

Python期末：小恐龍

報告人：資工三 4109056003 鄭佩綺



目錄

Part 1

介紹

Part 2

實現

Part 3

Demo



PART 1

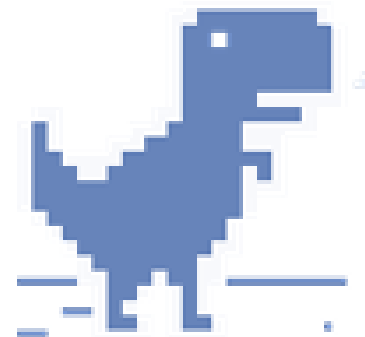
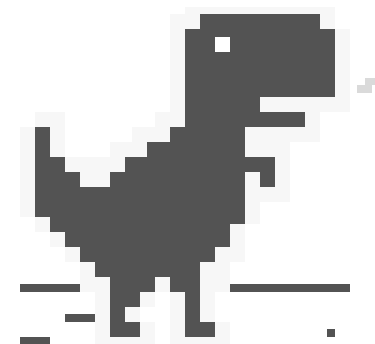
介紹



簡介

題目介紹：斷線小恐龍

- 可選擇恐龍顏色
- 背景隨著分數改變
- 可以發射火箭
- 吃到驚嘆號能增加火箭數量



PART 2

實現

實現

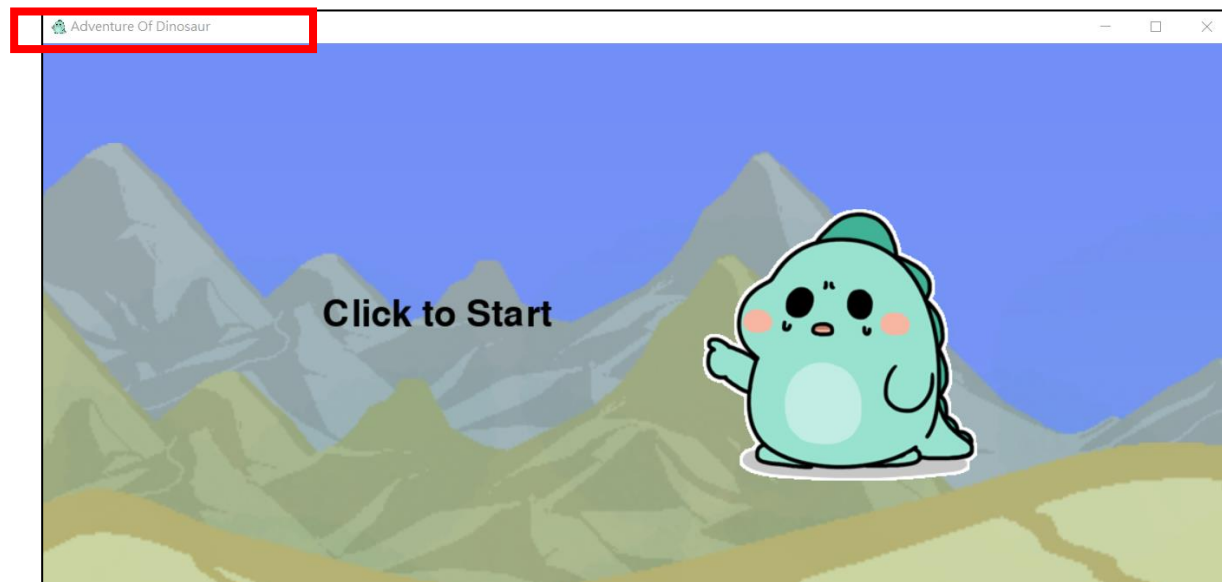
- 設置介面大小
- 載入背景和素材
- 顯示窗口標題(Adventure of Dinosaur)

```
width = 1100; height = 500
```

```
pygame.init()
pygame.mixer.init() # 設置窗口大小

screenIcon = pygame.image.load('./picture/menu.gif')
pygame.display.set_icon(screenIcon)

screen = pygame.display.set_mode((width, height),pygame.RESIZABLE)
pygame.display.set_caption("Adventure Of Dinosaur") # 使用系統自帶的
my_font = pygame.font.SysFont("Times New Roman", 20)
```



