cờ pos }

while còn lấy được một nước đi m do

begin

thực hiện nước đi m;

value := -Minimax (depth - 1);

bỏ thực hiện nước đi m;

if value > best then best := value;

end;

Minimax := best;

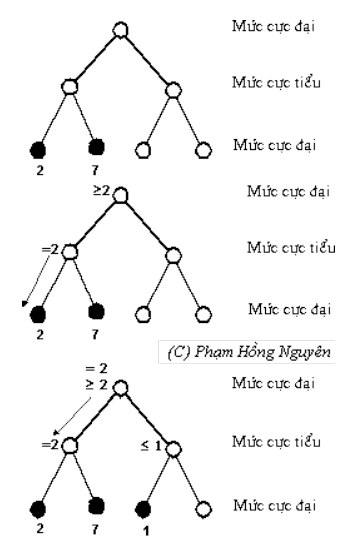
end;

end;

##### 

##### **2.2.1.6. Thủ tục AlphaBeta pruning:**

Thủ tục AlphaBeta là một cải tiến thuật toán Minimax nhằm tỉa bớt nhánh của cây trò chơi, làm giảm số lượng nút phải sinh và lượng giá, do đó có thể tăng độ sâu của cây tìm kiếm. Giả sử hình 2.7 là một thế cờ mà hai nút đầu tiên đã được lượng giá. Nếu thực hiện thủ tục Minimax đối với các nút đó sẽ cho thấy người chơi cực đại đã được đảm bảo nếu đi nước bên trái sẽ được ít nhất là 2 điểm dù là các lượng giá của các nút khác cho kết quả như thế nào đi nữa.



Hình 2.7. Thuật toán AlphaBeta

Bây giờ, ta lại giả sử nút tiếp theo được lượng giá và cho kết quả là 1. Nếu đi vào nhánh này thì đối phương sẽ đảm bảo làm điểm của người chơi cực đại không thể vượt quá được giá trị 1 dù là các lượng giá của các nút khác cho kết quả như thế nào đi nữa. Do đó đến đây, nước đi tốt nhất là chọn nước đi bên trái với đảm bảo là ít nhất đạt được 2 điểm. Và do đó, hoàn toàn không cần thiết phải lượng giá nút còn lại.   
  
**Nguyên tắc Alpha-Beta**   
  
Ý tưởng này được gọi là nguyên tắc Alpha-Beta do nó dùng trong thủ tục AlphaBeta (ta sẽ xét dưới đây). Hai tham số của thủ tục này (theo các đặt tên truyền thống) được gọi là alpha và beta và dùng để theo dõi các triển vọng - chúng cho biết các giá trị nằm ngoài khoảng [alpha, beta] là các điểm "thật sự tồi" và không cần phải xem xét nữa. Khoảng [alpha, beta] còn được gọi là cửa sổ alpha, beta. Trong ngữ cảnh của các trò chơi, nguyên tắc Alpha-Beta nói rằng, mỗi khi xem xét một nút bất kì, nên kiểm tra các thông tin đã biết về các nút cha, ông của nó. Rất có thể do có đủ thông tin từ cha, ông nên không cần phải làm bất cứ việc gì nữa cho nút này. Cũng vậy, nguyên tắc này cũng giúp chỉnh sửa hoặc xác định chính xác giá trị tại nút cha, ông nó. Như trên nói, một cách để tiện theo dõi quá trình tính toán là dùng các tham số alpha và beta để ghi lại các thông tin theo dõi cần thiết. Thủ tục AlphaBeta được bắt đầu tại nút gốc với giá trị của alpha là -vôcùng và beta là +vôcùng. Thủ tục sẽ tự gọi đệ quy chính nó với khoảng cách giữa các giá trị alpha và beta ngày càng hẹp hơn.   
  
**Viết chương trình cho thuật toán AlphaBeta** 

function AlphaBeta(alpha, beta, depth): integer;

begin

if depth = 0 then

AlphaBeta := Eval { Tính giá trị thế cờ pos }

else begin

best := -INFINITY;

Gen; { Sinh ra mọi nước đi từ vị trí pos }

while (còn lấy được một nước đi m) and (best < beta) do

begin

if best > alpha then alpha := best;

thực hiện nước đi m;

value := -AlphaBeta(-beta, -alpha, depth-1);

bỏ thực hiện nước đi m;

if value > best then best := value;

end;

AlphaBeta := best;

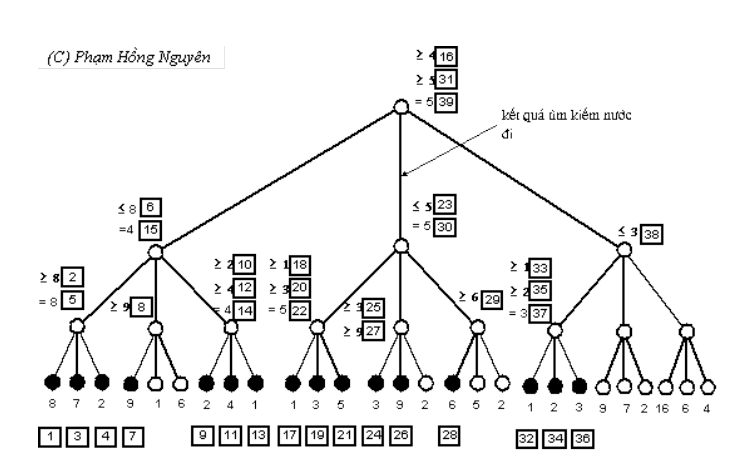
end;

end;

Lời gọi thủ tục AlphaBeta đầu tiên với độ sâu tìm kiếm 4 và thế cờ hiện tại pos có dạng như sau:   
  
AlphaBeta(-INFINITY, +INFINITY, 4);   
  
 Cũng tương tự như thuật toán Minimax ta đã gộp hai mục 2 và 3 làm một nhờ việc đổi dấu thích hợp. So với thuật toán Minimax thì trong thuật toán AlphaBeta đã đưa thêm hai biến alpha, beta làm hai mức ngưỡng. Ta thấy cứ mỗi khi best >= beta thì thuật toán không thực hiện tiếp vòng lặp, có nghĩa là nó không chịu mở rộng tiếp những nhánh còn lại nữa. Các nhánh đó đã bị cắt bỏ - và do đó ta sẽ tiết kiệm được thời gian. Việc cắt bỏ này hoàn toàn an toàn với những lí do ta đã xét ở trên. Ta thấy rằng mỗi lần hàm này được gọi thì chỉ có tham số beta được dùng để so sánh cắt bỏ, còn tham số alpha không được dùng. Tuy nhiên khi áp dụng cùng thuật toán cho cây con thì ta đã hoán vị hai giá trị alpha, beta cho nhau (và đảo cả dấu), do đó alpha sẽ có tác dụng trong độ sâu sau, rồi độ sâu sau nữa lại đến lượt beta... Nói cách khác, một giá trị chỉ luôn ảnh hưởng đến người chơi cực đại, còn giá trị kia lại luôn ảnh hưởng đến người chơi cực tiểu. Chúng là các ngưỡng của họ (ngưỡng giữa các nước đi được chấp nhận và không chấp nhận). Những nước đi cần quan tâm phải nằm lọt giữa hai giá trị này. Dần dần khoảng cách giữa hai giá trị alpha - beta càng ngày càng thu hẹp và dẫn đến các nhánh cây có giá trị nằm ngoài khoảng này nhanh chóng bị cắt bỏ.

