TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

A circular logo with a star and a blue star with a red and yellow circle

Description automatically generated with medium confidence

**BÀI TẬP LỚN**

**LẬP TRÌNH PYTHON**

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

**HỆ : ĐẠI HỌC CHÍNH QUY**

ĐỀ TÀI: SỬ DỤNG PYGAME XÂY DỰNG MỘT TỰA GAME PLATFORMER ĐƠN GIẢN

THÁI NGUYÊN - 2024

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ



**BÀI TẬP LỚN LẬP TRÌNH PYTHON**

**BỘ MÔN : CÔNG NGHỆ THÔNG TIN**

ĐỀ TÀI: SỬ DỤNG PYGAME XÂY DỰNG MỘT TỰA GAME PLATFORMER ĐƠN GIẢN

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn** | **:** | **TS. Nguyễn Tuấn Linh** |
| **Họ và tên sinh viên** | **:** | **La Đức Thắng** |
| **Ngành học** | **:** | **Kỹ thuật Máy tính** |
| **MSSV** | **:** | **K215480106120** |
| **Lớp** | **:** | **K57KMT.01** |

THÁI NGUYÊN - 2024

|  |  |
| --- | --- |
| TRƯỜNG ĐHKTCN | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM |
| KHOA ĐIỆN TỬ | Độc lập - Tự do - Hạnh phúc |

BÀI TẬP LỚN

LẬP TRÌNH PYTHON

**Sinh viên:** La Đức Thắng

**Lớp:** K57KMT **Khoá:** 2021-2025

**Bộ môn:** Công Nghệ Thông Tin

**Giáo viên hướng dẫn:** TS. Nguyễn Tuấn Linh

**Tên đề tài** : Xây dựng một tựa game platformer 2D đơn giản bằng Pygame.

Yêu cầu :

* Tạo nhân vật, vũ khí, kẻ thù
* Nhân vật có thể di chuyển, tấn công, có thanh HP
* Cuối mỗi chương có 1 màn boss

|  |  |  |
| --- | --- | --- |
| BCN KHOA | TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN |
| (Ký và ghi rõ họ tên) | (Ký và ghi rõ họ tên) | (Ký và ghi rõ họ tên) |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Xếp loại:……………… Điểm :……

Thái Nguyên, ngày….tháng…..năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 4](#_Toc192258276)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc192258277)

[LỜI NÓI ĐẦU 8](#_Toc192258278)

[LỜI CẢM ƠN 9](#_Toc192258279)

[CHƯƠNG I: KHÁI QUÁT VỀ BÀI TOÁN 10](#_Toc192258280)

[1.1. Đặt vấn đề 10](#_Toc192258281)

[1.2. Hướng giải quyết 10](#_Toc192258282)

[CHƯƠNG II: PHÂN TÍCH BÀI TOÁN 11](#_Toc192258283)

[2.1. Phân tích bài toán 11](#_Toc192258284)

[2.2. Phương pháp học máy sử dụng 11](#_Toc192258285)

[2.2.1. Phương pháp DeepLearning 11](#_Toc192258286)

[2.2.2. VGGFace2 12](#_Toc192258287)

[2.2.3. InceptionResNetV1 14](#_Toc192258288)

[CHƯƠNG III: HUẤN LUYỆN MÔ HÌNH 17](#_Toc192258289)

[3.1. Một số khái niệm trong huấn luyện mô hình 17](#_Toc192258290)

[3.1.1. Batch 17](#_Toc192258291)

[3.1.2. Epoch 17](#_Toc192258292)

[3.1.3. Weights 18](#_Toc192258293)

[3.1.4. Iteration 18](#_Toc192258294)

[3.1.5. Loss 19](#_Toc192258295)

[3.1.6. Accuracy 19](#_Toc192258296)

[3.1.7. Overfitting và underfitting 20](#_Toc192258297)

[3.2. Huấn luyện mô hình 21](#_Toc192258298)

[3.2.1. Thu thập dữ liệu 21](#_Toc192258299)

[3.2.2. Tiền xử lý dữ liệu 21](#_Toc192258300)

[3.2.3. Xây dựng dataset 22](#_Toc192258301)

[3.2.4. Tiến hành huấn luyện 24](#_Toc192258302)

[3.3. Đánh giá mô hình 25](#_Toc192258303)

