



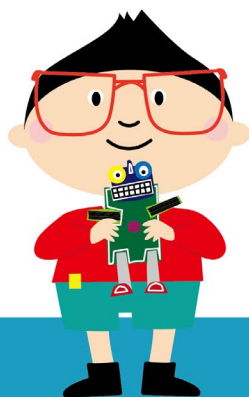
SiNGULAR 奇點創意

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 08



Maker + Coder =



Singular Super Inventor





跑跑小恐龍

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

ADV-06

- 新增 adv-06 資料夾
- 在 adv-06 資料夾內新增 image 資料夾
- 在 adv-06 資料夾內新增程式檔案
- 將slack模板貼近程式檔案
- 下載slack的素材並放入adv-06 資料夾/image資料夾裡面

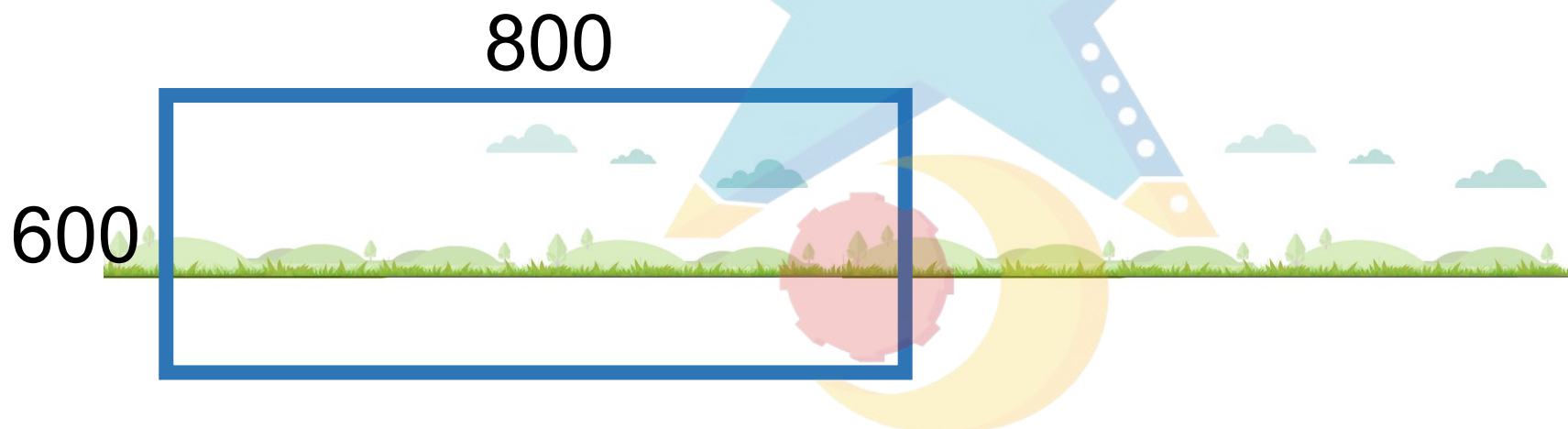
任務1

- 請設計一個捲動的背景。



捲動背景

- 利用兩張一樣的背景，利用背景在視窗中位移，制造出捲動的效果。



如何讓背景循環重複呢？

- 在畫背景的時候會一直改變x座標
 - 等到x座標超出視窗之後要重置x
 - 有哪一些方式可以讓x在視窗內循環撥放呢？
-
- 小練習：
 - 印出一個數字，從0~10循環重複顯示



循環數字方法

方法一：使用計數器

```
i = 0
while True:
    print(i)
    if i >= 10:
        i = 0
    else:
        i += 1

    time.sleep(0.3)
```

方法二：使用for迴圈

```
while True:
    for i in range(11):
        print(i)
        time.sleep(0.3)
```

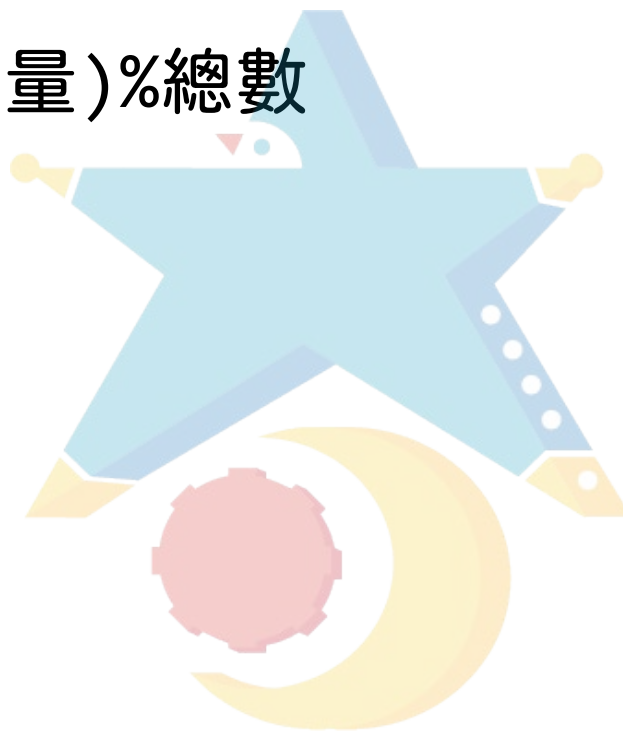
方法三：使用%餘數

```
cnt = 0
while True:
    print(cnt)
    cnt = (cnt + 1) % 11
    time.sleep(0.3)
```



