

Python 期末專題報告

109403526 黃彥程 資管二 A

小遊戲製作

目錄

專題說明-----p2

專題介紹-----p2

成果畫面-----p5

程式碼-----p5

專題說明

利用 python 模組 **pygame** 製作適合小朋友遊玩的簡單小遊戲，也可以放鬆時當作小遊戲遊玩。

內容為扮演一條小魚，躲避鯊魚群的攻擊，若遭到太多次鯊魚的攻擊則失敗結束遊戲。

專題介紹

整體架構:

採用 **pygame** 套件內的功能，一開始，初始化整個遊戲的遊戲畫面，設定他的長寬並使其展示在畫面上(**display**)，然後設定角色類別(玩家 **Player class**，鯊魚群 **Shark class**):

```
class Player(pygame.sprite.Sprite):  
class Shark(pygame.sprite.Sprite):
```

在類別裡面分別去做初始化這個類別，使他可以有個實體存在在畫面上:

```
def __init__(self):
```

並加入 **update function** 使他們分別可以以不同方式移動(玩家藉由設定的 **WASD** 移動，鯊魚則向左且上下隨機移動):

```
def update(self):
```

設立一個 **Group** 名 **all_sprites** 來裝所有物件(**Player** 與 8 個 **Shark**)，一個 **sharks** 來裝 8 個 **shark**:

```
all_sprites = pygame.sprite.Group()
sharks = pygame.sprite.Group()
player = Player()
all_sprites.add(player)
for i in range(8):
    shark = Shark()
    all_sprites.add(shark)
    sharks.add(shark)
```

利用 **while** 迴圈來使遊戲進行，因遊戲是一個不斷更新與使用者互動的程式，所以需使用迴圈來不斷更新達成目的。

而遊戲有所謂的更新率(每秒最多執行次數)，又稱 **FPS**，這邊設定為 **60**:

```
FPS = 60
clock.tick(FPS)
```

在知道此程式每秒執行 **60** 次之後，利用 **all_sprites.update()** 來使每個在 **all_sprites** 群組裡的物件每秒執行各自 **update()** 的內容，來達到畫面更新的效果。

再來是當玩家沒有躲掉鯊魚的攻擊時，鯊魚會消失並且玩家的血量會下降:

```
hits = pygame.sprite.spritecollide(player, sharks, True)
for hit in hits:
    shark = Shark()
    all_sprites.add(shark)
    sharks.add(shark)
    player.health -= 10
    if player.health <= 0:
        running = False
```

但是鯊魚消失後會一直呈現少一隻的狀態，所以會在重新

做一個新的 Shark 物件並加入群組

```
def new_shark():  
    shark = Shark()  
    all_sprites.add(shark)  
    sharks.add(shark)
```

血條的更新，上面有寫到當撞到就扣血且當玩家血量歸零時遊戲自動結束:

```
for hit in hits:  
    player.health -= 10  
    if player.health <= 0:  
        running = False
```

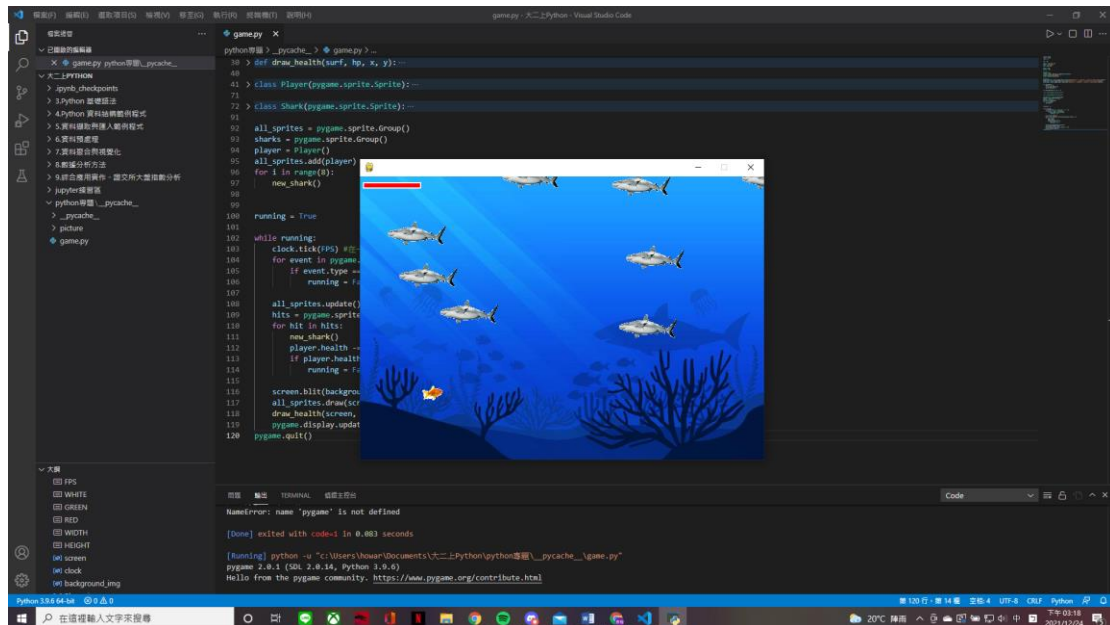
在同一個迴圈的底下則有負責將當前數據傳入計算血量長度的 function，與將更新結果顯在畫面上的兩條程式碼:

```
draw_health(screen, player.health, 5, 10)  
pygame.display.update()
```