Ta có thể nhận xét như sau:   
  
- Số lần tăng số nút khi tăng độ sâu của Minimax luôn là hệ số phân nhánh b, trong trường hợp này là 40. Số lần tăng của AlphaBeta ít hơn nhiều: chỉ cỡ 1.7 lần khi tăng từ d lẻ sang d chẵn và 23.2 lần khi từ d chẵn sang lẻ - trung bình chỉ tăng khoảng hơn 6 lần khi tăng d

- Số nút của AlphaBeta tăng chậm hơn rất nhiều lần so với Minimax. Tỉ số nút phải xét giữa hai thuật toán này càng cao khi d càng lớn.

Công thức tính số nút cho thấy số nút phải xét khi dùng AlphaBeta ít hơn nhiều so với Minimax nhưng vẫn là hàm số mũ và vẫn dẫn tới bùng nổ tổ hợp. Thuật toán AlphaBeta hoàn toàn không chống được bùng nổ tổ hợp mà chỉ làm giảm tốc độ bùng nổ. Tuy trong thực tế số nút phải xét (lượng giá) thường nhiều hơn trong điều kiện lí tưởng nhưng nó vẫn đủ để tiết kiệm khá nhiều thời gian. Trong cùng một khoảng thời gian, thuật toán AlphaBeta có thể tìm đến độ sâu gấp hai lần độ sâu tìm kiếm bằng Minimax.    
  
  
  
  
Ví dụ: Ta sẽ xem xét thuật toán AlphaBeta hoạt động như thế nào đối với cây trò chơi như trong hình 2.8.   
  
  
 

Hình 2.8. cây trò chơi  
  
  
Cây này có độ sâu bằng 3 và hệ số phân nhánh bằng 3. Các thứ tự kết luận (các con số bên trái) được đưa ra như sau:   
  
[1-2] Tìm kiếm đi xuống dưới theo nhánh trái cho đến lá. Ở đây giá trị tĩnh thu được là 8. Giá trị đầu tiên này do người chơi cực đại được phép chọn trong ba giá trị ở nhánh này đã đảm bảo rằng là kết quả thu được sẽ ít nhất là bằng 8. Điều lưu ý này được bước 2 ghi lại.   
  
[3-5] Để chắc chắn không còn có điểm nào cao hơn 8, người chơi cực đại phải xét cả hai thế cờ còn lại và thu được các giá trị 7 và 2. Do đó đến đây đã kết luận chính xác điểm cao nhất có thể đạt được ở cây con là đúng bằng 8.   
  
[6]. Leo lên một tầng cây. Đây là các nước đi của người chơi cực tiểu. Ta không hi vọng anh ta cho người chơi cực đại được nhiều điểm nên có thể tạm kết luận ở mức này là sẽ đạt được nhiều nhất là 8 điểm.   
  
[7-8]. Để xem người chơi cực tiểu còn lựa chọn nào tốt hơn (và tồi tệ hơn cho người chơi cực đại) ta phải xem xét cả hai nước đi còn lại. Nước đi còn lại đầu tiên dẫn đến giá trị lượng giá tĩnh là 9 - một giá trị lớn hơn 8. Như vậy nhánh giữa là tồi tệ hơn cho người chơi cực tiểu. Đến đây việc cắt bỏ được thực hiện - đừng hòng người chơi cực đại với tới được điểm đó khi đã có sẵn lựa chọn thấp hơn cho anh ta (là 8). Điều này cũng dẫn đến không cần thiết phải xét hai nút còn lại - đằng nào nhánh giữa cũng đủ tồi tệ rồi và người chơi cực tiểu sẽ không chọn nó để đi.   
  
[9-14]. Người chơi cực tiểu cần phải khảo sát tiếp lựa chọn cuối cùng. Cách làm tương tự như phần trên. Ở đây phải lượng giá cả ba nút cây và kết luận cuối cùng được đưa ra là người chơi cực đại đi giỏi lắm thì chỉ đạt được 4 điểm.   
  
[15]. Như vậy nhờ việc khảo sát nhánh cây bên phải người chơi cực tiểu thấy rằng nếu chọn đi theo nhánh này thì người chơi cực đại chỉ được có 4 điểm thay cho 8.   
  
[16]. Bây giờ ta có thể kết luận ở mức trên cùng. Mức này là của người chơi cực đại. Anh ta thấy rằng nếu chọn đi theo nhánh trái thì được 4 điểm. Như vậy anh ta đã chắc chắn điểm của mình sẽ ít nhất là 4 rồi. Để xem liệu có thể đạt được điểm cao hơn nữa hay không cần phải xem xét hai nhánh còn lại.   
  
[17-30]. Tương tự như phần trên, ta kết luận nhánh giữa sẽ mang lại cho người chơi cực đại 5 điểm. 31. Cũng tương tự như kết luận 16, ở đây ta kết luận khả quan hơn là người chơi cực đại đã cầm chắc 5 điểm và có thể còn cao hơn.   
  
[32-38] Ta kết luận được rất nhanh là cây con bên phải chỉ cho "thu hoạch" nhiều nhất là 3 điểm - một điểm số quá kém do đó thuật toán không buồn xem xét các trường hợp còn lại nữa. Do đó đã tiết kiệm được 6 nút không cần phải lượng giá và cũng không phải sinh nước đi cho hai trường hợp.   
  
[39]. Kết luận cuối cùng là điểm cao nhất mà người chơi cực đại có thể thu được là 5 điểm nhờ chọn đi theo nhánh giữa.

##### **2.2.1.7. Hướng cải thiện việc tỉa nhánh của thuật toán AlphaBeta :**

Thuật toán AlphaBeta nói chung giúp chúng ta tiết kiệm nhiều thời gian so với Minimax mà vẫn đảm bảo kết quả tìm kiếm chính xác. Tuy nhiên lượng tiết kiệm này không ổn định - phụ thuộc vào số nút mà nó cắt bỏ. Trong trường hợp xấu nhất thuật toán không cắt được một nhánh nào và phải xét số nút đúng bằng Minimax. Ta cần đẩy mạnh việc cắt bỏ nhờ đẩy nhanh sự thu hẹp của cửa sổ tìm kiếm alpha - beta. Cửa sổ này được thu hẹp một bước khi gặp một giá trị mới tốt hơn giá trị cũ. Khi gặp giá trị tốt nhất thì cửa sổ này thu hẹp nhất. Do đó nếu càng sớm gặp giá trị tốt nhất thì cửa sổ càng chóng thu hẹp. Như vậy phải làm sao cho các nút ở lá được sắp xếp theo trật tự từ cao xuống thấp. Trật tự này càng tốt bao nhiêu thì thuật toán chạy càng nhanh bấy nhiêu (các công thức về số nút phải lượng giá trong điều kiện lý tưởng ở trên tính được với trật tự là tốt nhất). Ta sẽ trở lại phần này trong một chương riêng.

##### **2.2.1.8.Tổng kết :**

Chương này trình bày những kiến thức chung về trò chơi cờ, các định nghĩa và thế nào là cây trò chơi. Do bùng nổ tổ hợp quá lớn của cây trò chơi mà cả người và máy không thể (và không bao giờ) có thể tìm kiếm vét cạn (hết mọi khả năng). Do đó phương pháp tìm kiếm duy nhất là chỉ tìm kiếm đến một độ sâu giới hạn nào đó và chọn nước đi dẫn đến một thế cờ có lợi nhất cho mình. Do phải tính cả khả năng chống trả của đối phương nên ta không dùng được các thuật toán tìm kiếm thông thường. Phải dùng một thuật toán tìm kiếm riêng cho cây trò chơi. Đó là thuật toán Minimax và cải tiến của nó là AlphaBeta. Tuy cả hai thuật toán đều không tránh được bùng nổ tổ hợp nhưng AlphaBeta làm chậm bùng nổ tổ hợp hơn nên được dùng nhiều trong các trò chơi cờ.