循環演算法

- **新數值** = (舊數值 - 變動量) % 總數



匯入模組

#####匯入模組#####

```
import pygame
import sys
import os
```



初始化

```
#####初始化#####  
os.chdir(sys.path[0])  
pygame.init()  
LIMIT_LOW = 140 # 地面高度  
clock = pygame.time.Clock()
```

載入圖片

#####載入圖片物件#####

```
img = pygame.image.load("image/bg.png") # 加載背景
```

```
bg_x = img.get_width()
```

```
bg_y = img.get_height()
```

```
bg_roll_x = 0 # 背景圖片滾動位置
```



建立視窗物件

#####建立視窗#####

```
screen = pygame.display.set_mode([bg_x, bg_y]) # 設定窗口  
pygame.display.set_caption("Dinosaur")
```

新增背景更新指令

#####定義函式#####

```
def bg_update():  
    """更新背景"""
```

```
    global bg_roll_x
```

```
    bg_roll_x = (bg_roll_x - 10) % bg_x # 背景移動
```

```
    screen.blit(img, (bg_roll_x - bg_x, 0)) # 背景圖左移
```

```
    screen.blit(img, (bg_roll_x, 0)) # 背景圖接續顯示
```

主程式

```
#####循環偵測#####
```

```
while True:
```

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

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

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

```
            sys.exit()
```

```
    bg_update()
```

```
    pygame.display.update()
```



任務2

- 請設計恐龍跑步的效果。



恐龍圖片

#####載入圖片物件#####

```
img = pygame.image.load("image/bg.png") # 加載背景
```

```
img_dinosaur = [ # 加載恐龍
```

```
    pygame.image.load("image/小恐龍1.png"),
```

```
    pygame.image.load("image/小恐龍2.png"),
```

```
]
```

...省略...



恐龍參數

#####恐龍物件#####

ds_x = 50 # 恐龍x位置

ds_y = LIMIT_LOW # 恐龍y位置

ds_index = 0 # 恐龍圖片編號

恐龍動作更新指令

- 請製作一個指令 `move_dinosaur()`
- 指令可以讓恐龍循環顯示2張圖片並畫在視窗上



恐龍動作更新指令

- 請製作一個指令`move_dinosaur()`
- 指令可以讓恐龍循環顯示2張圖片並畫在視窗上

#####定義函式#####

...省略...

```
def move_dinosaur():  
    """移動恐龍"""  
    global ds_y, ds_index  
  
    # 計算恐龍圖片編號  
    ds_index = (ds_index - 1) % len(img_dinosaur)  
    screen.blit(img_dinosaur[ds_index], (ds_x, ds_y))
```



更新主程式

#####循環偵測#####

```
while True:
```

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

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

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

```
            pygame.quit()
```

```
            sys.exit()
```

```
    bg_update()
```

```
    move_dinosaur()
```

```
    pygame.display.update()
```



任務3

- 請設計恐龍跳躍的效果。



更新恐龍參數

#####恐龍物件#####

...省略...

ds_index = 0 # 恐龍圖片編號

jumpState = False # 跳躍狀態

jumpValue = 0 # 跳躍值

jump_height = 13 # 跳躍高度

偵測鍵盤事件

#####匯入模組#####

```
import pygame
```

```
import sys
```

```
import os
```

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



偵測鍵盤事件

#####循環偵測#####

```
while True:
```

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

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

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

```
            ...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
```

```
    ...省略...
```

更新恐龍指令

#####定義函式#####

...省略...

```
def move_dinosaur():
```

```
    """移動恐龍"""
```

```
    global ds_y, jumpState, jumpValue, ds_index
```

```
    if jumpState: # 可以起跳
```

```
        if ds_y >= LIMIT_LOW:
```

```
            jumpValue = -jump_height
```

```
        if ds_y <= 0:
```

```
            jumpValue = jump_height
```

```
        ds_y += jumpValue
```

```
    if ds_y >= LIMIT_LOW:
```

```
        jumpState = False
```

```
        ds_y = LIMIT_LOW # 確保恐龍回到地面
```

...省略...



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

增加跳躍引力

```
def move_dinosaur():  
    """移動恐龍"""  
    global ds_y, jumpState, jumpValue, ds_index  
  
    if jumpState: # 可以起跳  
        ...省略...  
        ds_y += jumpValue  
  
        # 平滑跳躍  
        jumpValue += 1 # 上升速度逐漸減小  
  
        ...省略...
```



任務4

- 請設計仙人掌移動的效果。



仙人掌圖片載入

#####載入圖片物件#####

```
img = pygame.image.load("image/bg.png") # 加載背景
```

```
img_dinosaur = [ # 加載恐龍
```

```
    pygame.image.load("image/小恐龍1.png"),
```

```
    pygame.image.load("image/小恐龍2.png"),
```

```
]
```

```
img_cacti = pygame.image.load("image/cacti.png") # 加載仙人掌
```

```
bg_x = img.get_width()
```

```
...省略...
```



設定仙人掌參數

#####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



仙人掌更新指令

- 請製作一個指令`move_cacti()`
- 指令可以讓仙人掌在畫面上從右到左循環顯示



仙人掌更新指令

- 請製作一個指令`move_cacti()`
- 指令可以讓仙人掌在畫面上從右到左循環顯示

#####定義函式#####

...省略...

```
def move_cacti():  
    """移動仙人掌"""  
    global cacti_x
```

```
    cacti_x = (cacti_x - cacti_shift) % (bg_x - 100) # 仙人掌移動  
    screen.blit(img_cacti, (cacti_x, cacti_y))
```



更新主程式

```
#####循環偵測#####
```

```
while True:
```

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

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

```
    bg_update()
```

```
    move_cacti()
```

```
    move_dinosaur()
```

```
    pygame.display.update()
```

