**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍNH MINH**



**LẬP TRÌNH VỚI PYTHON**

**Đề tài: Sử dụng python và pygame**

**Để tạo game pacman**

Sinh viên thực hiện: Phạm Hữu Ngân Phương

Mã số sinh viên: N19DCCN149

Lớp: D19CQCN03-N

Sinh viên thực hiện:Nguyễn Trần Đức Thuận

Mã số sinh viên: N19DCCN203

Lớp: D19CQCN03-N

TP Hồ Chí Minh, ngày Tháng Năm 2021

### ***Mục lục***

### **Phụ lục Page**

**Chương 1: Giới thiệu đề tài** **5**

**Chương 2: Cơ sở lý thuyết** **9**

* **Các thư viện được sử dụng trong đề tài và chức năng của chúng** **9** 
  1. Thư viện pygame  **11**
  2. Thư viện sys **13**
  3. Thư viện random  **11**

**Chương 3: Thực nghiệm**  **14**

* **Các chức năng chính của đề tài** **9**

1. Màn hình bắt đầu trò chơi **13**
2. Bản đồ game  **11**
3. Nhân vật **13**
   1. Tạo nhân vật Pacman  **11**
   2. Tạo các Ghost  **11**
4. Hoạt ảnh của nhân vật (Animation) **13**
5. Các hàm phụ **13**
6. Chương trình chính để chạy trò chơi **13**

**Chương 4: Kết luận**  **15**

**Tài liệu tham khảo**  **16**

Link github: <https://github.com/thuanntd2001/PacmanGame/tree/master>

**Chương 1: Giới thiệu đề tài**

Game pacman đối với những người trẻ chắc cũng không còn xa lạ gì nữa, nhất là trong thời kì internet phát triển như bây giờ. Có rất nhiều kiểu biến tấu khác nhau và sau đây nhóm chúng em sẽ trình bày một trong số các kiểu biến tấu đó ( dựa trên game pacman gốc). Đề tài này vừa để chúng em thu thập thêm kiến thức và kiếm điểm cuối kì, cũng như để giải trí sau những giờ học mệt mỏi. Và sau đây chúng em kính mời cô xem về đề tài game pacman của nhóm chúng em.

**Chương 2: Cơ sở lý thuyết**

* **Các thư viện được sử dụng trong đề tài và chức năng của chúng**
  + - 1. **Thư viện pygame**
* Pygame là một thư viện của ngôn ngữ lập trình python dùng để hỗ trợ khởi tạo và phát triển các dạng trò chơi 2D.
* Trong đồ án này chúng em sử dụng thư viện này là chủ yếu, nhưng ở đây chúng em chỉ giới thiệu các chức năng cơ bản của thư viện pygame, còn về các chức năng cụ thể đã sử dụng trong đồ án chúng em xin phép được nêu cụ thể trong phần thực nghiệm ở dưới. Các chức năng cơ bản của thư viện pygame :
  + - Pygame.init(): khởi tạo tất cả các môđun cần thiết cho pygame
    - Pygame.display.set\_mode(width,height): dùng để khởi tạo một cửa sổ có kích thước như bạn mong muốn
* Pygame.display.set\_caption(): dùng để xét tiêu đề cho cửa sổ màn hình
* While True: vòng lặp game

for event in pygame.event.get(): dùng để gọi các sự kiện xảy ra trong game

Vẽ trạng thái của trò chơi lên màn hình

Xử lý các sự kiện

Cập nhật trạng thái của trò chơi

* If event.type == QUIT:

# nếu sự kiện là sự kiện thoát thì sẽ gọi hàm pygame.quit() để thoát các môđun trò chơi.

* Pygame.display.update(): dùng để cập nhật trạng thái trò chơi
* Pygame.time.clock.tick( ): ở đây được dùng để chỉ tốc độ khung hình mà chương trình vẽ mỗi giây.
  + - 1. **Thư viện Sys**
* Module sys cung cấp các hàm và biến được sử dụng để thao tác các môi trường khác nhau chạy trong python. Nó cho phép chúng ta truy cập các tham số và chức năng cụ thể của hệ thống.
  + - 1. **Thư viện random**
* Thư viện random dùng để tạo ngẫu nhiên một số, nhưng trong đồ án của chúng em, random được sử dụng để tính đường đi của ghost mỗi khi chạm tường hoặc mỗi khi đi vào vùng cần chuyển hướng.

