**TRƯỜNG ĐẠI HỌC KỸ THUẬT - CÔNG NGHỆ CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN HỌC PHẦN**

**ĐỀ TÀI**

**XÂY DỰNG CHƯƠNG TRÌNH ĐÁNH CỜ CA RÔ**

**BẰNG NGÔN NGỮ PYTHON**

**HỘI ĐỒNG:**

**GVHD: Phạm Yến Nhi**

**GVPB: ………………………………………**

**---o0o---**

**SVTH : Triệu Phương Nam (1500380)**

**Cần Thơ, ngày 18 tháng 12 năm 2017**

**NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN**

Hưng Yên, ngày … tháng 12 năm 2012

GIẢNG VIÊN HƯỚNG DẪN

**NHẬN XÉT GIẢNG VIÊN PHẢN BIỆN**

Hưng Yên, ngày … tháng 12 năm 2012

GIẢNG VIÊN PHẢN BIỆN

**LỜI CAM ĐOAN**

**LỜI CẢM ƠN/ LỜI NGỎ**

Sau thời gian thực hiện làm đề tài đến nay em đã hoàn thành đề tài của mình. Để có được kết quả như hôm nay, em thực hiện đề tài đã nhận được rất nhiều sự quan tâm, giúp đỡ và chỉ đạo nhiệt tình của các thầy, cô trong khoa công nghệ thông tin.

Đặc biệt em thực hiện đề tài xin gửi lời cảm ơn chân thành nhất tới cô **“ Phạm Yến Nhi”.** Cô là người đã tận tình chỉ bảo, chia sẻ cho em những kiến thức bổ ích hướng dẫn động viên trong suốt thời gian qua, tạo mọi điều kiện tốt nhất cho em trong quá trình thực hiện.

Mặc dù em thực hiện đề tài đã rất cố gắng trong quá trình làm đề tài song đồ án của em không thể tránh khỏi những thiếu sót, hạn chế. Em thực hiện đề tài rất mong nhận được sự đóng góp chân thành từ các thầy, cô và các bạn để đề tài của em được hoàn thiện hơn.

*Em thực hiện đề tài xin chân thành cảm ơn!*

**Mục lục**

[**NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN 2**](#_Toc502861321)

[**NHẬN XÉT GIẢNG VIÊN PHẢN BIỆN 3**](#_Toc502861322)

[**LỜI CAM ĐOAN 4**](#_Toc502861323)

[**I. Tóm tắt Đồ án 9**](#_Toc502861324)

[**1. Tổng quan 9**](#_Toc502861325)

[**2. Phạm vi nghiên cứu, phạm vi ứng dụng 10**](#_Toc502861326)

[**3. Đối tượng nghiên cứu 10**](#_Toc502861327)

[**4. Mục đích nghiên cứu 10**](#_Toc502861328)

[**5. Nhiệm vụ nghiên cứu 10**](#_Toc502861329)

[**6. Phương pháp nghiên cứu 10**](#_Toc502861330)

[**7.** **Chức năng, công việc đã làm được** 10](#_Toc502861331)

[**II.** **Tìm hiểu Python** 10](#_Toc502861332)

[**1. Tổng quan về Python** 10](#_Toc502861333)

[a) Lý do ra đời Python 10](#_Toc502861334)

[b) Python là gì? 11](#_Toc502861335)

[**2. Đặc điểm** 11](#_Toc502861336)

[**3. Python hướng đối tượng:** 12](#_Toc502861337)

[a. Khái niệm đối tượng 12](#_Toc502861338)

[b. Từ khóa class 13](#_Toc502861339)

[c. Thuộc tính 14](#_Toc502861340)

[d. Phương thức 17](#_Toc502861341)

[e. Kế thừa 19](#_Toc502861342)

[f. Đa hình 20](#_Toc502861343)

[g. Các phương thức đặc biệt 21](#_Toc502861344)

[**III.** **Lý thuyết game** 24](#_Toc502861345)

[a) Mở đầu 24](#_Toc502861346)

[c) Tại sao phải cần dùng minimax? 25](#_Toc502861347)

[d) Giải pháp: 25](#_Toc502861348)

[**IV.** **Xây dựng trò chơi** 25](#_Toc502861349)

[1. Tạo bàn cho bàn cờ caro 25](#_Toc502861350)

[2. File vẽ 28](#_Toc502861351)

[3. Tạo màng hình bắt đầu trò chơi 30](#_Toc502861352)

[4. game 34](#_Toc502861353)

[5. Giải thuật 35](#_Toc502861354)

[6. File caro 38](#_Toc502861355)

Mục lục hình ảnh

[Hinh : 1 Ví vụ khái niệm hướng đối tượng 13](#_Toc502136670)

[Hinh : 2 Ví vụ khái niệm hướng đối tượng 1.1 13](#_Toc502136671)

[Hinh : 3 ví vụ về từ khóa lớp 14](#_Toc502136672)

[Hinh : 4 ví vụ về từ khóa lớp 1.1 14](#_Toc502136673)

[Hinh : 5 ví vụ về từ khóa lớp 1.2 14](#_Toc502136674)

[Hinh : 6 ví vụ về thuộc tính 15](#_Toc502136675)

[Hinh : 7 ví vụ về thuộc tính 1.1 15](#_Toc502136676)

[Hinh : 8 ví vụ về thuộc tính 1.2 15](#_Toc502136677)

[Hinh : 9 ví vụ về thuộc tính 1.3 15](#_Toc502136678)

[Hinh : 10 ví vụ về thuộc tính 1.4 15](#_Toc502136679)

[Hinh : 11 ví dụ về thuộc tính 1.5 15](#_Toc502136680)

[Hinh : 12 ví vụ về thuộc tính 1.6 16](#_Toc502136681)

[Hinh : 13 ví vụ về thuộc tính 1.7 16](#_Toc502136682)

[Hinh : 14 ví vụ về thuộc tính 1.8 16](#_Toc502136683)

[Hinh : 15 ví vụ về thuộc tính 1.9 16](#_Toc502136684)

[Hinh : 16 ví vụ về thuộc tính 2.0 17](#_Toc502136685)

[Hinh : 17 ví vụ về thuộc tính 2.1 17](#_Toc502136686)

[Hinh : 18 ví vụ về phuong thức 18](#_Toc502136687)

[Hinh : 19 ví vụ về phuong thức 1.1 18](#_Toc502136688)

[Hinh : 20 ví vụ về phuong thức 1.2 18](#_Toc502136689)

[Hinh : 21 ví vụ về phuong thức 1.3 18](#_Toc502136690)

[Hinh : 22 ví vụ về phuong thức 1.4 19](#_Toc502136691)

[Hinh : 23 ví vụ về phuong thức 1.5 19](#_Toc502136692)

[Hinh : 24 ví dụ kế thừa 19](#_Toc502136693)

[Hinh : 25 ví vụ về kế thức 1.1 20](#_Toc502136694)

[Hinh : 26 ví vụ về đa thức 1..2 20](#_Toc502136695)

[Hinh : 27 ví vụ về đa thức 20](#_Toc502136696)

[Hinh : 28 ví vụ về đa thức 1.1 20](#_Toc502136697)

[Hinh : 29 ví vụ về đa thức 1.2 21](#_Toc502136698)

[Hinh : 30 ví vụ về đa thức 1.3 21](#_Toc502136699)

[*Hinh :* 31 Các phương thức đặc biệt 22](#_Toc502136700)

