HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN



****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN LẬP TRÌNH PYTHON**

**Giảng viên: Vũ Hoài Thư**

**Nhóm 21**

**Sinh viên: Vũ Minh Hoàng**

**Mã Sinh Viên: B19DCCN285**

***Hà Nội – 2021***

MUC LỤC

[1.Tổng quan 3](#_Toc89075854)

[1.1 Lý do chọn đề tài 3](#_Toc89075855)

[1.2 Các thành viên trong nhóm 3](#_Toc89075856)

[2.Chi tiết các bước làm và xây dựng chương trình 3](#_Toc89075857)

[2.1 Module setting 4](#_Toc89075858)

[2.2 Module Snake 4](#_Toc89075859)

[2.2.1 Hàm \_\_init\_\_() 4](#_Toc89075860)

[2.2.2 Hàm control() 5](#_Toc89075861)

[2.2.3 Hàm draw() 6](#_Toc89075862)

[2.2.4 Hàm move() 6](#_Toc89075863)

[2.3 Module Wall 7](#_Toc89075864)

[2.3.1 Hàm \_\_init\_\_() 7](#_Toc89075865)

[2.3.2 Hàm draw() 10](#_Toc89075866)

[2.4 Module Fruit 10](#_Toc89075867)

[2.4.1 Hàm \_\_init\_\_() 10](#_Toc89075868)

[2.4.2 Hàm draw() 10](#_Toc89075869)

[2.5 Module Sound 10](#_Toc89075870)

[2.6 Module chứa hàm main của game 11](#_Toc89075871)

[3.Tổng kết 18](#_Toc89075872)

# 1.Tổng quan

## 1.1 Lý do chọn đề tài

Ngày nay công nghệ thông tin trở thành một lĩnh vực mũi nhọn trong công cuộc phát triển kinh tế xã hội. Có thể nói game đã và đang trở thành một nhu cầu giải trí thiết yếu trong cuộc sống của con người hiện nay. Một trong những thể loại game kinh điển mà khi nhắc tới chắc không còn xa lạ gì với chúng ta nhất là đối với thế hệ 8x trở về sau này đó chính là game rắn săn mồi. Trong tiền bối công nghệ ngày càng được phát triển như hiện nay, nhiều thể loại game mới cũng ra đời ngày càng nhiều và thu hút được nhiều bạn trẻ. Tuy nhiên không vì vậy mà trò chơi rắn săn mồi bị mờ mà vẫn giữ được một vị trí trong lòng những người yêu thích có thể loại trò chơi này. Vì thế chủ đề “Game Rắn Săn Mồi” được tiến hành phục vụ nhu cầu giải trí cho chúng ta sau những giờ làm việc căng thẳng.

Con rắn là một trò chơi cổ điển. Xuất hiện lần đầu tiên trong thập niên 70, sau đó nó đã được đưa lên máy tính. Trong trò chơi này người chơi điều khiển một con rắn. Người dùng các mũi tên lên, xuống, trái, phải để điều khiển con rắn sao cho con rắn ăn mồi. Mỗi lần ăn mồi sẽ được cộng thêm một điểm. Trò chơi kết thúc khi con rắn di chuyển chạm vào chướng ngại vật hoặc chính thân con rắn.

## 1.2 Các thành viên trong nhóm

Nguyễn Tiến Hải Ninh (Code + Demo)

Vũ Minh Hoàng (Làm Silde + Thuyết Trình)

Phạm Văn Thái (Code + Viết Báo Cáo)

# 2.Chi tiết các bước làm và xây dựng chương trình

Các tính năng cơ bản:

+ Chơi mới

+ Chọn map phù hợp

+ Chơi lại map cũ hoặc thay đổi map mới khi kết thúc

+ Tạm dừng khi đang chơi

+ Hiển thị điểm số của người chơi

+ Kết hợp âm thanh sống động khi thao tác

Trước hết chúng ta hãy khởi tạo các module cần thiết trong quá trình xây dựng game nhé:

