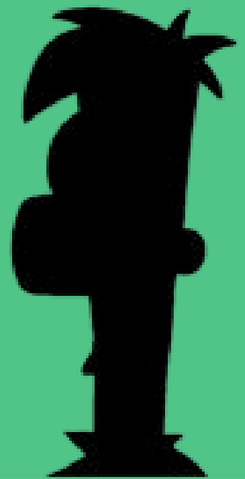
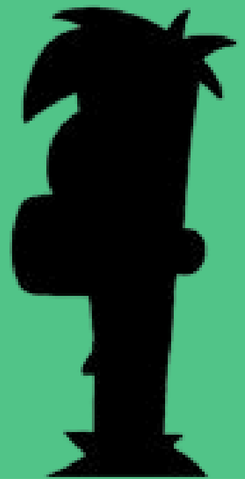
NTNU CSIE CAMP

PYGAME



NTNU CSIE CAMP

PYGAME



目錄

Pygame 介紹

Q: 什麼是 Pygame?

A: Pygame 是一個用於 Python 程式語言的 2D 遊戲開發庫。

Pygame 有哪些特點?

- 圖形處理
- 聲音處理
- 事件處理
- 碰撞檢測
- 時間管理
- 跨平台
- 開源且免費
- 簡單易用

做遊戲之前必須具備的觀念

Q: 遊戲畫面怎麼出現的?

A: 遊戲的畫面是由渲染而來

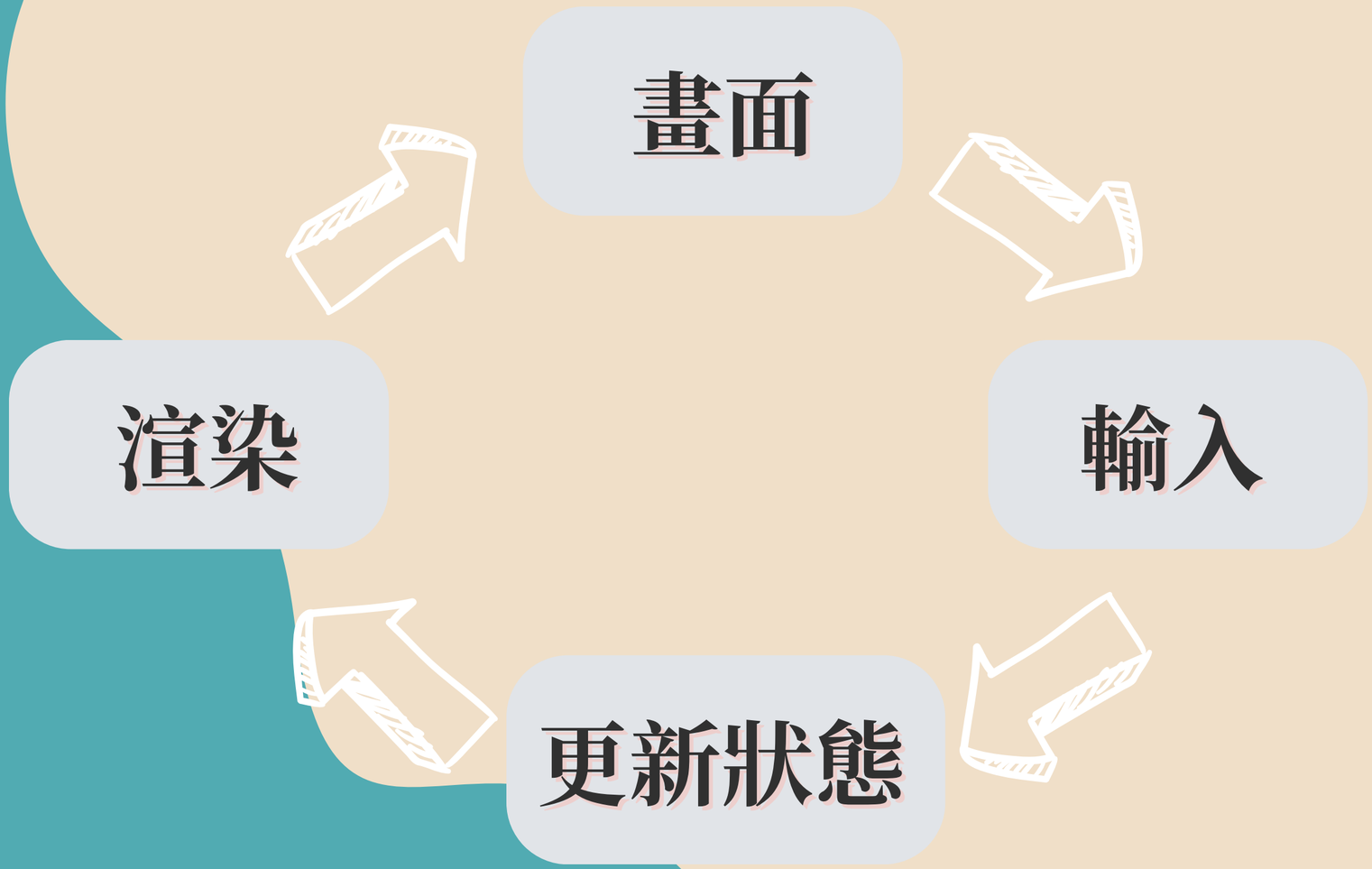
什麼是渲染?

圖片 → 電腦計算 → 顯示畫面

Q: 遊戲的畫面怎麼能一直變動?