實現

- 載入音樂
- 設定背景移動速度

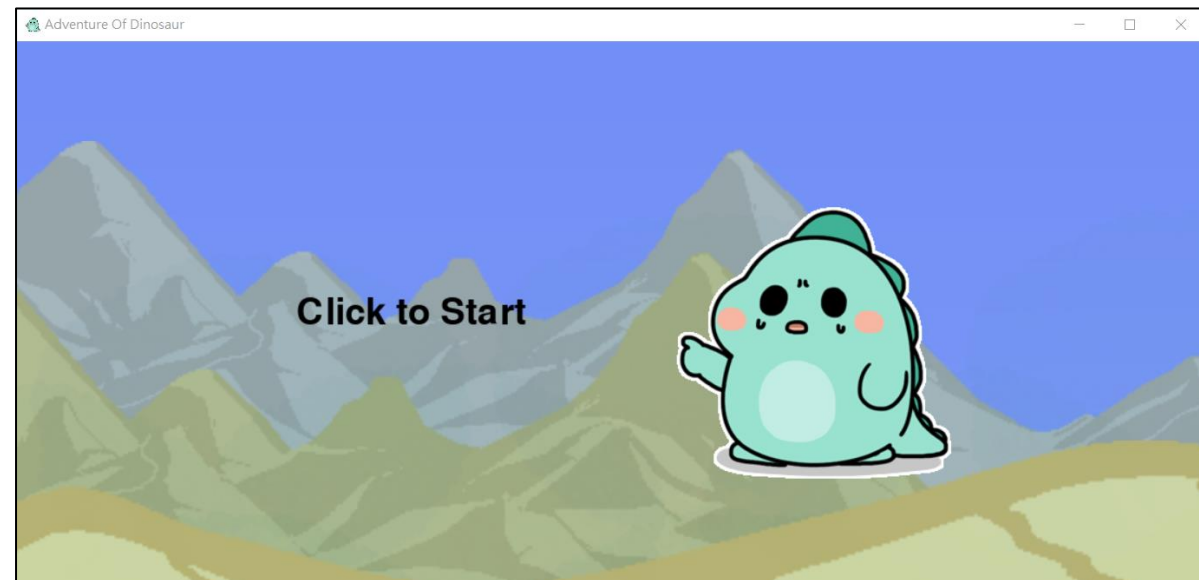
```
global width; global height
game_speed = 0.5 # 背景移動速度
x_pos_bg = 0 # 背景x座標
y_pos_bg = 0 # 背景y座標

bgd = pygame.image.load("./picture/dessert.png")
pic = pygame.image.load("./picture/menu.gif")

if flag == 0:
    pygame.mixer.music.load("menu(2).mp3")

    pygame.mixer.music.play(-1, 0)

run = True
while run:
    # background move
    image_width = bgd.get_width()
    screen.blit(bgd, (x_pos_bg, y_pos_bg))
    screen.blit(bgd, (image_width + x_pos_bg, y_pos_bg))
    x_pos_bg -= game_speed # 使背景移動
    if x_pos_bg <= -image_width:
        screen.blit(bgd, (image_width + x_pos_bg, y_pos_bg))
        x_pos_bg = 0
```