[Hinh : 32 Các phương thức đặc biệt 1.1 22](#_Toc502136701)

[Hinh : 33 Các phương thức đặc biệt 1.2 22](#_Toc502136702)

[Hinh : 34 các phương thức đặc biệt 1.3 23](#_Toc502136703)

[Hinh : 35 các phương thức đặc biệt 1.4 23](#_Toc502136704)

[Hinh : 36 các phương thức đặc biệt 1.5 24](#_Toc502136705)

[Hinh : 37 bàn cờ caro 35](#_Toc502136706)

[Hinh : 38 Chon che do tro choi 37](#_Toc502136707)

[Hinh : 39 huong dan choi voi may 38](#_Toc502136708)

[Hinh : 40 huong dan tro choi 38](#_Toc502136709)

[Hinh : 41 vẽ chọn trang 41](#_Toc502136710)

[Hinh : 42 vẽ màng hình bắt đầu trò chơi 42](#_Toc502136711)

[Hinh : 43 vẽ tấn công hay phòng thủ 43](#_Toc502136712)

[Hinh : 44 vẽ lượt chọn đánh với chế độ dễ và khó 44](#_Toc502136713)

1. **Tóm tắt Đồ án**
2. **Tổng quan**

Khi rảnh rỗi hoặc sau mỗi giờ làm việc căng thẳng thì mỗi người trong chúng trong chúng ta đều muốn tìm cho mình một trò chơi hoặc một hoạt động để giảm căng thẳng và lấy lại tinh thần. Có thể đó là các hoạt động thể thao, hoặc chơi các trò chơi giải trí tùy thuộc vào quỹ thời gian của từng người, và với những ai có ít quỹ thời gian nhưng muốn có một trò chơi mang tính giải trí cao thì game Cờ caro là một trong những game đáp ứng được nhu cầu đó:

* **Gomoku:** Hiện nay được chơi chính thức với bàn 13x13. Không có hoà. Nếu hết đất thì Trắng thắng. Chưa tìm được chứng minh nào cho thấy Đen chắc chắn thắng. Tuy nhiên Đen vẫn có ưu thế rất lớn.
* **ProGomoku.** Chơi trên bàn 15x15. Nước đầu của Đen đặt sẵn ở trung tâm. Nước thứ ba (nước thứ hai của Đen) phải đặt ngoài hình vuông cấm. Hình vuông cấm là hình vuông trung tâm kích thước 5x5. Không có hạn chế cho Trắng. Đã có chứng minh Đen chắc chắn thắng trong biến thể này.
* **Cờ caro** ban đầu loại cờ này được chơi bằng các quân cờ màu trắng và đen trên một bàn cờ vây (19x19). Quân đen đi trước và người chơi lần lượt đặt một quân cờ của họ trên giao điểm còn trống. Người thắng là người đầu tiên có được một chuỗi liên tục gồm 5 quân hàng ngang, hoặc dọc, hoặc chéo. Tuy nhiên, vì một khi đã đặt xuống, các quân cờ không thể di chuyển hoặc bỏ ra khỏi bàn, do đó loại cờ này có thể chơi bằng giấy bút.

Do sự thông dụng của cờ caro có thể chơi mỗi lúc mỗi nơi nên khi cờ caro xuất hiện phổ Việt Nam với những biến thể, hình thức được lan rộng từ đó em đã làm game Cờ Carô theo những quy tắc trong đó mỗi đối thủ trong mỗi lượt đi của mình sẽ chọn 1 đánh trên ô lưới còn lại trên bàn cờ (kẽ sẵn các ô lưới):

* Bàn cờ có kích thước tuỳ ý 15x15.
* Quân cờ đầu tiên được đánh tùy theo người đánh, nếu là máy thì đánh vào ô giữa của bàn cờ.
* Nếu tồn tại đúng 5 con liên tiếp trên 1 hàng là thắng (chéo, ngang, dọc).
* Nếu hết chỗ đi thì 2 bên hoà.

Mục tiêu chính của đề tài là giúp chơi tăng cường trí nhớ, giác quan và thấy thoải mái tinh thần sau khi đã dánh thắng đối thủ. Ngoài ra chương trình còn có chức năng đấu với máy với hai cấp độ chơi khi bạn chơi một mình và bạn muốn năng trình độ của mình.

Đồ án được xây dựng trò chơi “Cờ Caro” sử dụng ngôn ngữ Python và các cách tiếp cận giải quyết các vấn đề trong trò chơi cùng những kết quả đã đạt được dựa trên một số những ví dụ thử nghiệm trong khi lập trình.

1. **Phạm vi nghiên cứu, phạm vi ứng dụng**

* Phạm vi nghiên cứu : tự nghiên cứu.
* Phạm vi ứng dụng: mỗi người

1. **Đối tượng nghiên cứu**

* Python
* Chương trình chơi game caro
* Công cụ xây dựng phần mềm: IDLE 3.6

## Mục đích nghiên cứu

* Xây dựng được tài liệu giúp tìm hiểu về ngôn ngữ python
* Xây dựng được chương trình chơi game caro có sử dụng ngôn ngữ python

## Nhiệm vụ nghiên cứu

* Tìm hiểu giải thuật minimax.
* Tìm hiều Tkinter để áp dụng vào xây dựng chương trình game caro.

## Phương pháp nghiên cứu

* Tìm hiểu hết tổng hợp kiến thức chung của python , sau đó đi sâu vào những kiến thức cụ thể để xây dựng giao diện ứng dụng cờ caro.
* Nghiên cứu Tkinter để dựa vào đó thiết kế giao diện cho desktop.

1. **Chức năng, công việc đã làm được**

|  |  |  |
| --- | --- | --- |
| STT | Chức năng, công việc | Đã làm được |
| 1 | Tạo giao diện thiện, dễ sử dụng. | √ |
| 2 | Có thể chơi 1 người hoặc 2 người | √ |
| 3 | Reset lạ khi đánh xong | √ |
| 4 | Chọn mức độ dễ, khó | √ |
| 5 | Hướng dẫn chơi | √ |

1. **Tìm hiểu Python**
2. **Tổng quan về Python**
3. Lý do ra đời Python

**Python** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [thông dịch](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_th%C3%B4ng_d%E1%BB%8Bch) do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra năm [1990](https://vi.wikipedia.org/wiki/1990). Python hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl). Python được phát triển trong một dự án [mã mở](http://www.opensource.org/), do tổ chức phi lợi nhuận [Python Software Foundation](http://www.python.org/psf/about/) quản lý.

1. Python là gì?

* Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng, đa mục đích và cũng là một ngôn ngữ lập trình động.
* Cú pháp của Python là khá dễ dàng để học và ngôn ngữ này cũng mạnh mẽ và linh hoạt không kém các ngôn ngữ khác trong việc phát triển các ứng dụng. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục.
* Python không chỉ làm việc trên lĩnh vực đặc biệt như lập trình web, và đó là tại sao ngôn ngữ này là đa mục đích bởi vì nó có thể được sử dụng với web, enterprise, 3D CAD, …
* Với Python, việc phát triển ứng dụng và debug trở nên nhanh hơn bởi vì không cần đến bước biên dịch và chu trình edit-test-debug của Python là rất nhanh.

