

BÁO CÁO ĐỒ ÁN MÔN HỌC

Lớp: IE221. O21.CNCL

SINH VIÊN THỰC HIỆN:

Mã sinh viên: 21522885

Họ và tên: Phan Thị Cát Tường

TÊN ĐỀ TÀI: Xây dựng ứng dụng game vượt chướng ngại vật (Adventure Runner).

CÁC NỘI DUNG CẦN BÁO CÁO:

1. Giới thiệu đồ án

1. Giới thiệu đồ án:

Đồ án này tập trung vào việc xây dựng một ứng dụng game có tên gọi là "Adventure Runner" (Người chơi phiêu lưu). Đây là một game platformer được lấy cảm hứng từ trò chơi kinh điển Mario.

Game thuộc thể loại vượt chướng ngại vật, nơi người chơi sẽ phải điều khiển một nhân vật vượt qua các thử thách di chuyển từ điểm xuất phát đến điểm đích của mỗi màn chơi để dành chiến thắng. Không chỉ đơn giản là vượt qua, mà còn phải tránh né các chướng ngại vật đa dạng từ chướng ngại vật động đến chướng ngại vật tĩnh.

Điểm đặc biệt là bạn còn có thể thu thập các vật phẩm trên đường để tăng điểm số và cải thiện khả năng của nhân vật. Người chơi được cấp một thanh máu để vượt qua mỗi màn chơi. Mỗi màn được xây dựng theo mức độ khó tăng dần như là nhiều vật cản hơn và bản đồ game khó đi hơn. Đồng thời người chơi sẽ được lựa chọn nhân vật và các phím di chuyển theo sở thích của người chơi. Game sẽ chạy cho đến khi người chơi hoàn thành mọi màn chơi hoặc mất hết mạng sống hoặc người chơi bấm thoát.

2. Quá trình thực hiện

a. Tuần 1:

- Tìm hiểu về cách hoạt động của trò chơi platformer, xác định các tính năng và yêu cầu cụ thể của trò chơi. Lập kế hoạch cho quá trình phát triển.

- Phác thảo ý tưởng và giao diện trò chơi, bao gồm bố cục cơ bản và các yếu tố giao diện chính.
- Thiết kế sơ đồ lớp cơ bản trong trò chơi, bao gồm nhân vật, vật phẩm, chương ngại vật ...
- Tìm hiểu về thư viện để xây dựng trò chơi.

b. Tuần 2:

- Bắt đầu viết code để tạo ra cơ sở của trò chơi, bao gồm việc tạo ra màn hình chính, các màn chơi và giao diện người dùng cơ bản.
- Thêm nhân vật chính và các chương ngại vật vào trò chơi. Xử lý di chuyển của nhân vật và phản ứng khi va chạm với các chương ngại vật.

c. Tuần 3:

- Thực hiện hệ thống điểm số và thanh máu cho nhân vật, bao gồm việc cập nhật điểm số khi thu thập vật phẩm và quản lý thanh máu.
- Bổ sung các tính năng như âm thanh, lựa chọn nhân vật và lựa chọn phím di chuyển trên bàn phím.
- Kiểm tra trò chơi để phát hiện và sửa các lỗi hoặc vấn đề kỹ thuật. Đảm bảo rằng trò chơi hoạt động mượt mà và không có lỗi.