A: 遊戲畫面由多張圖片快速串連起來

遊戲循環



幀數 FPS

遊戲畫面的基本構成

UI 層

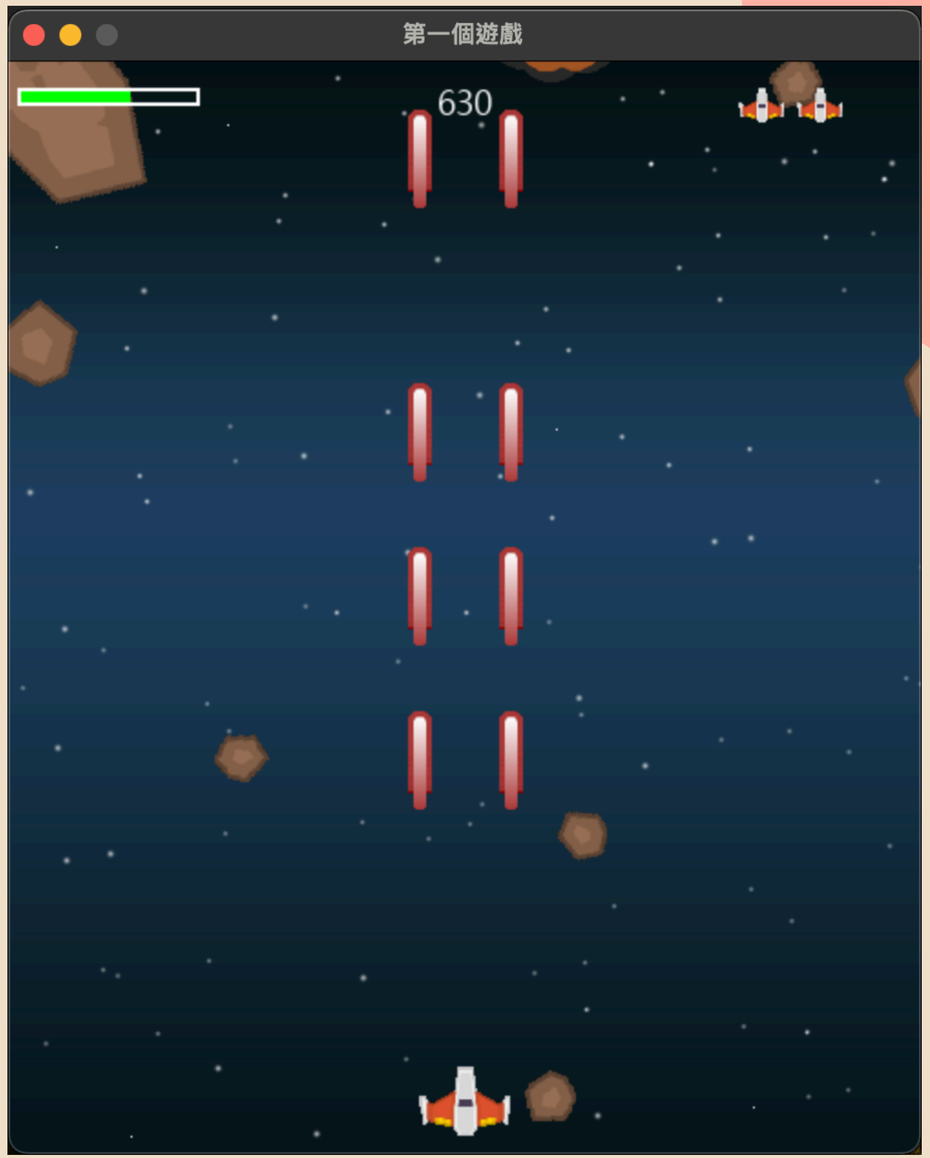
特效層

角色層

背景層

上層

下層



遊戲設計的流程

ex: 背景圖、分數　　ex: 子彈擊中敵人

思考遊戲規則 → 主畫面設計 → 元件功能設計 → 元件間的互動 → 遊戲的優化

ex: 遊戲勝利條件　ex: 角色發射子彈　ex: 背景樂、音效

Pygame的基本架構

Pygame 的引入與啟動

- 要記得引入Pygame套件
```
import pygame
```
- 常用來檢測退出事件
```
import sys
```
- 這個語法是用來初始化
```
pygame.init()
```