## Đặc điểm

Dưới đây là một số đặc điểm chính của Python:

* **Dễ dàng để sử dụng**: Python là một ngôn ngữ bậc cao rất dễ dàng để sử dụng. Python có một số lượng từ khóa ít hơn, cấu trúc của Python đơn giản hơn và cú pháp của Python được định nghĩa khá rõ ràng, …
* Bạn có thể **đọc code của Python khá dễ dàng**. Phần code của Python được định nghĩa khá rõ ràng và rành mạch.
* Python có một **thư viện chuẩn khá rộng lớn**.
* Có khả năng tương tác với các module khác viết trên **C/C++** (Hoặc **Java** cho **Jython**, hoặc **.Net** cho **IronPython**).
* Python cũng là một ngôn ngữ lập trình **hướng đối tượng**. Ngoài ra, Python còn hỗ trợ các phương thức lập trình theo hàm và theo cấu trúc.
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

#### **Python dễ dàng kết nối với các thành phần khác:**

* **Python** có thể kết nối với các đối tượng **COM**, **.NET** (**Ironpython, Python for .net**), và **CORBA**, **Java**… **Python** cũng được hỗ trợ bởi **Internet Communications Engine (ICE)** và nhiều công nghệ kết nối khác.
* Có thể viết các thư viện trên **C/C++** để nhúng vào **Python** và ngược lại.

#### **Python là ngôn ngữ có khả năng chạy trên nhiều nền tảng.**

* **Python** có cho mọi hệ điều hành: **Windows**, **Linux/Unix**, **OS/2**, **Mac**, **Amiga**, và những hệ điều hành khác. Thậm chí có cả những phiên bản chạy trên **.NET**, máy ảo **Java**, và điện thoại di động (Nokia Series 60). Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.

#### **Python rất đơn giản và dễ học**

* **Python** có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, và cả các thư viện mã nguồn mở được chia sẻ trên mạng.

#### **Python là ngôn ngữ mã nguồn mở**

* Cài đặt **Python** dùng giấy phép nguồn mở nên được sử dụng và phân tối tự do, ngay cả trong việc thương mại. Giấy phép **Python** được quản lý bởi **Python Software Foundation**.

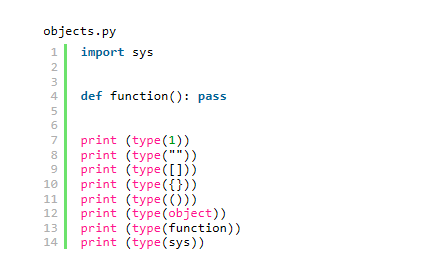
## Python hướng đối tượng:

**Lập trình hướng đối tượng** (gọi tắt là **OOP**, từ chữ Anh ngữ *object-oriented programming*), là kĩ thuật lập trình hỗ trợ công nghệ đối tượng. OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, nhiều người còn cho rằng OOP dễ tiếp thu hơn cho những người mới học về lập trình hơn là các phương pháp trước đó.

Trong OOP có các khái niệm cơ bản sau đây:

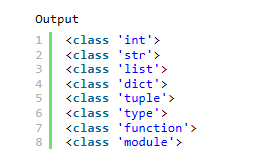
* Trừu tượng (**Abstraction**)
* Đa hình (**Polymorphism**)
* Đóng gói (**Encapsulation**)
* Kế thừa (**Inheritance**)
  1. **Khái niệm đối tượng**

Đối tượng là các thực thể của một lớp nào đó. Lớp là khuôn mẫu của các đối tượng. Ví dụ như list, tuple, dictionary, string, int… là các lớp. Khi chúng ta khai báo biến thuộc các lớp này thì chúng là các đối tượng. Tất cả mọi thứ trong Python đều là đối tượng.



Hinh : 1 Ví vụ khái niệm hướng đối tượng

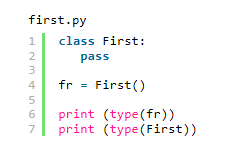
Trong ví dụ trên nhờ vào hàm type() mà chúng ta biết được thực chất tất cả các kiểu dữ liệu và các module mà chúng ta đã học thực chất đều là các đối tượng.



Hinh : 2 Ví vụ khái niệm hướng đối tượng 1.1

* 1. **Từ khóa class**

Trong ví dụ trên, các lớp mà chúng ta kiểm tra bằng hàm type là các lớp có sẵn trong Python. Khi chúng ta muốn định nghĩa một lớp cho riêng mình thì chúng ta dùng từ khóa class. Một lớp có thể hiểu là một bản thiết kế để tạo ra một thực thể (mà chúng ta gọi là đối tượng). Một đối tượng là một thực thể được xây dựng từ một bản thiết kế đó (mà chúng ta gọi là lớp). Ví dụ chúng ta định nghĩa lớp Dog, thì chúng ta có các đối tượng là KiKi, MeoMeo,..



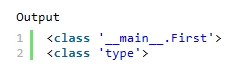
Hinh : 3 ví vụ về từ khóa lớp

Trong đoạn code trên chúng ta định nghĩa một lớp. Phần thân lớp tạm thời được để trống.



Hinh : 4 ví vụ về từ khóa lớp 1.1

Ở dòng trên chúng ta xây dựng nên thực thể fr từ bản thiết kế First. Hay nói cách khác là chúng ta đã tạo ra đối tượng fr từ lớp First.

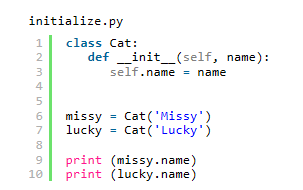


Hinh : 5 ví vụ về từ khóa lớp 1.2

Bên trong một lớp, chúng ta có thể định nghĩa các biến và hàm, biến và hàm bên trong một lớp được gọi là thuộc tính và phương thức. Chẳng hạn chúng ta định nghĩa lớp Employee (nhân viên), bên trong lớp này sẽ có thuộc tính Salary (lương) và phương thức calculateSalary (tính tiền lương).

* 1. **Thuộc tính**

Thuộc tính là biến nằm trong một lớp. Thuộc tính mô tả các đặc tính của một đối tượng. Trong Python có một phương thức đặc biệt gọi là  \_\_init\_\_() dùng để khởi tạo giá trị cho các thuộc tính của một đối tượng.



Hinh : 6 ví vụ về thuộc tính

Trong đoạn code trên, chúng ta định nghĩa lớp Cat. Bên trong lớp này, chúng ta định nghĩa phương thức \_\_init\_\_(). Phương thức \_\_init\_\_() là phương thức khởi tạo của tất cả các lớp, mỗi khi tạo một đối tượng phương thức này sẽ tự động được gọi.



Hinh : 7 ví vụ về thuộc tính 1.1

Bất cứ phương thức nào của Python cũng đều phải có tham số đầu tiên là self rồi mới đến các tham số khác. self thực ra chỉ là biến đối tượng đã gọi phương thức này mà thôi. Chẳng hạn trong ví dụ trên khi chúng ta gọi missy = Cat("Missy") thì self chính là missy.



Hinh : 8 ví vụ về thuộc tính 1.2

Ở dòng trên chúng ta khởi tạo thuộc tính name và gán giá trị cho nó.



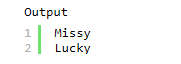
Hinh : ví vụ về thuộc tính 1.3

Trong hai dòng trên chúng ta dùng hàm Cat() để tạo ra hai đối tượng lớp thuộc lớp Cat. Chúng ta không trực tiếp đưa tham số self vào mà chỉ đưa các tham số thứ 2 trở đi thôi, self được tự động đưa vào bởi trình thông dịch. Trong ví dụ trên “Missy” và “Lucky” sẽ được gán cho thuộc tính name trong lớp Cat.