實現

- 設定文字和素材(恐龍)位置
- 點擊畫面進入選角畫面

```
# menu word and picture
death_text_position = (width // 3, height // 2) |
dino_position = (width // 2 + 15, height // 2 - 140)
start_text = Text("Click to Start", 50, 'black', death_text_position)
start_text.draw(screen)
screen.blit(pic, dino_position)

pygame.display.update()

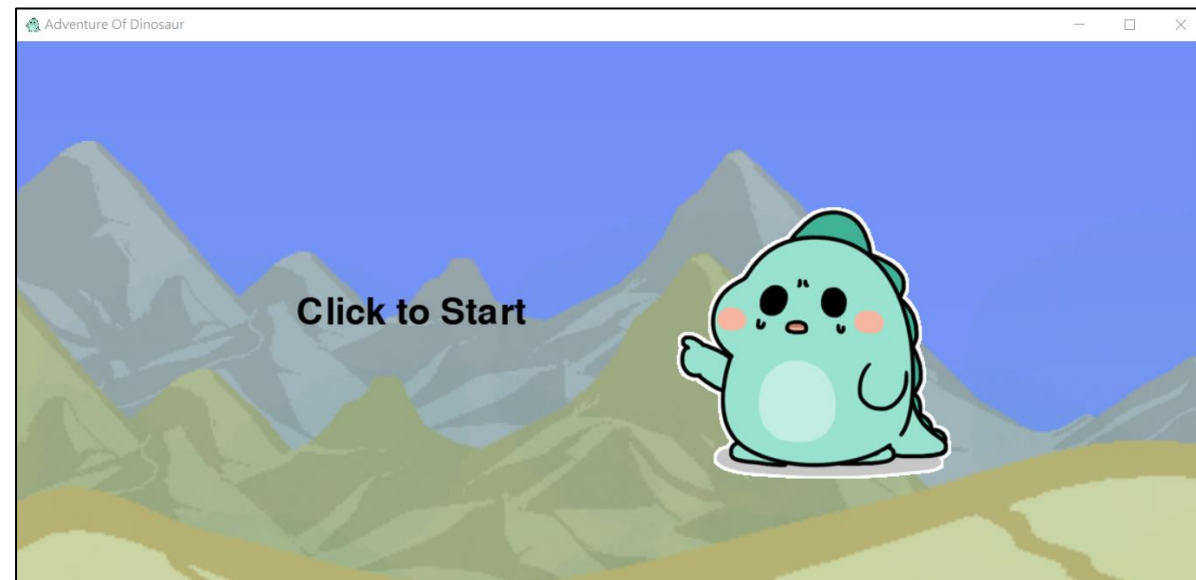
for event in pygame.event.get():

    if event.type == pygame.VIDEORESIZE:
        width = event.w
        height = event.h
        pygame.display.flip()
        menu(1)

    if event.type == pygame.MOUSEBUTTONDOWN:
        run = False

    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()
```

character()



實現

進入選角畫面

- 箭頭顯示位置
- Case紀錄選擇的角色

```
def character(flag = True):
    global width; global height
    start = [pygame.image.load("./dino/DinoStart.png") , pygame.image.load("./dino/b_DinoStart.png")]
    bgd = pygame.image.load("./picture/dessert.png")
    arrow = pygame.image.load("./picture/arrow.png")
    restart = pygame.image.load("./picture/Reset.png")
    clock = pygame.time.Clock()

    run = True

    global case

    while run:
        screen.blit(bgd, [0,0])

        if flag == True:
            start_text = Text("Click to choose character", 50,'black', (width // 2, height // 4) )
            x, y = pygame.mouse.get_pos()

            if x <= (width // 3 + 75) and x >= (width // 3 - 50):
                screen.blit(arrow, [width // 3, height // 3])
                case = 0
            if x <=(width // 2 + 75) and x >= (width // 2 - 50):
                screen.blit(arrow, [width // 2, height // 3])
                case = 1

            screen.blit(start[0], (width // 3, height // 2))
            screen.blit(start[1], (width // 2, height // 2))
```