**Chương 3: Thực nghiệm**

* **Các chức năng chính của đề tài**

1. **Màn hình bắt đầu trò chơi (main.py):**

* Khai báo (\_init\_( )) :
* Các lựa chọn (items) khi bắt đầu: Start, Level, Exit.
* Màu chữ dùng để in ra màn hình. (Font\_color)
* Màu chữ khi con trỏ chỉ vào các lựa chọn ( Select\_color)
* State được đặt ở vị trí 1 ( Start).
* Cập nhật trạng thái và xử lý các sự kiện (update( ) and event\_handler( )) :

Dùng enumerate () để xác định các index của từng item

Di chuyển bằng nút trái phải

Khi nhấn nút di chuyển qua trái nếu state >0 thì state – 1.

Khi nhấn nút di chuyển qua phải nếu state < độ dài của item -1 thì state +1.

Nếu con trỏ chỉ tới vị trí này

Nếu không

Cập nhật

Gán label bằng phần tử tại vị trí đó với màu font\_color

Gán label bằng phần tử tại vị trí đó với màu select\_color

In ra màn hình với toạ độ posX và poxY

1. **Bản đồ game (BanDo.py) :**

* Khai báo(\_init\_( )):
* Các màu sẽ sử dụng trong đồ án: BLACK, WHITE,YELLOW, DARKBLUE, GRAY tất cả các màu đều được khai báo bằng kiểu màu RGB.
* Khởi tạo độ rộng cạnh hình vuông bằng 1 px.
* Khởi tạo màu chữ, màu tường(COLOR\_WALL), màu của đường đi(COLOR\_GROUND) và màu của ô đã được active(COLOR\_ACTIVE).
* Khởi tạo kích thước của 1 khối (KTI)px và kích thước của dấu chấm(COCHU) ( check point) trên bản đồ.
* Khởi tạo chiều dài, chiều rộng của màn hình.
* Khởi tạo độ lùi của khối thức ăn khi in ra màn hình.
* Khởi tạo Font chữ với size bằng COCHU (độ lớn của chấm thức ăn) đã được khai báo ở trên.
* Hàm f2t1( ): truyền vào số cột của ma trận và vị trí i và j trong ma trận.
* Hàm này dùng để chuyển vị trí của ma trận 2 chiều thành vị trí tại ma trận 1 chiều.
* **Lớp Square (class Square)** :
* Dùng để cung cấp các pixel trên bản đồ để có thể tuỳ chỉnh màu sắc cũng như thay đổi các trạng thái.
* Khai báo(\_init\_( )) : truyền vào toạ độ x,y , kích thước của các khối (KTI), kiểu dùng để vẽ khối tường hoặc đường đi (style) , khoảng cách x, y giữa các khối.
* Khai báo các khối bằng pygame.rect() ( dùng để lưu trữ toạ độ hình chữ nhật) và truyền vào toạ độ left=x, top=y, width=KTI, height =KTI tương ứng
* Khởi tạo độ to nhỏ của khối (KTI).
* Khởi tạo kiểu dùng để vẽ khối, đường đi hay check point.

Style

Style=0 hoặc Style=3 hoặc Style =9

Style=1

Style=2

Xét color của ô đó bằng COLOR\_ACTIVE

Xét color của ô đó bằng COLOR\_WALL

Xét color của ô đó bằng COLOR\_GROUND

* Hàm vẽ (draw( )): truyền vào màn hình cửa sổ được khai báo ở chương trình chính (screen).
* Vẽ hình chữ nhật đã khai báo ở trên ra màn hình bằng pygame.draw.rect() với screen, rect( đã được khai báo ở lớp square phía trên) và color.

