

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
MÔN KỸ THUẬT LẬP TRÌNH PYTHON

Đề tài:

ỨNG DỤNG GAME CƠ BẢN BẰNG PYGAME

GVHD: Nguyễn Thanh Sơn

Sinh viên thực hiện:

MSSV	Tên	Vai trò
18520688	Trịnh Lê Giang	Trưởng nhóm

☞ Tp. Hồ Chí Minh, 5/2021 ☞

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

....., ngày.....tháng.....năm 2021

Người nhận xét

GIỚI THIỆU VỀ ĐỒ ÁN

Về đồ án pygame, em đã chọn hướng phát triển là tạo nên một game platformer 2D giống như tựa game huyền thoại Mario, thông qua một số tham khảo và tìm hiểu ở nhiều nguồn khác nhau, em đã cho ra được một thành phẩm gần giống với Mario bản gốc. Trong dự án, em sử dụng chủ yếu là các thư viện pygame hỗ trợ, áp dụng một số yêu cầu, quy định của thầy đã giao như: áp dụng mô hình hướng đối tượng bằng cách sử dụng các mô hình lớp, đối tượng, Upload project lên git, phân chia và tổ chức project gọn ghẽ và dễ nhìn.

MỤC LỤC

GIỚI THIỆU VỀ ĐỒ ÁN	3
MỤC LỤC	4
Chương 1: THÔNG TIN VỀ PYGAME	5
1.1Tìm hiểu chung	5
1.2Lý do chọn đề tài.....	5
Chương 2: NỘI DUNG ĐỒ ÁN.....	6
2.1Một số vấn đề chung về đồ án:	6
2.2Quá trình thực hiện:	6
Chương 3: KẾT LUẬN	11
3.1Ưu điểm.....	11
3.2Nhược điểm.....	11
3.3Kết luận	11
TÀI LIỆU THAM KHẢO	11
ĐƯỜNG DẪN ĐẾN DỰ ÁN.....	11

Chương 1: THÔNG TIN VỀ PYGAME

1.1 Tìm hiểu chung

- Pygame là một tập hợp các mô-đun Python được thiết kế để viết trò chơi điện tử. Pygame bổ sung thêm chức năng trên thư viện SDL tuyệt vời. Điều này cho phép bạn tạo các trò chơi và chương trình đa phương tiện đầy đủ tính năng bằng ngôn ngữ python.
- Pygame có tính di động cao và chạy trên hầu hết mọi nền tảng và hệ điều hành.
- Pygame miễn phí. Được phát hành theo giấy phép LGPL, bạn có thể tạo mã nguồn mở, phần mềm miễn phí, phần mềm chia sẻ và trò chơi thương mại với nó.
- Đây là một nền tảng dễ hiểu và thân thiện với người mới bắt đầu có thể giúp bạn phát triển một cách nhanh chóng.
- Pygame hỗ trợ đầy đủ các yếu tố để có thể hoàn thành được một tựa game theo ý tưởng của các bạn, chỉ cần các bạn sáng tạo và biết áp dụng những thứ có sẵn một cách tốt nhất.

1.2 Lý do chọn đề tài

- Pygame là một trong những thư viện dễ tiếp cận và sử dụng. Có các trang hướng dẫn syntax ngay tại trang chủ của pygame.
- Quá trình phát triển pygame không quá mất thời gian và có thể phát triển cá nhân
- Là một đề tài mới mẻ đối với bản thân em, phù hợp cho bản thân để trải nghiệm những việc mới mẻ.
- Pygame hỗ trợ hầu như đầy đủ các tính năng của một dạng trò chơi mà nó hỗ trợ (Thiết lập phím, va chạm, máu, nền....)
- Pygame miễn phí, vì thế có thể tìm các khóa giảng dạy ở tất cả mọi nền tảng từ dạng .doc cho đến những videos
- Đối với dự án này thì em làm một mình với mong muốn là có thể phát triển bản thân theo hướng không bị lệ thuộc và tự sắp xếp thời gian hợp lý hơn cho đề tài này.

Chương 2: NỘI DUNG ĐỒ ÁN

2.1 Một số vấn đề chung về đồ án:

- Game của em phát triển là một tựa game platformer với ý tưởng là tái tạo lại một game giống như Mario.
- Để có thể phát triển game sử dụng pygame một cách nhanh nhất, em đã tổng hợp kiến thức học được từ các nguồn: Trang chủ pygame, các video hướng dẫn và một số forum khác.

