BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC SAO ĐỎ**

\*\*\*\*\*

**BÁO CÁO BÀI TẬP LỚN**

**TÊN ĐỀ TÀI: Game bắn xe tăng**

**Học phần: Lập trình Python**

**Sinh viên thực hiện: Phan Văn Huân- Phạm Văn Hoàn**

**Lớp: DK11 – CNTT1**

**Khoa: Công nghệ thông tin**

**Năm 2023**

**MỤC LỤC**

[**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT** 2](#_Toc127796282)

[1.1. Python 2](#_Toc127796283)

[1.2. Thư viện Pygame 2](#_Toc127796284)

[**CHƯƠNG 2. THIẾT KẾ CHƯƠNG TRÌNH** 3](#_Toc127796285)

[2.1. Quy trình 3](#_Toc127796286)

[2.2. Giao diện 3](#_Toc127796287)

[2.3. Thiết kế chương trình 3](#_Toc127796288)

[2.3. Kết luận 3](#_Toc127796289)

[2.4. Tài liệu tham khảo 3](#_Toc127796290)

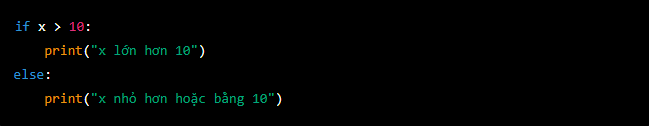
# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

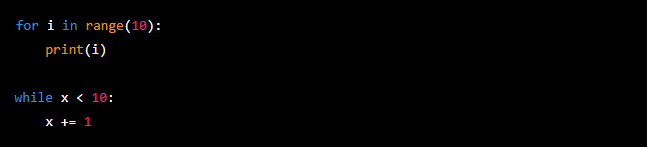
## Python

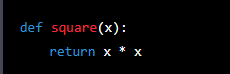
Python là một ngôn ngữ lập trình cao cấp và được sử dụng phổ biến cho các ứng dụng máy tính, trò chơi, web và khoa học dữ liệu. Với cú pháp đơn giản và dễ hiểu, Python rất phù hợp cho các người mới bắt đầu học lập trình.

1. Cú pháp cơ bản: Python có cú pháp đơn giản và dễ hiểu, thường sử dụng cấu trúc lệnh điều kiện và vòng lặp. Điều này giúp cho việc code của bạn có thể đọc được và dễ dàng bảo trì hơn.

* Lệnh điều kiện: if/else



* Vòng lặp: for/while
* Hàm: def



1. Kiểu dữ liệu: Python có nhiều kiểu dữ liệu khác nhau, bao gồm: số, chuỗi, danh sách, tuple, từ điển và boolean.

* Số: int, float



* Chuỗi: str



* Danh sách: list



* Tuple: tuple



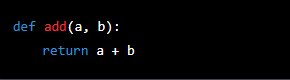
* Từ điển: dict



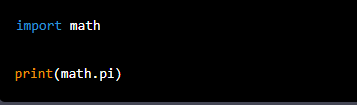
* Boolean: bool



1. Hàm: Python hỗ trợ việc định nghĩa và sử dụng hàm một cách dễ dàng. Hàm trong Python có thể có các tham số đầu vào và giá trị trả về.

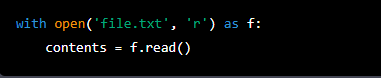


1. Module: Python có một thư viện lớn gồm các module, cung cấp nhiều tính năng cho lập trình viên. Các module có thể được import và sử dụng trong chương trình của bạn.