最後，當然還有自動結束遊戲與手動按右上的紅色 x 來結束遊戲的程式碼:

```
for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
        running = False  
pygame.quit()
```

成果畫面



程式碼

```
import pygame
import random
import os

FPS = 60

WHITE = (255,255,255)
GREEN = (0,255,0)
RED = (255,0,0)

WIDTH = 700
HEIGHT = 490

#初始化
pygame.init()
screen = pygame.display.set_mode((WIDTH,HEIGHT))
pygame.display.set_caption('')
clock = pygame.time.Clock()
```

```

#載入圖片
background_img = pygame.image.load(os.path.join("python 專題", "picture", "Ocean.jpg")).convert()
Player_img = pygame.image.load(os.path.join("python 專題", "picture", "Fish.jpg")).convert()
Shark_img = pygame.image.load(os.path.join("python 專題", "picture", "Sharkk.jpg")).convert()

def new_shark():
    shark = Shark()
    all_sprites.add(shark)
    sharks.add(shark)

def draw_health(surf, hp, x, y):
    if hp<0:
        hp = 0
    BAR_LENGTH = 100
    BAR_HEIGHT = 10
    fill = (hp/100)*BAR_LENGTH
    outline_rect = pygame.Rect(x, y, BAR_LENGTH, BAR_HEIGHT)
    fill_rect = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surf, RED, fill_rect)
    pygame.draw.rect(surf, WHITE, outline_rect,2)

class Player(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(Player_img, (50,40))
        self.image.set_colorkey(WHITE)
        self.rect = self.image.get_rect() #把這個 image 框起來(定位)
        self.rect.centery = HEIGHT/2
        self.rect.left = 50
        self.speedx = 8
        self.health = 100

    def update(self):
        key_pressed = pygame.key.get_pressed()
        if key_pressed[pygame.K_d]:

```

```

        self.rect.x += self.speedx
    if key_pressed[pygame.K_a]:
        self.rect.x -= self.speedx
    if key_pressed[pygame.K_s]:
        self.rect.y += self.speedx
    if key_pressed[pygame.K_w]:
        self.rect.y -= self.speedx

    if self.rect.right > WIDTH:
        self.rect.right = WIDTH
    if self.rect.left < 0:
        self.rect.left = 0
    if self.rect.bottom > HEIGHT:
        self.rect.bottom = HEIGHT
    if self.rect.top < 0:
        self.rect.top = 0

class Shark(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(Shark_img,(100,40))
        self.image.set_colorkey(WHITE)
        self.rect = self.image.get_rect() #把這個 image 框起來(定位)
        self.rect.x = random.randrange(700,800)
        self.rect.y = random.randrange(0,HEIGHT-self.rect.height)
        self.speedx = random.randrange(-10,-2)
        self.speedy = random.randrange(-3,3)

    def update(self):
        self.rect.x += self.speedx
        self.rect.y += self.speedy
        if self.rect.top > HEIGHT or self.rect.left > WIDTH or
self.rect.right < 0:
            self.rect.x = random.randrange(700,800)
            self.rect.y = random.randrange(0,HEIGHT-self.rect.height)
            self.speedx = random.randrange(-10,-2)
            self.speedy = random.randrange(-3,3)

```

```

all_sprites = pygame.sprite.Group()
sharks = pygame.sprite.Group()
player = Player()
all_sprites.add(player)
for i in range(8):
    new_shark()

running = True

while running:
    clock.tick(FPS) #在一秒之內最多被執行 60 次
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    all_sprites.update()
    hits = pygame.sprite.spritecollide(player, sharks, True)
    for hit in hits:
        new_shark()
        player.health -= 10
        if player.health <= 0:
            running = False

    screen.blit(background_img, (0,0))
    all_sprites.draw(screen)
    draw_health(screen, player.health, 5, 10)
    pygame.display.update()
pygame.quit()

```

(載入圖片那邊的路徑需要改成自己的路徑!)

