



SiNGULAR 奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

*Join Singular!
Be a super inventor!*

AnimeGames 13



Maker + Coder =



Singular Super Inventor



更隨機生成怪物機制

- 資料結構: Queue
- 就像超市的結帳隊伍。在這個隊伍裡，最先排隊的人會最先結帳離開，新來的人則排在隊伍的後面。
- 就像在遊戲中，最先“排隊”的怪物會是第一個出現，而新出現的怪物則加入到列表的末尾。
- 隨機生成新的怪物加入排隊



Queue小練習

- 一個點心店，學生們排隊等候購買點心。隊伍的前面的學生將會先得到點心並離開隊伍，而新加入的學生將會排在隊伍的尾端。



Queue小練習

```
from collections import deque
```

```
# 創建一個空的隊列
```

```
snack_queue = deque()
```

```
# 向隊列中加入學生
```

```
snack_queue.append("小明")
```

```
snack_queue.append("小華")
```

```
snack_queue.append("小強")
```

```
print(f"初始隊列：{snack_queue}")
```

```
# 第一位學生購買點心並離開隊列
```

```
first_student = snack_queue.popleft()
```

```
print(f"{first_student} 已經購買點心並離開隊列。")
```

```
print(f"現在的隊列：{snack_queue}")
```

```
# 新的學生加入隊列
```

```
snack_queue.append("小美")
```

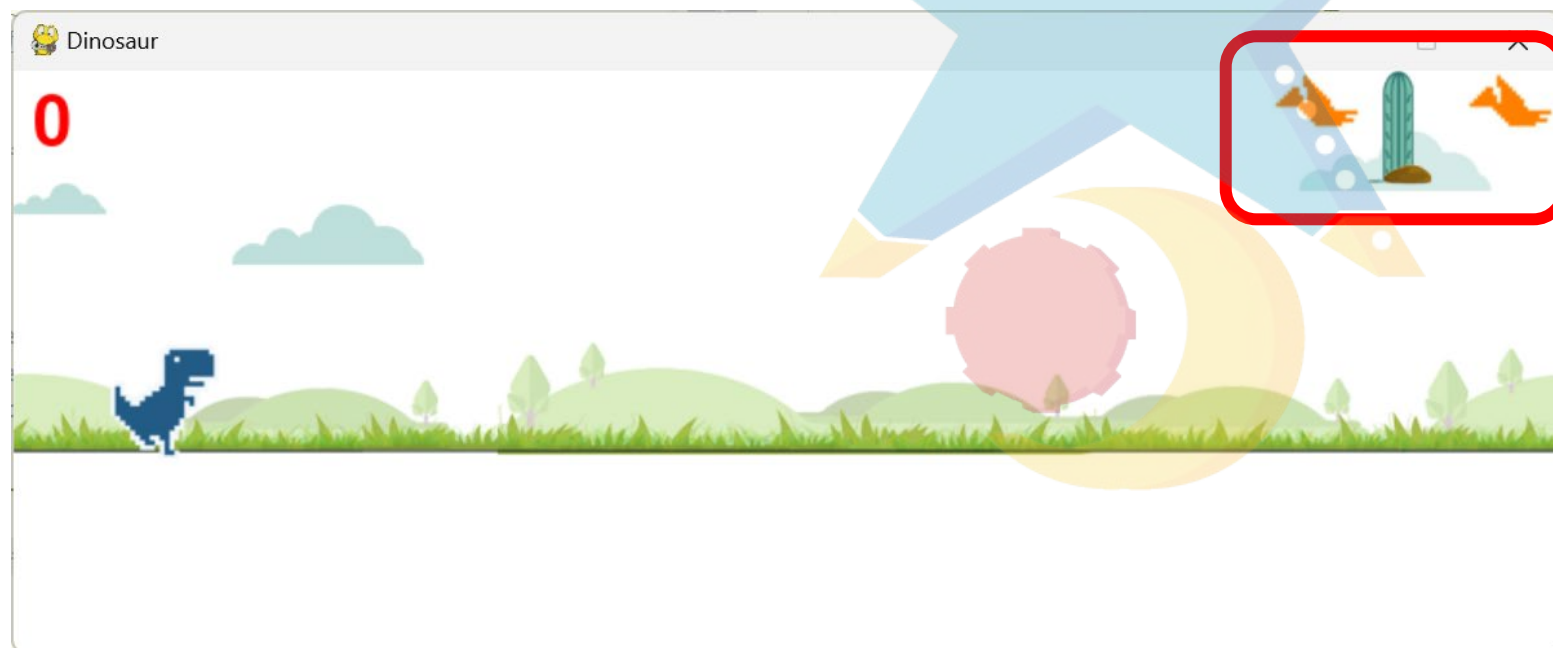
```
print(f"小美加入隊列。")
```

```
print(f"最終隊列：{snack_queue}")
```