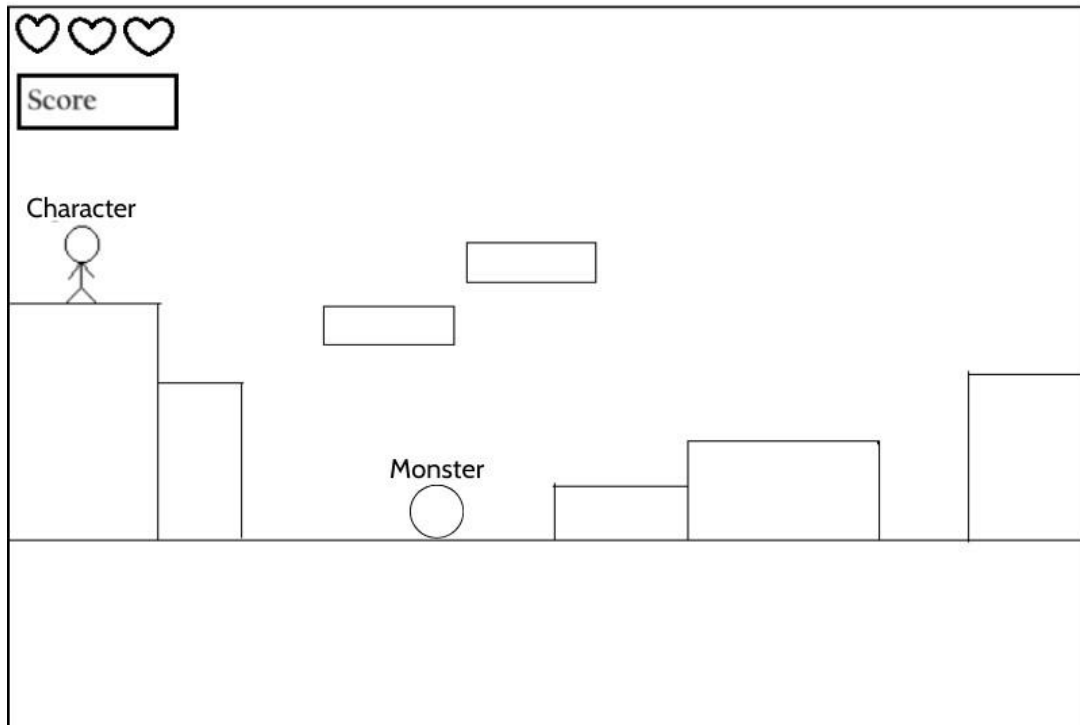
d. Tuần 4:

- Thử nghiệm trò chơi để thu thập ý kiến và phản hồi. Ghi nhận các lỗi và ý kiến đóng góp để sửa đổi và cải thiện trò chơi.
- Hoàn thiện viết báo cáo về quá trình phát triển trò chơi và làm slide thuyết trình.

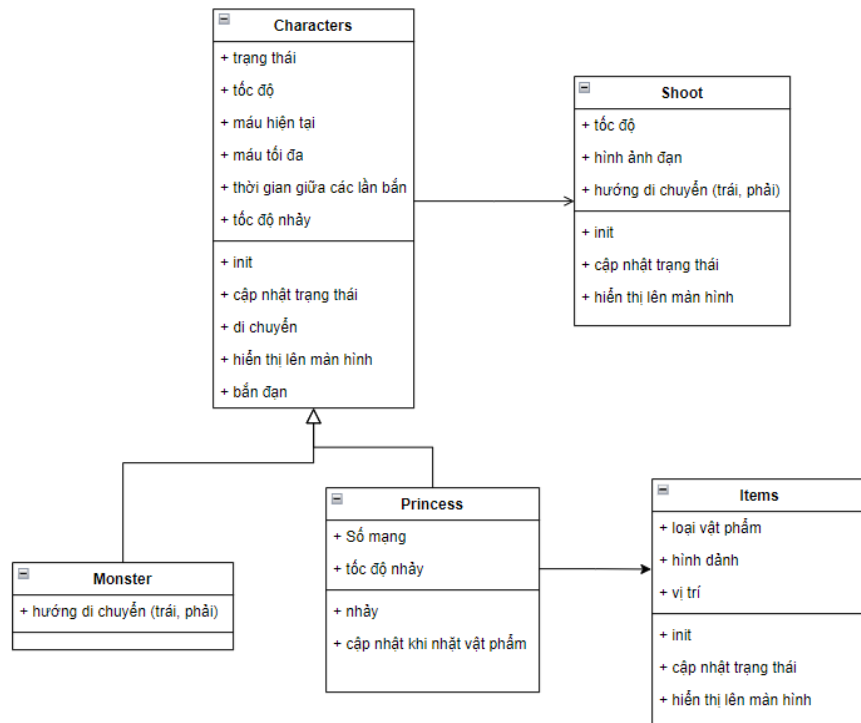
3. Kết quả đạt được

a. Tuần 1:

- Phác thảo giao diện cơ bản (dự kiến):



- Sơ đồ lớp cơ bản của game (dự kiến):



- Tìm hiểu về thư viện để xây dựng trò chơi.