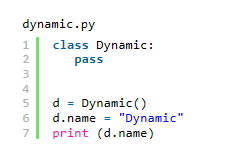
Hinh : 10 ví vụ về thuộc tính 1.4

Trong hai dòng code trên chúng ta in ra giá trị của thuộc tính name trong hai đối tượng Cat. Để truy xuất đến thuộc tính của một đối tượng thì chúng ta ghi tên đối tượng, dấu chấm và tên thuộc tính.



Hinh : 11 ví dụ về thuộc tính 1.5

Ngoài ra bạn có thể gán giá trị cho các thuộc tính ở bất cứ đâu sau phần định nghĩa lớp chứ không chỉ riêng bên trong phương thức khởi tạo.



Hinh : 12 ví vụ về thuộc tính 1.6

Trong ví dụ trên chúng ta định nghĩa lớp Dynamic.



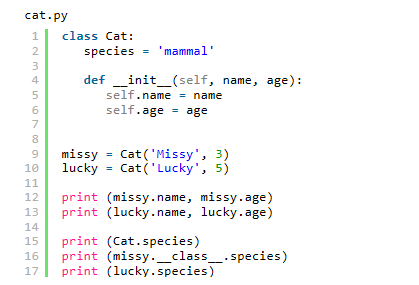
Hinh : 13 ví vụ về thuộc tính 1.7

Ở dòng này chúng ta khai báo và gán giá trị cho thuộc tính name.



Hinh : 14 ví vụ về thuộc tính 1.8

Một đặc điểm nữa trong Python bạn có thể định nghĩa các thuộc tính chung cho mọi đối tượng.



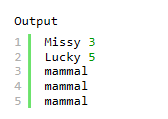
Hinh : 15 ví vụ về thuộc tính 1.9

Trong ví dụ trên, chúng ta tạo ra hai đối tượng Cat với thuộc tính name và age, hai thuộc tính này được khai báo bên trong phương thức \_\_init\_\_() và được gọi là thuộc tính đối tượng. Ngoài ra hai đối tượng này còn có một thuộc tính chung là thuộc tính species, thuộc tính này được khai báo ở ngoài phương thức \_\_init\_\_() và được gọi là thuộc tính lớp. Thuộc tính lớp được chia sẻ chung cho mọi đối tượng của lớp đó, trong khi thuộc tính đối tượng chỉ dành riêng cho đối tượng. Nghĩa là khi bạn thay đổi giá trị của thuộc tính chia sẻ trong một đối tượng thì tất cả các thuộc tính chia sẻ đó trong các đối tượng khác cũng thay đổi theo.



Hinh : 16 ví vụ về thuộc tính 2.0

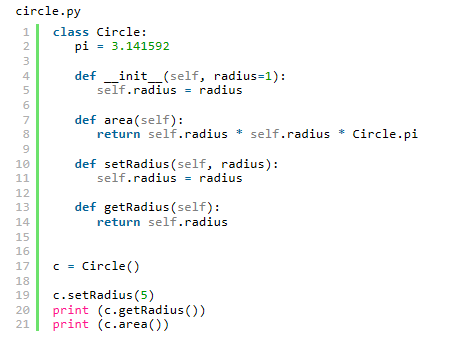
Có hai cách để truy xuất thuộc tính lớp, thứ nhất là thông qua tên lớp, cách thứ hai là thông qua một thuộc tính đặc biệt nữa là thuộc tính \_\_class\_\_.



Hinh : ví vụ về thuộc tính 2.1

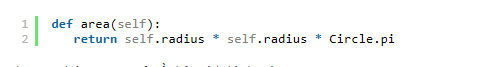
* 1. Phương thức

Phương thức chẳng qua là các hàm, chỉ khác hàm bình thường ở chỗ chúng được định nghĩa bên trong một lớp. Các phương thức được sử dụng để thực hiện các công việc cụ thể. Phương thức là một thành phần quan trọng trong khái niệm Encapsulation (đóng gói). Chẳng hạn như bạn có một lớp tên là Network, lớp này có phương thức connect() dùng để kết nối đến các máy tính trong mạng LAN, thì bạn chỉ quan tâm rằng phương thức này làm công việc kết nối mạng chứ không quan tâm đến việc nó kết nối như thế nào.



Hinh : 18 ví vụ về phuong thức

Trong ví dụ trên, chúng ta định nghĩa lớp Circle (hình tròn) và 3 phương thức mới.



Hinh : 19 ví vụ về phuong thức 1.1

Phương thức area() trả về diện tích hình tròn.



Hinh : 20 ví vụ về phuong thức 1.2

Phương thức setRadius() thiết lập giá trị cho thuộc tính radius (bán kính).



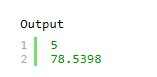
Hinh : 21 ví vụ về phuong thức 1.3

Phương thức getRadius() trả về lấy giá trị bán kính.



Hinh : 22 ví vụ về phuong thức 1.4

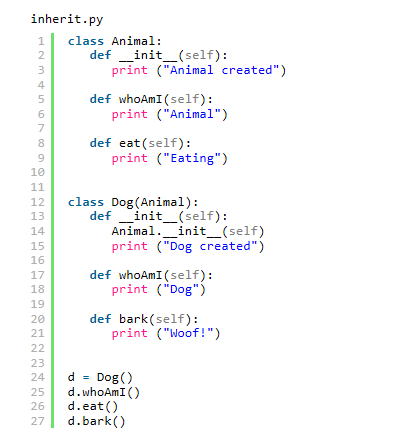
Chúng ta gọi các phương thức thông qua biến đối tượng.



Hinh : 23 ví vụ về phuong thức 1.5

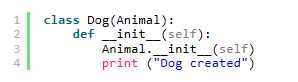
* 1. Kế thừa

Kế thừa là định nghĩa một lớp dựa trên một lớp đã được định nghĩa trước đó. Lớp kế thừa từ lớp khác được gọi là lớp dẫn xuất, lớp được các lớp khác kế thừa mình thì gọi là lớp cơ sở. Kế thừa trong lập trình hướng đối tượng cho phép chúng ta sử dụng lại mã nguồn và giảm độ phức tạp của chương trình. Lớp dẫn xuất có thể kế thừa hoặc mở rộng các tính năng của lớp cơ sở.



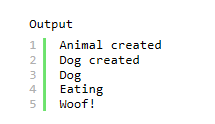
Hinh : 24 ví dụ kế thừa

Trong ví dụ trên chúng ta định nghĩa hai lớp là là Animal và lớp Dog. Lớp Dog kế thừa từ lớp Animal, thừa hưởng một số phương thức của lớp Animal và có các phương thức của riêng nó. Ở trên lớp Dog kế thừa phương thức eat(), kế thừa và thay đổi phương thức whoAmI(), ngoài ra lớp Dog còn có phương thức của riêng nó là phương thức bark().



Hinh : 25 ví vụ về kế thức 1.1

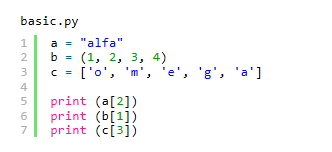
Để kế thừa một lớp thì chúng ta đặt tên lớp đó bên trong cặp dấu ngoặc tròn () ngay phía sau phần định nghĩa tên lớp. Nếu bên trong lớp cơ sở đã định nghĩa phương thức \_\_init\_\_(), chúng ta phải gọi lại phương thức \_\_init\_\_() từ lớp cơ sở.