[CHƯƠNG IV: XÂY DỰNG CHƯƠNG TRÌNH 28](#_Toc192258304)

[4.1. Thiết kế hệ thống 28](#_Toc192258305)

[4.1.1. Phân tích chương trình 28](#_Toc192258306)

[4.1.2. Phân tích hệ thống 29](#_Toc192258307)

[4.1.3. Cơ sở dữ liệu 29](#_Toc192258308)

[4.1.4. Giao diện chương trình 30](#_Toc192258309)

[4.2. Kiểm thử chương trình 31](#_Toc192258310)

[CHƯƠNG V: TỔNG KẾT 34](#_Toc192258311)

[5.1. Kết quả đã đạt được 34](#_Toc192258312)

[5.2. Hướng phát triển của đề tài 34](#_Toc192258313)

[KẾT LUẬN 35](#_Toc192258314)

[TÀI LIỆU THAM KHẢO 36](#_Toc192258315)

# DANH MỤC TỪ VIẾT TẮT

1. CSDL: Cơ sở dữ liệu.
2. CNTT: Công nghệ thông tin

LỜI NÓI ĐẦU

Ngày nay, ngành công nghiệp trò chơi điện tử (game) đang phát triển mạnh mẽ, trở thành một lĩnh vực không chỉ mang tính giải trí mà còn đóng vai trò quan trọng trong giáo dục, truyền thông và sáng tạo nội dung. Trong đó, thể loại game platformer 2D – nơi người chơi điều khiển nhân vật vượt qua chướng ngại vật, chiến đấu với kẻ địch và hoàn thành các nhiệm vụ – vẫn giữ được vị trí đặc biệt nhờ tính đơn giản, hấp dẫn và khả năng phát triển ý tưởng sáng tạo không giới hạn.

Đề tài này được thực hiện nhằm mục tiêu tìm hiểu quy trình xây dựng một trò chơi platformer 2D đơn giản bằng ngôn ngữ lập trình Python với thư viện Pygame. Thông qua việc xây dựng các chức năng như di chuyển nhân vật, nhảy, tương tác với nền tảng, tấn công kẻ địch và quản lý trạng thái của đối tượng trong trò chơi, người thực hiện đề tài có thể vận dụng và củng cố các kiến thức về lập trình hướng đối tượng, xử lý va chạm, điều khiển thời gian, và thiết kế trò chơi.

Ngoài ra, việc xây dựng một trò chơi hoàn chỉnh còn giúp nâng cao tư duy logic, khả năng thiết kế hệ thống và giải quyết vấn đề, từ đó góp phần phục vụ cho các định hướng nghề nghiệp liên quan đến công nghệ phần mềm, đặc biệt trong lĩnh vực phát triển trò chơi và ứng dụng tương tác.

Tuy đề tài còn nhiều điểm đơn giản và mang tính chất học thuật, nhưng đây là bước đầu cần thiết để tiếp cận các kỹ thuật và tư duy trong lĩnh vực lập trình game – một lĩnh vực hấp dẫn, giàu tiềm năng và đầy thử thách.

Trong quá trình thực hiện đề tài, em đã cố gắng tìm hiểu, nghiên cứu và áp dụng những kiến thức, kỹ năng đã học để xây dựng một tựa game hoàn chỉnh, có khả năng hoạt động ổn định. Tuy nhiên, do kiến thức và kinh nghiệm còn hạn chế, chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, đánh giá từ quý thầy cô và các bạn để đề tài được hoàn thiện hơn.

LỜI CẢM ƠN

Trong suốt quá trình học tập và thực hiện bài tập lớn, em đã nhận được sự giúp đỡ tận tình của thầy giáo TS. Nguyễn Tuấn Linh trong bộ môn Tin học công nghiệp – Khoa Điện tử - Trường Đại học Kỹ thuật Công Nghiệp – Đại học Thái Nguyên. Em bày tỏ lòng biết ơn thầy đã tận tình giúp đỡ, hướng dẫn em trong thời gian thực hiện đề tài này.

Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm thực tế còn ít, cho nên đề tài không thể tránh khỏi thiếu sót. Vì vậy, em rất mong nhận được sự đóng góp ý kiến của các thầy giáo, cô giáo và các bạn.

Em xin chân thành cảm ơn!