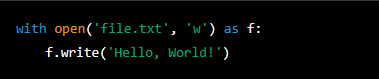


1. Thao tác trên file: Python cung cấp các phương thức cho phép đọc và ghi file.

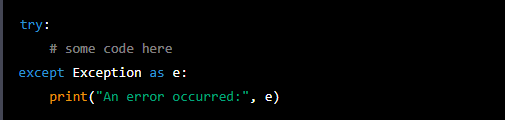
* Đọc file:



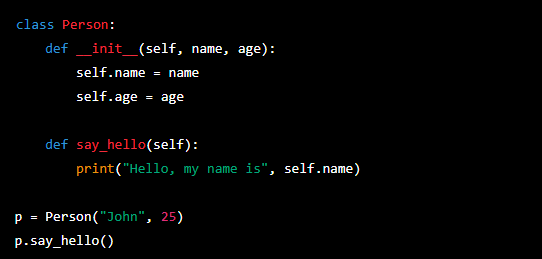
* Ghi file:



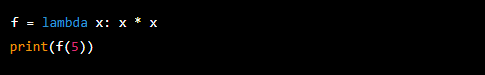
1. Xử lý ngoại lệ: Python cho phép xử lý ngoại lệ bằng cách sử dụng các khối try/except, giúp ngăn chặn các lỗi xảy ra trong quá trình thực thi chương trình.



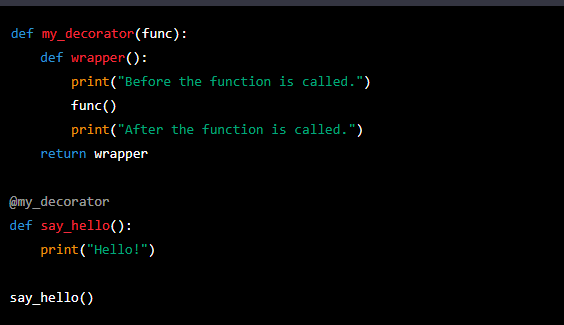
1. Lập trình hướng đối tượng: Python hỗ trợ lập trình hướng đối tượng, cho phép định nghĩa các lớp và đối tượng. Điều này giúp cho việc code của bạn trở nên dễ dàng bảo trì hơn.



1. Lambda: Lambda là một loại hàm ẩn danh, giúp cho việc code của bạn trở nên ngắn gọn hơn.



1. Decorator: Decorator là một cách để thay đổi hoặc mở rộng chức năng của một hàm.



Trên đây là những kiến thức cơ bản về ngôn ngữ Python. Python có nhiều tính năng và thư viện hỗ trợ mạnh mẽ, giúp cho lập trình viên có thể xây dựng các ứng dụng phức tạp. Nếu bạn đang muốn học lập trình Python, đây là một ngôn ngữ rất phù hợp để bắt đầu.

1. Thư viện chuẩn của Python: Python đi kèm với một số thư viện chuẩn cung cấp các công cụ hữu ích như xử lý chuỗi, đọc và ghi tệp tin, xử lý thời gian, thao tác với đường dẫn, v.v. Một số thư viện phổ biến nhất của Python là:

* os: hỗ trợ thao tác với hệ thống tệp và thư mục.
* sys: cung cấp thông tin về phiên bản Python đang sử dụng và cấu hình hệ thống.
* datetime: hỗ trợ xử lý thời gian và ngày tháng.
* re: cung cấp các công cụ để xử lý chuỗi bằng các biểu thức chính quy.
* math: cung cấp các hàm toán học phổ biến như lấy căn bậc hai, tính số mũ, v.v.

1. Thư viện bên thứ ba của Python: Ngoài các thư viện chuẩn, Python còn có một số thư viện bên thứ ba được phát triển bởi cộng đồng lập trình viên. Các thư viện này cung cấp các tính năng đa dạng và hữu ích như đồ họa, xử lý dữ liệu, máy học, v.v. Một số thư viện bên thứ ba phổ biến nhất của Python là:

* NumPy: hỗ trợ các phép toán số học và thống kê trên mảng nhiều chiều.
* Pandas: hỗ trợ xử lý và phân tích dữ liệu có cấu trúc dưới dạng bảng.
* Matplotlib: cung cấp các công cụ để tạo ra biểu đồ và đồ thị.
* TensorFlow: thư viện hỗ trợ xây dựng mô hình máy học và deep learning.
* Django: framework web cho phép phát triển các ứng dụng web phức tạp.

