import pygame #

from pygame import mixer

from pygame.locals import \*

import random

import sys

# Khởi tạo Pygame và âm thanh #

pygame.mixer.pre\_init(44100, -16, 2, 512)

mixer.init()

pygame.init() # khởi tạo các module pygame đã nhập, gọi nó là vị trí đầu tiên

# Tải âm thanh

explosion\_fx = pygame.mixer.Sound('boom.mp3')

explosion\_fx.set\_volume(0.335)

explosion2\_fx = pygame.mixer.Sound('boom1.mp3')

explosion2\_fx.set\_volume(0.335)

shot\_fx = pygame.mixer.Sound('shot.mp3')

shot\_fx.set\_volume(0.005)

winning = pygame.mixer.Sound('win.mp3')

winning.set\_volume(0.25)

loser = pygame.mixer.Sound('lose.mp3')

loser.set\_volume(0.25)

nhac\_nen = pygame.mixer.Sound('music.mp3')

nhac\_nen.set\_volume(0.33)

# Cài đặt chung #

clock = pygame.time.Clock() # giới hạn số khung hình trên giây để đảm các biến tronh đó không chạy quá nhanh or quá chậm tùy loại máy tính khác nhaunhau

fps = 60

screen\_width = 736

screen\_height = 421

screen = pygame.display.set\_mode((screen\_width, screen\_height)) # tạo mà hình trờ chơi

pygame.display.set\_caption('CYBER SHIELD') # đặt tên của sổ game

# Xác định phông chữ

font30 = pygame.font.SysFont('Constantia', 30)

font40 = pygame.font.SysFont('Constantia', 40)

font50 = pygame.font.SysFont('Bold System Font', 40)

font60 = pygame.font.SysFont('Bold System Font', 50)

# Xác định các biến trò chơi #

rows = 5

cols = 5

virus\_cooldown = 1000

last\_virus\_shot = pygame.time.get\_ticks()

countdown = 3

last\_count = pygame.time.get\_ticks()

score = 0  # Biến lưu điểm số

virus\_grid\_size = 1  # Kích thước ban đầu của nhóm virus (1x1)

max\_virus\_grid = 5

game\_over = 0

# Xác định màu sắc

red = (255, 0, 0)

green = (0, 255, 0)

white = (255, 255, 255)

yellow = (255, 255, 0)

# Tải ảnh

icon = pygame.image.load('car1.png')

bg = pygame.image.load('dg1.jpg')

intro\_background = pygame.image.load("nen.jpg")

end\_background = pygame.image.load("end.jpg")

pygame.display.set\_icon(icon)

# Chức năng hiển thị văn bản

def draw\_text(text, font, text\_col, x, y):

    img = font.render(text, True, text\_col)

    screen.blit(img, (x, y))

def draw\_bg():

    screen.blit(bg, (0, 0))

# Màn hình chờ

def intro\_screen():

    while True:

        screen.blit(intro\_background, (0, 0))

        draw\_text("PRESS ANY KEY TO START", font50, yellow, screen\_width // 2 - 180, screen\_height // 2 + 150)

        pygame.display.update()

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                sys.exit()

            if event.type == pygame.KEYDOWN:

                return

# Màn hình kết thúc #

def game\_over\_screen(score, result):

    while True:

        # Vẽ background kết thúc

        screen.blit(end\_background, (0, 0))

        # Hiển thị kết quả

        if result == 'win':

            draw\_text("YOU WIN!", font60, yellow, screen\_width // 2 - 150, screen\_height // 2 - 100)

            winning.play()

        else:

            draw\_text("YOU DIED!!!", font40, red, screen\_width // 2 - 100, screen\_height // 2 - 100)

        # Hiển thị điểm số

        draw\_text(f"SCORE: {score}", font30, white, screen\_width // 2 - 50, screen\_height // 2 - 30)

        # Hiển thị hướng dẫn chơi lại

        draw\_text("PRESS '1' TO RESTART", font50, red, screen\_width // 2 - 145, screen\_height // 2 + 100)

        pygame.display.update()

        # Chờ người chơi nhấn '1'

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                sys.exit()

            if event.type == pygame.KEYDOWN:

                if event.key == pygame.K\_1:

                    return  # Quay lại màn hình chờ

# Spaceship class #

class Spaceship(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y, health):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.image = pygame.image.load('car1.png')

        self.rect = self.image.get\_rect()

        self.rect.center = [x, y]

        self.health\_start = health

        self.health\_remaining = health

        self.last\_shot = pygame.time.get\_ticks()

    def update(self):

        speed = 5

        cooldown = 300

        game\_over = 0

        key = pygame.key.get\_pressed()

        if key[pygame.K\_LEFT] and self.rect.left > 0:

            self.rect.x -= speed

        if key[pygame.K\_RIGHT] and self.rect.right < screen\_width:

            self.rect.x += speed

        time\_now = pygame.time.get\_ticks()

        if key[pygame.K\_UP] and self.rect.top > 0:

            self.rect.y -= speed

        if key[pygame.K\_DOWN] and self.rect.bottom < screen\_height:

            self.rect.y += speed

        if key[pygame.K\_SPACE] and time\_now - self.last\_shot > cooldown:

            shot\_fx.play()

            bullet = Bullets(self.rect.centerx, self.rect.top)

            bullet\_group.add(bullet)

            self.last\_shot = time\_now

        self.mask = pygame.mask.from\_surface(self.image)

        pygame.draw.rect(screen, red, (self.rect.x, (self.rect.bottom + 10), self.rect.width, 10))

        if self.health\_remaining > 0:

            pygame.draw.rect(screen, green, (self.rect.x, (self.rect.bottom + 10), int(self.rect.width \* (self.health\_remaining / self.health\_start)), 10))

        elif self.health\_remaining <= 0:

            explosion = Explosion(self.rect.centerx, self.rect.centery, 3)

            explosion\_group.add(explosion)

            self.kill()

            game\_over = -1

        return game\_over

# Bullet class #

class Bullets(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.image = pygame.image.load('dan1.png')

        self.rect = self.image.get\_rect()

        self.rect.center = [x, y]

    def update(self):

        self.rect.y -= 5

        if self.rect.bottom < 0:

            self.kill()

        if pygame.sprite.groupcollide(bullet\_group, virus\_group, True, True):

            global score

            score += 10  # Tăng điểm khi tiêu diệt virus

            self.kill()

            explosion\_fx.play()

            explosion = Explosion(self.rect.centerx, self.rect.centery, 2)

            explosion\_group.add(explosion)

# Virus class #

class Viruss(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.image = pygame.image.load('covid' + str(random.randint(1, 4)) + '.png')

        self.rect = self.image.get\_rect()

        self.rect.center = [x, y]

        self.move\_counter = 0

        self.move\_direction = 1

    def update(self):

        # Di chuyển sang trái và phải

        self.rect.x += self.move\_direction

        self.move\_counter += 1

        if abs(self.move\_counter) > 145:

            self.move\_direction \*= -1

            self.move\_counter \*= self.move\_direction

# Virus Bullet class #

class Virus\_Bullets(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.image = pygame.image.load('virus\_bullet.png')

        self.rect = self.image.get\_rect()

        self.rect.center = [x, y]

        self.speed = 5

    def update(self):

        self.rect.y += 2

        if self.rect.top > screen\_height:

            self.kill()

        if pygame.sprite.spritecollide(self, spaceship\_group, False, pygame.sprite.collide\_mask):

            self.kill()

            explosion2\_fx.play()

            spaceship.health\_remaining -= 1

            explosion = Explosion(self.rect.centerx, self.rect.centery, 1)

            explosion\_group.add(explosion)

# Explosion class #

class Explosion(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x, y, size):

        pygame.sprite.Sprite.\_\_init\_\_(self)

        self.images = []

        for num in range(7, 13):

            img = pygame.image.load('ex' + str(num) + '.png')

            if size == 1:

                img = pygame.transform.scale(img, (20, 20))

            if size == 2:

                img = pygame.transform.scale(img, (40, 40))

            if size == 3:

                img = pygame.transform.scale(img, (160, 160))

            self.images.append(img)

        self.index = 0

        self.image = self.images[self.index]

        self.rect = self.image.get\_rect()

        self.rect.center = [x, y]

        self.counter = 0

    def update(self):

        explosion\_speed = 3

        self.counter += 1

        if self.counter >= explosion\_speed and self.index < len(self.images) - 1:

            self.counter = 0

            self.index += 1

            self.image = self.images[self.index]

        if self.index >= len(self.images) - 1 and self.counter > -explosion\_speed:

            self.kill()

# Tạo nhóm sprite #

spaceship\_group = pygame.sprite.Group()

bullet\_group = pygame.sprite.Group()

virus\_group = pygame.sprite.Group()

virus\_bullet\_group = pygame.sprite.Group()

explosion\_group = pygame.sprite.Group()

# Tạo xe cứu thương

spaceship = Spaceship(screen\_width // 2, screen\_height - 50, 3)

spaceship\_group.add(spaceship)

def create\_virus\_group(grid\_size):

    virus\_group.empty()  # Xóa nhóm virus cũ

    # Tính toán kích thước và khoảng cách của lưới

    virus\_width = 30  # Giả định chiều rộng của virus

    virus\_height = 30  # Giả định chiều cao của virus

    spacing\_x = 40  # Khoảng cách ngang giữa các virus

    spacing\_y = 25  # Khoảng cách dọc giữa các virus

    total\_width = grid\_size \* (virus\_width + spacing\_x) - spacing\_x

    start\_x = (screen\_width - total\_width) // 2  # Bắt đầu ở giữa màn hình

    total\_height = grid\_size \* (virus\_height + spacing\_y) - spacing\_y

    start\_y = (screen\_height // 3 - total\_height // 2)  # 2/3 phần trên của màn hình