Hinh : 26 ví vụ về đa thức 1..2

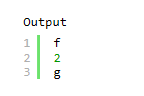
* 1. **Đa hình**

Đa hình là bạn có thể sử dụng các toán tử hay các hàm trên nhiều kiểu dữ liệu khác nhau.



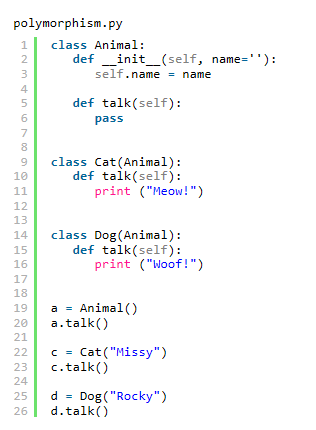
Hinh : 27 ví vụ về đa thức

Bản thân các hàm có sẵn trong Python cũng có tính chất đa hình. Chẳng hạn như hàm print() mà chúng ta hay dùng, mỗi lần gọi hàm này bạn có thể đưa vào hầu hết các kiểu dữ liệu khác nhau.



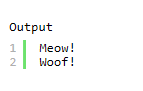
Hinh : 28 ví vụ về đa thức 1.1

Trong hướng đối tượng thì tính đa hình càng được phát huy tác dụng.



Hinh : 29 ví vụ về đa thức 1.2

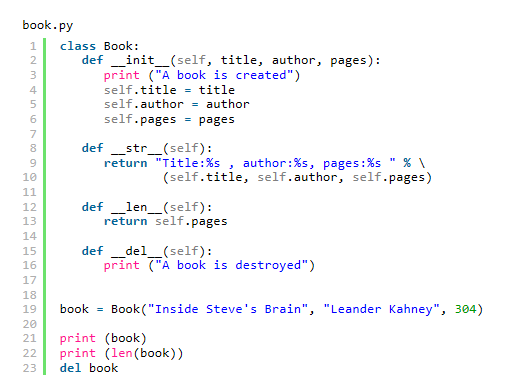
Trong ví dụ trên, chúng ta định nghĩa hai lớp chó (Dog) và mèo (Cat) kế thừa từ lớp Animal. Do đó cả hai lớp này đều kế thừa phương thức talk() của lớp Animal, nhưng mỗi lớp lại in ra hai dòng text khác nhau.



Hinh : 30 ví vụ về đa thức 1.3

* 1. Các phương thức đặc biệt

Tất cả các lớp dù là có sẵn hay do chúng ta định nghĩa đều kế thừa từ một lớp gốc trong Python có tên là object. Lớp này có sẵn một số phương thức và đương nhiên là các lớp do chúng ta định nghĩa đều kế thừa các phương thức này, ví dụ như phương thức \_\_init\_\_()… Trong Python khi chúng ta gọi đến các hàm hay toán tử được xây dựng sẵn như print(), del… chúng sẽ gọi đến các phương thức gốc của lớp object. Chính vì các lớp do chúng ta định nghĩa đều được kế thừa từ lớp object nên chúng ta cũng có thể dùng các hàm hay toán tử có sẵn trong Python với các lớp của chúng ta. Trong các ngôn ngữ như C++ thì tính chất này được gọi là quá tải toán tử (operator overloading).



*Hinh : 31 Các phương thức đặc biệt*

Trong ví dụ trên, chúng ta định nghĩa lớp Book. Bên trong lớp này chúng ta định nghĩa 4 phương thức được kế thừa từ lớp object là \_\_init\_\_(),\_\_str\_\_(), \_\_len\_\_() và \_\_del\_\_().



Hinh : 32 Các phương thức đặc biệt 1.1

Khi sử dụng hàm print(), hàm này sẽ gọi đến phương thức \_\_str\_\_(), do đó khi chúng ta kế thừa lại phương thức này, hàm print() sẽ gọi đến phương thức \_\_str\_\_() trong lớp Book của chúng ta.



Hinh : 33 Các phương thức đặc biệt 1.2

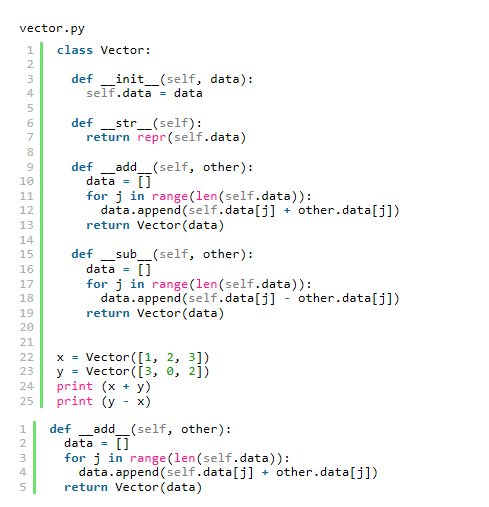
Tương tự, hàm len() gọi đến phương thức \_\_len\_\_()



Hinh : 34 các phương thức đặc biệt 1.3

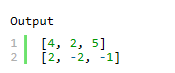
Từ khóa del gọi đến phương thức \_\_del\_\_() và trên thực tế là phương thức này làm công việc hủy bỏ một đối tượng ra khỏi bộ nhớ. Nhưng ở đây chúng ta kế thừa phương thức này chỉ làm công việc là in ra một đoạn text.

Trong ví dụ dưới đây, chúng ta sẽ kế thừa các toán tử cộng và trừ.



Hinh : 35 các phương thức đặc biệt 1.4

Chúng ta định nghĩa lớp Vector, lớp vector này kế thừa hai phương thức \_\_add\_\_() và \_\_sub\_\_(). phương thức \_\_add\_\_() sẽ được gọi khi sử dụng toán tử + và phương thức \_\_sub\_\_() sẽ được gọi khi sử dụng toán tử -.



Hinh : 36 các phương thức đặc biệt 1.5

1. **Lý thuyết game**
2. Mở đầu

Khi nghiên cứu một số bài toán thực tế trong các lĩnh vực kinh tế, xã hội hay trong chiến tranh người ta thường gặp những tình huống trong đó hai bên có quyền đối lập nhau. Mỗi bên đều tìm cách hành động sao cho mình có lợi nhất. Những tình huống như thế được gọi là các tình huống xung đột. Có rất nhiều tình huống xung đột như: xung đột vũ trang giữa hai quốc gia, giữa hai phe phái, cạnh tranh kinh tế giữa hai cuộc mặc cả giữa bên mua và bên bán.

Việc nghiên cứu, phân tích những tình huống như thế nhằm nắm được bản chất của nó và để ra những giải pháp tốt nhất đối với mỗi bên tham gia đã dẫn đến sự ra đời của một môn toán học mới Lý thuyết trò chơi. Lý thuyết trò chơi là lý thuyết toán về các tình huống xung đột.

Để nắm bắt được bản chất của vấn đề cần nghiên cứu, một cách quen thuộc là mô hình dạng toán học. người ta loại bỏ những nhân tố thứ yếu, các trường hợp ngoại lệ, chỉ giữ lại những điểm chính, những quy tắc, điều kiện quan trọng nhất chi phối diễn biến của tình huống xung đột đang nghiên cứu. Những mô hình nhận được như vậy sẽ rất giống với mô hình trò chơi mà ta thường gặp hằng ngày. Vì lẽ đó mà ta gọi là các trò chơi và sử dụng các thuật ngữ của trò chơi thường nhật.