1. Virtual environment: Virtual environment là một cách để tạo ra môi trường lập trình Python độc lập để phát triển các ứng dụng mà không ảnh hưởng đến môi trường Python toàn cục. Điều này giúp cho việc quản lý các thư viện và phát triển ứng dụng trở nên dễ dàng hơn.
2. Công cụ phát triển Python: Python có nhiều công cụ phát triển hỗ trợ như PyCharm, Visual Studio Code, Jupyter Notebook, v.v. Những công cụ này cung cấp môi trường phát triển tích hợp và giúp cho lập trình viên dễ dàng phát triển và quản lý mã ngu

## Thư viện Pygame

1. Pygame Zero được tạo ra để cung cấp trải nghiệm code game tốt hơn cho các lập trình viên Python. Với Pygame Zero, bạn có thể bỏ qua việc tạo lại các code phổ biến và dành thời gian đó cho việc sáng tạo.

2. Thư viện random trong Python là một thư viện được sử dụng để tạo ra các giá trị ngẫu nhiên. Nó cung cấp một số hàm để tạo ra các giá trị ngẫu nhiên cho các số nguyên, số thực và chuỗi.

Các hàm cơ bản của thư viện random bao gồm:

* random(): Tạo ra một số ngẫu nhiên từ 0 đến 1.
* randint(a, b): Tạo ra một số ngẫu nhiên trong khoảng từ a đến b.
* uniform(a, b): Tạo ra một số ngẫu nhiên trong khoảng từ a đến b với bất kỳ giá trị thập phân nào.
* choice(seq): Lấy một phần tử ngẫu nhiên từ một chuỗi cho trước.
* shuffle(seq): Xáo trộn các phần tử trong chuỗi cho trước.

Ngoài ra, thư viện random còn cung cấp một số hàm khác để tạo ra các giá trị ngẫu nhiên, như seed(), getstate(), setstate(), và các hàm liên quan đến phân phối như gauss() và triangular().

Thư viện random được sử dụng rất phổ biến trong các ứng dụng thống kê, mô phỏng, trò chơi và kiểm thử. Nó cũng là một trong những thư viện cơ bản và hữu ích trong Python.

3. Thư viện time trong Python cung cấp các chức năng để xử lý thời gian và đo thời gian. Nó cung cấp các chức năng để truy cập hệ thống thời gian, tạo độ điều chỉnh và đo lường thời gian thực hiện các tính năng được phép. Dưới đây là một số hàm cơ bản của thư viện thời gian:

* time(): return number seconds kể từ thời điểm 00:00:00 UTC ngày 1/1/1970.
* sleep(): tạm dừng chương trình trong một số giây được định trước.
* perf\_counter(): trả về thời gian hiện tại trong giây, với độ chính xác cao hơn so với thời gian hàm.
* process\_time(): trả về thời gian CPU sử dụng bởi chương trình hiện tại.

Ngoài các hàm này, thư viện thời gian còn cung cấp các hàm để chuyển đổi định dạng thời gian, ví dụ như hàm gmtime() để lấy thời gian hiện tại theo định dạng UTC, localtime() để lấy thời gian hiện tại theo định dạng format address, and strftime() to a time string format.

Thời gian thư viện được sử dụng rất phổ biến trong các ứng dụng thời gian thực, đo thời gian và thực hiện các nhiệm vụ đồng bộ hóa thời gian trong các ứng dụng Python.

# **CHƯƠNG 2. THIẾT KẾ CHƯƠNG TRÌNH**

## 2.1. Quy trình

Bước 1: Import các thư viện cần thiết

* **pygame zero**: Thư viện đồ họa cho Python
* **random**: Thư viện sinh số ngẫu nhiên

Bước 2: Khai báo lớp **Game** từ **pgzrun**

* Khởi tạo các biến xe tăng và đạn cảu mình và của địch
* Sử dụng hàm **Actor()** để thêm hình ảnh và âm thanh
* Sử dụng hàm **draw()** để vẽ giao diện chương trình

Bước 3: Tạo hàm **collidelist** để sét va chạm

* tank.collidelist(walls) 1 xe tăng va chạm với tường
* wallS\_index=bullet.collidelist(walls) đạn va chạm với tường
* enemy\_index=bullet.collidelist(enemies) đạn va chạm với xe địch

Bước 6: Gọi hàm **pgzrun.go()** để chạy chương trình.