- + Pygame

- + Sys

- + Math

- + OS

b. Tuần 2:

- Viết code tạo ra màn hình chính và giao diện người dùng cơ bản.
- Thêm nhân vật chính và các chướng ngại vật vào trò chơi. Xử lý di chuyển của nhân vật và phản ứng khi va chạm với các chướng ngại vật.
- Xây dựng nhân vật và quái vật có thể bắn đạn, khi nhân vật đến gần thì quái vật sẽ tự động tấn công, quái vật bị tiêu diệt khi bị bắn đạn. Khi nhân vật bị quái tấn công sẽ bị giảm thanh máu, khi thanh máu về 0 thì thua cuộc
- Xây dựng các màn chơi với địa hình khác nhau.
- Xây dựng các vật phẩm trên đường, khi nhặt có thể tăng máu hoặc tăng số lượng đạn.
- **Link github:** <https://github.com/ctuongphan/Game-Adventure-Runner>

c. Tuần 3:

- Tạo ra menu cho màn hình chính chi tiết hơn: thêm menu có 2 mục cho phép thay đổi nhân vật và bàn phím.
- Xây dựng tính năng âm thanh cho nhân vật lúc di chuyển, chiến đấu.
- Xây dựng thành phần điểm thu thập được khi tiêu diệt quái.
- Chỉnh sửa chi tiết map trò chơi.
- Kiểm tra lại trò chơi và chỉnh sửa.

d. Tuần 4:

- Thử nghiệm trò chơi, thu thập ý kiến và phản hồi từ mọi người xung quanh.
- Ghi nhận các lỗi và ý kiến đóng góp, sửa đổi và cải thiện trò chơi.

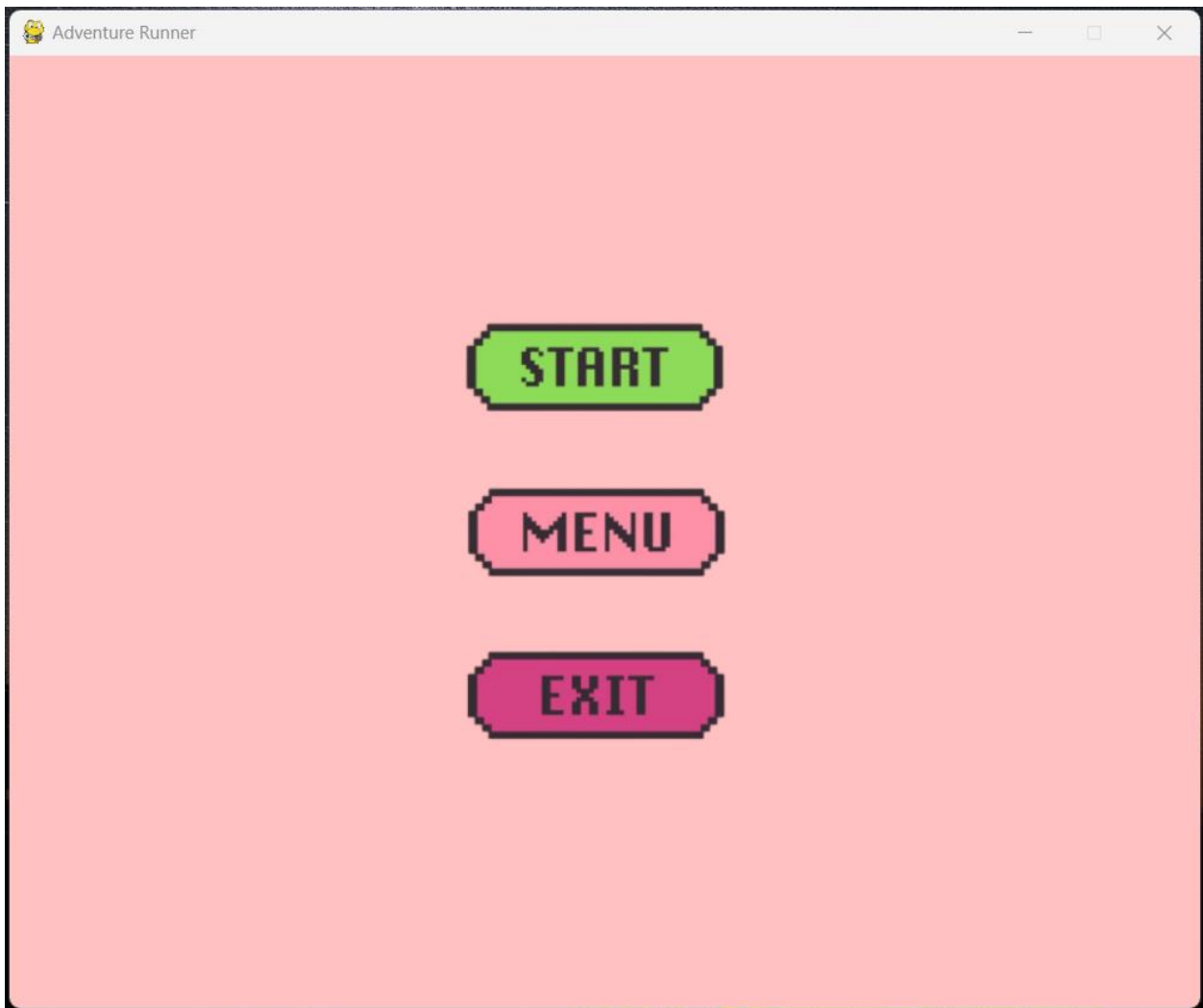
- Viết báo cáo về quá trình phát triển trò chơi và làm slide thuyết trình về những gì ban đầu muốn làm và bây giờ làm được những gì.