1. Thuật toán minimax là gì?

Minimax là giải thuật 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:

* **Min** thể hiện cho giá trị tối thiểu mà máy tính có thể đi trong nước hoặc vài nước tiếp theo. Người chơi thường chọn nước đi có giá trị **Min**
* **Max**thể hiện cho giá trị lớn nhất mà máy tính có thể đi trong nước hoặc một vài nước tiếp theo. Máy sẽ thường chọn nước đi được đánh giá là **Max**.

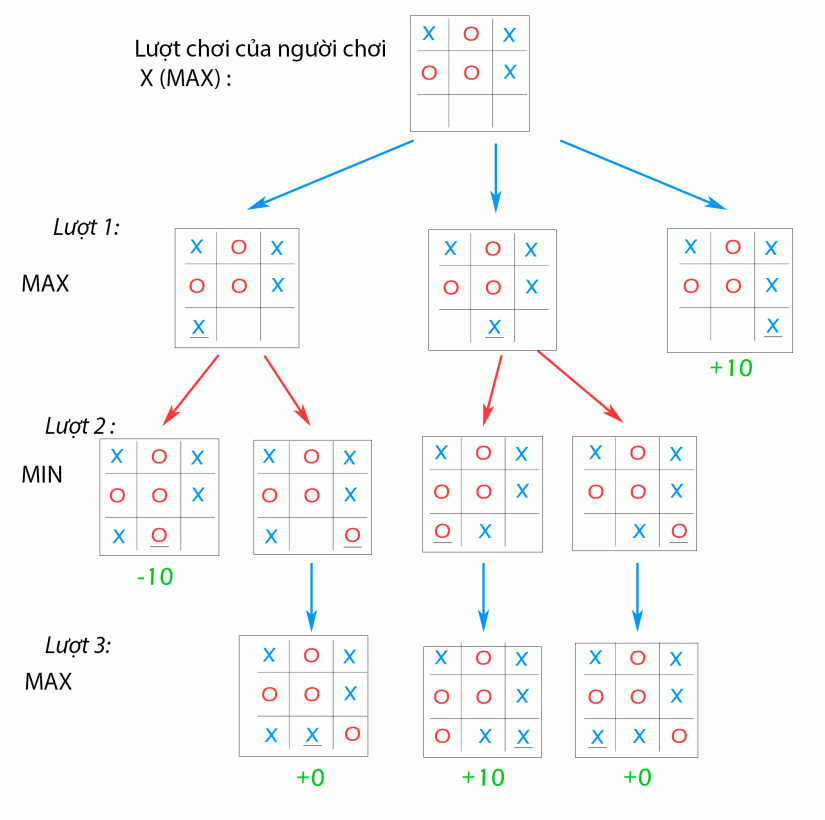
Thực tế không đơn giản như vậy vì khi máy tính mong muốn chọn được giá trị **max** nhất (có lợi nhất cho mình) nhưng đó cũng là thiệt hại nhất cho đối thủ vì thế đối thủ sẽ chọn nước đi sao cho giảm độ thiệt hại nhất có thể tức là sẽ luôn chọn **Min** giá trị mà máy tính sẽ chọn.

1. Tại sao phải cần dùng minimax ?

Như các bạn đã biết thì có rất nhiều thuật toán tìm kiếm để làm AI trong game như A, Heuristic... Mỗi thuật toán thì sẽ phù hợp với từng loại game cho nó. Những game đối kháng trong đối người chơi luân phiên đánh như cờ vua, cờ tường, caro... Khi chơi bạn có thể khai triển hết không gian trạng thái nhưng khó khăn chủ yếu là bạn phải tính toán được phản ứng và nước đi của đối thủ mình như thế nào? Cách xử lý đơn giản là bạn giả sử đối thủ của bạn cũng sử dụng kiến thức về không gian trạng thái giống bạn. Giải thuật Minimax áp dụng giả thuyết này để tìm kiếm không gian trạng thái của trò chơi. Trường hợp này thuật toán minimax sẽ đáp ứng những gì mình cần.

1. Giải pháp:

Máy phải luôn luôn mặc định là người chơi sẽ chọn giá trị tối thiểu (**MIN**), máy cần chọn trong các nước đi của mình sao cho ở trạng tháicuối, máy luôn có được giá trị tốt nhất (**MAX**) trong các trường hợp tính được:(-10 là thua, + 10 là trắng, +0 là hòa)



*Hình 37: mô phỏng game cờ ca rô 3x3*

1. **Xây dựng trò chơi**
2. Tạo bàn cho bàn cờ caro

class Board(object): ## Lớp Tạo Bàn cờ

begin

Phương thức\_\_init\_\_(self,

Cho r = 600, c = 600)

self.hàng =15, self.cột =15

lề = 13

ô =41

self.width,self.height =r,c

gọi self.initBoard()

self.now = 'white'

self.isWin = False

self.pieceCount = 0

End

Bengin

Hàm initBoard(self):

Cho self.board = []

self.count = []

For hàng in range (self.hàng): #15

Thêm vào mảng self.banrd.append([-1] \* self.cột) #[-1]\*15

for hàng in range (self.hàng): #15

Thêm vào mảng self.count.append([0] \* self.cột) #[-1]\*15

End

Bengin

Hàm drawCrossLines(self,canvas):# vẽ đường ngang

For rowLine in range(self.hàng):#15

canvas. create\_line(self.lề,self.lề+rowLine\*self.ô,

self.cao-self.lề,self.lề+rowLine\*self.ô)#(13,13+row.Line\*13,15-13,13+rowLine\*13)

End

Bengin

hàm drawVerticalLines(self,canvas):# vẽ các đường thẳng đứng

for (colLine) in range(self.cột):#15

canvas. create\_line (self.lề+colLine\*self.ô,self.lề,

self.lề+colLine\*self.ô,self.cao-self.lề) )#(13,13+row.Line\*13,15-13,13+rowLine\*13)

End

Bengin

hàm drawSmallDots(self,canvas):# 5 dấu chấm đen

Mảng l = []

Input ‘ = 2.5

Thêm vào l.append((self.lề+3\*self.ô,self.lề+3\*self.ô)) #0

Thêm vào l.append((self.lề+11\*self.ô,self.lề+3\*self.ô)) #1

Thêm vào l.append((self.lề+3\*self.ô,self.lề+11\*self.ô)) #2

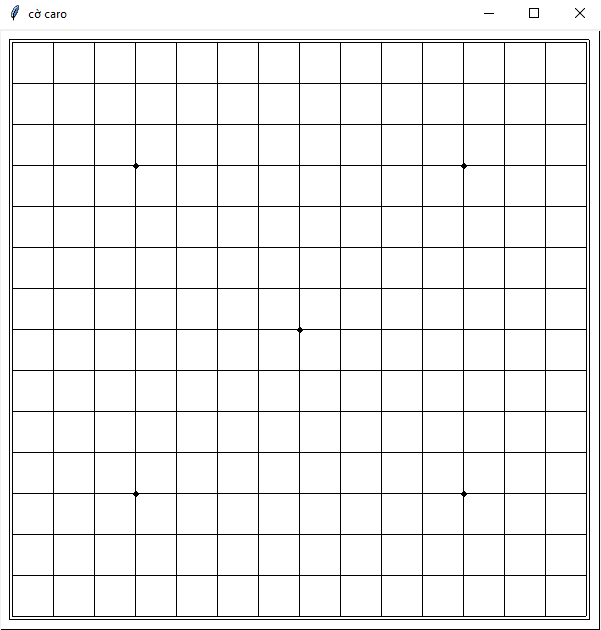
Thêm vào l.append((self.lề+11\*self.ô,self.lề+11\*self.ô)) #3