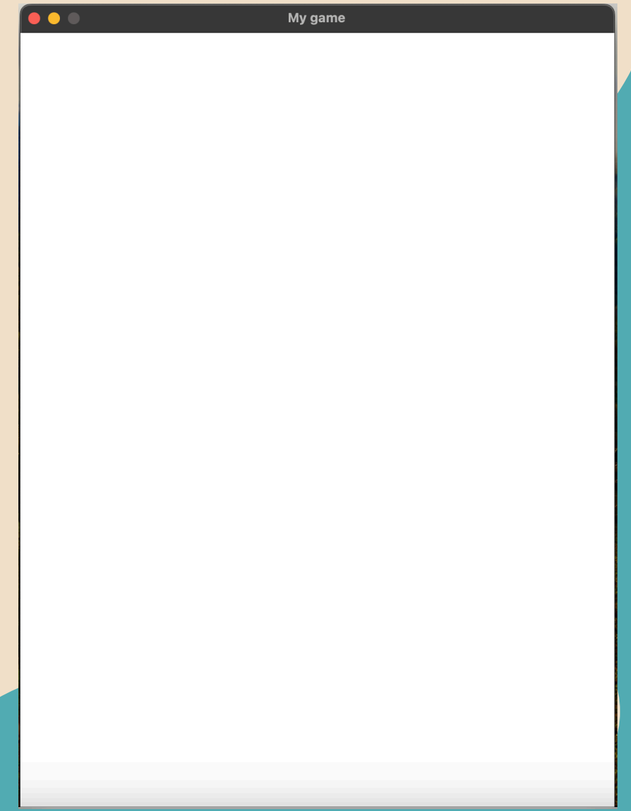
遊戲視窗大小

（寬度, 高度）

```
# 視窗的大小
screen = pygame.display.set_mode((600,800))
```

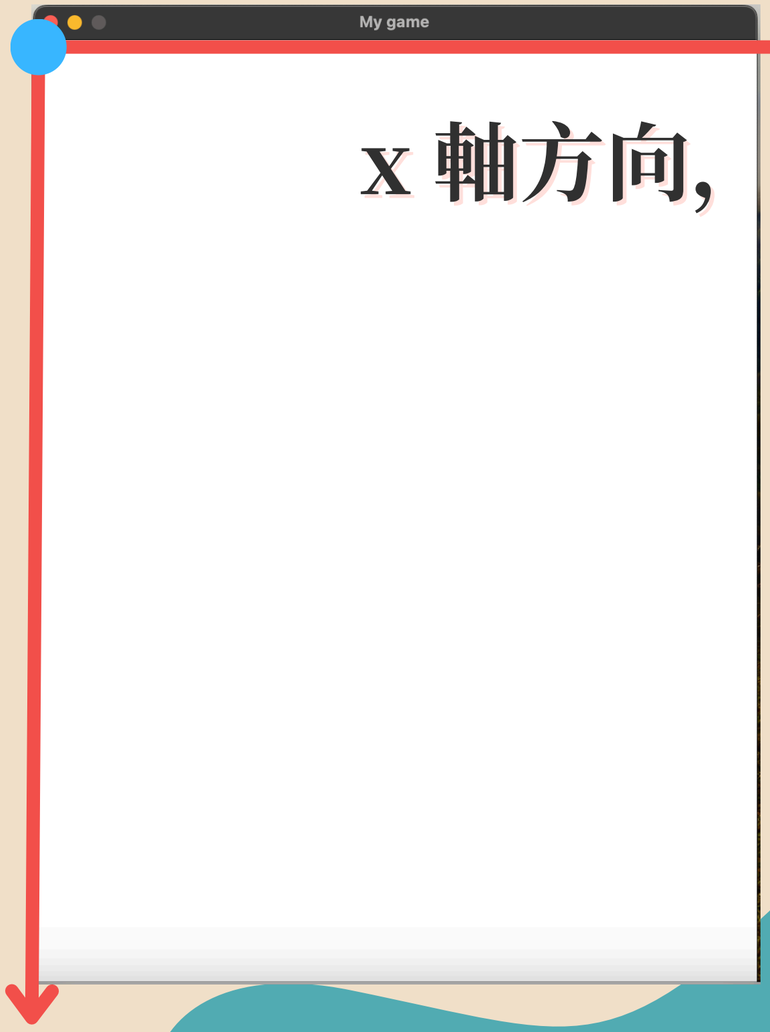
（寬度）

（高度）



遊戲的座標

起始點(0,0)



x軸

x 軸方向，向右遞增

y 軸方向，向下遞增

y 軸

遊戲視窗大小

```
# 視窗的大小
screen = pygame.display.set_mode((600,800))
```

```
# 寬度與長度
WIDTH = 600
HEIGHT = 800
```

原本的寬度和長度可以用變數來更好的呈現

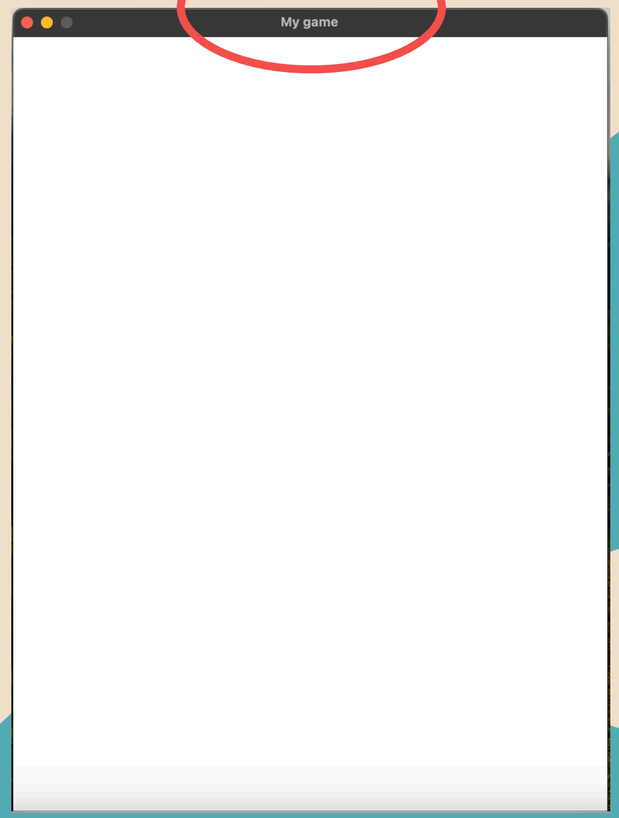
```
screen = pygame.display.set_mode((WIDTH, HEIGHT))
```

遊戲標題

My game

遊戲標題會出現在這裡

```
# 遊戲的標題
pygame.display.set_caption("My game")
```



遊戲的主迴圈

- 遊戲的畫面需要一直變化，所以需要用無限迴圈來不段更新遊戲畫面

```
# 遊戲迴圈
# 創建一個時鐘
clock = pygame.time.Clock()
while True:
    # 處理遊戲的幀數
    clock.tick(60)
    # 取得使用者的所有輸入
    for event in pygame.event.get():
        # 當使用者按下關閉視窗，便退出遊戲
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # 遊戲的畫面更新
    pygame.display.update()
```

遊戲的幀數

時鐘用來處理後面想要的幀數

這裡處理遊戲的幀數，設定為每秒六十幀

```
# 遊戲迴圈
# 創建一個時鐘
clock = pygame.time.Clock()
while True:
    # 處理遊戲的幀數
    clock.tick(60)
    # 取得使用者的所有輸入
    for event in pygame.event.get():
        # 當使用者按下關閉視窗，便退出遊戲
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # 遊戲的畫面更新
    pygame.display.update()
```

取得使用者的輸入

```
# 遊戲迴圈
# 創建一個時鐘
clock = pygame.time.Clock()
while True:
    # 處理遊戲的幀數
    clock.tick(60)
    # 取得使用者的所有輸入
    for event in pygame.event.get():
        # 當使用者按下關閉視窗，便退出遊戲
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # 遊戲的畫面更新
    pygame.display.update()
```

得到使用者的輸入

判斷是否點擊關閉視窗

取得使用者的輸入

```python
# 遊戲迴圈
# 創建一個時鐘
clock = pygame.time.Clock()
while True:
    # 處理遊戲的幀數
    clock.tick(60)
    # 取得使用者的所有輸入
    for event in pygame.event.get():
        # 當使用者按下關閉視窗，便退出遊戲
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # 遊戲的畫面更新
    pygame.display.update()
```

退出遊戲

更新遊戲畫面

繪製背景的顏色

這裡元組內依序是（紅，綠，藍）

```
screen.fill((255, 255, 255))
```

為了提高可讀性和易於維護，會這樣寫

```
WHITE = (255, 255, 255)
```

```
screen.fill(WHITE)
```

如果想要找其他配色可以點下方的連結

點我

Pygame文本創建

文字創建

```
# 創建文字格式
font = pygame.font.Font(None, 36)  # 使用預設的字體，並設置字體大小

# 創建文字
text = font.render("Hello, Pygame!", True, BLACK)

# 得到文字的矩形位置
text_rect = text.get_rect()

# 設置文字的位置 (x, y)
text_rect.center = (300, 400)

# 繪製出文字
screen.blit(text, text_rect)
```

文字創建

特別說明 font.render 括號中代表的東西

依序為 (文字，是否開啟抗鋸齒，顏色)

```
# 創建文字
text = font.render("Hello, Pygame!", True, BLACK)
```