KẾT QUẢ ĐẠT ĐƯỢC:

Sau đây là thành quả mà em đã xây dựng một ứng dụng game.

Giao diện home của game:

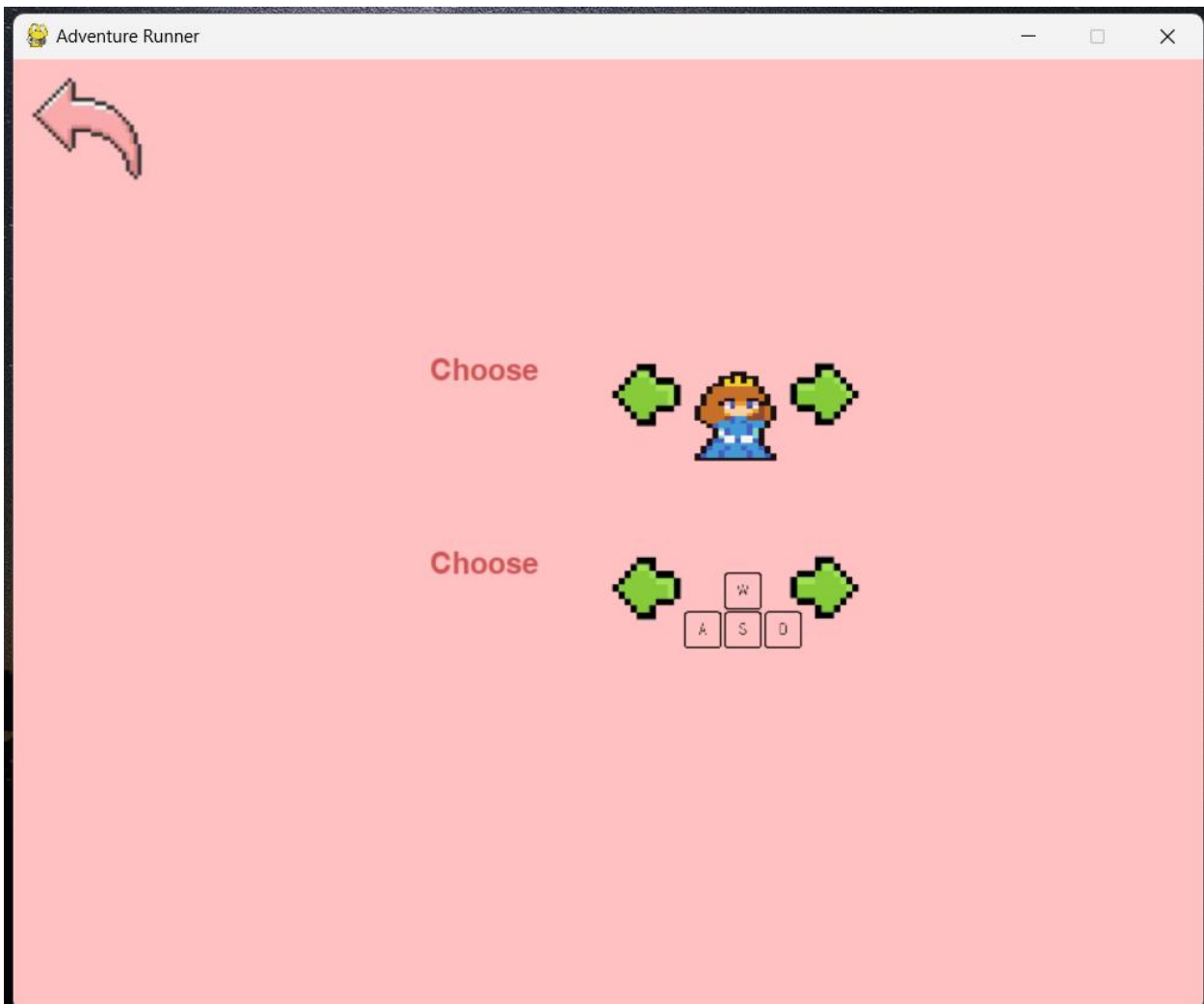
- Nhấn “Start” để bắt đầu chơi.
- Nhấn “Menu” để có thể lựa chọn nhân vật khác và các phím di chuyển khác theo sở thích.
- Nhấn “Exit” để thoát game.



Hình 1. Màn hình chính

Giao diện “Menu” của game:

- Lựa chọn nhân vật khác và các phím di chuyển khác theo ý muốn.



Hình 2. Màn hình Menu

Giao diện khi bắt đầu chơi game:

- Sử dụng các phím trên bàn phím để di chuyển nhân vật vượt qua các chướng ngại vật, chạm vật phẩm để sử dụng được chức năng của vật phẩm.