Active = False

Style=9

Style=3

Style=0

In dấu chấm (tượng trưng cho check point) ra màn hình bằng hàm printTT( )

In hình ellipse bằng pyame.ellipse với màu được truyền vào là màu trắng cùng với toạ độ x,y được xác định bằng hàm chuyenTT ( )

In chữ o (tượng trưng cho check point) ra màn hình bằng hàm printTT ( )

* Hàm chuyenTT( ): truyền vào độ lùi của khối thức ăn.
* Khi dùng hàm này sẽ trả về một tuple dùng để xác định x,y để khi in hình nó sẽ ở trung tâm hình vuông.
* Hàm printTT( ): truyền vào màn hình cửa sỏ cần in chữ và nội dung cần in.
* Khi dùng hàm này nó sẽ in chữ ở trung tâm hình vuông.
* Hàm isIn( ): truyền vào toạ độ x,y.
* Hàm này dùng để kiểm tra toạ độ mình truyền vào có nằm trong hình vuông hay không.
* **Lớp BanDo (class BanDo( )) :**
* Lớp BanDo cung cấp các phương thức tạo, sửa, di chuyển đến 1 toạ độ trên map để vẽ. chỉ có bản đồ hình vuông được sử dụng.
* Khai báo(\_init\_( )): truyền vào ma trận cần dùng để tạo map và kích thước 1 ô (KTI).
  + - *Giải thích cấu trúc ma trận dùng để truyền vào: ma trận dùng để chuyền vào là ma trận hai chiều i và j. Có thể là ma trận vuông hoặc ma trận hình chữ nhật đều có thể truyền vào được.*
* Khởi tạo ma trận cần dùng.
* Khai báo IPS là số hàng của ma trận và IPSR là số cột của ma trận.
* Khai báo kích thước của 1 ô (KTI).
* Hàm chuyen(): truyền vào toạ độ x,y.
* Hàm này sẽ trả về toạ độ của các khối trên bản đồ.
* Hàm sinhBD(): truyền vào matran để sinh bản đồ từ ma trận đó.
* Tạo một list để lưu các giá trị của square (img)

Hàng của ma trận (i)

Cột của ma trận (j)

Vị trí của khối trong ma trận Xij và Yij

Tạo các khối bằng lớp square tại vị trí Xij và Yij

Nếu tại vị trí i và j trên bản đồ không phải là 1

Thêm khối vừa được tạo vào img

Thì end + 1

* Hàm vẽ (draw( )) : truyền vào screen.

Số cột \* số hàng (IPS \*IPSR)

Tại vị trí i

Vẽ khối ra màn hình

1. **Nhân vật(char.py):**

* Khai báo(\_init\_( )):
* Truyền vào bản đồ từ đối tượng bản đồ
* Xét toạ độ x,y thành các toạ độ trên bản đồ
* Hàm vẽ (draw( )): truyền vào mà hình cần in (screen ở đây là màn hình cửa sổ được khai báo bằng hàm pygame.display.set\_mode((x,y)), trong đó x là chiều rộng của màn hình còn y là chiều dài của màn hình
* Vẽ hình ảnh ra màn hình với toạ độ x,y đã xét ở trên ( dùng hàm screen.blit(image,(x,y)) để in image tương ứng ra màn hình)
* Hàm setPos( ): truyền vào i,j của hình ảnh và mặc định mode=0 nếu bạn không truyền mode vào