CHƯƠNG I: KHÁI QUÁT BÀI TOÁN

## Khái niệm game platformer

Platformer là thể loại trò chơi mà nhân vật chính di chuyển trên mặt phẳng 2 chiều, thường có khả năng:

* Di chuyển trái/phải.
* Nhảy và chịu ảnh hưởng bởi trọng lực.
* Tương tác với các vật thể như platform, kẻ địch, vật phẩm.

Một số tựa game platformer nổi tiếng có thể kể đến như: Super Mario Bros, Sonic The Hedgehog, Hollow Knight, Blasphemous,…

A screenshot of a video game

AI-generated content may be incorrect.

Hình 1.1: Game Blasphemous

## 1.2. Thư viện Pygame

Pygame là một thư viện Python mã nguồn mở được sử dụng để phát triển game 2D hoặc các ứng dụng đa phương tiện một cách dễ dàng và nhanh chóng.

Pygame cung cấp các module để làm việc với đồ họa 2D (hiển thị hình ảnh, vẽ đối tượng), Âm thanh (phát nhạc, hiệu ứng), sự kiện bàn phím, chuột, tay cầm (gamepad), vòng lặp game (game loop).

Ứng dụng:

* Tạo game 2D như: platformer, đua xe, bắn súng, giải đố,...
* Làm phần mềm mô phỏng đơn giản.
* Học lập trình game căn bản (logic, sự kiện, vật lý, xử lý va chạm,...).

## Mục tiêu của hệ thống

* Tạo một môi trường 2D có nền, platform và vật thể.
* Cho phép người chơi điều khiển nhân vật:
* Di chuyển, nhảy, tấn công.
* Quản lý trạng thái game: máu, cooldown, animation, va chạm, v.v.

CHƯƠNG II: PHÂN TÍCH BÀI TOÁN

## 2.1. Dữ liệu bài toán

Dữ liệu đầu vào:

* Bàn phím
* Sự kiện (Event Inputs)
* Tài nguyên game

Dữ liệu đầu ra:

* Hiển thị đồ họa (Nhân vật chính, kẻ địch, nền, platform, đạn, hiệu ứng, HUD animation khi di chuyển, nhảy, tấn công.)
* Thay đổi trạng thái game

## 2.2. Phân tích hệ thống

### 2.2.1. Các yêu cầu của hệ thống

Yêu cầu chức năng:

* Người chơi có thể điều khiển nhân vật di chuyển, tấn công. Nhân vật có thể nhận sát thương, game kết thúc khi nhân vật hết máu.
* Các thực thể có một hitbox cố định. Chỉ khi tấn công vào hitbox đó mới có thể gây sát thương
* Kẻ địch có thể di chuyển và bắn đạn, gây sát thương cho người chơi.
* Boss có nhiều máu hơn, phải đánh và né tránh nhiều đòn để thắng.
* Hiển thị HUD cho biết số màn chơi và số máu còn lại của người chơi.

Yêu cầu phi chức năng:

* Tách rõ các class: Player, Enemy, Boss, Platform, Bullet.
* Mỗi level có cấu trúc riêng (platforms + enemies) dễ mở rộng.
* Code dễ bảo trì, dễ thêm level mới, boss mới hoặc hiệu ứng.
* Dễ mở rộng với tính năng âm thanh, item, animation,…

### 2.2.2. Các thực thể tồn tại trong chương trình

Chương trình sẽ có các thực thể như sau:

* Player: Nhân vật người chơi, có thể di chuyển trái/phải, nhảy, tấn công bằng tay hoặc bắn súng nếu nhặt được Gun. Có animation đi, đứng, tấn công.
* Enemy: Kẻ địch di chuyển ngang trong khoảng 300-700, có thể nhảy, rơi xuống platform, bắn đạn về phía người chơi.
* Boss: Phiên bản mạnh hơn Enemy, có nhiều pattern tấn công: di chuyển, bắn đạn 8 hướng, nhảy.
* Platform: Nền tảng để người chơi và kẻ địch đứng, nhảy lên.
* Bullet và BossBullet: Đạn của người chơi và boss với tốc độ, hướng riêng.

Các thực thể này sẽ được khởi tạo bằng các class trong file entities.py.