2.2 Quá trình thực hiện:

- Các thư viện sử dụng: Pygame (module: locals, mixer)
 - Pygame.locals: Là một module được sử dụng để thiết lập hiển thị, các thao tác (tắt, bấm chuột, phím...) và theo dõi thời gian.
 - Mixer: Là một module để tải và phát các trình âm thanh được truyền vào.
- Quá trình thực hiện được chia làm các phần:
 - Phần 1: Khởi tạo thế giới và nhân vật
 - Phần 2: Thêm các phím chức năng cho nhân vật. Thêm vào va chạm của nhân vật đối với các mặt phẳng theo phương x và y. Thêm vào hệ thống kẻ thù và dung nham
 - Phần 3: Đưa vào hệ thống kiểm tra va chạm với kẻ thù, tạo cử động cho kẻ thù qua phải qua trái theo nơi spawn. Thêm vào list ảnh nhân vật để có thể mô tả hành động đi lại của nhân vật.
 - Phần 4: Đưa vào list hình ảnh cho kẻ thù, dung nham cho game thêm sinh động. Thêm xu, tính năng nhặt xu và tính điểm. Thêm vào chức năng Win và màn hình khởi động game.
 - Phần 5: Thêm âm thanh. Tạo chức năng giết kẻ địch bằng cách nhảy lên đầu. Cập nhật lại map cho phù hợp với lối chơi.
- Các class được sử dụng trong dự án gồm: ***Class Player, Class World, Class Enemy, Class Lava, Class Coin, Class Button*** và ***Class Myster_Box***
- Các chức năng, tính năng chính trong game:

- Thiết lập keymap:

```
#Set key
key = pygame.key.get_pressed()
if key[pygame.K_q]:
    pygame.quit()
if key[pygame.K_r]:
    reset()
    score = 0
if key[pygame.K_UP] and self.jumped == False and self.floating == False:
    jump_fx.play()
    self.vel_y = -19
    self.jumped = True
if key[pygame.K_UP] == False:
    self.jumped = False
if key[pygame.K_LEFT]:
    self.direction = -1
    dx -= 6
    self.counter += 1
if key[pygame.K_RIGHT]:
    self.direction = 1
    dx += 6
    self.counter += 1
if key[pygame.K_LEFT] == False and key[pygame.K_RIGHT] == False:
    self.counter = 0
    self.index = 0
    if self.direction == 1:
        self.image = self.images_right[self.index]
    if self.direction == -1:
        self.image = self.images_left[self.index]
```

- Trọng lực:

```
#Thêm trọng lực
self.vel_y += 1
if self.vel_y > 10:
    self.vel_y = 20
dy += self.vel_y
```

- Va chạm của nhân vật với các object:

```
#Them va cham
self.floating = True
for tile in world.tile_list:
    # #Kiem tra va cham ngang
    if tile[1].colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
        dx = 0
    #kiem tra va cham doc
    if tile[1].colliderect(self.rect.x, self.rect.y +dy, self.width, self.height):
        #Check block cham dau khi nhay
        if self.vel_y < 0:
            dy = tile[1].bottom - self.rect.top
            self.vel_y = 0

        #Check cham dat khi nguoi choi roi xuong
    elif self.vel_y >= 0:
        dy = tile[1].top - self.rect.bottom
        self.vel_y = 0
        self.floating = False
```

- Va chạm với chướng ngại vật

```
else:
    if pygame.sprite.spritecollide(self, Goomba_group, False,pygame.sprite.collide_rect_ratio(0.8)):
        game_over = -1
        die_fx.play()
        pygame.mixer.music.stop()
    if pygame.sprite.spritecollide(self, Lava_Group, False):
        game_over = -1
        die_fx.play()
        pygame.mixer.music.stop()

    if pygame.sprite.spritecollide(self, Mystery_Group, False, pygame.sprite.collide_rect_ratio(0.7)):
        game_over = -2
        mystery_fx.play()
        pygame.mixer.music.stop()
```

- Nhảy lên đầu kẻ thù:

```
if self.floating == True:
    if pygame.sprite.spritecollide(self, Goomba_group, True):
        self.vel_y = -16
        self.jumped = True
        stomp_fx.play()
```

- Nhặt coin và hiển thị điểm:

```
#Check nhặt coin
if pygame.sprite.spritecollide(player, Coin_Group, True):
    score +=1
    coin_fx.play()
draw_text('x' +str(score), font_score, white, tile_size -10, 10)
```