## 2.1 Module setting

Khởi tạo kích thước cửa sổ màn hình, lưới màn hình và FPS

*SCREEN\_WIDTH = 900*

*SCREEN\_HEIGHT = 600*

*UNIT = 30*

*GRID = []*

*FPS = 8*

*for i in range(0, SCREEN\_HEIGHT, UNIT):*

*for j in range(0, SCREEN\_WIDTH, UNIT):*

*GRID.append([j, i])*

## 2.2 Module Snake

Module chứa đối tượng Snake với các thuộc tính: màu sắc, độ dài,... và các phương thức: định hướng, di chuyển, tạo hình

Trong đó:

### 2.2.1 Hàm \_\_init\_\_()

Khởi tạo một số thuộc tính của đối tượng Snake như color, direction, length, score, alive,…

*def \_\_init\_\_(self, map):*

*self.color = (148, 57, 107)*

*self.direction = RIGHT*

*if map == 3:*

*self.direction = UP*

*self.body = [[st.SCREEN\_WIDTH/2, st.SCREEN\_HEIGHT/2]]*

*self.saveDir = [self.direction[2]]*

*self.length = 1*

*self.score = 0*

*self.alive = True*

*self.pause = False*

### 2.2.2 Hàm control()

Sử dụng các mũi tên nhập từ bàn phím, định hướng di chuyển cho đối tượng Snake**.**

*def control(self):*

*for event in pygame.event.get():*

*if event.type == pygame.KEYDOWN:*

*if event.key == pygame.K\_UP and self.direction != DOWN:*

*self.direction = UP*

*break*

*elif event.key == pygame.K\_DOWN and self.direction != UP:*

*self.direction = DOWN*

*break*

*elif event.key == pygame.K\_LEFT and self.direction != RIGHT:*

*self.direction = LEFT*

*break*

*elif event.key == pygame.K\_RIGHT and self.direction != LEFT:*

*self.direction = RIGHT*

*break*

*elif event.type == pygame.MOUSEBUTTONDOWN:*

*pos = event.pos*

*if pos[0] in range(840, 881) and pos [1] in range(10, 511):*

*pygame.mixer.Channel(0).play(sound.click)*

*self.pause = True*

*elif event.type == pygame.QUIT:*

*pygame.quit()*

*sys.exit()*

### 2.2.3 Hàm draw()

Sử dụng để hiển thị đối tượng Snake trên surface. Nếu các bạn có biết qua photoshop thì chắc hẳn các bạn biết đến khái niệm layer. Surface trong pygame cũng tương tự như thế, nó là một “lớp” ảnh trong suốt. Các surface có thể được vẽ chồng lên nhau.

*def draw(self, surface):*

*flag = True*

*for point, dir in zip(self.body, self.saveDir):*

*img = pygame.Surface((st.UNIT, st.UNIT))*

*if flag:*

*img = pygame.image.load("assets/images/head\_snake1.png")*

*flag = False*

*else:*

*img = pygame.image.load("assets/images/body\_snake.png")*

*img = pygame.transform.scale(img, (st.UNIT, st.UNIT))*

*img = pygame.transform.rotate(img, dir)*

*surface.blit(img, (point[0], point[1]))*

### 2.2.4 Hàm move()

Sử dụng để cập nhật vị trí của Snake theo vị trí đầu của Snake, đồng thời kiểm tra Snake đã đâm vào chướng ngại vật hay tự đâm vào thân mình hay là chưa.

*def move(self, wall):*

*old = self.body[0]*

*new = [(old[0]+self.direction[0])%st.SCREEN\_WIDTH, (old[1]+self.direction[1])%st.SCREEN\_HEIGHT]*

*if (new in wall) or (self.length > 2 and new in self.body[2:]):*

*self.alive = False*

*pygame.mixer.Channel(1).play(sound.gameover)*

*else:*

*self.saveDir.insert(0, self.direction[2])*

*self.body.insert(0, new)*

*if len(self.body) > self.length:*

*self.body.pop()*

## 2.3 Module Wall

Module chứa đối tượng Wall, là tập hợp các ô vuông mà khi Snake chạm vào, game sẽ kết thúc.