CHƯƠNG III: THIẾT KẾ CHƯƠNG TRÌNH

## 3.1. Texture

### 3.1.1. Platform

Để hiển thị các đối tượng trên màn hình game, chúng ta cần có texture. Đối với các đối tượng như đơn giản như Platform, ta có thể dùng Pygame để khởi tạo.

class Platform:

def \_\_init\_\_(self, x, y, w, h):

self.rect = pygame.Rect(x, y, w, h)

def draw(self, screen):

pygame.draw.rect(screen, (100, 100, 100), self.rect)

def get\_hitbox(self):

return self.rect

Khi tạo màn chơi, các Platform sẽ được đặt ở các vị trí chỉ định. Ví dụ: [Platform(200, 450, 200, 20) sẽ tạo ra một Platform có diện tích 200\*20 (h\*w) ở vị trí x=200, y=450.

### 3.1.2. Player và Enemy

Các texture cho Player và Enemy sẽ được lưu vào thư mục "textures".

A screenshot of a video game

AI-generated content may be incorrect.

Hình 3.1: Thư mục lưu trữ các texture của game

Player sẽ có animation mỗi khi di chuyển hoặc tấn công. Enemy sẽ chỉ di chuyển và bắn đạn đơn giản.

def load\_image(name):

path = os.path.join("textures", name)

return pygame.transform.scale(pygame.image.load(path), (64, 64))

Hàm trên là hàm load\_image để tự động thêm tên thư mục "textures" vào trước các path. Hàm này sẽ được sử dụng nhiều trong code nhằm rút gọn code.

self.idle\_img = load\_image("char.png")

self.idle\_img\_left = pygame.transform.flip(self.idle\_img, True, False)

self.move\_images\_right = [load\_image(f"char\_move\_{i}.png") for i in range(1, 4)]

self.move\_images\_left = [pygame.transform.flip(img, True, False) for img in self.move\_images\_right]

self.attack\_images\_right = [load\_image("char\_atk\_1.png"), load\_image("char\_atk\_2.png")]

self.attack\_images\_left = [pygame.transform.flip(img, True, False) for img in self.attack\_images\_right]

Các hình ảnh sẽ được tải vào chương trình như trên.

## 3.2. Khởi tạo đối tượng

### 3.2.1. Platform

Platform được khởi tạo như đã nêu trong phần 3.1. Tuy nhiên, để chương trình nhận biết được nhân vật có thể đứng trên platform, ta cần thiết lập một số thứ.

Trước tiên, ta cần lấy hitbox của đối tượng.

def get\_hitbox(self):

return pygame.Rect(self.x, self.y, self.width, self.height)

Sử dụng hàm get\_hitbox này, ta sẽ kiểm tra va chạm giữa nhân vật và Platform:

on\_platform = False

rect = self.get\_hitbox()

for plat in platforms:

if rect.colliderect(plat.rect):

if self.vel\_y >= 0 and rect.bottom - self.vel\_y <= plat.rect.top:

self.y = plat.rect.top - self.height

self.vel\_y = 0

self.is\_jumping = False

on\_platform = True

Code trên kiểm tra va chạm giữa nhân vật và Platform. Khi phần đáy của hitbox tiếp xúc với Platform, vị trí y của nhân vật sẽ thay đổi thành ở ngay trên Platform đó.

### 3.2.2. Bullet

Bullet là những viên đạn mà kẻ thù sẽ bắn ra nhằm gây sát thương cho người chơi. Tùy thuộc vào đối tượng thì Bullet sẽ khác nhau.

* Nếu là kẻ thù bình thường, đạn sẽ bay theo đường thẳng.
* Nếu là boss, đạn sẽ bắn ra nhiều phía.

### 3.2.3. Character

Character là một lớp chung được kế thừa bởi 2 lớp Player và Enemy. Nó lưu trữ các thuộc tính và phương thức chung của cả 2 lớp.

Các thuộc tính chung:

* x, y: Vị trí của nhân vật
* width: Chiều rộng của nhân vật
* height: Chiều cao của nhân vật
* vel\_y: Vận tốc theo trục y (nhảy, rơi). Trong game 2D, trục Y hướng xuống (ngược với vật lý thông thường), nên càng tăng y thì nhân vật càng rơi xuống.
* gravity: Trọng lực để kéo nhân vật lại, nếu không nhân vật sẽ bị nhảy mà không có điểm rơi.
* is\_jumping: Trạng thái nhảy (True/False)
* alive: Trạng thái sống/chết

Các phương thức chung:

* get\_hitbox: Lấy hitbox của nhân vật
* apply\_gravity\_and\_platform\_collision: Dùng để áp dụng trọng lực và kiểm tra va chạm với các platform hoặc mặt đất.