Thêm vào l.append((self.lề+7\*self.ô,self.lề+7\*self.ô)) #4

for c in l:

canvas.create\_oval(c[0]-radius,c[1]-radius,c[0]+radius,c[1]+radius,fill = 'black')

End



Hinh : 37 bàn cờ caro

1. File vẽ

Bengin

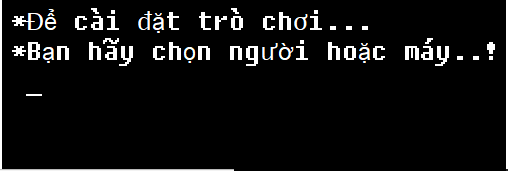
Hàm drawChooseOppo(self,canvas):## Chon che do tro choi

input self.height = 180

display self.text = "\*Để cài đặt trò chơi...\n\*Bạn hãy chọn người hoặc máy..!\n "

self.drawBox(canvas)

End



Hinh : 38 Chon che do tro choi

Bengin

hàm drawChooseDiff(self,canvas):## huong dan choi voi may

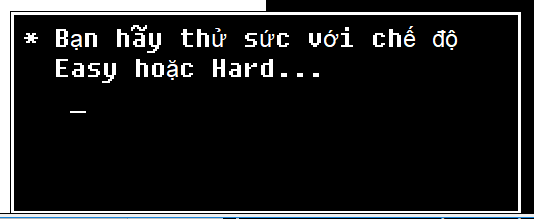
input self.height = 200

display self.text = "\* Bạn hãy thử sức với chế độ \n Easy hoặc Hard...\n "

self.drawBox(canvas)

self.drawText(canvas)

End



Hinh : 39 huong dan choi voi may

Bengin

hàm drawChooseColor(self,canvas):## huong dan tro choi

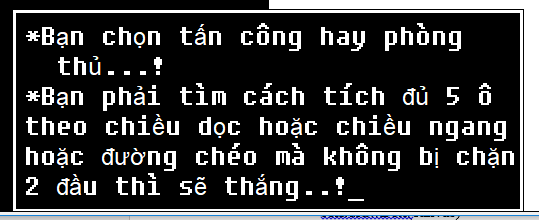
input self.height = 200

display self.text = "\*Bạn chọn tấn công hay phòng\n thủ...!\n\*Bạn phải tìm cách tích đủ 5 ô \ntheo chiều dọc hoặc chiều ngang \nhoặc đường chéo mà không bị chặn \n2 đầu thì sẽ thắng..!"

self.drawBox(canvas)

self.drawText(canvas)

end



Hinh : 40 huong dan tro choi

1. Tạo màng hình bắt đầu trò chơi

Bengin

Hàm drawChoicePage(self):# vẽ chọn trang

input self.canvas.create\_rectangle(0,0,self.width/2,self.height,fill = "white")

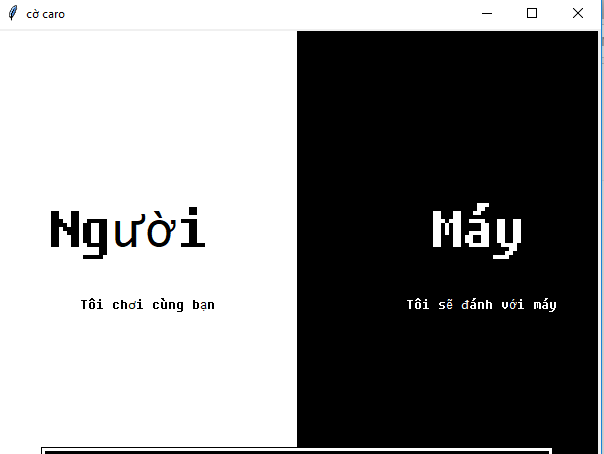
input self.canvas.create\_rectangle(self.width/2,0,self.width,self.height,fill = "black")

input self.canvas.create\_text(50,self.height/3,text = "Người", fill = "black",anchor = W,font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/4,self.height/2.5+35,text = """Tôi chơi cùng bạn""", fill = "black",font = ("Fixedsys", 8))

inpun self.canvas.create\_text(self.width/5\*4,self.height/3,text = "Máy", fill = "white",font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/5\*4,self.height/2.5+35,text = """ Tôi sẽ đánh với máy""", fill = "white",font = ("Fixedsys", 8))



Hinh : 41 vẽ chọn trang

Begin

Hàm drawStartScreening(self):#vẽ màng hình bắt đầu trò chơi

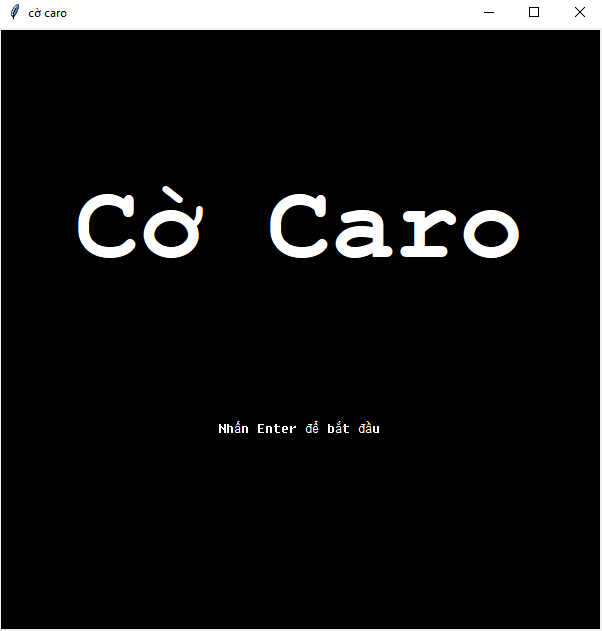
input self.canvas.create\_rectangle(0,0,self.width,self.height,fill = "black")

display self.canvas.create\_text(self.width/2,self.height/3,text = 'Cờ Caro', font = ('Courier',self.fontSize,'bold'),fill = "white")

if self.textcolor thì fill = "white" ngược lại fill = "black"

display self.canvas.create\_text(self.width/2,self.height\*2/3,text = 'Nhấn Enter để bắt đầu', font = ('Fixedsys',15), fill = fill)

end



Hinh : 42 vẽ màng hình bắt đầu trò chơi

Bengin

hàm drawChoiceColorPage(self):# vẽ tấn công hay phòng thủ

ipunt self.canvas.create\_rectangle(0,0,self.width/2,self.height,fill = "black")

input self.canvas.create\_rectangle(self.width/2,0,self.width,self.height,fill = "white")

display self.canvas.create\_text(50,self.height/3,text = "White", fill = "white",anchor = W,font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/4,self.height/2.5+35,text = """Tôi rất giỏi về phòng thủ,\nmáy đi trước""", fill = "white",font = ("Fixedsys", 8))

display self.canvas.create\_text(self.width/5\*4,self.height/3,text = "Black ", fill = "Black",font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/5\*4,self.height/2.5+35,text = """Tấn công tui rất giỏi ,\ntui đi trước""", fill = "black",font = ("Fixedsys", 8))

end 

Hinh : 43 vẽ tấn công hay phòng thủ

Begin

hàm drawDiffcultPage(self):#vẽ lượt chọn đánh với chế độ dễ và khó

input self.canvas.create\_rectangle(0,0,self.width/2,self.height,fill = "white")