Mode

Mode = 1

Mode = 2

Mode = 3

Đặt nhân vật tại vị trí hàng i trên bản đồ, giữ nguyên toạ độ trục x cũ

Đặt nhân vật tại vị trí cột j trên bản đồ, giữ nguyên toạ độ trục y cũ

Đặt nhân vật tại vị trí i và j trên bản đồ

* 1. **Nhân Vật Pacman**
* Set hiệu ứng nổ và game\_over bằng False
* Khai báo(\_init\_( )): truyền vào toạ độ i,j mà pacman đứng lúc bắt đầu trò chơi, bando trong đối tượng bản đồ và speed nếu không được truyền vào thì sẽ mặc định bằng 5.
* Kế thừa thuộc tính của lớp nhân vật
* Truyền ảnh vào bằng hàm pygame.image.load() và gán cho nó là image, sau đó dùng . convert() tăng hiệu suất.
* Dùng .set\_colorkey: đặt colorkey cho ảnh hiện thị ra màn hình, nếu màu cùng màu với colorkey thì chỗ đó sẽ trong suốt.
* Dùng hàm .get\_rect() để tạo một ô hình chữ nhật bao phủ lấy hình và cũng để dễ dàng hơn trong việc xác định vị trí cũng như xử lý va chạm của hình với các bức tường.
* Xét lại kích thước 1 ô bằng kích thước cạnh của ô vuông chứa hình ảnh .
* Set speed bằng speed được truyền vô nếu không được truyền sẽ mặc định speed bằng 5
* Khởi tạo dx,dy bằng 0 ( dx là sự thay đổi của toạ độ x, còn dy là sự thay đổi của toạ độ y, ví dụ nếu đi qua phải dx sẽ cộng 1 số dương)
* Khởi tạo hướng bằng “s”, “s” ở đây có nghĩa là đứng yên (stand).
* Khởi tạo point bằng 0.
* Khởi tạo nhạc game mỗi khi pacman ăn được point cũng như nhạc mỗi khi game over(ở đây dùng pygame.mixer.sound(file-sound) để thêm âm thanh vào).
* Khởi tạo background end game và thêm ảnh vào cho nó
* Khởi tạo font chữ cho trò chơi bằng hàmpygame.font.sysfont(Name-font, size).
* Thêm các hình ảnh vào và gán cho chúng bằng các tên img, img1, img2,img3.
* Sau đó khởi tạo các animation qua trái, qua phải, lên, xuống cũng như hiệu ứng biến mất.(ở đây chúng em có dùng 3 hàm thêm để điều chỉnh hình ảnh:
* Đầu tiên là pygame.scale(image,(x,y)) dùng để thu phóng image theo x và y tương ứng. Ví dụ ảnh ban đầu có kích thước (900,600) khi truyền vào hàm này với x=600 và y =300 sẽ thu nhỏ hình lại với kích thước(600,300).
* Thứ 2 là hàm pygame.transform.flip(image,flip\_x,flip\_y) dùng để xoay hình theo chiều dọc ( flip\_x ) và theo chiều ngang ( flip\_y).
* Thứ 3 là hàm pygame.transform.rotate(image,angle) dùng để xoay hình theo số độ angle.
* Khởi tạo mode, bd, end bằng 0.
* Hàm qua phải(moveR( )):
* Khi đó dx= speed
* Áp dụng animation moveRight đã được xét ở trên và dùng hàm update của đối tượng animation với fps bằng 10.
* Sau đó ảnh sẽ được xét lại bằng hàm currentImage() trong đối tượng animation
* Hàm qua trái(moveL( )):
* Khi đó dx= -speed
* Áp dụng animation moveLeft đã được xét ở trên và dùng hàm update của đối tượng animation với fps bằng 10.
* Sau đó ảnh sẽ được xét lại bằng hàm currentImage() trong đối tượng animation
* Hàm đi lên(moveU( )):
* Khi đó dy= -speed
* Áp dụng animation moveUp đã được xét ở trên và dùng hàm update của đối tượng animation với fps bằng 10.
* Sau đó ảnh sẽ được xét lại bằng hàm currentImage() trong đối tượng animation
* Hàm đi xuống(moveD( )):
* Khi đó dy= speed
* Áp dụng animation moveDown đã được xét ở trên và dùng hàm update của đối tượng animation với fps bằng 10.
* Sau đó ảnh sẽ được xét lại bằng hàm currentImage() trong đối tượng animation
* Hàm đứng yên(stand( )):khi vào trò chơi mặc định sẽ làm hàm này
* Khi đứng yên dx,dy sẽ bằng 0.
* Ảnh sẽ được xét là ảnh nhân vật gốc.
* Hàm dừng di chuyển qua trái(stop\_move\_left( )):
* Nếu pacman đang di chuyển qua trái mà thấy dx < 0 (dừng qua trái và chuẩn bị đi qua phải) thì hình ảnh sẽ được lật lại đồng thời sẽ xét dx lại bằng 0.
* Hàm dừng di chuyển qua phải(stop\_move\_right( )):
* Nếu pacman đang di chuyển qua phải mà thấy dx > 0 (dừng qua phải và chuẩn bị đi qua trái) thì hình ảnh sẽ được lật lại đồng thời sẽ xét dx lại bằng 0.
* Hàm dừng di chuyển lên trên(stop\_move\_up( )):
* Nếu pacman đang di chuyển lên trên mà thấy dy < 0 (dừng lên trên và chuẩn bị đi xuống) thì hình ảnh sẽ được quay 90 độ lại đồng thời sẽ xét dy lại bằng 0.
* Hàm dừng di chuyển đi xuống(stop\_move\_down( )):
* Nếu pacman đang di chuyển đi xuống mà thấy dy> 0 (dừng qua đi xuống và chuẩn bị đi lên) thì hình ảnh sẽ được quay 90 độ lại đồng thời sẽ xét dy lại bằng 0.
* Hàm update( ): truyền vào hướng và screen
* Xét pos bằng toạ độ trên dưới trái phải và các góc của hình pacman bằng hàm getPosB ( )
* Xét i và j là toạ độ cảu hình pacman trên bản đổ bằng hàm getCoo( )