### **2.3. Thuật toán xây dựng trò chơi**

#### **2.3.1. Tìm nước đi cho máy**:

Sau khi người chơi đi nước đi của mình, sẽ tới lượt đi quân cờ cho máy. Lúc này máy cần tìm nước đi phù hợp với thế cờ hiện tại có lợi cho máy tính:

**a) Lượng giá thế cờ:**

1. **B1:** khởi tạo giá các ô vuông trong bàn cờ, mỗi ô vuông có giá trị là 0
2. **B2:** lượng giá cho thế cờ để lấy được 10 nước đi tốt nhất cho máy.

Giải thuật lượng giá thế cờ (tính giá trị từng ô vuông còn trống trong bàn cờ). Ở đây xét lượt đi cho máy tính ( lượt đi quân O)

Chú ý: Các ô vuông còn trồng được lượng giá với 2 giá trị là :

* + - Tấn công hoặc phòng thủ.
    - Trong tấn công có các mức tấn công tùy thuộc vào thế cờ.
    - Trong phòng thủ có các mức phòng thủ tùy thuộc vào thế cờ.
  + **Cách xét:** 
    - máy duyệt từng cặp 5 ô vuông liên tiếp nhau trong 1 hàng, tuần tự đến hết hàng này, chuyển xuống hàng dưới, cho tới khi hết hàng 15 trên bàn cờ.
    - Tiếp đó, máy tính duyệt 5 ô vuông liên tiếp nhau trong 1 cột, hết cột này tới cột khác, đến hết cột 15 trong bàn cờ.
    - Máy duyệt 5 ô vuông liên tiếp trên đường chéo dưới ( tương tự như cột và hàng nói ở trên).
    - Máy duyệt 5 ô vuông liên tiếp trên đường chéo trên ( tương tự như cột và hàng nói ở trên).
  + **Cách duyệt 5 ô vuông liên tiếp trong 1 hàng:**
* Ban đầu đếm trong 5 ô vuông liên tiếp có số quân máy( cntM ), số quân người chơi (cntH) là bao nhiêu.
* Xét 5 ô vuông này với điều kiện trong 5 ô vuông

cntM khác cntH và (cntM = 0 hoặc cntM =0), hay số quân máy khác số quân người chơi và số quân trong 5 ô vuông được xét của 1 trong 2 bên phải = 0 (cntH =0 hoặc cntM=0 );

* ờ đây ta giả sử cntM = 3, cntH =0; và lượng giá cho máy (lượt máy đánh)

 O: quân máy

Vậy ở đây thấy rằng quân máy có 3 quân liên tiếp. Ở đây là lượng giá thế cờ cho lượt đi của máy.vậy lượng giá (tính giá trị) cho 2 ô trống còn lại là tấn công.

* Giả sử với cntM = 0; cntH = 3;

 X: quân người chơi

Thấy rằng có 3 quân người chơi trong 5 ô trống, đang xét lượng giá cho máy, vậy cần phải phòng thủ hay 2 ô trống còn lại có giá trị phòng thủ cho máy.

***Sơ đồ thuật toán lượng giá cho thế cờ***

**Chú thích:**

+ rw : (row) là biến đếm cho số hàng.

+ cl : (column) là biến đếm cho số cột.

+ cntM: (countMachine) là biến đếm số quân máy trong 5 ô vuông liên tiếp.

+ cntH: (countHuman) là biến đếm số quân người chơi trong 5 ô vuông liên tiếp.

+ ov: là ký hiệu chỉ ô vuông

+ M: (Machine) là ký hiệu chỉ quân cờ máy tính.

+ ov == M : xét tại ô vuông này có phải là quân cờ của máy tính đã đánh.

+ H: (Human) là ký hiệu chỉ quân cờ của người chơi.

+ ov == H : xét tại ô vuông này có phải là quân cờ của người chơi đã đánh.

+ ot : ô vuông trống (ô vuông chưa được quân nào đánh vào).

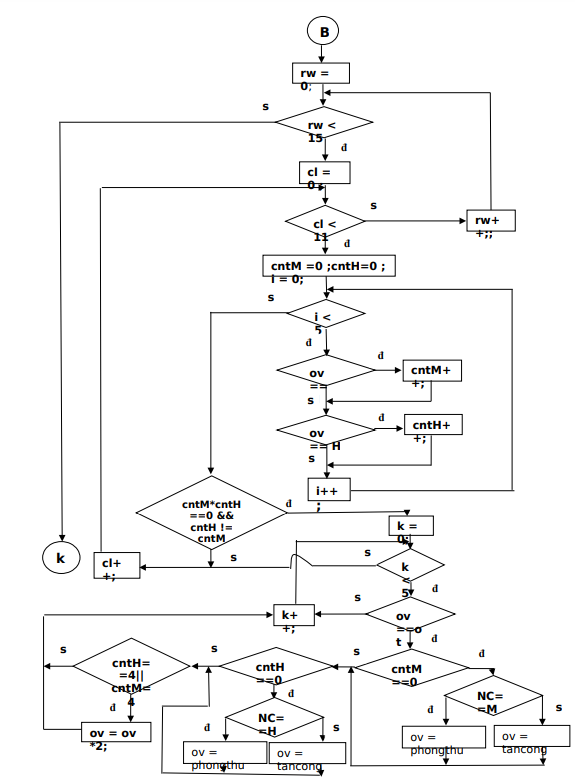
+ ov == ot : xét ô vuông hiện tại có phải là ô vuông trống không ?

+ phongthu: phòng thủ.

+ tancong: tấn công.

+ ov = phongthu là giá trị ô vuông trống trong 5 ô vuông đang xét được gán là 1 giá trị phòng thủ.

+ ov = tancong là giá trị ô vuông trống trong 5 ô vuông đang xét được gán là 1 giá trị tấn công.

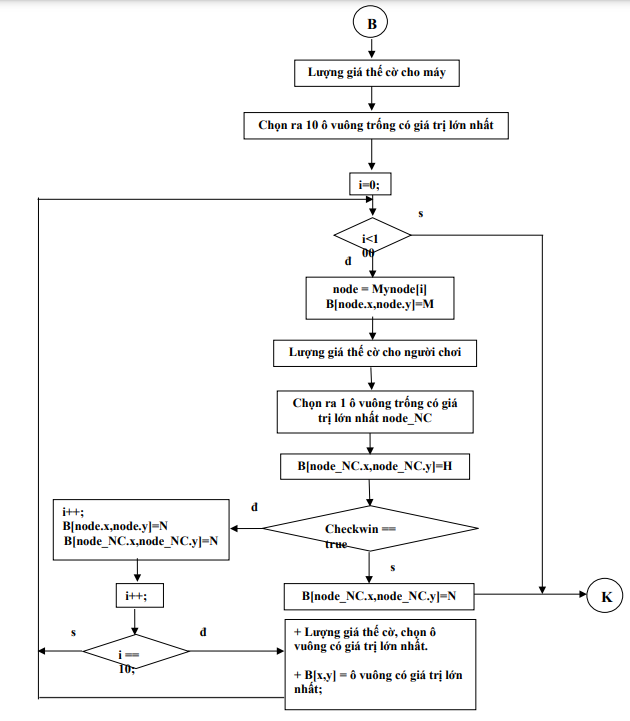


Hình 2.9. *Sơ đồ thuật toán lượng giá cho thế cờ*

1. **B3:** Với thuật toán trên thì lượng giá cho thế cờ (tính giá trị cho các ô vuông còn trống) theo hàng. Còn với cột, đường chéo xuống , đường chéo lên cũng áp dụng tương tự.

**b) Tìm nước đi quân cho máy tính:**

***Sơ đồ thuật toán tính nước cờ của máy***



Hình 2.10. *Sơ đồ thuật toán tính nước cờ của máy*

**Chú thích:**

+ lượng giá cho máy: là lượng khi tới lượt máy đi quân.