實現

進入確定遊戲的畫面

- 設定素材(香菇)的移動
- 設定文字位置和內容

```
def home():  
    global width; global height; global case  
    bgd = pygame.image.load("./picture/day.png")  
    cha = pygame.image.load("./picture/msh.png")  
    arrow = pygame.image.load("./picture/arrow.png")  
  
    clock = pygame.time.Clock()  
  
    pos = [(480,250),(500,220),(520,250),(500,220),(480,250)]  
  
    cnt = 0  
  
    run = True  
    while run:  
        clock.tick(5)  
        screen.blit(bgd, [0,0])  
        screen.blit(cha, pos[cnt])  
        cnt = (cnt+1) % 5  
  
        start_text2 = Text("Press enter : Start the game", 50, 'black', (width // 2, height // 3-25))  
        start_text3 = Text("Press esc : Exit the game", 50, 'black', (width // 2, height // 3+20))  
  
        start_text2.draw(screen)  
        start_text3.draw(screen)
```



實現

- 設定 Enter 和 esc 按鍵功能

```
for event in pygame.event.get():
    x, y = pygame.mouse.get_pos()

    if event.type == pygame.VIDEORESIZE:
        width = event.w
        height = event.h
        pygame.display.flip()

    if event.type == pygame.KEYDOWN :
        if event.key == pygame.K_KP_ENTER or event.key == pygame.K_RETURN:
            # print('2')
            play(case)

        if event.key == pygame.K_ESCAPE:
            pygame.quit()
            sys.exit()

    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()
```