Trong đó:

### 2.3.1 Hàm \_\_init\_\_()

Sử dụng để khởi tạo thuộc tính position của đối tượng Wall, cũng như tạo ra các map có thể lựa chọn khi chơi game.

*def \_\_init\_\_(self, map):*

*self.position = []*

*if map == 1:*

*for i in range(0, st.SCREEN\_HEIGHT, st.UNIT):*

*for j in range(0, st.SCREEN\_WIDTH, st.UNIT):*

*if i == 0 or i == st.SCREEN\_HEIGHT-st.UNIT:*

*self.position.append([j, i])*

*else:*

*if j == 0 or j == st.SCREEN\_WIDTH-st.UNIT:*

*self.position.append([j, i])*

*elif map == 2:*

*for i in range(0, st.SCREEN\_HEIGHT, st.UNIT):*

*if i < st.SCREEN\_HEIGHT\*7//20 or i >= st.SCREEN\_HEIGHT\*13//20:*

*for j in range(0,st.SCREEN\_WIDTH, st.UNIT):*

*if i == st.SCREEN\_HEIGHT\*7//20-st.UNIT or i == st.SCREEN\_HEIGHT\*13//20:*

*if j >= st.SCREEN\_WIDTH//3 and j < st.SCREEN\_WIDTH\*2//3:*

*self.position.append([j, i])*

*elif i == 0 or i == st.SCREEN\_HEIGHT-st.UNIT:*

*if j < st.SCREEN\_WIDTH//3 or j >= st.SCREEN\_WIDTH\*2//3:*

*self.position.append([j, i])*

*else:*

*if j == 0 or j == st.SCREEN\_WIDTH-st.UNIT:*

*self.position.append([j, i])*

*elif map == 3:*

*for i in range(0, st.SCREEN\_HEIGHT, st.UNIT):*

*for j in range(0, st.SCREEN\_WIDTH, st.UNIT):*

*if i <= st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//5 - st.UNIT and j == st.SCREEN\_WIDTH//3 + st.UNIT:*

*self.position.append([j, i])*

*if i >= st.SCREEN\_HEIGHT//5 and j == st.SCREEN\_WIDTH - st.SCREEN\_WIDTH//3 - 2 \* st.UNIT:*

*self.position.append([j, i])*

*if i == st.SCREEN\_HEIGHT//5 and j <= st.SCREEN\_WIDTH//3 - 2 \* st.UNIT:*

*self.position.append([j, i])*

*if i == st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//5 - st.UNIT and j >= st.SCREEN\_WIDTH - st.SCREEN\_WIDTH//3 + st.UNIT:*

*self.position.append([j, i])*

*elif map == 4:*

*for i in range(0, st.SCREEN\_HEIGHT, st.UNIT):*

*for j in range(0, st.SCREEN\_WIDTH, st.UNIT):*

*if i == 0 or i == st.SCREEN\_HEIGHT-st.UNIT:*

*self.position.append([j, i])*

*if i <= st.SCREEN\_HEIGHT//5 or i >= st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//5 - st.UNIT:*

*if j == 0 or j == st.SCREEN\_WIDTH-st.UNIT:*

*self.position.append([j, i])*

*if i == st.SCREEN\_HEIGHT//5 + st.UNIT or i == st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//5 - 2 \* st.UNIT:*

*if j > st.SCREEN\_HEIGHT//5 and j < st.SCREEN\_WIDTH - st.SCREEN\_HEIGHT//5 - st.UNIT:*

*self.position.append([j, i])*

*elif map == 5:*

*for i in range(0, st.SCREEN\_HEIGHT, st.UNIT):*

*for j in range(0, st.SCREEN\_WIDTH, st.UNIT):*

*if i == st.SCREEN\_HEIGHT//2 - st.UNIT:*

*if j <= st.SCREEN\_WIDTH//2 - 4 \* st.UNIT or j >= st.SCREEN\_WIDTH//2:*

*self.position.append([j, i])*

*if i == st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//4 - st.UNIT:*

*self.position.append([j, i])*

*if i <= st.SCREEN\_HEIGHT//2 - 5 \* st.UNIT:*

*if j == st.SCREEN\_WIDTH//2:*

*self.position.append([j, i])*

*if i >= st.SCREEN\_HEIGHT - st.SCREEN\_HEIGHT//4 - st.UNIT:*

*if j == st.SCREEN\_WIDTH//2:*

*self.position.append([j, i])*

### 2.3.2 Hàm draw()

Sử dụng để hiển thị đối tượng Wall trên surface.