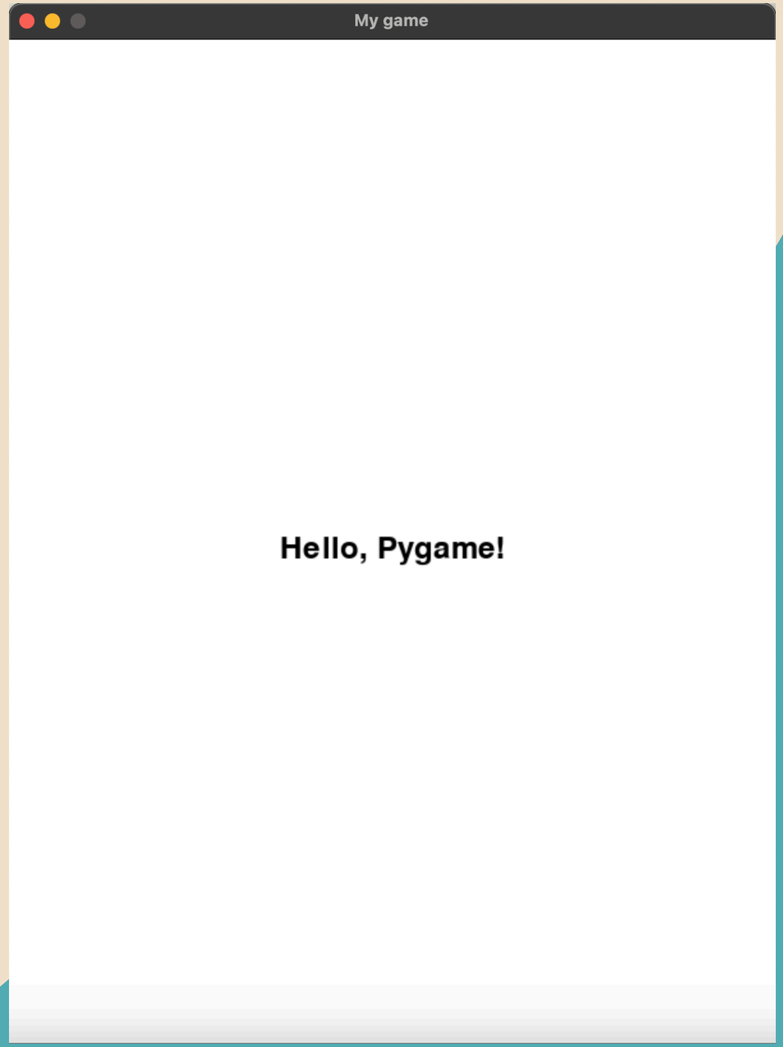
抗鋸齒讓原本顯示斜線或曲線時的鋸齒狀圖像
變得自然和平滑

文字創建練習（一）

從資料夾中，打開“2.text.py”檔案

在這行下面開始打，執行後做出右圖

```
39      # 遊戲的畫面更新
40      pygame.display.update()
41
42      # 視窗顏色
43      # screen.fill((255, 255, 255))
44      screen.fill(WHITE)
45
46      # 以下請寫出文字創建的函式
47
48
49      # 遊戲的畫面更新
50      pygame.display.update()
```



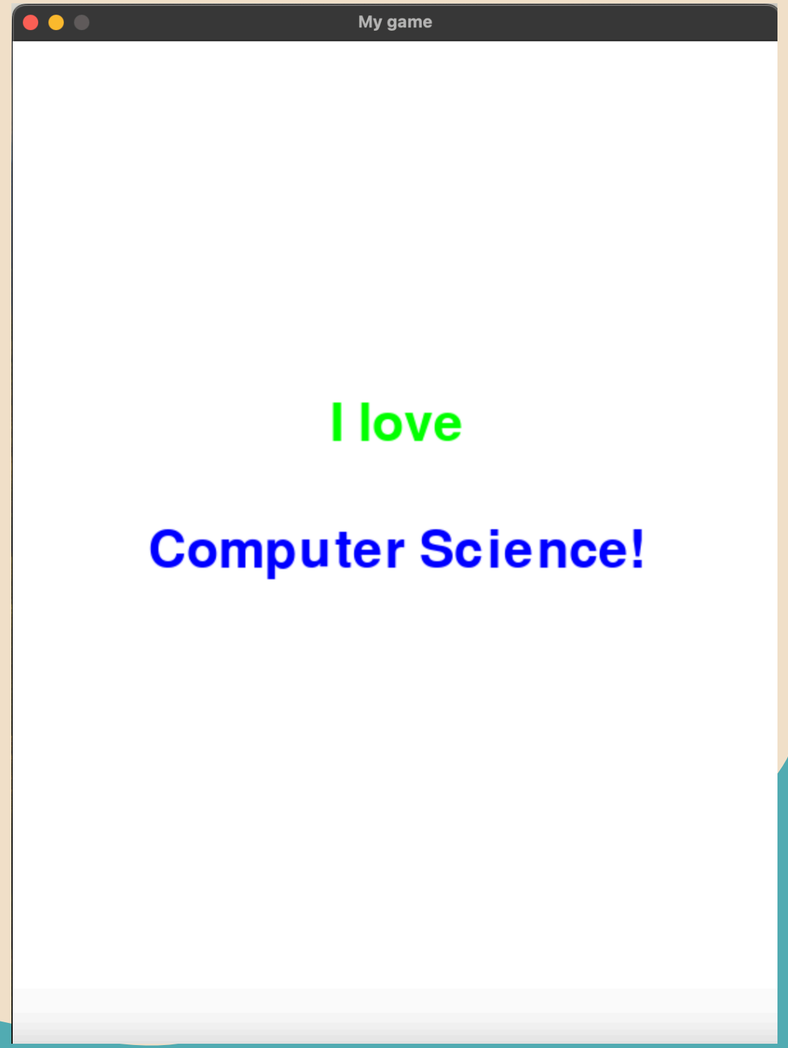
文字創建練習（二）

試試看做出右邊的畫面

字體大小：60

其它的改動該在哪裡改變呢?

ps: 顏色的變數已經在
程式碼的開頭了喔!