## 2.2. Giao diện

Chương trình trò chơi Bắn xe tăng được thiết kế với giao diện sử dụng thư viện pygame zero của Python. Khi chạy chương trình, người dùng sẽ nhìn thấy 1 chiếc xe tăng màu xanh là xe tăng bên mình và 6 chiếc xe tăng màu đỏ là của địch.

Người chơi sử dụng 4 nút di chuyển trên bàn phím để di chuyển xe tăng mình ,ấn nút cách để bắn đạn.

Khi tiêu diệt được hết xe tăng bên địch sẽ thắng

## 2.3. Thiết kế chương trình

* **Import thư viện :**

import pygzrun

import random

* **Ở trong lớp Game, chúng ta có một số thuộc tính, bao gồm:**

WIDTH =800

HEIGHT = 600

#cài kích thước cửa sổ trò chơi

SIZE\_TANK=25

#kích thước xe tăng

enemies=[]

#mảng chứa các xe tăng địch

walls=[]

#mảng gồm các bức tường

bullets=[]

#mảng chứa các viên đạn xe mình

bullets\_holdoff=0

# độ trễ khi bắn của xe mình

enemy\_bullets\_holdoff=0

# độ trễ khi bắn của xe địch

enemy\_move\_count=0

# thời gian xe địch thay đổi trạng thái

enemy\_bullets=[]

# mảng chứa đạn của xe địch

game\_over=False

* **Định dạng xe mình:**

- Sử dụng hàm Actor để lấy file ảnh

- Phương thức pos để cài đặt vị trí cho xe

- Phương thức angle để cài đặt hướng góc cho xe

tank = Actor("tank\_blue")

tank.pos=(WIDTH/2,HEIGHT-SIZE\_TANK)

tank.angle=90

* **Định dạng xe tăng địch**

-Sử dụng for để khởi tạo 6 chiếc xe của địch

for i in range(6):

    enemy=Actor("tank\_red")

    enemy.x=i\*100+100

    enemy.y=SIZE\_TANK

    enemy.angle=270

    enemies.append(enemy)

* **Cài đặt nền và tường**

-Chia chiều ngang cửa sổ thành 16 và chiều dọc thành 10 ô gạch và sử dụng 2 vòng for lồng nhau để cài đặt từng ô

-Sử dụng random để ngẫu nhiên tại vị trí đang cài đặt có tường hoặc không

background = Actor('grass')

for x in range(16):

    for y in range(10):

        if random.randint(0,100)<50:

            wall=Actor("wall")

            wall.x=x\*50+SIZE\_TANK

            wall.y=y\*50+50+SIZE\_TANK

            walls.append(wall)

* **Cài dặt xe tăng mình**

-Khởi tạo hai biến original x-y để lưu lại vị trí hiện tại của xe

- keyboard để bắt sự kiện ấn nút trên bàn phím máy tính và angle để thay đổi hướng quay của xe

-tank.collidelist(walls)!=-1 bắt sự kiện va chạm của xe với tường và tank.x<SIZE\_TANK or tank.x>(WIDTH-SIZE\_TANK) or tank.y<SIZE\_TANK or tank.y>(HEIGHT-SIZE\_TANK): xe va chạm với giới hạn cửa sổ nếu đúng sẽ gán vị trí xe bằng vị trí khi bắt đầu va chạm

def tank\_set():

    original\_x=tank.x

    original\_y=tank.y

    if keyboard.left:

        tank.x=tank.x-2

        tank.angle=180

    elif keyboard.right:

        tank.x=tank.x+2

        tank.angle=0

    elif keyboard.up:

        tank.y=tank.y-2

        tank.angle=90

    elif keyboard.down:

        tank.y=tank.y+2

        tank.angle=270

    if tank.collidelist(walls)!=-1:

        tank.x=original\_x

        tank.y=original\_y

    if tank.x<SIZE\_TANK or tank.x>(WIDTH-SIZE\_TANK) or tank.y<SIZE\_TANK or tank.y>(HEIGHT-SIZE\_TANK):

        tank.x=original\_x

        tank.y=original\_y

* **Cài đặt đạn xe mình**

-Khi bullets\_holdoff==0 tức là độ trễ khi bắn về 0 mà ấn nút space sẽ khởi tạo ra một viên đạn được lấy ra từ file ảnh bằng hàm Actor ,vị trí đạn và hướng sẽ cùng với vị trí và hướng của xe