*def draw(self, surface):*

*for point in self.position:*

*img = pygame.image.load("assets/images/wall.png")*

*img = pygame.transform.scale(img, (st.UNIT, st.UNIT))*

*surface.blit(img, point)*

## 2.4 Module Fruit

Module chứa đối tượng Fruit, là một ô vuông được sinh ngẫu nhiên (không trùng với các ô có Wall và Snake), khi chạm vào Fruit, Snake được tăng 1 điểm và 1 đơn vị chiều dài

Trong đó:

### 2.4.1 Hàm \_\_init\_\_()

Sử dụng để khởi tạo ngẫu nhiên thuộc tính position của đối tượng Fruit

*def \_\_init\_\_(self, wall, snake):*

*self.position = random.choice([p for p in st.GRID if (p not in wall) and (p not in snake)])*

*img = pygame.image.load(random.choice(["assets/images/apple.png", "assets/images/banana.png", "assets/images/grape.png"])).convert\_alpha()*

*self.img = pygame.transform.scale(img, (st.UNIT, st.UNIT))*

### 2.4.2 Hàm draw()

Sử dụng để hiển thị đối tượng Fruit trên surface.

*def draw(self, surface):*

*surface.blit(self.img, (self.position[0] , self.position[1]))*

## 2.5 Module Sound

Module khởi tạo các file âm thanh:

Âm thanh nền (background)

Âm thanh khi click chuôt vào cửa sổ đa nhiệm trên màn hình

Âm thanh khi Snake chạm đến Fruit

Âm thanh khi Snake chạm vào Wall hoặc vào chính mình

*pygame.mixer.init()*

*background1 = pygame.mixer.Sound("assets/sound/background1.mp3")*

*background2 = pygame.mixer.Sound("assets/sound/background2.mp3")*

*background3 = pygame.mixer.Sound("assets/sound/background3.mp3")*

*click = pygame.mixer.Sound("assets/sound/click.mp3")*

*eat = pygame.mixer.Sound("assets/sound/eat\_fruit.wav")*

*gameover = pygame.mixer.Sound("assets/sound/gameover.mp3")*

## 2.6 Module chứa hàm main của game

Khởi tạo cửa sổ game với kích thước và tiêu đề

Tạo hình nền menu, chèn background

Tạo các button với các chức năng bắt đầu game, chọn map, tạm dừng,

Thao tác các tính năng trong game như tạm dừng, chơi lại map cũ khi kết thúc, thoát ra menu chọn map mới, thoát game

*import pygame, sys, random, fruit, snake, wall, sound*

*from pygame.locals import \**

*import setting as st*

*def play(SCREEN, map):*

*SCREEN.fill((255, 255, 255))*

*pygame.mixer.Channel(1).play(random.choice([sound.background1, sound.background2]), -1)*

*#Khoi tao doi tuong*

*s = snake.Snake(map)*

*w = wall.Wall(map)*

*f = fruit.Fruit(w.position, s.body)*

*#Khoi tao hinh anh*

*background\_game = pygame.transform.scale(pygame.image.load("assets/images/background\_game.jpg").convert\_alpha(), (st.SCREEN\_WIDTH, st.SCREEN\_HEIGHT))*

*background\_over = pygame.transform.scale( pygame.image.load("assets/images/gameover.jpg").convert\_alpha(), (200, 200))*

*replay = pygame.transform.scale(pygame.image.load("assets/images/replay.png").convert\_alpha(), (40, 40))*

*back1 = pygame.transform.scale(pygame.image.load("assets/images/quit.png").convert\_alpha(), (40, 40))*