Pygame繪製圖形

繪製矩形

依序為 (畫布，顏色，[x 座標, y 座標, 寬度, 高度]，線寬)

```
pygame.draw.rect(screen, BLACK, [100, 100, 200, 100], 2)
```

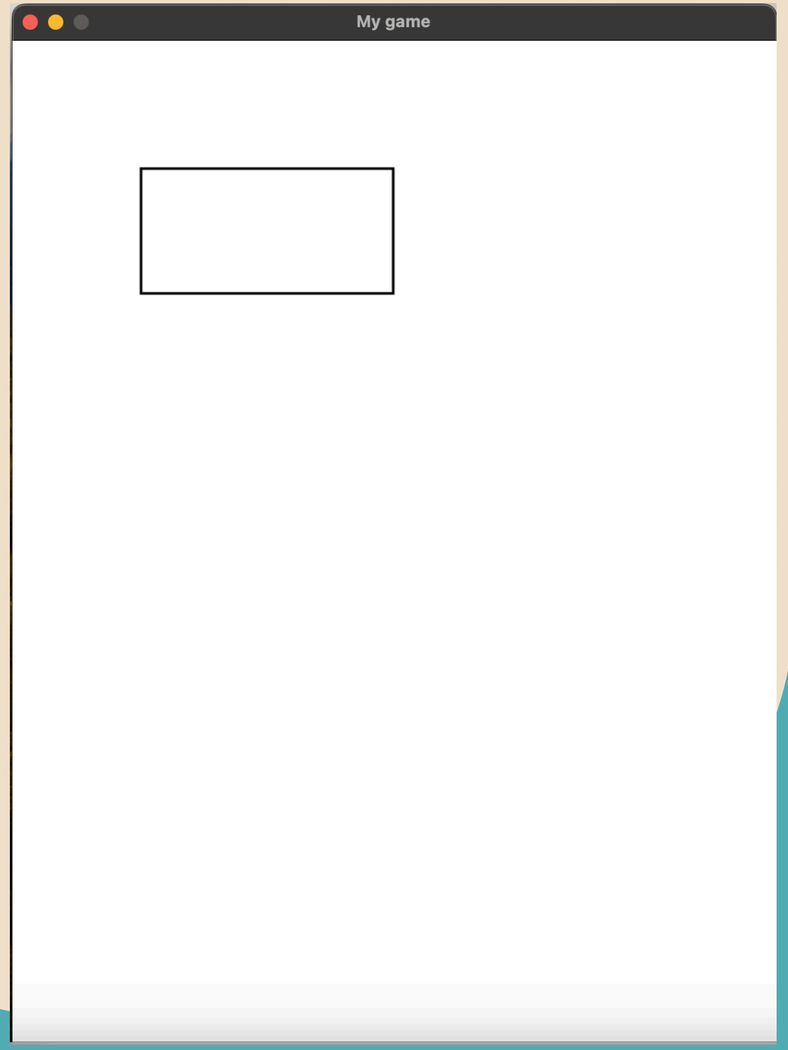
如果想畫實心矩形，就不要加上線寬

記得 (x, y) 是矩形的左上角

繪製矩形練習(一)

從資料夾中，打
開“3.rectangle.py”檔案

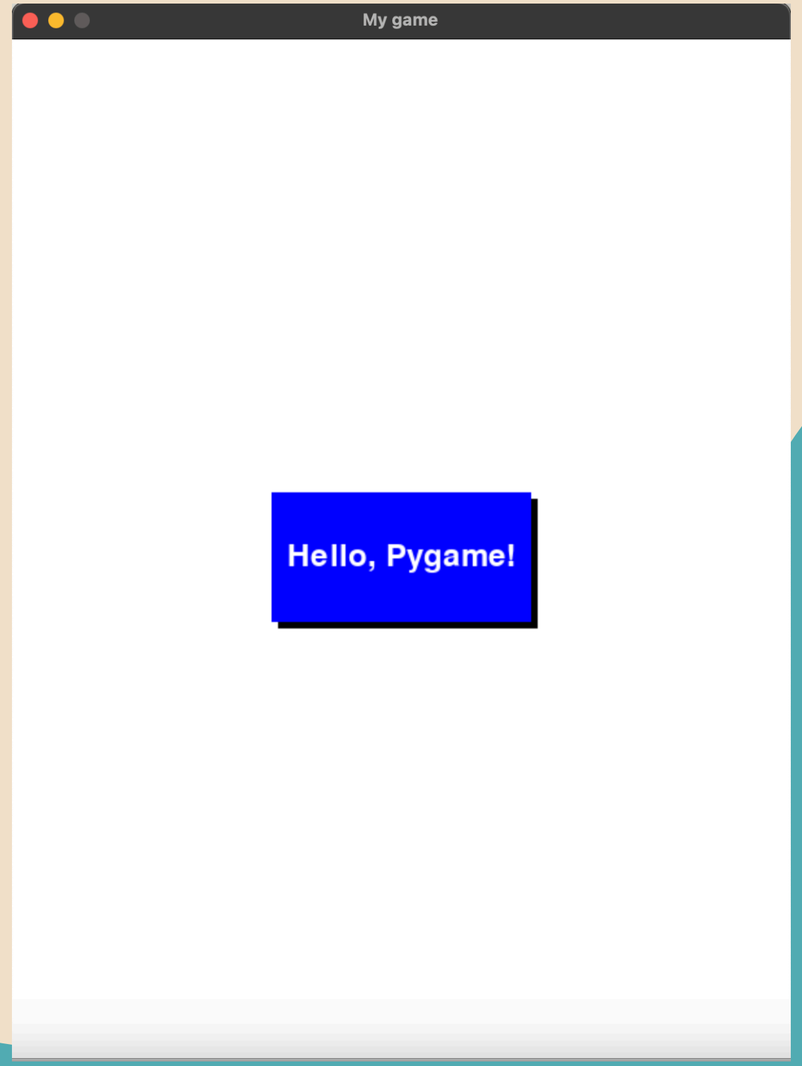
先試著畫出右邊的矩形吧!



繪製矩形練習(二)

結合前面的文字做出右邊的畫面吧!

(1) 試著調整線寬，做出實心矩形
(2)如何做出「視覺」上的立體呢?