任務

- 建立隨機怪物出現的預覽



建立怪物Queue

#####匯入模組#####

```
import pygame
```

```
import sys
```

```
import os
```

```
from pygame.locals import *
```

```
import random
```

```
from collections import deque
```



新增目前場上的怪物清單及間隔計數

#####敵人出現Queue#####

`max_enemies = 3` # 最大敵人數量，可以根據需要修改這個值，總共可以放幾個敵人在隊列中

`enemies_queue = deque(maxlen=max_enemies)` # 使用deque來創建一個有最大長度的隊列

`active_enemies = []`

`enemies_delay = 0` # 敵人出現間隔計數

`enemies_delay_max = 20` # 敵人出現間隔計數最大值

移除多餘程式

```
#####初始化#####  
os.chdir(sys.path[0])  
pygame.init()  
LIMIT_LOW = 140 # 地面高度  
PTERA_LIMIT_LOW = 110 # 翼龍高度  
clock = pygame.time.Clock()  
RED = (255, 0, 0) # 紅色  
# enemy_random = 0 # 隨機敵人
```



移除多於程式

```
#####仙人掌物件#####  
# cacti_x = bg_x - 100 # 障礙物x位置  
# cacti_y = LIMIT_LOW # 障礙物y位置  
# cacti_shift = 10 # 仙人掌移動量  
# cacti_center_x = cacti_x + img_cacti.get_width() / 2 # 障礙物中心x位置  
# cacti_center_y = cacti_y + img_cacti.get_height() / 2 # 障礙物中心y位置  
# cacti_detect_r = max(img_cacti.get_width(), img_cacti.get_height()) / 2  
# 15 # 障礙物偵測半徑  
  
cacti = Cacti(bg_x - 100, LIMIT_LOW, [img_cacti], 10)
```



移除多於程式

```
#####翼龍物件#####  
# ptera_x = bg_x - 100 # 障礙物x位置  
# ptera_y = PTERA_LIMIT_LOW # 障礙物y位置  
# ptera_index = 0 # 翼龍圖片編號  
# ptera_shift = 10 # 翼龍移動量  
# ptera_center_x = ptera_x + img_ptera[0].get_width() / 2 # 翼龍中心x位置  
# ptera_center_y = ptera_y + img_ptera[0].get_height() / 2 # 翼龍中心y位置  
# ptera_detect_r = max(img_ptera[0].get_width(),  
img_ptera[0].get_height()) / 2 - 10 # 翼龍偵測半徑  
  
ptera = Ptera(bg_x - 100, PTERA_LIMIT_LOW, img_ptera, 10)
```



移除多餘程式

#####循環偵測#####

while True:

...省略...

if event.key == **K_RETURN** and gg:

score = 0

gg = **False**

~~# cacti_x = bg_x - 100~~

~~# ptera_x = bg_x - 100~~

ds_y = **LIMIT_LOW**

jumpState = **False**

~~cacti.initial()~~

~~ptera.initial()~~



移除多於程式

#####循環偵測#####

while True:

...省略...

~~if cacti.x <= 0 or ptera.x <= 0:~~

~~score += 1~~

~~# cacti.move()~~

~~# gg = is_hit(ds_center_x, ds_center_y, cacti.center_x, cacti.center_y, cacti.detect_r + ds_detect_r)~~

~~ptera.move()~~

~~gg = is_hit(ds_center_x, ds_center_y, ptera.center_x, ptera.center_y, ptera.detect_r + ds_detect_r)~~

~~# if enemy_random == 0:~~

~~# move_cacti()~~

~~# gg = is_hit(ds_center_x, ds_center_y, cacti_center_x, cacti_center_y, cacti_detect_r + ds_detect_r)~~

~~# else:~~

~~# move_ptera()~~

~~# gg = is_hit(ds_center_x, ds_center_y, ptera_center_x, ptera_center_y, ptera_detect_r + ds_detect_r)~~

pygame.display.update()



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

Maker + Coder = Singular Super Inventor

新增加入隊伍清單指令

#####定義函式#####

...省略...

```
def add_enemy_to_queue():
```

```
    """隨機選擇一個敵人加入到隊列中"""
```

```
    enemy_type = random.choice(["cacti", "ptera"]) # 隨機選擇仙人掌或翼龍
```

```
    if len(enemies_queue) < max_enemies: # 如果隊列中的敵人數量小於最大敵人數量
```

```
        enemies_queue.append(enemy_type) # 將新敵人加入隊列
```

```
# 在畫面右上角顯示目前對列中的敵人縮圖
```

```
for i, enemy in enumerate(enemies_queue):
```

```
    # enumerate(enemies_queue)的意思是將enemies_queue中的元素依序取出，並且給予一個編號
```

```
    if enemy == "cacti":
```

```
        screen.blit(img_cacti, (bg_x - max_enemies * 50 + i * 50, 0))
```

```
    elif enemy == "ptera":
```

```
        screen.blit(img_ptera[0], (bg_x - max_enemies * 50 + i * 50, 0))
```



