

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO CUỐI KỲ**

**Kỹ Thuật Lập Trình Python – IE221.L21.CNCL**

**Ứng dụng game cơ bản bằng Pygame**



Giảng viên hướng dẫn: Nguyễn Thanh Sơn

Danh sách thành viên nhóm:

Lương Duy Bảo – 18520484

## **Mục lục**

I. Giới thiệu sơ lược về python .....	4
II. Giới thiệu pygame .....	4
1. Pygame là gì?.....	4
2. Lịch sử .....	4
3. Kiến trúc và tính năng .....	4
4. Cách cài đặt .....	5
III. Các thư viện được sử dụng: .....	5
IV. Quá trình làm việc: .....	5
V. Code game Space Invaders .....	7
1. Tạo cửa sổ game .....	7
2. Load các hình ảnh.....	7
3. Thiết lập title, icon.....	8
4. Tạo background .....	8
5. Class enemy .....	8
6. Tạo player, thiết lập di chuyển cho player .....	9
7. Xử lý va chạm .....	11

8. Xử lí bắn đạn .....	11
9. Hiện thị lives, level:.....	12
10. Game over: .....	12

## **I. Giới thiệu sơ lược về python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

## **II. Giới thiệu pygame**

### **1. Pygame là gì?**

Pygame là một bộ mô-đun Python đa nền tảng được thiết kế để viết trò chơi điện tử. Nó bao gồm đồ họa máy tính và thư viện âm thanh được thiết kế để sử dụng với ngôn ngữ lập trình Python.

### **2. Lịch sử**

Pygame ban đầu được viết bởi Pete Shinnars để thay thế PySDL sau khi quá trình phát triển của nó bị đình trệ. Đây là một dự án cộng đồng từ năm 2000 và được phát hành theo phần mềm miễn phí mã nguồn mở GNU Lesser General Public License.

### **3. Kiến trúc và tính năng**

Pygame sử dụng thư viện Simple DirectMedia Layer (SDL), với mục đích cho phép phát triển trò chơi máy tính trong thời gian thực mà không cần cơ chế bậc thấp của ngôn ngữ lập trình C và các dẫn xuất của nó. Điều này dựa trên giả định rằng các chức năng đất tiên nhất bên trong trò chơi có thể được trừu tượng hóa khỏi logic trò chơi, do

đó có thể sử dụng ngôn ngữ lập trình bậc cao, chẳng hạn như Python, để cấu trúc trò chơi.

Các tính năng khác mà SDL không có bao gồm toán học vectơ, phát hiện va chạm, quản lý độ họa 2d, hỗ trợ MIDI, camera, thao tác mảng pixel, chuyển đổi, lọc, hỗ trợ phông chữ freetype nâng cao và vẽ.

Các ứng dụng sử dụng pygame có thể chạy trên điện thoại và máy tính bảng Android với việc sử dụng Bộ phụ pygame cho Android (pgs4a). Âm thanh, rung, bàn phím và gia tốc kế được hỗ trợ trên Android.

#### 4. Cách cài đặt

nếu bạn đã cài đặt python, bạn có thể cài đặt pygame bằng pip:

```
pip install pygame
```

### III. Các thư viện được sử dụng:

STT	Thư viện	Chức năng
1	pygame	Tạo cửa sổ, render ảnh, dùng hàm load của pygame để load ảnh, xử lý sự kiện bắt phím, sử dụng pygame.mixer để chèn nhạc
2	os	Để xử lý các đường dẫn
3	math	Dùng để tính toán khoảng cách dựa vào tọa độ của 2 vật
4	random	Random vị trí xuất hiện của enemy

### IV. Quá trình làm việc:

Nhóm 22		Game Space Invaders bằng pygame		Date of Preparation		Activity Plan Prepared by Lương Duy Bảo
ST T	Tên task	Planned Start Date	Planned Completed Date	Tên thành viên	Tình trạng	
1	Tìm hiểu pygame, cách render ảnh, key press	May 2, 2021	May 8, 2021	Lương Duy Bảo	Hoàn thành	
2	Tạo background, thiết lập title, icon		May 9 ,2021		Hoàn thành	
3	Tạo player, xử lí key press		May 10,2020		Hoàn thành	
4	Tạo 1 enemy, thiết lập di chuyển cho enemy		May 10 ,2021		Hoàn thành	
5	Tạo bullet, thiết lập bắn đạn cho player		May 12,2021		Hoàn thành	

6	Xử lí va chạm, giết enemy bằng đạn, player mất máu khi chạm enemy, chèn âm thanh, hiện số mạng, hiện level		May 12,2020		Hoàn thành
7	Tạo nhiều enemy, game over, xử lí bắn đạn cho enemy		May 20,2020		Hoàn thành

## V. Code game Space Invaders

### 1. Tạo cửa sổ game

Trước tiên cần khởi tạo để sử dụng các hàm của pygame

```
pygame.init()
```

Tạo cửa sổ game:

```
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    pygame.display.update()
```