繪製圓形

依序為 (畫布，顏色，(x 座標, y 座標)，半徑，線寬)

```
pygame.draw.circle(screen, BLACK, (300, 150), 50, 2)
```

如果想畫實心圓，就不要加上線寬

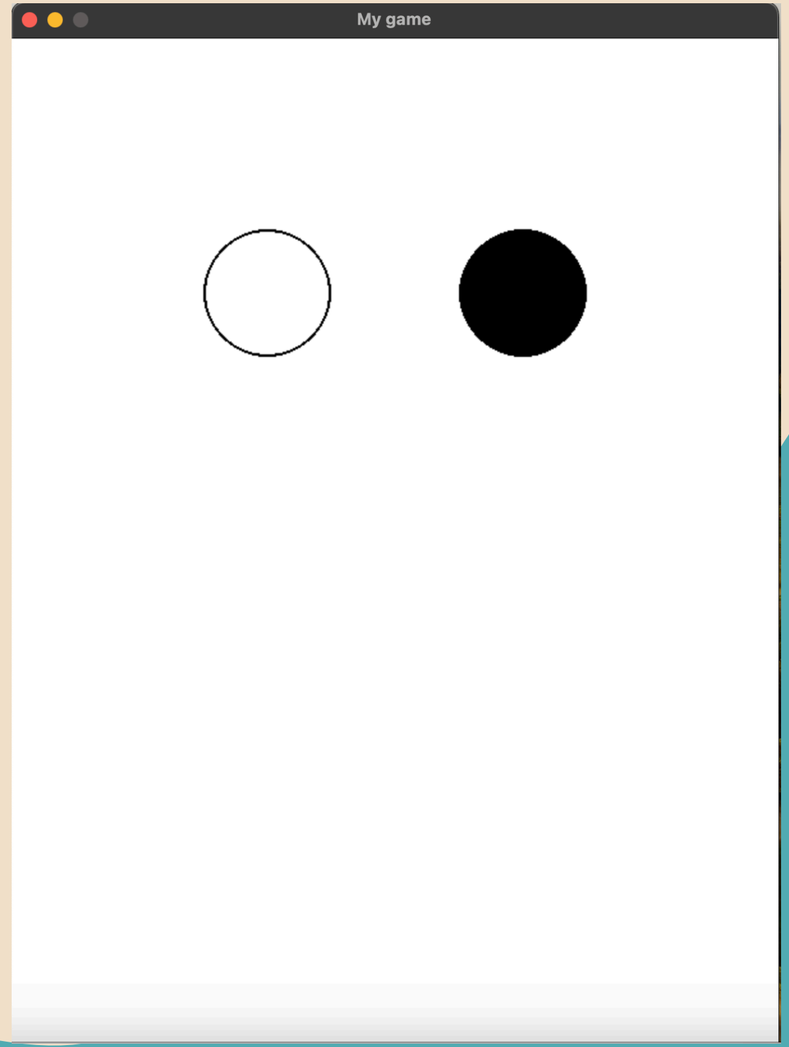
```
pygame.draw.circle(screen, BLACK, (500, 150), 50)
```

注意：圓形的 (x, y) 是「圓心座標」

繪製圓形練習（一）

從資料夾中，打
開“4.circle.py”檔案

先試著畫出右邊的兩個圓形吧!



繪製橢圓形

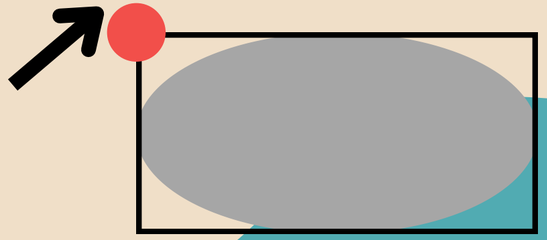
依序為 (畫布，顏色，[x 座標, y 座標, x軸半徑, y軸半徑]，線寬)

```
pygame.draw.ellipse(screen, BLACK, [100, 300, 150, 80], 2)
```

如果想畫實心橢圓，就不要加上線寬

```
pygame.draw.ellipse(screen, BLACK, [350, 300, 150, 80])
```

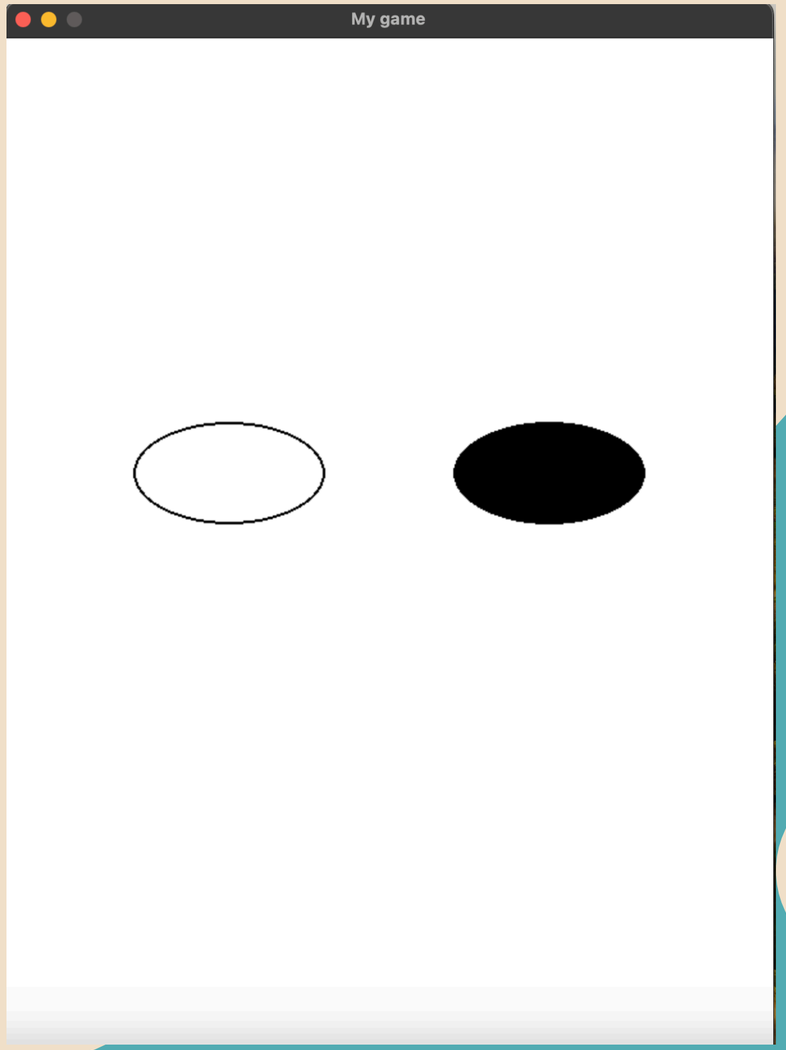
注意：橢圓形的 (x, y) 在這



繪製橢圓形練習（一）

從資料夾中，打
開“5.ellipse.py”檔案

先試著畫出右邊的兩個橢圓形吧!