### 3.2.3. Player

Player sẽ có những đặc tính như sau:

* Có thể di chuyển trái phải và nhảy
* Có thể tấn công. Tấn công có thời gian hồi giữa các đòn.
* Khi bị dính đòn, trừ 1 máu. Game kết thúc khi Player hết máu.
* Sau khi bị dính đòn sẽ được bất tử trong 1 giây sau đó để chuẩn bị lại.

#### 3.2.3.1. Di chuyển

Các thuộc tính liên quan đến di chuyển:

* speed: Tốc độ di chuyển mỗi frame.
* facing\_right: Trạng thái quay mặt sang phải (Phục vụ cho animation)
* idle\_img, move\_images: Các hình ảnh dùng cho hoạt ảnh di chuyển.
* anim\_index, anim\_timer, anim\_speed: Điều khiển hoạt ảnh di chuyển

Mặc định Player quay mặt sang phải. Khi ấn nút mũi tên chỉ trái trên bàn phím, Player sẽ di chuyển ngược lại, đồng thời quay mặt về phía bên trái bằng cách lật ngược lại các ảnh texture.



Hình 3.2: Player di chuyển sang trái và sang phải

Hoạt ảnh di chuyển:

if moving:

self.anim\_timer += 1

if self.anim\_timer >= self.anim\_speed:

self.anim\_timer = 0

self.anim\_index = (self.anim\_index + 1) % len(self.move\_images\_right)

self.current\_img = (

self.move\_images\_right[self.anim\_index]

if self.facing\_right

else self.move\_images\_left[self.anim\_index]

)

#### 3.2.3.2. Nhảy

Các thuộc tính liên quan đến nhảy:

* jump\_power: Sức bật. Là vận tốc âm ban đầu khi nhấn nhảy, giúp nhân vật bật lên cao

Công thức tổng quát cho mô phỏng nhảy:

​

Code thực thi chức năng nhảy:

if keys[pygame.K\_UP] and not self.is\_jumping:

self.is\_jumping = True

self.vel\_y = -self.jump\_power

Khi ấn nút arrow up (nút mũi tên chỉ trên). Nhân vật sẽ thay đổi vị trí theo từng frame. Khi đạt đến độ cao cực đại, nhân vật sẽ từ từ rơi xuống theo từng frame.

#### 3.2.3.3. Tấn công

Các thuộc tính liên quan tới tấn công:

* is\_attacking, attack\_duration, attack\_timer, attack\_cooldown, attack\_ready, attack\_key\_released: Các thuộc tính liên quan đến thực thi tấn công.
* attack\_images\_right, attack\_images\_left, attack\_anim\_index, attack\_anim\_timer, attack\_anim\_speed: Các thuộc tính liên quan đến hoạt ảnh tấn công.

Xử lý tấn công: Khi nhấn Z, nếu nhân vật đã sẵn sàng (attack\_ready = True) và chưa tấn công, bắt đầu đòn đánh:

* Đặt trạng thái is\_attacking = True.
* Đặt thời gian đánh (attack\_timer).
* Đặt attack\_ready = False để không đánh liên tục.
* Đặt attack\_cooldown = 10 (chờ 10 khung hình mới được đánh tiếp).
* Nếu người chơi thả phím Z, gán attack\_key\_released = True để cho phép bấm lại ở lần sau, tránh trường hợp giữ phím để spam đòn tấn công.
* Nếu người chơi ấn lại phím Z sau khi đã thả ra và đã hết cooldown: Bắt đầu lại tấn công như cũ. Reset attack\_key\_released = False để chờ thả ra lần nữa.
* Mỗi khung hình, nếu attack\_cooldown > 0 thì giảm đi 1 cho đến khi về 0. Cho phép tấn công lại khi cooldown đã hết.

### 3.2.4. Enemy

Về cơ bản, Enemy vẫn giống như Player vì cùng kế thừa từ class. Tuy nhiên các hành động của đối tượng này sẽ do chương trình điều khiển.

Di chuyển:

* Di chuyển enemy theo hướng direction (1 hoặc -1) và tốc độ speed.
* Nếu enemy đi quá bên trái (x < 300): quay sang phải.
* Nếu enemy đi quá bên phải (x > 700 - width): quay sang trái.