*back2 = pygame.transform.scale(pygame.image.load("assets/images/quit.png").convert\_alpha(), (150, 150))*

*pause = pygame.transform.scale(pygame.image.load("assets/images/pause.png").convert\_alpha(), (40, 40))*

*unpause = pygame.transform.scale(pygame.image.load("assets/images/unpause.png").convert\_alpha(), (150, 150))*

*#Vong lap game khi start*

*while True:*

*#Vong lap khi Snake con song*

*while s.alive:*

*SCREEN.blit(background\_game, (0, 0))*

*s.control()*

*s.move(w.position)*

*if s.body[0][:2] == f.position:*

*pygame.mixer.Channel(0).play(sound.eat)*

*f = fruit.Fruit(w.position, s.body)*

*s.score += 1*

*s.length += 1*

*#Tam dung*

*while s.pause:*

*pygame.mixer.Channel(1).pause()*

*w.draw(SCREEN)*

*s.draw(SCREEN)*

*f.draw(SCREEN)*

*SCREEN.blit(unpause, (st.SCREEN\_WIDTH/2-200, st.SCREEN\_HEIGHT/2-100))*

*SCREEN.blit(back2, (st.SCREEN\_WIDTH/2+50, st.SCREEN\_HEIGHT/2-100))*

*SCREEN.blit(pygame.font.SysFont("times new roman", 20).render(f"Điểm: {s.score}", True, (255, 255, 0)), (10, 20))*

*pygame.display.update()*

*for event in pygame.event.get():*

*if event.type == pygame.QUIT:*

*pygame.quit()*

*sys.exit()*

*elif event.type == pygame.MOUSEBUTTONDOWN:*

*pos = event.pos*

*if pos[0] in range(250, 401) and pos [1] in range(200, 401):*

*pygame.mixer.Channel(0).play(sound.click)*

*pygame.mixer.Channel(1).unpause()*

*s.pause = False*

*if pos[0] in range(500, 651) and pos [1] in range(200, 401):*

*pygame.mixer.Channel(0).play(sound.click)*

*pygame.mixer.Channel(1).unpause()*

*main()*

*w.draw(SCREEN)*

*s.draw(SCREEN)*

*f.draw(SCREEN)*

*SCREEN.blit(pygame.font.SysFont("times new roman", 20).render(f"Điểm: {s.score}", True, (255, 255, 0)), (10, 20))*

*SCREEN.blit(pause ,(840, 10))*

*pygame.display.update()*

*pygame.time.Clock().tick(st.FPS)*

*#Gameover: hien modal thong bao*

*SCREEN.blit(background\_over, (st.SCREEN\_WIDTH/2-100, st.SCREEN\_HEIGHT/2-100))*

*SCREEN.blit(pygame.font.SysFont("times new roman", 17).render(f"Thành tích: {s.score} điểm", True, (0, 0, 0)), (380, 320))*

*SCREEN.blit(replay, (380, 350))*

*SCREEN.blit(back1, (470, 350))*

*for event in pygame.event.get():*

*#Thoat game*

*if event.type == pygame.QUIT:*

*pygame.quit()*

*sys.exit()*

*if event.type == pygame.MOUSEBUTTONDOWN:*

*pos = event.pos*

*#Choi lai map nay*

*if pos[0] in range(380, 421) and pos [1] in range(350, 391):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, map)*

*#Thoat ra man hinh chon map*

*if pos[0] in range(470, 511) and pos [1] in range(350, 391):*

*pygame.mixer.Channel(0).play(sound.click)*

*main()*

*pygame.display.update()*

*def main():*

*#Khoi tao cua so game*

*pygame.init()*

*SCREEN = pygame.display.set\_mode((st.SCREEN\_WIDTH, st.SCREEN\_HEIGHT))*

*pygame.display.set\_caption("SNAKE GAME")*

*FONT = pygame.font.SysFont("times new roman", 20)*

*#Hinh nen menu*

*BACKGROUND = pygame.image.load("assets/images/background\_menu.jpg").convert\_alpha()*