Explosion = False

Huong ==’s’

Huong ==’r’

Huong ==’d’

Huong ==’u’

Huong ==’l’

Stand

X+=dx

Y+=dy

MoveU

MoveL

MoveD

MoveR

Nếu cạnh trái của hình ảnh chạm biên

Nếu đang đi qua trái cạnh bên trái là tường bằng cách xét bản đồ và khi cạnh trái nằm trong tường

Nếu đang đi lên cạnh trên là tường bằng cách xét bản đồ và khi đỉnh trên nằm trong tường

Nếu cạnh dưới của hình chạm biên

Nếu cạnh phải của hình chạm biên

Nếu cạnh trên của hình ảnh chạm biên

Stop\_move\_left

Nếu đang đi xuống dưới xét có đi quá bản đồ và xét ô bên dưới có phải ô cuối cùng bên dưới và có phải là tường hay không

Nếu đang đi qua phải xét có đi quá bản đồ và xét ô bên phải có phải ô cuối cùng bên phải và có phải là tường hay không

Stop\_move\_up

Stop\_move\_down

Stop\_move\_right

* **Các điều kiện dùng để điều khiển hình ảnh để chống đè hình**
* Khi đi lên:

A= chạm cạnh biên trên

B=C=FALSE

Nếu i và j nằm vùng trừ ô sát cạnh biên

B= góc trên trái đụng tường

C= góc trên trái đụng tường

Nếu huong == ‘u’

Và (A hoặc B hoặc C)

Stop\_move\_up

Sử dụng hàm getPos để đưa hình ảnh vào giữa ô

* Khi qua trái :

A= chạm cạnh biên trái

B=C=FALSE

Nếu i và j nằm vùng trừ ô sát cạnh biên

B= góc trên trái đụng tường

C= góc dưới trái đụng tường

Nếu huong == ‘l’

Và (A hoặc B hoặc C)

Stop\_move\_left

Sử dụng hàm getPos để đưa hình ảnh vào giữa ô

* Khi qua phải:

A= chạm cạnh biên phải

B=C=FALSE

Nếu i và j nằm vùng trừ ô sát cạnh biên

B= góc trên phải đụng tường

C= góc dưới phải đụng tường

Nếu huong == ‘r’

Và (A hoặc B hoặc C)

Stop\_move\_right

Sử dụng hàm getPos để đưa hình ảnh vào giữa ô

* Khi đi xuống:

A= chạm cạnh biên dưới

B=C=FALSE

Nếu i và j nằm vùng trừ ô sát cạnh biên

B= góc dưới trái đụng tường

C= góc dưới phải đụng tường

Nếu huong == ‘d’

Và (A hoặc B hoặc C)

Stop\_move\_down

Sử dụng hàm getPos để đưa hình ảnh vào giữa ô

Dùng hàm f2t1 để tính toạ độ của pacman trên mảng 1 chiều

Nếu ô pacman đang đứng chưa được active (active = False )

Style của ô đó là 9

Style của ô đó là 0 hoặc 3

Khi pacman đi

Khi pacman đi

Ô đó sẽ được active (active = True )

Ô đó sẽ được active (active = True )

Nhạc sẽ được kêu khi pacman ăn được điểm

Nhạc sẽ được kêu khi pacman ăn được điểm

Point và end + 1

Point và end + 1

Gán time = thời gian kết thúc hiệu ứng bất tử

Gán thời gian bd tại thời gian ăn được điểm 9

Nếu time bd – time < 10s

Khởi động mode bất tử

(pacman rape ghost)

Nếu time bd – time > 10s

Trả lại mode bình thường

Image sẽ được gán hình bất tử

* Hàm rape( ) ( pacman ăn được ghost):

Dùng hàm getPosB để tính toạ độ 8 cạnh của ghost

Nếu 1 trong 8 cạnh của ghost chạm vào pacman

Point sẽ được cộng 10

* Hàm biRape( ) ( pacman bị ghost ăn):

Game over = True

Nhạc game over sẽ được phát

Màn hình sẽ được tô đen

In ra hiệu ứng nổ ra màn hình

Màn hình sẽ được tô đen

In ra màn hình hình background endgame và score của người chơi

* 1. **Tạo hình các ghost (Slime.py)**
* Khai báo(\_init\_( )) : truyền vào vị trí bắt đầu của ghost, hình ảnh của ghost, bản đồ từ lớp bando và speed ( nếu không truyền vào speed thì speed của ghost sẽ mặc định bằng giá trị được định sẵn ) bằng 5.
* Kế thừa lớp nhân vật
* Truyền ảnh vào bằng hàm pygame.image.load() và gán cho nó là image, sau đó dùng . convert() để tạo bản sao mới có cùng định dạng với hình cũ.
* Dùng .set\_colorkey: đặt colorkey cho ảnh hiện thị ra màn hình, nếu màu cùng màu với colorkey thì chỗ đó sẽ trong suốt.
* Dùng hàm .get\_rect() để tạo một ô hình chữ nhật bao phủ lấy hình và cũng để dễ dàng hơn trong việc xác định vị trí cũng như xử lý va chạm của hình với các bức tường.
* Xét lại kích thước 1 ô bằng chiều rộng của ô chữ nhật chứa hình ảnh .
* Xét speed bằng speed được truyền vô nếu không được truyền sẽ mặc định speed bằng 5
* Khởi tạo dx,dy bằng 0 ( dx là sự thay đổi của toạ độ x, còn dy là sự thay đổi của toạ độ y, ví dụ nếu đi qua phải dx sẽ cộng 1 số dương)
* Khởi tạo hướng bằng “r”, “r” ở đây có nghĩa là đi qua phải (right)
* Khởi tạo đổi hướng của ghost và xét nó bằng True.
* Khởi tạo hiệu ứng nổ và xét nó bằng False.
* Hàm qua phải(moveR( )):
* Khi đó dx= speed
* Hàm qua trái(moveL( )):
* Khi đó dx= -speed
* Hàm đi lên(moveU( )):
* Khi đó dy= -speed
* Hàm đi xuống(moveD( )):
* Khi đó dy= speed
* Hàm setPosSlime( ): truyền vào vị trí i và j của ghost trong ma trận bản đồ và mode =0.
* Hàm này dùng để xác định toạ độ i và j của ghost trên bản đồ
* Hàm rape( ): truyền vào nhân vật pacman.