+ lượng giá cho người chơi: là lượng khi tới lượt người chơi đi quân.

+ chọn ra 10 ô vuông có giá trị lớn nhất , lưu 10 ô vuông đó vào Mynode[10];

+ node = Mynode[i], chọn ra node có giá trị lớn thứ I trong mang Mynode[10];

+ M: (Machine) là ký hiệu quân cờ bên máy tính.

+ H: (Human) là ký hiệu quân cờ bên người chơi.

+ N : (None) là kí hiệu ô vuông trống

+ B[node.x,node.y] = M; gán quân cờ của máy vào ô vuông có tọa độ mà theo node.

+ B[node\_NC.x,node\_NC.y] = H; gán quân cờ của người chơi vào ô vuông có tọa độ mà theo node\_NC.

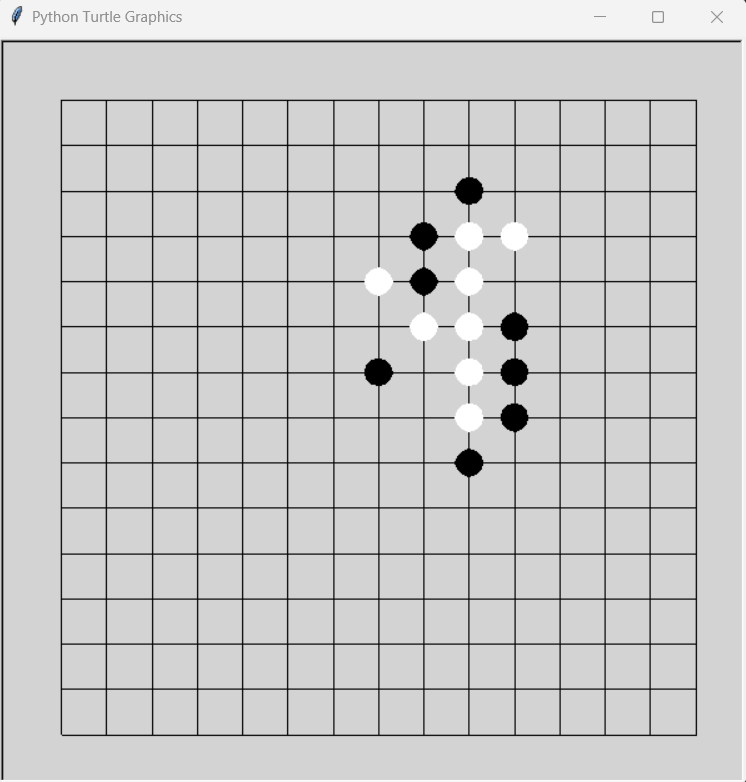
+ B[node.x,node.y]=N : gán ô trống vào ô vuông có tọa độ node.x và node.y

+ B[node\_NC.x,node\_NC.y]=N : gán ô trống vào ô vuông có tọa độ node\_NC.x và node\_NC.y.

### **2.4. Thiết kế giao diện:**

Xây dựng giao diện của trò chơi:

- Trò chơi được thiết kế với 1 Form: Form frmMain,( Form Chính)



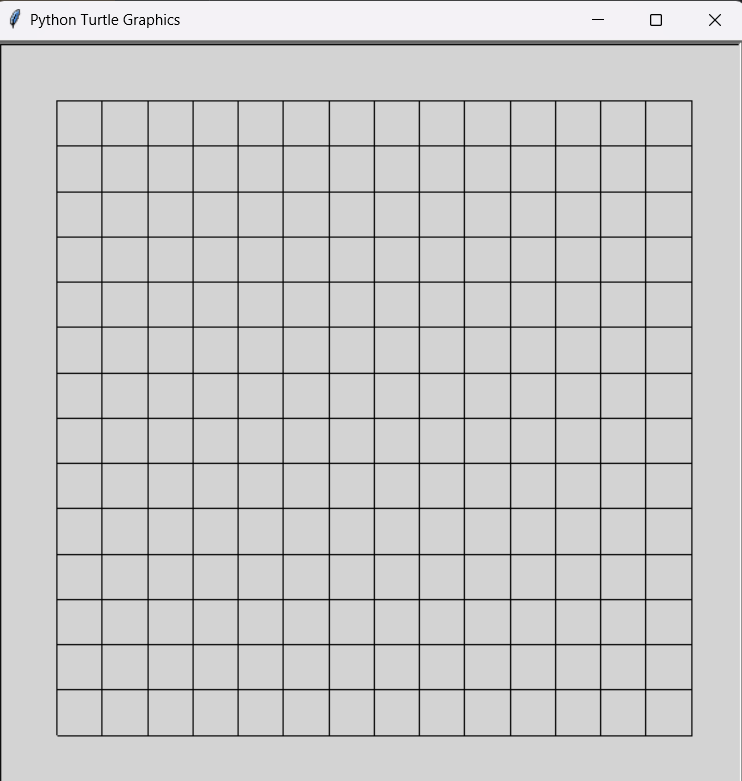
Hình 2.11. Giao diện chính

* cờ bắt đầu lại ( ván chơi mới)
* Chơi với máy **:** chế độ chơi cờ giữa người và máy.

\*Phần 2:

Là phần giao diện được chia thành 14x14 đường kẻ ngang và dọc cách

đều nhau (tạo thành 196 ô vuông xếp liền nhau theo từng hàng, mỗi hàng có 14 ô vuông, và có 14 hàng), 1 ô vuông = 30px.



Hình 2.12. Giao diện bàn cờ

# **CHƯƠNG III. Cài đặt và kiểm thử**

## **Môi trường cài đặt**

### **1.1. Cấu hình máy:**

### **1.2. Hệ điều hành:**

Tương thích với các hệ điều hành Microsoft Window 98, Window 2000, Window xp,Window 7, 10, 11 …