*BACKGROUND = pygame.transform.scale(BACKGROUND, (st.SCREEN\_WIDTH, st.SCREEN\_HEIGHT))*

*SCREEN.blit(BACKGROUND, (0, 0))*

*#Button play game*

*BUTTON = pygame.image.load("assets/images/button\_play.png").convert\_alpha()*

*BUTTON = pygame.transform.scale(BUTTON, (100, 50))*

*possition\_button = pygame.Rect(st.SCREEN\_WIDTH/2-50, st.SCREEN\_HEIGHT/2-25, 100, 50)*

*SCREEN.blit(BUTTON, possition\_button)*

*#Hinh map*

*map0 = pygame.transform.scale(pygame.image.load("assets/images/map0.png"), (120, 80))*

*map1 = pygame.transform.scale(pygame.image.load("assets/images/map1.png"), (120, 80))*

*map2 = pygame.transform.scale(pygame.image.load("assets/images/map2.png"), (120, 80))*

*map3 = pygame.transform.scale(pygame.image.load("assets/images/map3.png"), (120, 80))*

*map4 = pygame.transform.scale(pygame.image.load("assets/images/map4.png"), (120, 80))*

*map5 = pygame.transform.scale(pygame.image.load("assets/images/map5.png"), (120, 80))*

*#Nhac nen*

*pygame.mixer.Channel(1).play(sound.background3)*

*pygame.mixer.Channel(0).set\_volume(0.2)*

*pygame.mixer.Channel(1).set\_volume(0.2)*

*start = False*

*#Vong lap game*

*while True:*

*pygame.display.update()*

*for event in pygame.event.get():*

*#Thoat game*

*if event.type == pygame.QUIT:*

*pygame.quit()*

*sys.exit()*

*if event.type == pygame.MOUSEBUTTONDOWN:*

*pos = event.pos*

*#Event: An vao nut start*

*if pos[0] in range(possition\_button.left, possition\_button.right+1) and pos[1] in range(possition\_button.top, possition\_button.bottom+1):*

*pygame.mixer.Channel(0).play(sound.click)*

*SCREEN.blit(FONT.render(f"Chọn map:", True, (255, 0, 0)), (400, 330))*

*SCREEN.blit(map0, (250, 360))*

*SCREEN.blit(map1, (390, 360))*

*SCREEN.blit(map2, (530, 360))*

*SCREEN.blit(map3, (250, 450))*

*SCREEN.blit(map4, (390, 450))*

*SCREEN.blit(map5, (530, 450))*

*start = True*

*#Event: An vao hinh map de chon map va choi game*

*if start:*

*if pos[0] in range(250, 371) and pos[1] in range(360, 441):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 0)*

*if pos[0] in range(390, 511) and pos[1] in range(360, 441):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 1)*

*if pos[0] in range(530, 651) and pos[1] in range(360, 441):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 2)*

*if pos[0] in range(250, 371) and pos[1] in range(450, 531):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 3)*

*if pos[0] in range(390, 511) and pos[1] in range(450, 531):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 4)*

*if pos[0] in range(530, 651) and pos[1] in range(450, 531):*

*pygame.mixer.Channel(0).play(sound.click)*

*play(SCREEN, 5)*

*main()*

# 3.Tổng kết

Trong nghiên cứu thời gian và vận dụng các kiến thức đã được trang bị ở nhà trường và các lĩnh vực khác nhau. Chúng tôi đã viết được trò chơi rắn săn mồi nhắm mục tiêu giải trí lành mạnh. Chúng tôi hy vọng rằng những khái niệm này là tiền đề giúp chúng tôi phát triển mạnh hơn về những thể loại trò chơi khác chất lượng hơn, hay hơn để phục vụ cho nhu cầu trong cuộc sống của chúng ta.

Vì thời gian làm báo cáo hạn chế, kinh nghiệm thực tế không nhiều, nên báo cáo không tránh những sai sót, rất mong sự góp ý của cô và các bạn.

Chúng tôi xin chân thành cảm ơn !