Mode của pacman = 0

Trả về False

Nếu không

Vị trí x, y của ghost ( dùng hàm getPosB để xác định toạ độ của các góc cũng như các cạnh của hình ảnh ghost rồi lấy giá trị của nó )

Nếu vị trí của ghost trùng với vị trí của pacman

Trả về True

* Hàm biRape( ):

Ghost bị rape

Dùng hàm setPosSlime để xét ghost xuất hiện lại tại vị trí i=9 và j=10

Trả về explosion = False

* Hàm update( ):
* Xét pos bằng toạ độ trên dưới trái phải và các góc của hình ghost bằng hàm getPosB ( )
* Xét i và j là toạ độ cảu hình ghost trên bản đổ bằng hàm getCoo( )

Explosion = False

Huong ==’r’

Huong ==’d’

Huong ==’u’

Huong ==’l’

X+=dx

Y+=dy

MoveU

MoveL

MoveD

MoveR

Nếu đang đi lên cạnh trên là tường bằng cách xét bản đồ và khi đỉnh trên nằm trong tường

Nếu cạnh trái của hình ảnh chạm biên

Nếu đang đi qua trái cạnh bên trái là tường bằng cách xét bản đồ và khi cạnh trái nằm trong tường

Nếu cạnh trên của hình ảnh chạm biên

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Nếu đang đi qua phải xét có đi quá bản đồ và xét ô bên phải có phải ô cuối cùng bên phải và có phải là tường hay không

Nếu đang đi xuống dưới xét có đi quá bản đồ và xét ô bên dưới có phải ô cuối cùng bên dưới và có phải là tường hay không

Nếu cạnh dưới của hình chạm biên

Nếu cạnh phải của hình chạm biên

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Xét lại hướng bằng cách random ba hướng còn lại : ‘u’,’l’,’d’ bằng random.choice()

Xét lại hướng bằng cách random ba hướng còn lại : ‘r’,’l’,’d’ bằng random.choice()

Xét lại hướng bằng cách random ba hướng còn lại : ‘r’,’l’,’u’ bằng random.choice()

Xét lại hướng bằng cách random ba hướng còn lại : ‘r’,’d’,’u’ bằng random.choice()

Vị trí i và j trên ma trận = 3

daDoi = False

Random một trong 4 hướng ‘r’, ‘l’, ‘u’, ‘d’ bằng random.choice( )

Huong ==’d’

Huong ==’r’

Vị trí bên phải j không phải là tường

Vị trí bên dưới i không phải là tường

Huong ==’u’

Huong ==’l’

Vị trí i lớn hơn biên trái

Vị trí j lớn hơn biên trái

Vị trí j nằm trong vùng trừ cạnh sát biên

Vị trí i nằm trong vùng trừ cạnh sát biên

Vị trí bên trên i không phải là tường

Vị trí bên trái j không phải là tường

daDoi = True

daDoi = True

daDoi = True

daDoi = True

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

Dùng hàm setPosSlime để canh ghost vào giữa ô vuông với vị trí i và j trên bản đồ

X+=dx

Y+=dy

Xét lại dx=0

dy=0

1. **Hoạt ảnh của nhân vật Pacman ( Animation).**

* Khai báo:
* Truyền vào một imglist và sử dụng vòng lặp for để truyền từng hình ảnh trong imglist vào trong image\_list đã khai báo.
* Khai báo giá trị index bằng
* Khai bao giá trị clock bằng 0
* Hàm xét hình ảnh tại vị trí hiện tại:
* Sẽ trả về hình ảnh có vị trí index trong image\_list
* Hàm update:

Truyền vào số fps bạn muốn

Step = 30/fps

L=range(1,30,step)