### **1.3. Các phần mềm hệ thống:**

* Pycharm

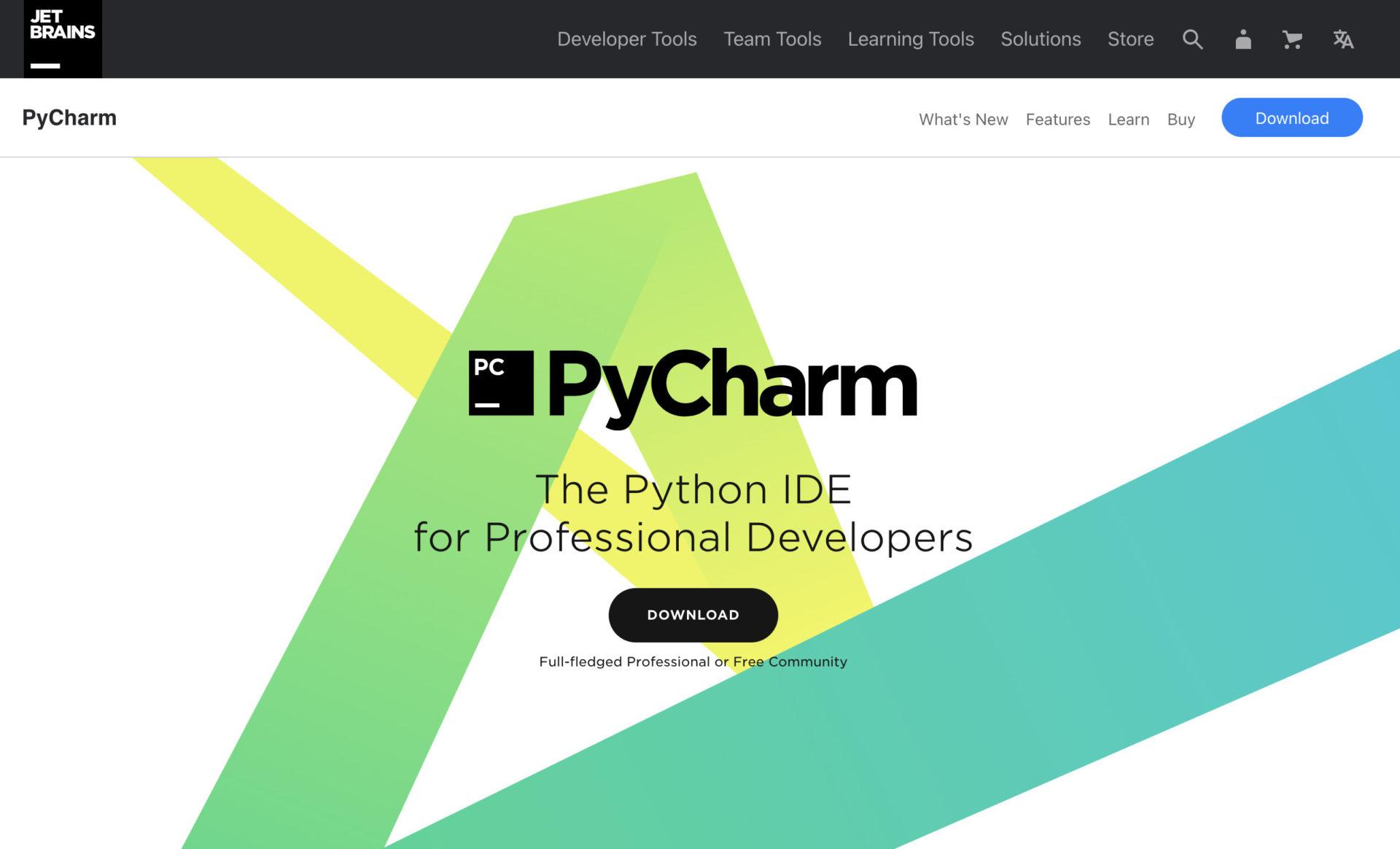
## **Cách cài đặt và kiểm thử**

### **2.1. Cài đặt môi trường Pycharm**

**Bước 1:** Truy cập vào website của

JetBrains: <https://www.jetbrains.com/pycharm/>

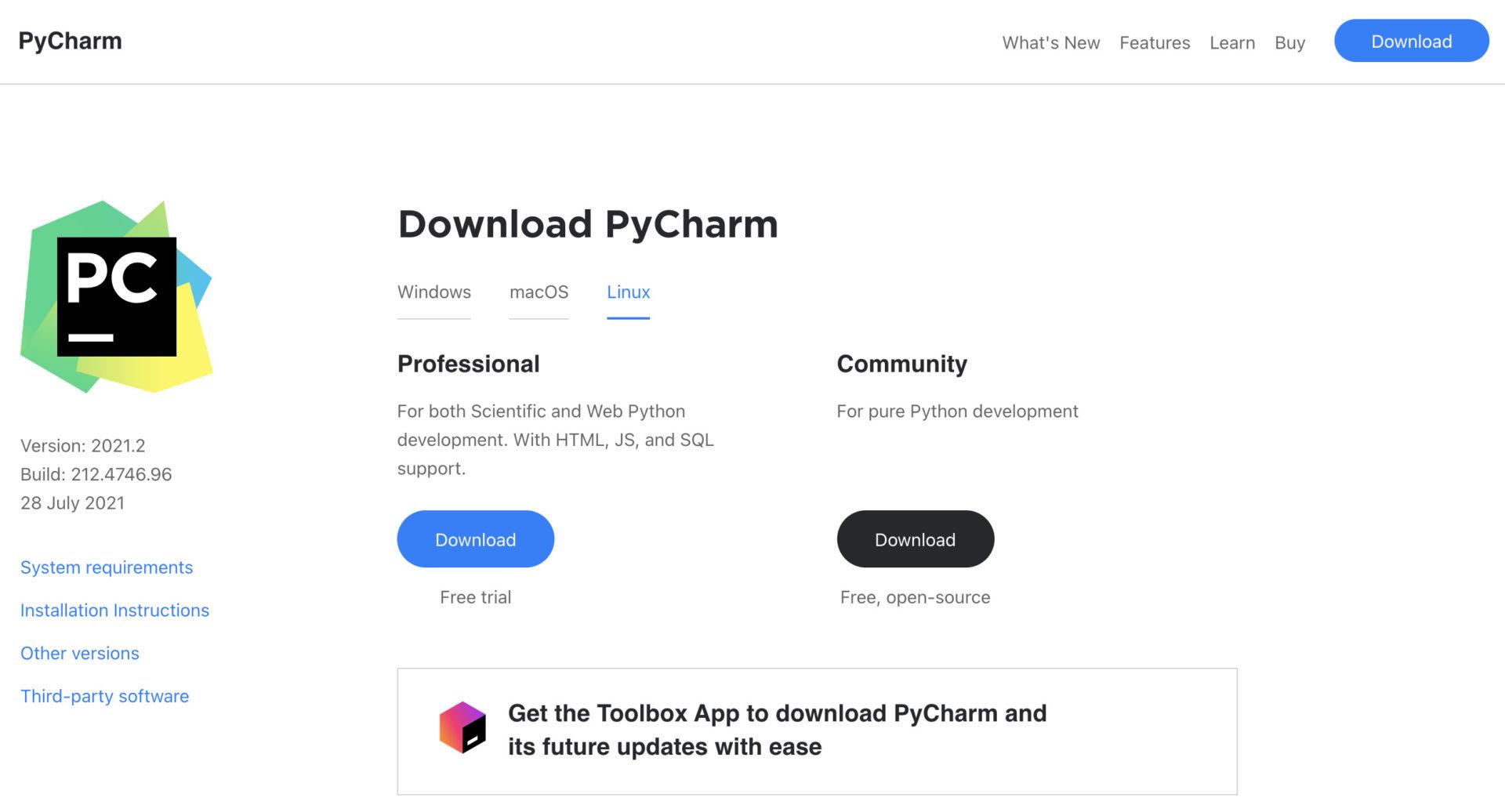
**Bước 2:** Nhấn chọn “Download”