- if bullet.angle==0,180,90,270 Ở các hướng khác nhau vị trí đạn sẽ được tăng thêm đơn vị bằng kích thước của xe để đạn xuất hiện tại ví trí đầu nòng

-Lưu vào mảng và gán độ trễ bằng 20

-Nếu độ trễ chưa bằng 0 sẽ giảm độ trễ đi 1 giá trị

- for bullet in bullets: duyệt các viên đạn chứa trong mảng và bullet.angle==0: tại các hướng khác nhau sẽ bullet.x=bullet.x+5 di chuyển viên đạn theo hướng đó mỗi lần 5 đơn vị

- tạo một vòng for khác cũng duyệt các viên đạn đó

- wallS\_index=bullet.collidelist(walls) Tạo biến lưu lại giá trị của hàm va chạm giữa đạn với tường .Nếu khác -1 tức là va chạm ,sẽ sounds.gun9.play() lấy âm thanh từ file và phát sau đó xóa tường tại vị trí va chạm và viên đạn đó

- if bullet.x<0 or bullet.x>WIDTH or bullet.y<0 or bullet.y>HEIGHT:nếu vị trí viên đạn không nằm trong phạm vi kích thước cửa sổ sẽ xóa viên đạn đó đi

- enemy\_index=bullet.collidelist(enemies) tạo một biến lưu giá trị va chạm của đạn với xe của địch nếu khác -1 sẽ phát âm thanh và xóa viên đạn cùng chiếc xe tăng đó

def tank\_bullets\_set():

    global bullets\_holdoff

    if bullets\_holdoff==0:

        if keyboard.space:

            bullet=Actor("bulletblue2")

            bullet.angle=tank.angle

            bullet.pos=tank.pos

            if bullet.angle==0:

                bullet.x=tank.x+SIZE\_TANK

            if bullet.angle==180:

                bullet.x=tank.x-SIZE\_TANK

            if bullet.angle==90:

                bullet.y=tank.y-SIZE\_TANK

            if bullet.angle==270:

                bullet.y=tank.y+SIZE\_TANK

            bullets.append(bullet)

            bullets\_holdoff=20

    else:

        bullets\_holdoff=bullets\_holdoff-1

    for bullet in bullets:

        if bullet.angle==0:

            bullet.x=bullet.x+5

        if bullet.angle==180:

            bullet.x=bullet.x-5

        if bullet.angle==90:

            bullet.y=bullet.y-5

        if bullet.angle==270:

            bullet.y=bullet.y+5

    for bullet in bullets:

        wallS\_index=bullet.collidelist(walls)

        if wallS\_index!=-1:

            sounds.gun9.play()

            del walls[wallS\_index]

            bullets.remove(bullet)

        if bullet.x<0 or bullet.x>WIDTH or bullet.y<0 or bullet.y>HEIGHT:

            bullets.remove(bullet)

        enemy\_index=bullet.collidelist(enemies)

        if enemy\_index !=-1:

            sounds.exp.play()

            bullets.remove(bullet)

            del enemies[enemy\_index]

* Cài đặt xe tăng địch

-Cũng cài đặt va chạm với tường và giới hạn cửa sổ trò chơi như xe mình

-random 3 trường hợp nếu >0 sẽ di chuyển,=1 sẽ đổi hướng,=2 sẽ bắn và cũng cài đặt như xe mình bắn nếu va chạm với xe mình sẽ cho gameOver=true

def enemy\_set():

    global enemy\_move\_count

    global enemy\_bullets\_holdoff

    for enemy in enemies:

        original\_x=enemy.x

        original\_y=enemy.y

        choice=random.randint(0,2)

        if enemy\_move\_count>0:

            enemy\_move\_count=enemy\_move\_count-1

            if enemy.angle==0:

                enemy.x=enemy.x+2

            elif enemy.angle==180:

                enemy.x=enemy.x-2

            elif enemy.angle==90:

                enemy.y=enemy.y-2

            elif enemy.angle==270:

                enemy.y=enemy.y+2

            if enemy.x<SIZE\_TANK or enemy.x>(WIDTH-SIZE\_TANK) or enemy.y<SIZE\_TANK or enemy.y>(HEIGHT-SIZE\_TANK):

                enemy.x=original\_x

                enemy.y=original\_y

                enemy\_move\_count=0

            if enemy.collidelist(walls)!=-1:

                enemy.x=original\_x

                enemy.y=original\_y

                enemy\_move\_count=0

        if choice==0: #xe tăng địch di chuyển

            enemy\_move\_count=30

        if choice==1: # xe tăng địch đổi hướng

            enemy.angle=random.randint(0,3)\*90

        else: # xe tăng địch bắn

            if enemy\_bullets\_holdoff==0:

                bullet=Actor("bulletred2")

                bullet.angle=enemy.angle

                bullet.pos=enemy.pos

                if bullet.angle==0:

                    bullet.x=enemy.x+SIZE\_TANK

                if bullet.angle==180:

                    bullet.x=enemy.x-SIZE\_TANK

                if bullet.angle==90:

                    bullet.y=enemy.y-SIZE\_TANK

                if bullet.angle==270:

                    bullet.y=enemy.y+SIZE\_TANK

                enemy\_bullets.append(bullet)

                enemy\_bullets\_holdoff=60

            else:

                enemy\_bullets\_holdoff=enemy\_bullets\_holdoff-1

# đạn xe địch

def anemy\_bullets\_set():

    global enemies,game\_over

    for bullet in enemy\_bullets:

        if bullet.angle==0:

            bullet.x=bullet.x+5

        if bullet.angle==180:

            bullet.x=bullet.x-5

        if bullet.angle==90:

            bullet.y=bullet.y-5

        if bullet.angle==270:

            bullet.y=bullet.y+5

        #đạn địch phá tường ,phá xe

        for bullet in enemy\_bullets:

            wall\_index=bullet.collidelist(walls)

            if wall\_index!=-1:

                sounds.gun10.play()

                del walls[wall\_index]

                enemy\_bullets.remove(bullet)

            if bullet.x<0 or bullet.x>WIDTH or bullet.y<0 or bullet.y>HEIGHT:

                enemy\_bullets.remove(bullet)

            if bullet.colliderect(tank):

                game\_over=True

                enemies=[]

* **Hàm update() để các hàm chạy liên tục**

def update():

    tank\_set()

    tank\_bullets\_set()

    enemy\_set()

    anemy\_bullets\_set()

* **Vẽ lên trên màn hình**

-Sử dụng hàm draw để vẽ mọi thứ lên trên màn hình

def draw():

    if game\_over:

        screen.fill((0,0,0))

        screen.draw.text('YOU LOSE',(260,250),color=(255,255,255),fontsize=100)

    elif len(enemies)==0:

        screen.fill((0,0,0))

        screen.draw.text('YOU WIN',(260,250),color=(255,255,255),fontsize=100)

    else:

        background.draw()

        tank.draw()

        for wall in walls:

            wall.draw()

        for bullet in bullets:

            bullet.draw()

        for enemy in enemies:

            enemy.draw()

        for bullet in enemy\_bullets:

            bullet.draw()

## 2.3. Kết luận

Mã này là một trò chơi bắn xe tăng được viết bằng Python và sử dụng giao thư viện của pygame zero. Đối với độ chính xác nếu bài toán học máy, mã này không liên quan đến học máy và không có yếu tố đánh giá độ chính xác. Tuy nhiên, nếu đánh giá độ chính xác về tính đúng đắn của mã, thì có thể nói mã này là đúng và hoạt động tốt..

## 2.4. Tài liệu tham khảo

https://www.youtube.com/watch?v=Nnx16zgamZQ&t=1736s