### 2. Load các hình ảnh

```
#load image
PLAYER = pygame.image.load(os.path.join("Image", "player.png"))
ENEMY = pygame.image.load(os.path.join("Image", "enemy.png"))
UFO = pygame.image.load(os.path.join("Image", "ufo.png"))
UFO1 = pygame.image.load(os.path.join("Image", "ufo1.png"))
BULLET = pygame.image.load(os.path.join("Image", "bullet.png"))
ALIEN = pygame.image.load(os.path.join("Image", "alien.png"))
ALIEN1 = pygame.image.load(os.path.join("Image", "alien1.png"))
```

### 3. Thiết lập title, icon

```
# title and icon
pygame.display.set_caption('Space Invaders')
icon = pygame.image.load('./Image/spaceship.png')
pygame.display.set_icon(icon)
```

### 4. Tạo background

```
#background
background = pygame.image.load('./Image/background.png')
```

Vẽ background lên screen:

```
#background Image
screen.blit(background, (0,0))
```

### 5. Class enemy



```

class Enemy(Ship):
    TYPE = {
        "enemy": (ENEMY, ALIEN),
        "ufo": (UFO, ALIEN),
        "ufo1": (UFO1, ALIEN1)
    }
    def __init__(self, x, y, type, health = 100):
        super().__init__(x, y, health)
        self.ship_img, self.bullet_img = self.TYPE[type]
        self.mask = pygame.mask.from_surface(self.ship_img)

    def move(self, changeY):
        self.y += changeY

```

6. Tạo player, thiết lập di chuyển cho player  
class player:

```

class Player(Ship):
    def __init__(self, x, y, health = 100):
        super().__init__(x, y, health)
        self.ship_img = PLAYER
        self.bullet_img = BULLET
        self.mask = pygame.mask.from_surface(self.ship_img)
        self.max_health = health

    def move_bullet(self, vel, objs):
        self.cooldown()
        for bullet in self.bullets:
            bullet.move(vel)
            if bullet.off_screen(screen_height):
                self.bullets.remove(bullet)
            else:
                for obj in objs:
                    if bullet.collison(obj):
                        objs.remove(obj)
                        self.bullets.remove(bullet)
                        explosion_sound = mixer.Sound('./music/explosion.wav')
                        explosion_sound.play()

    def draw(self, screen):
        super().draw(screen)
        self.healthbar(screen)

    def healthbar(self, screen):
        pygame.draw.rect(screen, (255, 0, 0), (self.x, self.y+64+10, 64, 10))
        pygame.draw.rect(screen, (0, 255, 0), (self.x, self.y + 64 + 10, 64*(self.health/self.max_health), 10))

```

You, 3 days ago • update

Xử lí di chuyển cho player:

```

    running = False
    if lost == False:
        keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]:
        player.x -= playerX_change

    if keys[pygame.K_RIGHT]:
        player.x += playerX_change

    if keys[pygame.K_UP]:
        player.y -= playerY_change

    if keys[pygame.K_DOWN]:
        player.y += playerY_change

    if keys[pygame.K_SPACE]:
        laser_sound = mixer.Sound('./music/laser.wav')
        laser_sound.play()
        player.shoot()

```

## 7. Xử lí va chạm

Sử dụng công thức tính khoảng cách giữa 2 điểm A(x1,y1) và B(x2,y2)

$$D = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

```

def collide(obj1,obj2):
    dist = math.sqrt(math.pow(obj1.x-obj2.x,2) + math.pow(obj1.y-obj2.y,2))
    if dist < 27:
        return True
    else:
        return False

```

## 8. Xử lí bắn đạn

class bullet

```

class Bullet:
    def __init__(self,x,y,img):
        self.x = x
        self.y = y
        self.img = img
        self.mask = pygame.mask.from_surface(self.img)

    def draw(self,screen):
        screen.blit(self.img,(self.x,self.y))

    def move(self,vel):
        self.y += vel

    def off_screen(self,height):
        return not (self.y <= height and self.y >= 0)

    def collision(self,obj):
        return collide(self,obj)

```

Sự kiện nhấn phím space bắn đạn:

```

if keys[pygame.K_SPACE]:
    laser_sound = mixer.Sound('./music/laser.wav')
    laser_sound.play()
    player.shoot()

```

9. Hiện thị lives, level:

```

level_label = main_font.render(f"Level: {level}",1,(255,255,255))
lives_label = main_font.render(f"Lives: {lives}",1,(255,255,255))
screen.blit(level_label,(650,0))
screen.blit(lives_label,(10,10))

```

10.Game over:

```
def game_over_text():  
    over_text = over_font.render("GAME OVER", True, (255, 255, 255))  
    screen.blit(over_text, (200, 250))
```

```
if lives <= 0 or player.health <= 0:  
    lost = True
```

```
if lost:  
    game_over_text()
```