Enemy sẽ tấn công bằng đòn bắn ra đạn. Có các thuộc tính giống với các thuộc tính tấn công của Player, khác ở chỗ là tầm đánh của Enemy sẽ bao quát toàn bản đồ.

def shoot(self):

direction = self.direction

bullet\_x = self.x + (self.width if direction == 1 else -16)

bullet\_y = self.y + self.height // 2

self.bullets.append(Bullet(bullet\_x, bullet\_y, direction))

Code trên sẽ xác định hướng bắn (direction) của đạn. Vị trí xuất phát của viên đạn như sau:

* Nếu enemy quay phải → bắn từ bên phải self.x + width.
* Nếu quay trái → bắn từ bên trái self.x - 16.
* Thêm viên đạn mới vào danh sách.

Trong trường hợp này, Enemy bắn ra đạn theo đường thẳng.

### 3.2.5. Boss

Vì boss cũng là một kiểu kẻ thù, vậy nên lớp Boss sẽ kế thừa lớp Enemy. Tuy nhiên, Boss là một kẻ thù mạnh hơn nên sẽ phải sửa đổi sao cho khác biệt và mạnh hơn so với kẻ thù bình thường.

Đầu tiên, boss sẽ có nhiều máu hơn, trong bài toán này là 100 máu. Hitbox của boss cũng sẽ to hơn kẻ thù thường.

self.width = 96

self.height = 96

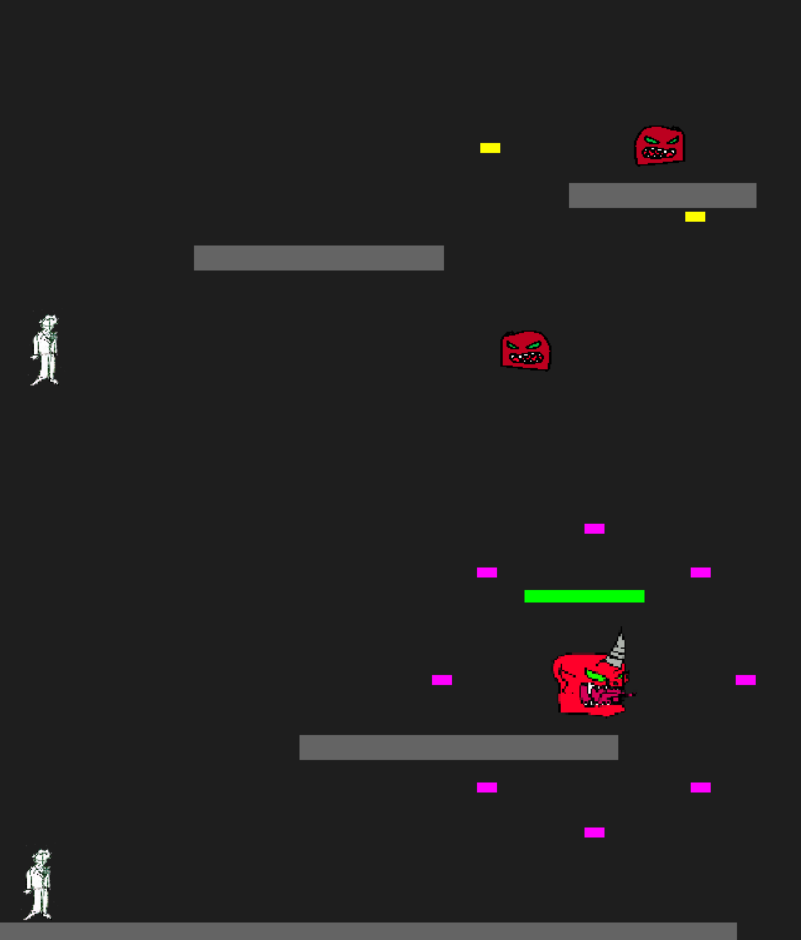
self.health = 100

self.max\_health = 100

Boss sẽ có nhiều pattern di chuyển hơn.

* Di chuyển
* Nhảy
* Bắn đạn

Thuật toán xử lý di chuyển và nhảy sẽ tương tự như đối với Player và Enemy, chỉ riêng xử lý đạn bắn ra sẽ khác một chút. Ở đây em sẽ cho boss bắn đạn ra nhiều hướng.