    # Tạo lưới virus #

    for row in range(grid\_size):

        for col in range(grid\_size):

            virus\_x = start\_x + col \* (virus\_width + spacing\_x)

            virus\_y = start\_y + row \* (virus\_height + spacing\_y)

            virus = Viruss(virus\_x, virus\_y)

            virus\_group.add(virus)

def virus\_shoot(): #

    global last\_virus\_shot

    time\_now = pygame.time.get\_ticks()  # Lấy thời gian hiện tại

    if time\_now - last\_virus\_shot > virus\_cooldown:  # Kiểm tra cooldown bắn đạn

        if len(virus\_group) > 0:  # Kiểm tra xem còn virus không

            attacking\_virus = random.choice(virus\_group.sprites())  # Chọn 1 virus ngẫu nhiên

            virus\_bullet = Virus\_Bullets(attacking\_virus.rect.centerx, attacking\_virus.rect.bottom)

            virus\_bullet\_group.add(virus\_bullet)  # Thêm đạn vào nhóm đạn

            last\_virus\_shot = time\_now

# Vòng lặp chính #

run = True

while run: # để giữ màn hình đứng in, không biến mất tức thì

    intro\_screen()

    # Reset mọi thứ

    nhac\_nen.play(-1)  # Chạy nhạc nền liên tục

    score = 0  # Reset điểm số

    countdown = 3  # Reset thời gian đếm ngược

    spaceship\_group.empty()  # Làm trống nhóm sprite

    bullet\_group.empty()

    virus\_group.empty()

    explosion\_group.empty()

    virus\_bullet\_group.empty()

    # Tạo lại đối tượng chính #

    spaceship = Spaceship(screen\_width // 2, screen\_height - 100, 3)

    spaceship\_group.add(spaceship)

    # Tạo lại viruses #

    create\_virus\_group(virus\_grid\_size)

    game\_over = 0

    while game\_over == 0:

        clock.tick(fps)

        draw\_bg()

        for event in pygame.event.get():

            if event.type == pygame.QUIT: # thoát khỏi của sổ game trên máy tính

                run = False

                game\_over = -1

        if countdown > 0:

            draw\_bg()

            draw\_text('GET READY!', font40, red, screen\_width // 2 - 110, screen\_height // 2 - 50)

            draw\_text(str(countdown), font40, red, screen\_width // 2, screen\_height // 2)

            pygame.display.update()

            count\_timer = pygame.time.get\_ticks()

            if count\_timer - last\_count > 1000:

                countdown -= 1

                last\_count = count\_timer

            continue

        virus\_shoot()

        # Cập nhật và hiển thị các đối tượng

        spaceship\_group.update()

        bullet\_group.update()

        virus\_group.update()

        explosion\_group.update()

        virus\_bullet\_group.update()

        spaceship\_group.draw(screen)

        bullet\_group.draw(screen)

        virus\_group.draw(screen)

        explosion\_group.draw(screen)

        virus\_bullet\_group.draw(screen)

        # Hiển thị điểm

        draw\_text(f"SCORE: {score}", font30, white, 10, 10)

        pygame.display.update() # cập nhật mới game

        # Kiểm tra kết thúc game

        if len(virus\_group) == 0:

            game\_over = 1

        elif spaceship.health\_remaining <= 0:

            game\_over = -1

    nhac\_nen.stop()

    if game\_over == -1:

        loser.play()

        game\_over\_screen(score, "lose")

        virus\_grid\_size = 1

        spaceship.health\_remaining = 3  # Hồi lại máu

        create\_virus\_group(virus\_grid\_size)

        game\_over = 0

    else:

        virus\_grid\_size += 1  # Tăng kích thước lưới virus

        if virus\_grid\_size > max\_virus\_grid:  # Nếu đạt kích thước tối đa

            screen.blit(end\_background, (0, 0))

            winning.play()

            draw\_text("YOU WIN ALL LEVELS!", font40, red, screen\_width // 2 - 200, screen\_height // 2 - 50)

            draw\_text(f"SCORE:{score}", font30, white, screen\_width // 2 - 50, screen\_height // 2+100)

            pygame.display.update()

            pygame.time.wait(5000)  # Hiển thị thông báo trong 3 giây

            run = False

        else:

            game\_over = 0  # Reset trạng thái game

            spaceship.health\_remaining = 3  # Hồi lại máu

            create\_virus\_group(virus\_grid\_size)  # Tạo virus mới