Hình 3.1. Download PyCharm

**Bước 3:**Tại đây, bạn sẽ thấy 2 phiên bản PyCharm:

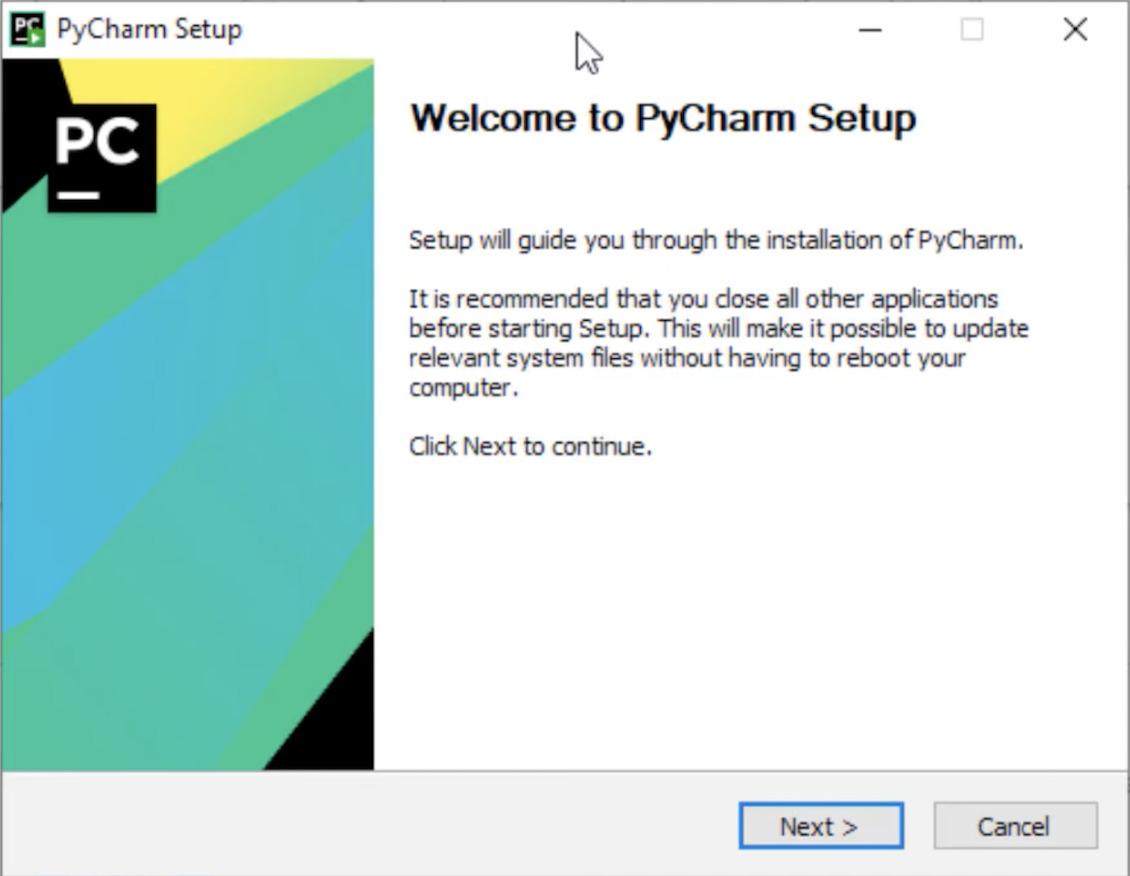
* Bản Professional: Chứa đầy đủ các tính năng từ cơ bản đến nâng cao để lập trình Python. Tuy nhiên, phiên bản này bạn sẽ phải mua bản quyền để sử dụng.
* Bản Community: Chỉ chứa những tính năng cơ bản để lập trình Python. Tuy nhiên, phiên bản này hoàn toàn miễn phí, bạn không cần trả phí như bản Professional.



Hình 3.2. Các phiên bản PyCharm

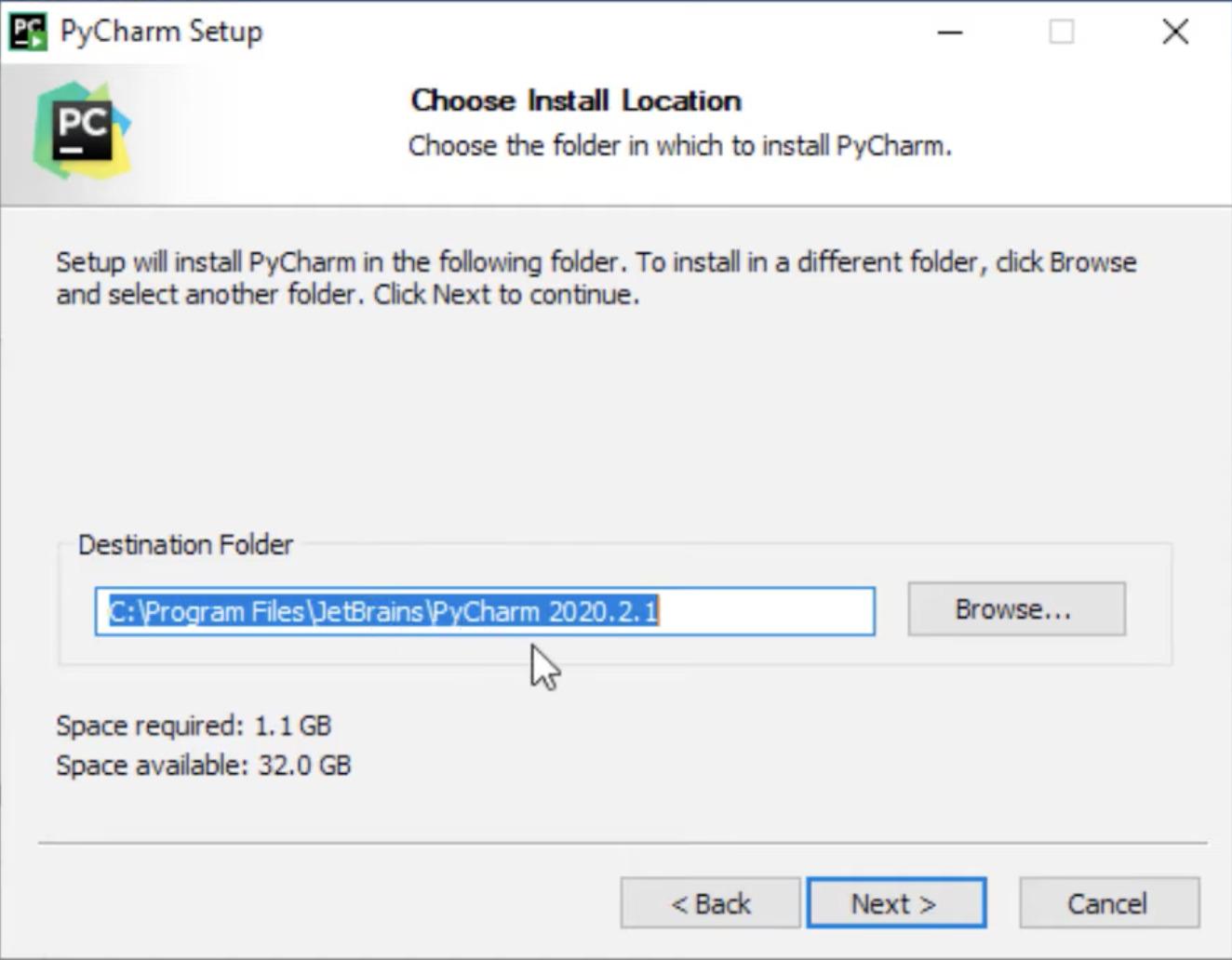
**Bước 4:** Sau khi tải xuống thành công, PyCharm sẽ được lưu tại thư mục Download của máy tính. Lúc này, bạn nhấn đúp chuột vào thư mục để bắt đầu cài đặt PyCharm.

Giao diện chào mừng như sau được hiển thị, chọn Next để tiếp tục cài đặt.