Hình 3.3: Đường đạn của kẻ thù thường và đường đạn của boss

Phương trình chuyển động của đường đạn boss:

Trong đó:

* là tọa độ trung tâm của boss
* là góc bắn
* là bận tốc ban đầu
* là thời điểm từ lúc viên đạn được bắn ra (trong trường hợp này thường sẽ tính bằng frame)

CHƯƠNG IV: XÂY DỰNG CHƯƠNG TRÌNH

## 4.1. Xây dựng giao diện

Sử dụng Pygame, ta sẽ thiết kế một giao diện cho trò chơi.

Đầu tiên là tạo 1 cửa sổ.

pygame.init()

WIDTH, HEIGHT = 800, 600

screen = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("Platformer Game")

clock = pygame.time.Clock()

Tạo menu chính:

if in\_menu:

screen.blit(load\_image(f"textures/title.jpg"), (0, 0)) # Dùng title.jpg làm nền menu

draw\_text("Press ENTER to start", WIDTH // 2, 550, (255, 0, 0), center=True)

pygame.display.flip()

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

elif event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN:

in\_menu = False

continue

Các hình nền kết thúc game (thắng và thua)

if game\_over:

screen.fill((0, 0, 0))

draw\_text("YOU LOST!", WIDTH // 2, HEIGHT // 2 - 40, (255, 0, 0), center=True, font\_obj=big\_font)

draw\_text("Press ESC to quit", WIDTH // 2, HEIGHT // 2 + 20, (255, 255, 255), center=True)

pygame.display.flip()

if keys[pygame.K\_ESCAPE]:

pygame.quit()

sys.exit()

continue

if game\_win:

screen.fill((0, 0, 0))

draw\_text("YOU WON!", WIDTH // 2, HEIGHT // 2 - 40, (0, 255, 0), center=True, font\_obj=big\_font)

draw\_text("Press ESC to quit", WIDTH // 2, HEIGHT // 2 + 20, (255, 255, 255), center=True)

pygame.display.flip()

if keys[pygame.K\_ESCAPE]:

pygame.quit()

sys.exit()

continue

## 4.2. Kiểm thử chương trình

A screenshot of a video game

AI-generated content may be incorrect.

Hình 4.1: Level 1

A screenshot of a video game

AI-generated content may be incorrect.

Hình 4.2: Nhân vật di chuyển

A screenshot of a video game

AI-generated content may be incorrect.

Hình 4.3: Màn boss

A screen shot of a computer

AI-generated content may be incorrect.

Hình 4.4: Màn hình chiến thắng game

Link GitHub của chương trình:

A qr code with a dinosaur

AI-generated content may be incorrect.

CHƯƠNG V: TỔNG KẾT

## 5.1. Kết quả đã đạt được

Các kết quả đã đạt được:

* Đã tạo được những tính năng cơ bản nhất của một tựa game platformer.
* Tạo được giao diện tương tác với người chơi.

## 5.2. Hướng phát triển của đề tài

Trong tương lai, em dự kiến sẽ có một số hướng phát triển như sau:

* Cải thiện chất lượng các mô hình nhân vật.
* Thiết kế màn chơi hoàn chỉnh hơn.
* Phát triển lên một trò chơi hoàn chỉnh bao gồm các nâng cấp, đa dạng nhân vật, vũ khí, kẻ thù,... và có một cốt truyện rõ ràng.

Link Github của đề tài:

A qr code with a dinosaur

AI-generated content may be incorrect.

KẾT LUẬN

Qua quá trình thực hiện đề tài xây dựng trò chơi dạng platformer sử dụng thư viện Pygame, em đã xây dựng được một trò chơi đơn giản nhưng có chiều sâu trong gameplay với các yếu tố hành động, né tránh và chiến đấu. Tuy nhiên do trình độ và kiến thức còn hạn chế nên đề tài của em vẫn chưa thể đáp ứng được nhu cầu thực tế, còn nhiều những thiếu sót hạn chế nhất định. Nhưng qua đó em đã rút ra rất nhiều kinh nghiệm cho bản thân.

Em xin chân thành cảm ơn thầy Nguyễn Tuấn Linh đã giúp em hoàn thành đề tài này.

TÀI LIỆU THAM KHẢO

<https://chatgpt.com>