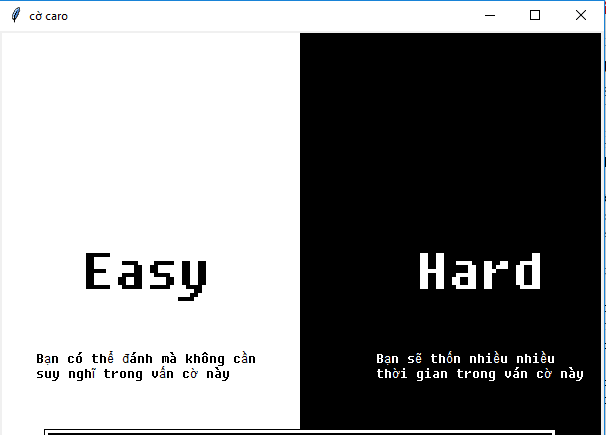
input self.canvas.create\_rectangle(self.width/2,0,self.width,self.height,fill = "black")

display self.canvas.create\_text(50,self.height/2.5,text = " Easy", fill = "black",anchor = W,font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/4,self.height/2+35,text = """Bạn có thể đánh mà không cần \nsuy nghĩ trong vấn cờ này""", fill = "black",font = ("Fixedsys", 8))

display self.canvas.create\_text(self.width/5\*4,self.height/2.5,text = "Hard", fill = "white",font = ("Fixedsys", 40))

display self.canvas.create\_text(self.width/5\*4,self.height/2+35,text = """Bạn sẽ thốn nhiều nhiều \nthời gian trong ván cờ này""", fill = "white",font = ("Fixedsys", 8))

end 

Hinh : 44 vẽ lượt chọn đánh với chế độ dễ và khó

1. game

from tkinter import \*

class Window(object):

def \_\_init\_\_(self,width,height):

self.root = Tk()

self.root.title("Co ca ro")

def quit(self):

self.root.destroy

1. Giải thuật

Bengin

Hàm minimaxWrapper(self):Giải thuật minimax dễ

input playerSaver = self.currentPlayer #nguoi choi hien tai

if self.rowcolSeq == []:

input self.inputInfo(7, 7)

return

if len(self.rowcolSeq)== 1:

input self.inputInfo(self.rowcolSeq[0][0], self.rowcolSeq[0][1]+1)

return

if self.currentPlayer[1] == 'black':

Input depth = 0

else

Input depth = 1

listOfMoves = self.getPossibleMoves()

self.decisionList = []

input row,col = self.minimax(listOfMoves,depth)

self.currentPlayer = playerSaver

self.inputInfo(row, col)

end

Bengin

Hàm minimax(self,l,depth): tìm đường minimax

input scores = {}

self.scores = []

tempL = {}

for moves in l:

input self.board.board[moves[0]][moves[1]] = self.currentPlayer[1]

if self.board.checkBoard() == True thì

self.board.isWin = False

self.board.board[moves[0]][moves[1]] = -1

return (moves[0],moves[1])

self.alternateTurn()

lnext = self.getPossibleMoves()

self.scores = []

for m in lnext:

input self.board.board[m[0]][m[1]] = self.currentPlayer[1]

self.board.now = self.currentPlayer[1]

self.scores.append((self.board.getScore()))

self.board.board[m[0]][m[1]] = -1

if self.currentPlayer[1] == 'black' thì

score = max(self.scores)

else

score = min(self.scores)

scores[moves] = score

self.alternateTurn()#luot danh

self.board.board[moves[0]][moves[1]] = -1

inverse = [(value, key) for key, value in scores.items()]

if self.currentPlayer[1] == 'white' thì

return min(inverse)[1]

else

return max(inverse)[1]

end

Bengin

Hàm computerWrapper(self): # minimax khó

input self.scores = []

if self.rowcolSeq == []:

input self.inputInfo(7, 7)

return

if len(self.rowcolSeq) == 1:

input self.inputInfo(self.rowcolSeq[0][0], self.rowcolSeq[0][1]+1)

return

for row in range(len (self.board.board)):

for col in range (len(self.board.board[0])):

if self.board.board[row][col] ==-1: thì

input self.board.board[row][col] = self.currentPlayer[1]

self.board.now = self.currentPlayer[1]

self.scores.append((self.board.getScore(),row,col))

self.board.board[row][col] = -1

if self.currentPlayer[1]== 'black':

input maxValue = self.scores[0][0]

res = []

for tup in self.scores:

if tup[0] > maxValue thì

input maxValue = tup[0]

input res = [(tup[1],tup[2])]

elseif tup[0] == maxValue thì

input res+=[(tup[1],tup[2])]

row,col = random.choice(res)

self.inputInfo(row, col)

return

else

input maxValue = self.scores[0][0]

res = []

for tup in self.scores:

if tup[0] > maxValue thì

input maxValue = tup[0]

res = [(tup[1],tup[2])]

esleif tup[0] == maxValue thì

res+=[(tup[1],tup[2])]

row,col = random.choice(res)

self.inputInfo(row, col)

dừng return

end

1. File caro

Bengin

hàm keyPressed(event, data):#bấm r để reset

if event.keysym == "r" thì

input init(data)

if not data.s.isOver thì

data.s.keyPressedWrapper(event)

redrawAll(data.canvas, data)

End

1. Kết luận và hướng phát triển

Xây dựng trò chơi giải trí là một trong những nhu cầu xuất phát từ thực tế, nhất là trong thời đại ngày nay, khi rất nhiều người làm việc bên máy tính suốt ngày. Do đó, con người cũng có thể thư giãn ngay trên máy tính của mình sau những giờ học và làm việc căng thẳng.

1. Kết quả đạt được:

* Phân tích cách chơi và thiết kế giao diện tổng quan.
* Tạo giao diện thân thiện giữa người chơi với chương trình.
* Xây dựng được trò chơi không đòi hỏi cao về cấu hình máy.
* Tìm hiểu được thuật toán Minimax.
* Xây dựng được thuật toán cho trò chơi.
* Trò chơi với máy với trí tuệ nhân tạo với hai cấp độ phù hợp với người chơi.

1. Hạn chế:

* Chưa làm hàm tính điểm số lần thắng thua.
* Chưa làm lưu game chơi.
* Lỗi khi đang vào trò chơi mà thoát game không tắt được nhạc.

1. Hướng phát triển:

* Từ kinh nghiệm với Python em tin chất những chương trình sau này sẽ hơn tốt do trong quá trình tìm hiểu được hiểu kiến thức mới cho đồ án 2 sau này.

1. Tài liệu kham khảo:
2. <https://www.scribd.com/doc/174108305/Thuat-toan-minimax-pdf>.
3. Slide bài giảng trí tuệ nhân tạo: https://drive.google.com/file/d/1WBP7dWGnikyK4oCoGZs-D1UzOX8H-S4z/view?usp=sharing
4. Youtube dạy lập trình hướng đối tượng: https://www.youtube.com/watch?v=yvKlkpG06Iw