- Win và Game Over:

```
#check trạng thái người chơi thì
if game_over == -1: #Chet
    if restart_button.draw():
        world = reset()
        game_over = 0
        score = 0
        victory_fx.stop()
        die_fx.stop()
        pygame.mixer.music.load('img/mario_theme.wav')
        pygame.mixer.music.play(-1 , 0.0, 500)
        pygame.mixer.music.set_volume(0.3)
if game_over == 1: #Win
    draw_text('YOU WIN!', font, blue, Width //2 - 110, Height //2 -100)

    if restart_button.draw():
        world = reset()
        game_over = 0
        score = 0
        victory_fx.stop()
        die_fx.stop()
        pygame.mixer.music.load('img/mario_theme.wav')
        pygame.mixer.music.play(-1 , 0.0, 500)
        pygame.mixer.music.set_volume(0.3)
if game_over == -2: #Curiosity
    draw_text('Why did you do that?', font, white, Width //2 - 280, Height //2 + 60)
    if restart_button.draw():
        world = reset()
        game_over = 0
        score = 0
        victory_fx.stop()
        die_fx.stop()
        mystery_fx.stop()
        pygame.mixer.music.load('img/mario_theme.wav')
        pygame.mixer.music.play(-1 , 0.0, 500)
        pygame.mixer.music.set_volume(0.3)
```

- Hệ thống map dựa theo tile:

```

world_data = [
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 0, 0, 0, 0, 5, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 5, 5, 0, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 5, 0, 0, 0, 5, 5, 6, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 2, 0, 0, 2, 5, 5, 0, 0, 0, 0, 3, 0, 0, 0, 0, 3, 0, 0, 0, 1, 1, 1, 2],
[2, 0, 0, 2, 0, 0, 0, 0, 2, 2, 2, 0, 0, 2, 2, 2, 0, 0, 2, 2, 2, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 7, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2],
[2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 5, 5, 0, 0, 0, 0, 0, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 5, 5, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 2],
[2, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 2, 2, 0, 0, 0, 5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 1, 1, 2],
[2, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 5, 5, 5, 0, 0, 0, 0, 1, 1, 1, 2],
[2, 0, 0, 0, 0, 1, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 5, 3, 5, 0, 0, 0, 1, 1, 1, 1, 2],
[1, 1, 1, 1, 1, 1, 1, 2, 4, 4, 4, 4, 4, 4, 4, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
]

```

○ Reset:

```

def reset():
    """ Ham reset sau khi game over se tra lai cac bien so va tai tao lai the gioi"""
    player.reset(100, Height - 170)
    Goomba_group.empty()
    Lava_Group.empty()
    Coin_Group.empty()
    Coin_Group.add(score_coin)
    score = 0

    world = World(world_data)
    return world

```

Chương 3: KẾT LUẬN

3.1 Ưu điểm

- Dự án đã đáp ứng đầy đủ yêu cầu của một game platformer.
- Sử dụng mô hình class hướng đối tượng.
- Có sound effect để game được sinh động hơn
- Tại các mục đều được ghi comment để định nghĩa cho từng mục và tác dụng của những mục đó

3.2 Nhược điểm

- Vì không có nhiều thời gian nên không thể tái tạo lại hoàn toàn map của Mario nên màn chơi chỉ gói gọn trong một màn hình
- Vẫn còn một số lỗi nhỏ như hoạt ảnh của dung nham không load, hoạt ảnh khi chết của nhân vật không được chạy khi bị chết bởi 2 Goomba ở giữa màn hình
- Hệ thống tính điểm còn chưa được tối ưu
- Hệ thống thiết kế map chẳng chịt khó nhìn
- Không có các class kế thừa

3.3 Kết luận

Pygame rất tốt để sử dụng cho việc tiếp cận lập trình game, với tính dễ dàng và tiện lợi của pygame, ta có thể tự học và làm một tựa game mong muốn của mình rất nhanh chóng, tạo thuận lợi cho những người mới muốn tiếp cận dần vào việc lập trình game.

TÀI LIỆU THAM KHẢO

[Pygame Front Page — pygame v2.0.1.dev1 documentation](#)

[Pygame Platformer - Game Development - CodersLegacy](#)

ĐƯỜNG DẪN ĐẾN DỰ ÁN

https://github.com/lordsobz/DoAn_Pygame_18520688.git