實現

- 設定

(1)背景 (2)背景移動速度 (3)分數

```
x_pos_bg -= game_speed # 使背景移動
if x_pos_bg <= -image_width:
    screen.blit(bgd, (image_width + x_pos_bg, y_pos_bg))
    x_pos_bg = 0

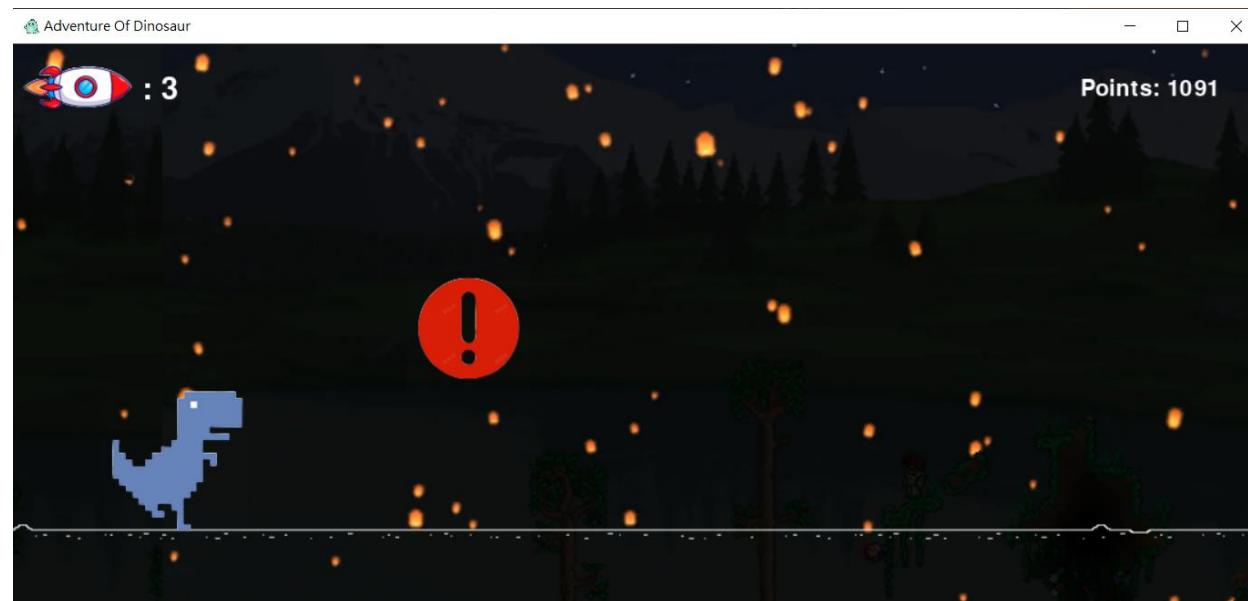
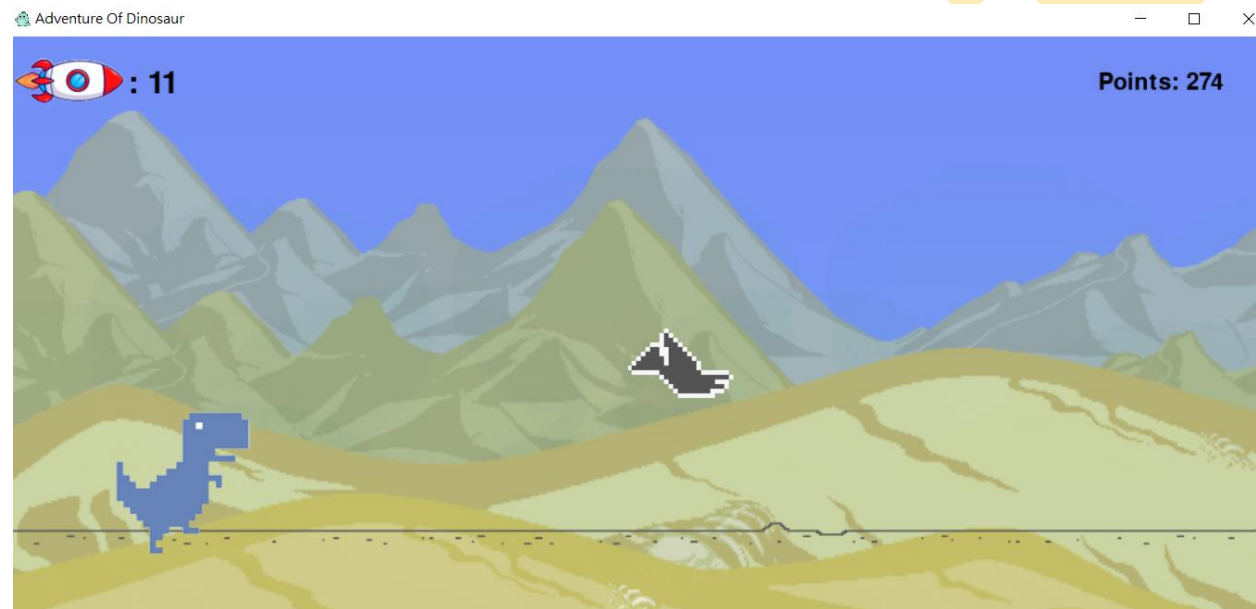
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()

points += 1
if points % 100 == 0:
    game_speed += 1
text_position = (1000, 40)

if change % 2 == 0:
    text = Text("Points: " + str(points), 30, 'black', text_position)
    text_r = Text(': ' + str(rocket_num), 40, 'black', (130, 40))
else:
    text = Text("Points: " + str(points), 30, 'white', text_position)
    text_r = Text(': ' + str(rocket_num), 40, 'white', (130, 40))

screen.blit(rocket, (-45, -45))

text.draw(screen)
text_r.draw(screen)
```