繪製直線

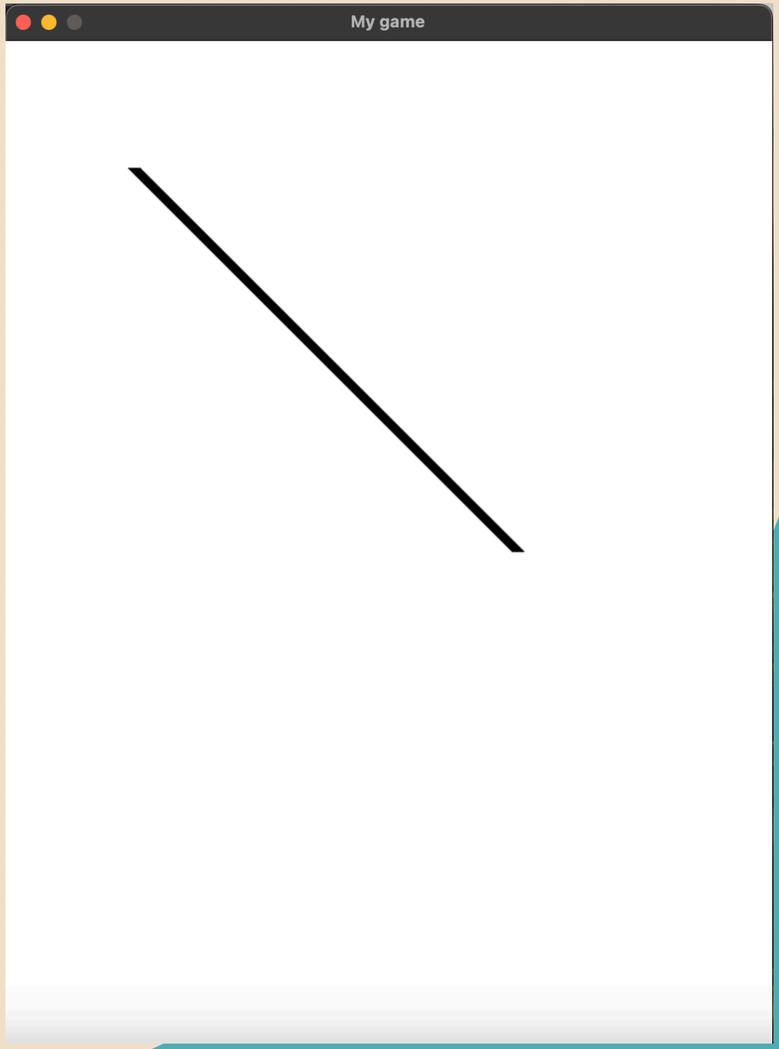
依序為 (畫布, 顏色, 起點 (x, y), 終點 (x, y), 線寬)

```
pygame.draw.line(screen, BLACK, (100, 600), (400, 500), 10)
```

繪製直線練習（一）

從資料夾中，打開「6.line.py」檔案

先試著畫出右邊的直線吧！



繪製圓弧

依序為 (畫布, 顏色, 起始角度, 結束角度, 線寬)

```
# 繪製圓弧
rect = pygame.Rect(100, 300, 400, 300)  # 矩形範圍（x座標, y座標, 長, 寬）
start_angle = 0  # 起始角度（度數）
end_angle = 180  # 結束角度（度數）
line_thickness = 3  # 弧線厚度

# 將度數轉換為弧度
start_angle_rad = math.radians(start_angle)
end_angle_rad = math.radians(end_angle)

pygame.draw.arc(screen, BLACK, rect, start_angle_rad, end_angle_rad, line_thickness)
```

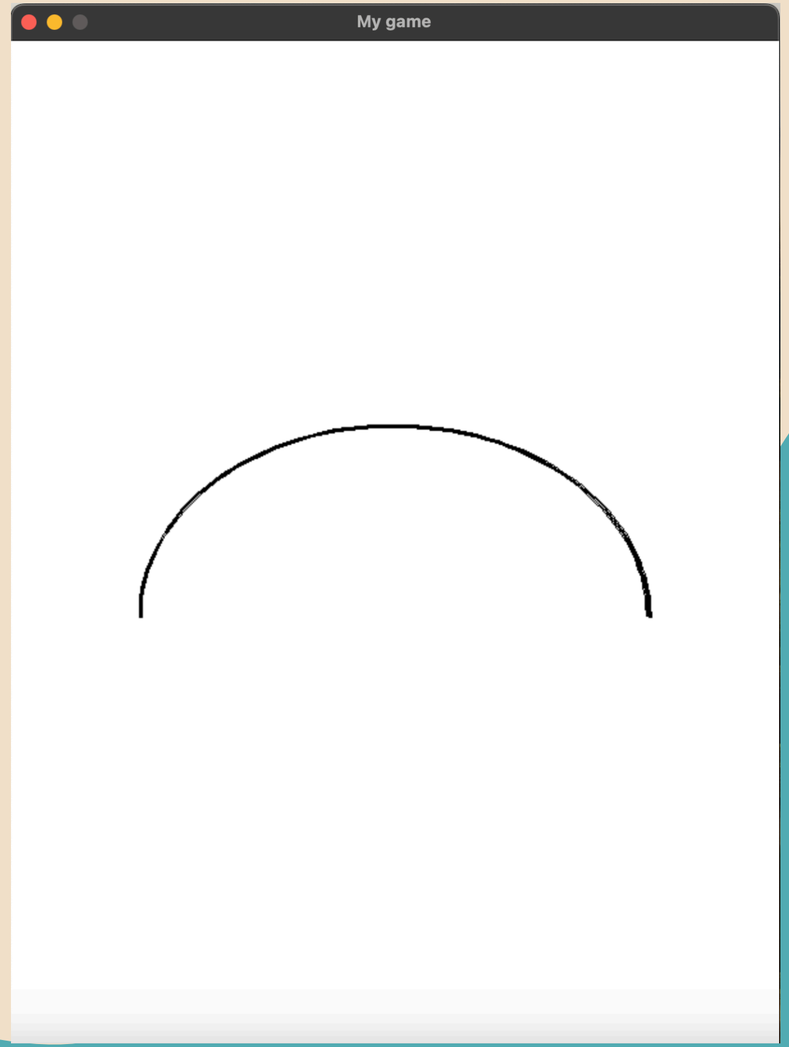
3.14 弧度
(180 度)