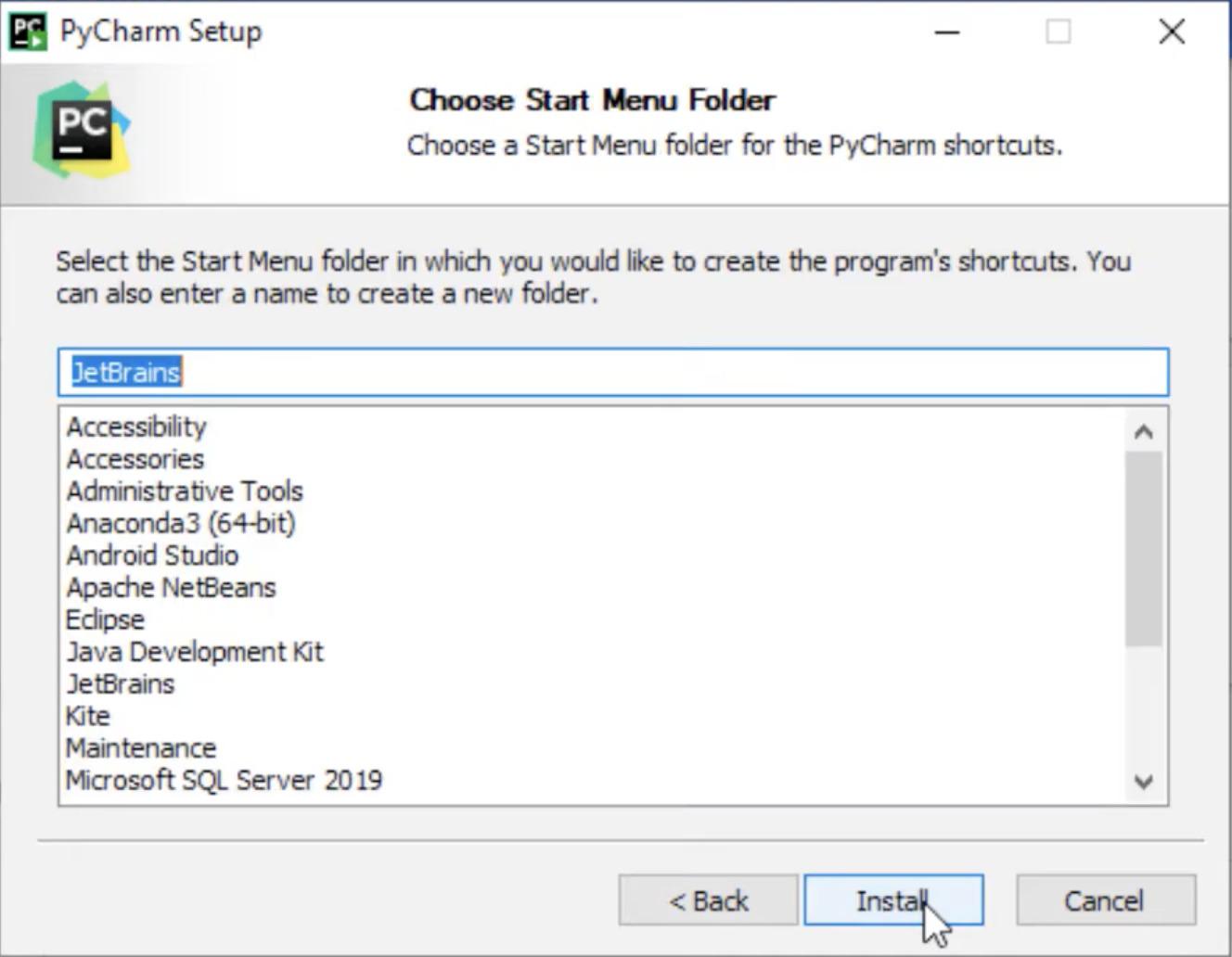
Hình 3.3 Setup PyCharm

**Bước 5:** Cửa sổ thiết lập vị trí cài đặt sẽ xuất hiện, bạn có thể chọn một thư mục cho vị trí cài đặt hoặc giữ lại đường dẫn mặc định. Sau đó, tiếp tục chọn Next để đến bước tiếp theo.