實現

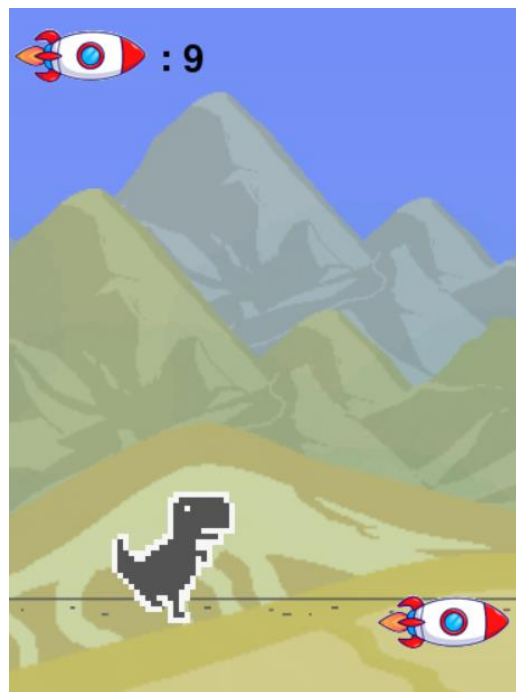
- 設定上、下、右鍵，小恐龍動作

```
if user_input[pygame.K_RIGHT] and not self.dino_shoot and rocket_num > 0:
    if not self.dino_shoot :
        if self.case == 1:
            self.rocket_rect.y = self.dino_rect.y + 20
        else:
            self.rocket_rect.y = self.dino_rect.y + 10
    rocket_num -= 1
    self.dino_shoot = True
    self.shoot_vel = 6

if user_input[pygame.K_UP] and not self.dino_jump:
    self.dino_duck = False
    self.dino_run = False
    self.dino_jump = True
elif user_input[pygame.K_DOWN] and not self.dino_jump:
    self.dino_duck = True
    self.dino_run = False
    self.dino_jump = False
elif not (self.dino_jump or user_input[pygame.K_DOWN]):
    self.dino_duck = False
    self.dino_run = True
    self.dino_jump = False
```



：發射火箭



：跳



：躲避



實現

- 選擇出現物品(障礙物或增加火箭的驚嘆號)
- 處理碰撞事件

```
# p = Prize(prize,random.randint(0,2200), random.randint(0,500))
for obstacle in obstacles:

    obstacle.update() # 障礙物移動
    obstacle.draw(screen) # 更新動畫

    if player.rocket_rect.colliderect(obstacle.rect) and rand != 3:
        # crash_result = pygame.sprite.collide_rect (player.rocket_re

        screen.blit(smoke, (obstacle.rect.x-60, obstacle.rect.y))
        obstacles.pop()
        player.dino_shoot = False
        break

    if player.dino_rect.colliderect(obstacle.rect):

        if rand != 3:
            pygame.time.delay(20) # 延遲0.02秒
            character(False)
        else:
            obstacles.pop()
            flag = True

    if obstacle.rect.x < -obstacle.rect.width:
        obstacles.pop()
```