更新主程式

```
#####循環偵測#####
```

```
while True:
```

```
    clock.tick(20)
```

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

```
        ...省略...
```

```
    if gg:
```

```
        game_over()
```

```
    else:
```

```
        bg_update()
```

```
        move_dinosaur()
```

```
        score_update()
```

```
        add_enemy_to_queue()
```



任務

- 依照Queue召喚怪物並登記到畫面上怪物出現清單



Maker + Coder = Singular Super Inventor



SINGULAR
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

新增召喚怪物指令

#####定義函式#####

...省略...

```
def create_enemies():
```

```
    """隨機決定是否召喚隊列中的敵人"""
```

```
    global active_enemies, score, gg, enemies_delay
```

```
    enemies_delay = (enemies_delay - 1) % enemies_delay_max # 敵人出現間隔計數
```

```
    if len(enemies_queue) > 0 and enemies_delay == 0:
```

```
        enemy_type = enemies_queue.popleft() # 將隊列中的敵人取出
```

```
        if enemy_type == "cacti":
```

```
            active_enemies.append(Cacti(bg_x - 100, LIMIT_LOW, [img_cacti], 10))
```

```
        elif enemy_type == "ptera":
```

```
            active_enemies.append(Ptera(bg_x - 100, PTERA_LIMIT_LOW, img_ptera, 10))
```

```
    for enemy in active_enemies:
```

```
        enemy.move()
```

```
        gg = is_hit(ds_center_x, ds_center_y, enemy.center_x, enemy.center_y, enemy.detect_r + ds_detect_r)
```

```
        if gg:
```

```
            break
```

```
    if enemy.x <= 0:
```

```
        score += 1
```

```
        active_enemies.remove(enemy)
```


更新事件偵測

#####循環偵測#####

```
while True:
```

```
    ...省略...
```

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

```
        ...省略...
```

```
        if event.type == KEYDOWN:
```

```
            if event.key == K_SPACE and ds_y <= LIMIT_LOW: # 判斷恐龍是否在地上
```

```
                jumpState = True # 開啟跳躍狀態
```

```
            elif event.key == K_DOWN:
```

```
                ...省略...
```

```
            if event.key == K_RETURN and gg:
```

```
                score = 0
```

```
                gg = False
```

```
                ds_y = LIMIT_LOW
```

```
                jumpState = False
```

```
                active_enemies.clear()
```



更新主程式

```
#####循環偵測#####
```

```
while True:
```

```
    clock.tick(20)
```

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

```
        ...省略...
```

```
    if gg:
```

```
        game_over()
```

```
    else:
```

```
        bg_update()
```

```
        move_dinosaur()
```

```
        score_update()
```

```
        add_enemy_to_queue()
```

```
        create_enemies()
```



任務

- 增加難度

```
#####初始化#####  
os.chdir(sys.path[0])  
pygame.init()  
LIMIT_LOW = 140 # 地面高度  
PTERA_LIMIT_LOW = 110 # 翼龍高度  
clock = pygame.time.Clock()  
RED = (255, 0, 0) # 紅色  
FPS = 20 # 遊戲更新畫面的時間  
level_up = False # 升級狀態
```



難度升級

```
#####循環偵測#####
```

```
while True:
```

```
    clock.tick(FPS)
```

```
    if score % 5 == 0 and score != 0 and not level_up:
```

```
        # 每得到5分，敵人出現間隔減少1，20為最小間隔
```

```
        enemies_delay_max = max(20, enemies_delay_max - 1)
```

```
        if enemies_delay_max == 20:
```

```
            # 如果敵人出現間隔已經最小，則遊戲更新畫面的時間加快
```

```
            FPS += 10
```

```
            level_up = True # 避免重複升級
```

```
    elif score % 5 != 0:
```

```
        level_up = False # 重置升級狀態
```



更新重新開始

#####循環偵測#####

```
while True:
```

```
    ...省略...
```

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

```
        ...省略...
```

```
            if event.key == K_RETURN and gg:
```

```
                score = 0
```

```
                gg = False
```

```
                ds_y = LIMIT_LOW
```

```
                jumpState = False
```

```
                active_enemies.clear()
```

```
                FPS = 20
```