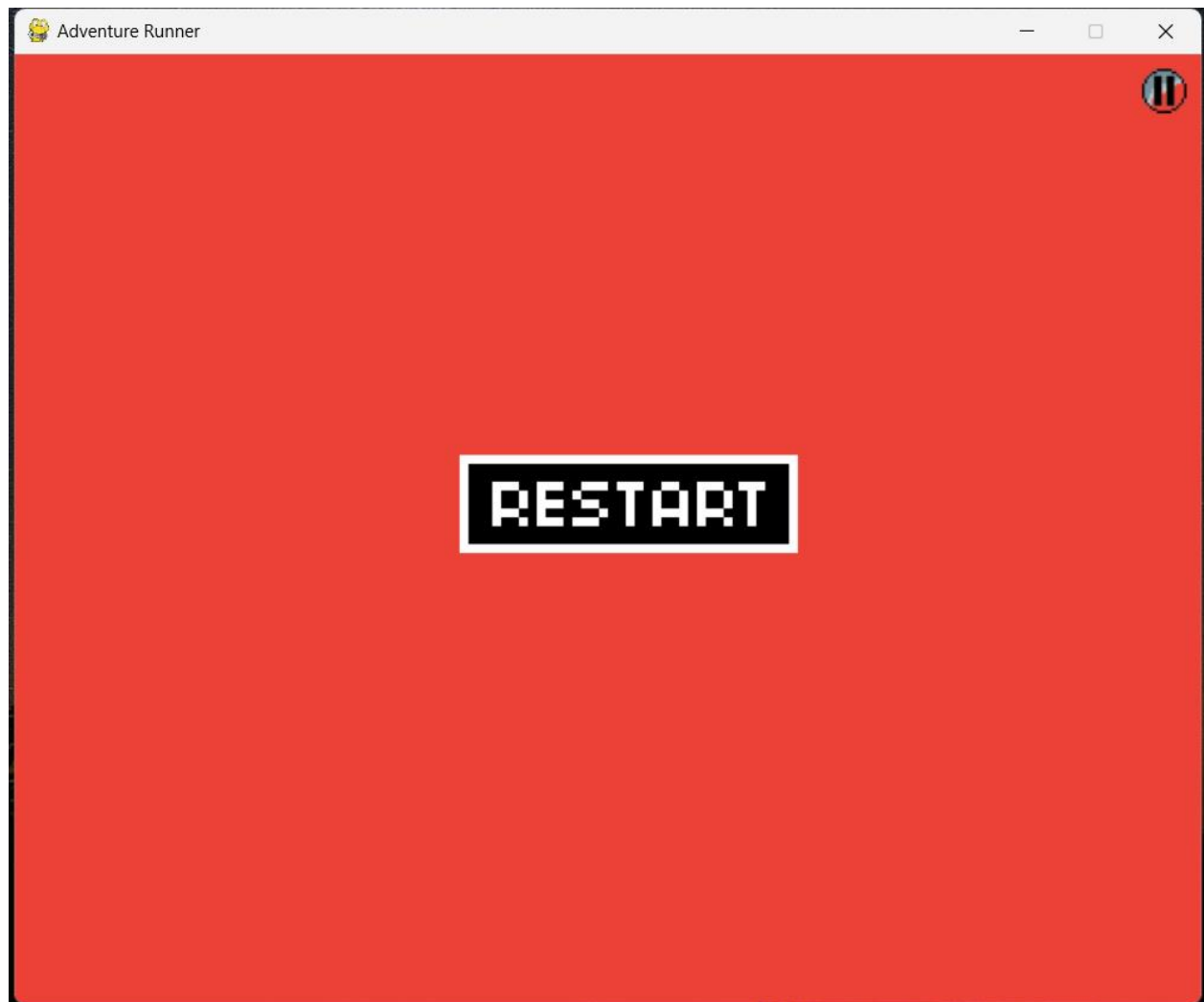
- Có thanh máu, thanh đạn và số điểm của nhân vật.



Hình 3. Màn hình bắt đầu chơi

Giao diện khi thua:

- Hiện lên màn hình có nút “Restart” để có thể chơi lại từ đầu.



Hình 4. Màn hình khi thua

Giao diện khi thắng:



Hình 5. Màn hình khi thắng

4. Tài liệu tham khảo

- Tìm hiểu về pygame: <https://www.geeksforgeeks.org/pygame-tutorial/>
- Install và sử dụng pygame: <https://www.pygame.org/docs/>
- Sử dụng thư viện OS: <https://viettuts.vn/python/mo-dun-os-trong-python>
- Tìm hiểu về CSV trong python: <https://realpython.com/python-csv/>
- Sử dụng các hình ảnh pixel cho game: <https://opengameart.org/>

5. Phụ lục 1: Giới thiệu (demo) kết quả

Em đã demo ứng dụng game và tải video demo lên drive.

Dưới đây là link dẫn tới video.

<https://drive.google.com/drive/folders/1PFqzrpHSSdvrqgXhYHf5HHDLGmtR6X5c>

6. Phụ lục 2: docstring

Class Princess:

Đại diện cho một nhân vật trong trò chơi.

Tham số:

char_type (str): Loại nhân vật ('player' hoặc 'Monster').

x (int): Tọa độ X của vị trí bắt đầu của nhân vật.

y (int): Tọa độ Y của vị trí bắt đầu của nhân vật.

scale (float): Hệ số thu phóng cho kích thước của nhân vật.

speed (int): Tốc độ di chuyển của nhân vật.

ammo (int): Số lượng đạn của nhân vật.

grenades (int): Số lượng lựu đạn của nhân vật.

index (int, tùy chọn): Chỉ số của nhân vật, mặc định là 1.