Nếu không

Nếu clock =30

Clock + thêm 1

Xét Clock =1

Nếu giá trị clock nằm trong l

Index của hình trong image\_list sẽ cộng thêm 1

Nếu nếu giá trị của index = độ dài của image\_list

Index sẽ trả về giá trị bằng 0

1. **Các hàm phụ.**

* Hàm tính thời gian thực của hệ thống //tên hàm: sử dụng hàm time.time trong thư viện time để trả về giá trị thời gian hiện tại sau đó dùng hàm round để làm tròn giá trị đó.
* Hàm getPosB: dùng để tính toạ độ trên, dưới, trái ,phải, và các góc của ảnh theo KTI.
* Hàm getCen: dùng để tính toạ độ chính xác của ảnh.
* Hàm getCoo: đùng để trả về toạ độ của ảnh trên bản đồ
* Minh hoạ hàm getPosB

U

C

UL

UR



R

L

DR

DL

D

1. **Chương trình chính để chạy trò chơi**

* Hàm rapePM( ) : truyền vào nhân vật pacman và các ghost được chứa trong slimeTup.

Mỗi ghost trong tất cả các ghost

Nếu ghost rape được pacman

Explosion của pacman sẽ được chuyển sang True

* Hàm rapeSlime( ) : truyền vào nhân vật pacman và các ghost được chứa.

Nếu pacman rape được bất kì ghost nào

Explosion của ghost sẽ được chuyển sang True

* Hàm chính để chạy game (main( )):
  + - Pygame.init(): khởi tạo tất cả các môđun cần thiết cho pygame
    - Nhập vào ma trận bạn cần dùng để khởi tạo bản đồ.
    - Khai báo huong = ‘s’ cũng như các hướng right, left, up down.
    - Pygame.display.set\_mode(): dùng để khởi tạo một cửa sổ đủ lớn để chứa map
    - Khởi tạo font chữ bằng pygame.font.Sysfont( ) với font chữ vnibendigo và size chữ là 80.
    - Khởi tạo sound\_start bằng pygame.mixer.Sound( ) và chơi sound\_start ngay khi bắt đầu trò chơi.
    - Khởi tạo bg\_end\_game bằng pygame.image.load( ) và nắn lại kích thước của hình bằng pygame.transform.scale( ) với kích thước (700px,438px).
    - Khởi tạo map0 ( hay còn gọi là bản đồ chính của trò chơi) bằng lớp BanDo (BanDo.py) được nói ở phần trên.
    - Khởi tạo các ghost bằng lớp Slime (Slime.py) và lưu trữ tất cả các ghost trong list tupSlime.
    - Khởi tạo pacman bằng lớp pacman (Char.py).
    - Khởi tạo done=false.

Khởi tạo các event trong trò chơi bằng pygame.event.get( )

Event = pygame.KEYDOWN( )

Event = pygame.Quit( )

Event = Right

Done=True

Event = Down

Event = Left

Event = Up

Huong ==’u’

Huong ==’r’

Huong ==’d’

Huong ==’l’

* Vẽ map lên màn hình screen.
* Vẽ các ghost trong list tupSlime lên màn hình và dùng hàm update( ) trong Slime.py để cập nhật trạng thái cũng như toạ độ của các ghost.
* Vẽ pacman lên màn hình dùng hàm update( ) trong Char.py để cập nhật hướng, trạng thái cũng như toạ độ của pacman .
* Nếu pacman bị game over ( tức là pacman bị ghost rape) done = True cũng như nếu point của pacman = tổng point của map thì done cũng sẽ bằng True.
* Cuối cùng là cập nhật các sự kiện cũng như chỉnh tốc độ khung hình là 20 khung hình trên giây.

**Chương 4: Kết luận**

Qua đề tài này chúng em cám ơn cô đã đi theo và truyền dạy kiến thức về python cho chúng em học kì vừa rồi. Nếu chúng em còn sai sót ở đâu mong cố sẽ chỉ dạy thêm để nhóm chúng em có thể khắc phục nhằm tạo ra một project hoàn hảo nhất.

Chúng em xin cám ơn cô

**Tài liệu tham khảo**

* Making Game With Python & Pygame ( by Al Sweigart)
* Invent Your Own Computer Games with Python 4th Edition (by Al Sweigart)