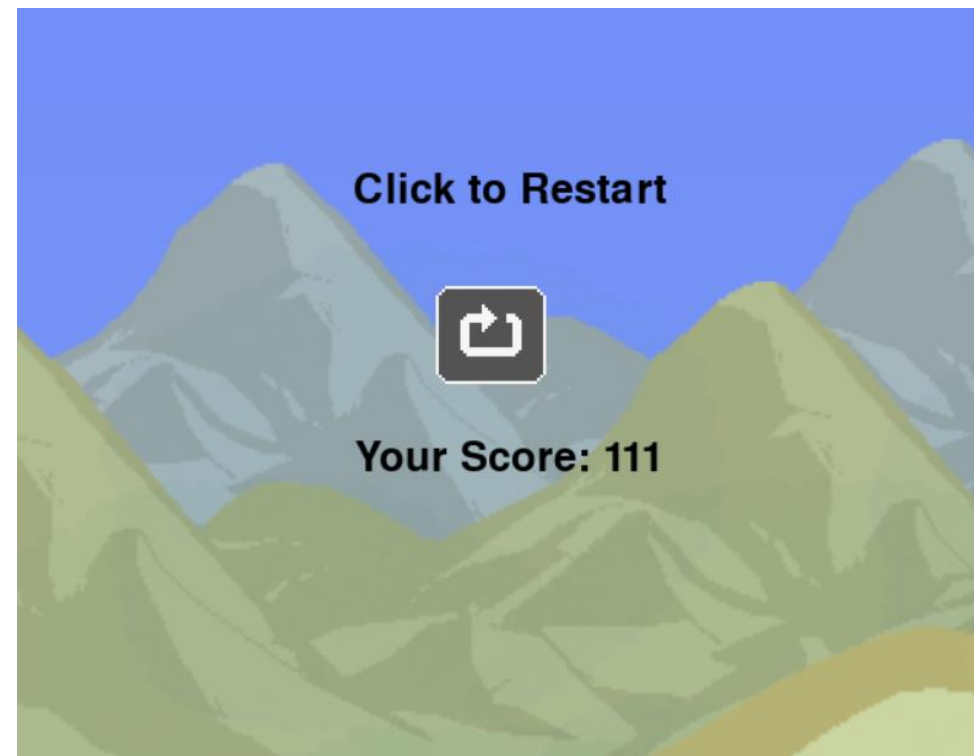
實現

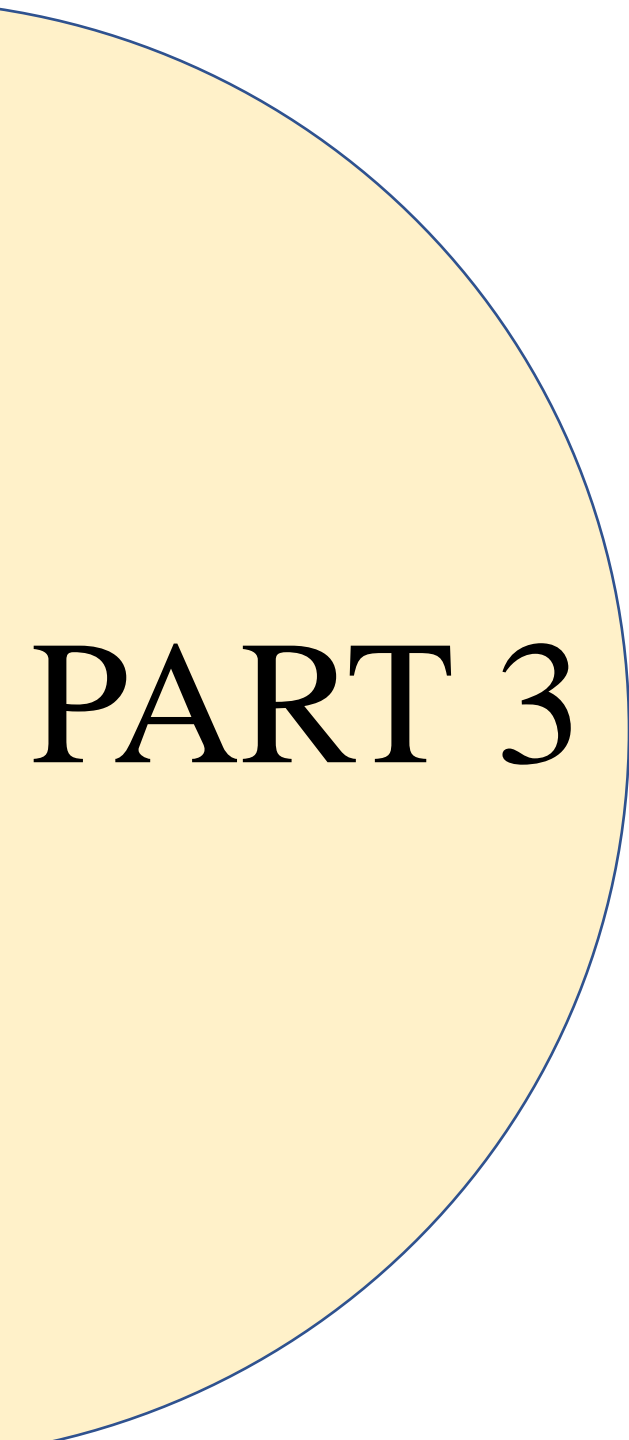
- 碰撞障礙物後，顯示重來和分數

```
elif flag == False:
    score_text_position = (width // 2, height // 2 + 50)
    start_text = Text("Click to Restart", 40, 'BLACK', (width // 2, height // 4))
    score_text = Text("Your Score: " + str(points), 40, 'BLACK', score_text_position)
    score_text.draw(screen)
    screen.blit(restart, (width // 2 - 50, height // 3 + 20))

start_text.draw(screen)

clock.tick(60)
pygame.display.update()
```





PART 3

Demo