Thuộc tính:

alive (bool): Cho biết liệu nhân vật còn sống hay không.

char_type (str): Loại nhân vật ('player' hoặc 'Monster').

speed (int): Tốc độ di chuyển của nhân vật.

ammo (int): Số lượng đạn của nhân vật.

start_ammo (int): Số đạn ban đầu.

shoot_cooldown (int): Thời gian nghỉ sau mỗi lần bắn.

grenades (int): Số lượng lựu đạn của nhân vật.

health (int): Sức khỏe hiện tại của nhân vật.

max_health (int): Sức khỏe tối đa của nhân vật.

direction (int): Hướng mà nhân vật đang nhìn (-1 cho trái, 1 cho phải).

vel_y (int): Vận tốc theo chiều dọc của nhân vật.

jump (bool): Cho biết liệu nhân vật đang nhảy hay không.

in_air (bool): Cho biết liệu nhân vật có đang ở trạng thái nảy hay không.

flip (bool): Cho biết liệu hình ảnh của nhân vật có bị lật ngược hay không.

`animation_list` (list): Danh sách chứa các khung hình của hoạt hình của nhân vật.
`frame_index` (int): Chỉ số của khung hình hiện tại trong danh sách hoạt hình.
`action` (int): Hành động hiện tại của nhân vật (0 cho đứng yên, 1 cho chạy, 2 cho nhảy, 3 cho chết).
`update_time` (int): Thời gian cập nhật hoạt hình lần cuối.
`move_counter` (int): Bộ đếm cho việc di chuyển của nhân vật.
`vision` (pygame.Rect): Hình chữ nhật đại diện cho phạm vi tầm nhìn của nhân vật.
`idling` (bool): Cho biết liệu nhân vật có đang đứng yên hay không.
`idling_counter` (int): Bộ đếm cho thời gian nhân vật đứng yên.
`index` (int): Chỉ số của nhân vật.
`vel_yy` (int): Bộ điều chỉnh vận tốc theo chiều dọc cho việc nhảy.

Phương thức:

`update(self)`: Cập nhật trạng thái và hoạt hình của nhân vật.
`move(self, moving_left, moving_right)`: Di chuyển nhân vật theo chiều ngang và xử lý việc nhảy.
`shoot(self)`: Bắn đạn nếu điều kiện đủ.
`ai(self)`: Xử lý hành vi trí tuệ nhân tạo cho nhân vật.
`update_animation(self)`: Cập nhật các khung hình hoạt hình của nhân vật.
`update_action(self, new_action)`: Cập nhật hành động của nhân vật.
`check_alive(self)`: Kiểm tra xem nhân vật còn sống hay không.
`draw(self)`: Vẽ nhân vật lên màn hình.

Class Water:

Đại diện cho một vùng nước trong trò chơi.

Tham số:

`img` (pygame.Surface): Hình ảnh đại diện cho vùng nước.
`x` (int): Tọa độ X của vị trí của vùng nước.

y (int): Tọa độ Y của vị trí của vùng nước.

Thuộc tính:

image (pygame.Surface): Hình ảnh của vùng nước.

rect (pygame.Rect): Hình chữ nhật giới hạn vùng nước trên màn hình.

Phương thức:

update(self): Cập nhật vị trí của vùng nước dựa trên sự cuộn màn hình.

Class World:

Đại diện cho thế giới trong trò chơi.

Thuộc tính:

obstacle_list (list): Danh sách chứa các vật cản trong thế giới.

Phương thức:

__init__(self): Khởi tạo một thế giới mới.

process_data(self, data, index): Xử lý dữ liệu cấp độ để tạo ra các đối tượng trong thế giới.

draw(self): Vẽ các đối tượng trong thế giới lên màn hình.

Class Decoration:

Đại diện cho một đối tượng trang trí trong trò chơi.

Tham số:

img (pygame.Surface): Hình ảnh đại diện cho đối tượng trang trí.

x (int): Tọa độ X của vị trí của đối tượng trang trí.

y (int): Tọa độ Y của vị trí của đối tượng trang trí.