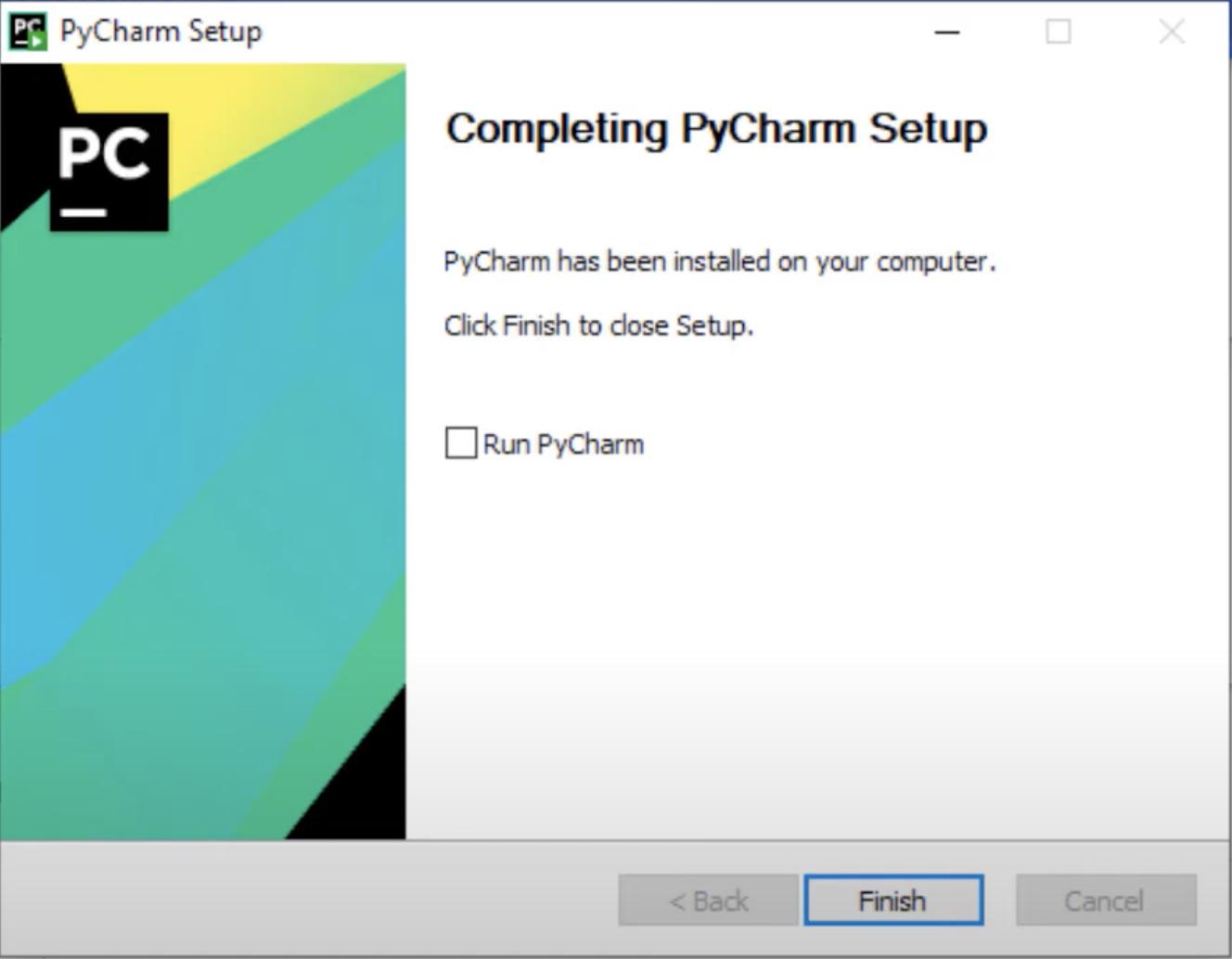
Hình 3.4. Install Location PyCharm

**Bước 6:** Chọn Install để thực hiện quá trình cài đặt. Sau khi cài đặt xong thì nhấn Finish để hoàn tất



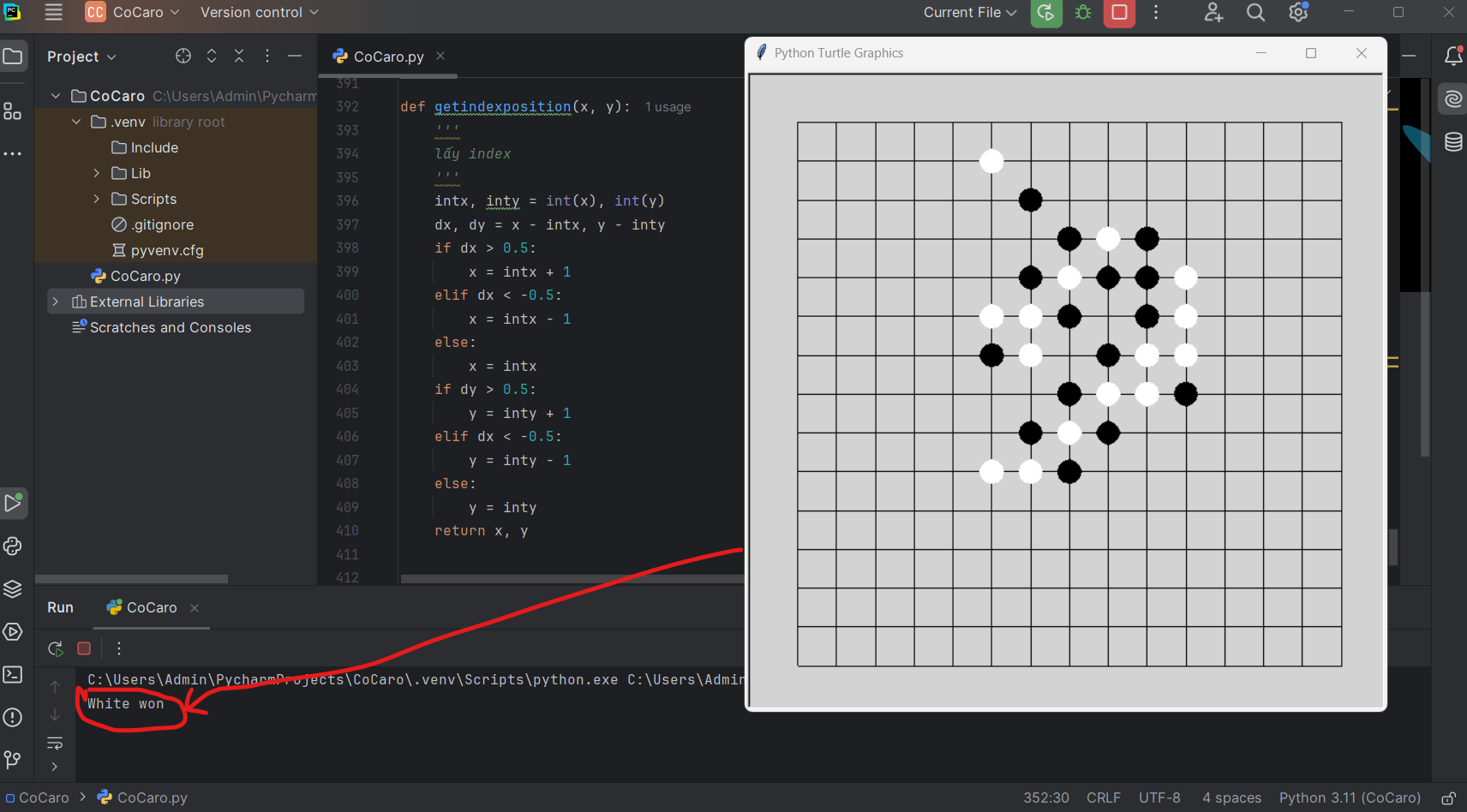
Hình 3.5 Install PyCharm

***Chọn Install để bắt đầu cài đặt***

**

Hình 3.6. Hoàn tất cài đặt PyCharm

Cài đặt chương trình trò chơi caro bằng Pycharm và tiến hành kiểm thử



Hình 3.7 chạy chương trình cờ caro

Trong trường hợp có kết quả thì sẽ hiển thị ở trong hộp thông báo như hình. Chương trình hoạt động bình thường.

# **KẾT LUẬN**

**Những kết quả đạt được:**

* Xây dựng thành công trò chơi Caro trên PC.
* Học tập và rút ra được một số kinh nghiệm về xây dựng Game
* Củng cố thêm kiến thức về kỹ thuật lập trình, một số thuật toán để xây dựng các trò chơi có tính đối kháng : BFS và DFS, thủ tục cắt tỉa alpha beta, min – max.
* Nắm rõ hơn về lập trình hướng đối tượng, phân tích và thiết kế hệ thống
* Game Caro này được xây dựng trên môi trường Pycharm, em đã tìm hiểu và học hỏi được thêm về công cụ xây dựng:Ngôn ngữ lập trình Python.
* Dễ sử dụng: trò chơi rất đơn giản, chỉ cần tuân theo quy luật là có thể thành công
* Tính tiện dụng: vì trò chơi rất nhỏ gọn có thể download dễ dàng, cài đặt nhanh chóng người chơi chỉ mất tâm 1 đến 2 phút là có thể cài xong trò chơi.
* Vì game có dung lượng nhỏ và giao diện đơn giản nên thích hợp với nhiều loại máy thông dụng
* Game có nhiều cấp độ và kiểu chơi nên phù hợp với nhiều người sử dụng.

**Hướng phát triển trong tương lai:**

**Trí tuệ nhân tạo** hay **trí thông minh nhân tạo** ([tiếng Anh](http://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *artificial intelligence* hay *machine intelligence*, thường được viết tắt là **AI**) là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói đến các [máy tính](http://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh) có mục đích không nhất định và ngành [khoa học](http://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc) nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo.

Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong [khoa học viễn tưởng](http://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_vi%E1%BB%85n_t%C6%B0%E1%BB%9Fng), nó là một trong những ngành trọng yếu của [tin học](http://vi.wikipedia.org/wiki/Tin_h%E1%BB%8Dc). Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc. Các ví dụ ứng dụng bao gồm các tác vụ điều khiển, lập kế hoạch và lập lịch (*scheduling*), khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty, nhận dạng chữ viết tay, nhận dạng tiếng nói và khuôn mặt. Bởi vậy, trí thông minh nhân tạo đã trở thành một môn học, với mục đích chính là cung cấp lời giải cho các vấn đề của cuộc sống thực tế. Ngày nay, các hệ thống nhân tạo được dùng thường xuyên trong kinh tế, y dược, các ngành kỹ thuật và quân sự, cũng như trong các [phần mềm](http://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) máy tính thông dụng trong gia đình và [trò chơi điện tử](http://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD).

Vì vậy em xây dựng Game Caro này cũng có 1 mục đích bước đầu tìm hiểu về mảng **Trí tuệ nhân tạo.** Tiếp tục tìm hiểu và xây dựng Game Caro với 1 số tính năng như: máy có thể học cách đi của người, đưa ra phán đoán về nước đi…..Và để Game có thể được nhiều người biết tới, em sẽ cố gắng tích hợp vào di động hay đưa lên website bằng cách sử dụng công cụ khác để xây dựng.

Trong thời gian làm đồ án, em đã cố gắng hết mình để hoàn thành đồ án tốt nhất. Nhưng với khả năng hiện tại và kinh nghiệm còn hạn chế, đồ án sẽ không tránh khỏi những thiếu sót. Em mong quý thầy cô cùng bạn bè thông cảm và đóng góp ý kiến để em hoàn thiện thêm kiến thức cho mình và đồ án đạt hiệu quả cao hơn.

Xin cảm ơn các quý thầy, cô đã chỉ bảo cho em kiến thức ban đầu về ngành CNTT trong 3 năm học qua. !

Cảm ơn các bạn và gia đình đã giúp đỡ trong quá trình làm đồ án này !

# **TÀI LIỆU THAM KHẢO**

[1]. Prentice.Hall.Visual.C.Sharp.2005.How.to.Program.2nd.Edition.Dec.

2005.part 2

[2]. Giáo trình trí tuệ nhân tạo (các tài liệu tham khảo qua mạng)

[3]. <https://pdfcoffee.com/thuat-toan-minimaxpdf-pdf-free.html>