Thuộc tính:

image (pygame.Surface): Hình ảnh của đối tượng trang trí.

rect (pygame.Rect): Hình chữ nhật giới hạn đối tượng trang trí trên màn hình.

Phương thức:

update(self): Cập nhật vị trí của đối tượng trang trí dựa trên sự cuộn màn hình.

Class Exit:

Đại diện cho một đối tượng cổng thoát trong trò chơi.

Tham số:

img (pygame.Surface): Hình ảnh đại diện cho đối tượng cổng thoát.

x (int): Tọa độ X của vị trí của đối tượng cổng thoát.

y (int): Tọa độ Y của vị trí của đối tượng cổng thoát.

Thuộc tính:

image (pygame.Surface): Hình ảnh của đối tượng cổng thoát.

rect (pygame.Rect): Hình chữ nhật giới hạn đối tượng cổng thoát trên màn hình.

Phương thức:

update(self): Cập nhật vị trí của đối tượng cổng thoát dựa trên sự cuộn màn hình.

Class ItemBox:

Đại diện cho một hộp vật phẩm trong trò chơi.

Tham số:

item_type (str): Loại vật phẩm trong hộp ('Health', 'Ammo', 'Grenade', 'Shoe').

x (int): Tọa độ X của vị trí của hộp vật phẩm.

y (int): Tọa độ Y của vị trí của hộp vật phẩm.

Thuộc tính:

item_type (str): Loại vật phẩm trong hộp ('Health', 'Ammo', 'Grenade', 'Shoe').

image (pygame.Surface): Hình ảnh của hộp vật phẩm.

rect (pygame.Rect): Hình chữ nhật giới hạn hộp vật phẩm trên màn hình.

Phương thức:

update(self): Cập nhật vị trí của hộp vật phẩm dựa trên sự cuộn màn hình và xử lý khi người chơi nhận được hộp vật phẩm.

Class HealthBar:

Đại diện cho thanh máu trong trò chơi.

Tham số:

x (int): Tọa độ X của thanh máu trên màn hình.

y (int): Tọa độ Y của thanh máu trên màn hình.

health (int): Sức khỏe hiện tại.

max_health (int): Sức khỏe tối đa.

Phương thức:

draw(self, health): Vẽ thanh máu trên màn hình với sức khỏe mới cập nhật.

Class Bullet:

Đại diện cho một viên đạn trong trò chơi.

Tham số:

x (int): Tọa độ X ban đầu của viên đạn.

y (int): Tọa độ Y ban đầu của viên đạn.

direction (int): Hướng di chuyển của viên đạn (-1 hoặc 1).

Thuộc tính:

speed (int): Tốc độ di chuyển của viên đạn.

image (pygame.Surface): Hình ảnh của viên đạn.

rect (pygame.Rect): Hình chữ nhật giới hạn viên đạn trên màn hình.

direction (int): Hướng di chuyển của viên đạn (-1 hoặc 1).

Phương thức:

update(self): Cập nhật vị trí của viên đạn và xử lý va chạm với các đối tượng trong trò chơi.

Class ScreenFade:

Đại diện cho hiệu ứng làm mờ màn hình trong trò chơi.

Tham số:

direction (int): Hướng của hiệu ứng làm mờ màn hình (1 hoặc 2).

colour (tuple): Màu của hiệu ứng làm mờ màn hình (dạng (R, G, B)).

speed (int): Tốc độ của hiệu ứng làm mờ màn hình.

Thuộc tính:

direction (int): Hướng của hiệu ứng làm mờ màn hình (1 hoặc 2).

colour (tuple): Màu của hiệu ứng làm mờ màn hình (dạng (R, G, B)).

speed (int): Tốc độ của hiệu ứng làm mờ màn hình.

fade_counter (int): Bộ đếm để điều chỉnh tốc độ và kích thước của hiệu ứng làm mờ.

Phương thức:

fade(self): Thực hiện hiệu ứng làm mờ màn hình và trả về True nếu hiệu ứng đã hoàn thành.