0 弧度
(0度)

繪製圓弧練習（一）

從資料夾中，打開「7.arc.py」檔案

先試著畫出右邊的圓弧吧！

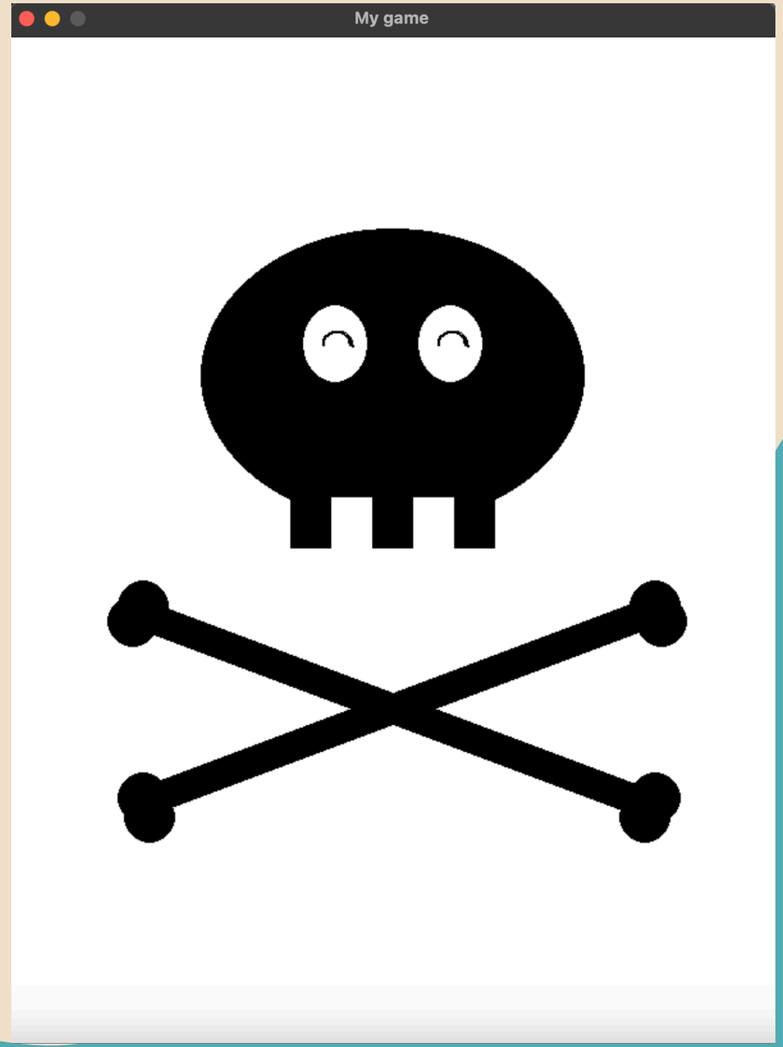


綜合練習

想做這頁練習的同學，請確認自己
前面都會了，再做喔！

從資料夾中，打
開“8.review.py”檔案

先試著畫出右邊的圖案吧！

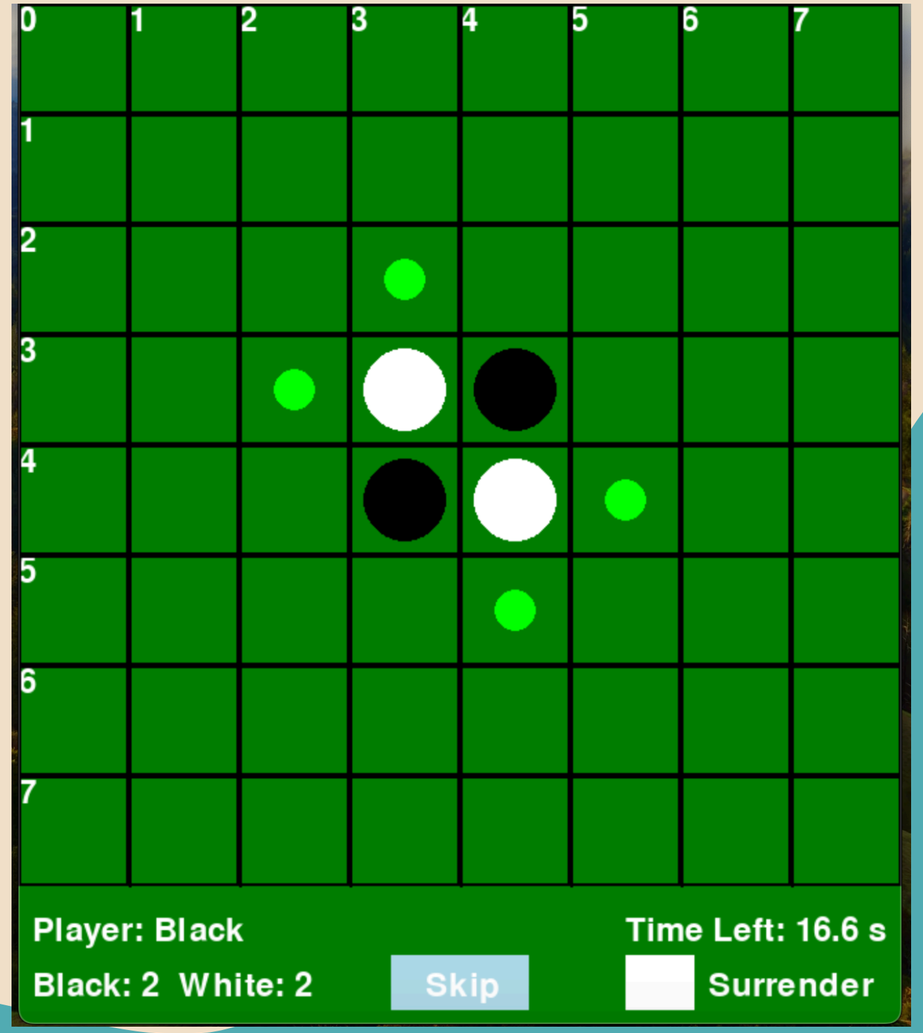


黑白棋

黑白棋的規則介紹

開局：

- 四枚棋子以交叉的方式放在棋盤中央，形成一個小的 2x2 的正方形，黑白各兩枚，黑色棋子置於右上和左下，白色棋子置於左上和右下。
- 黑棋先下



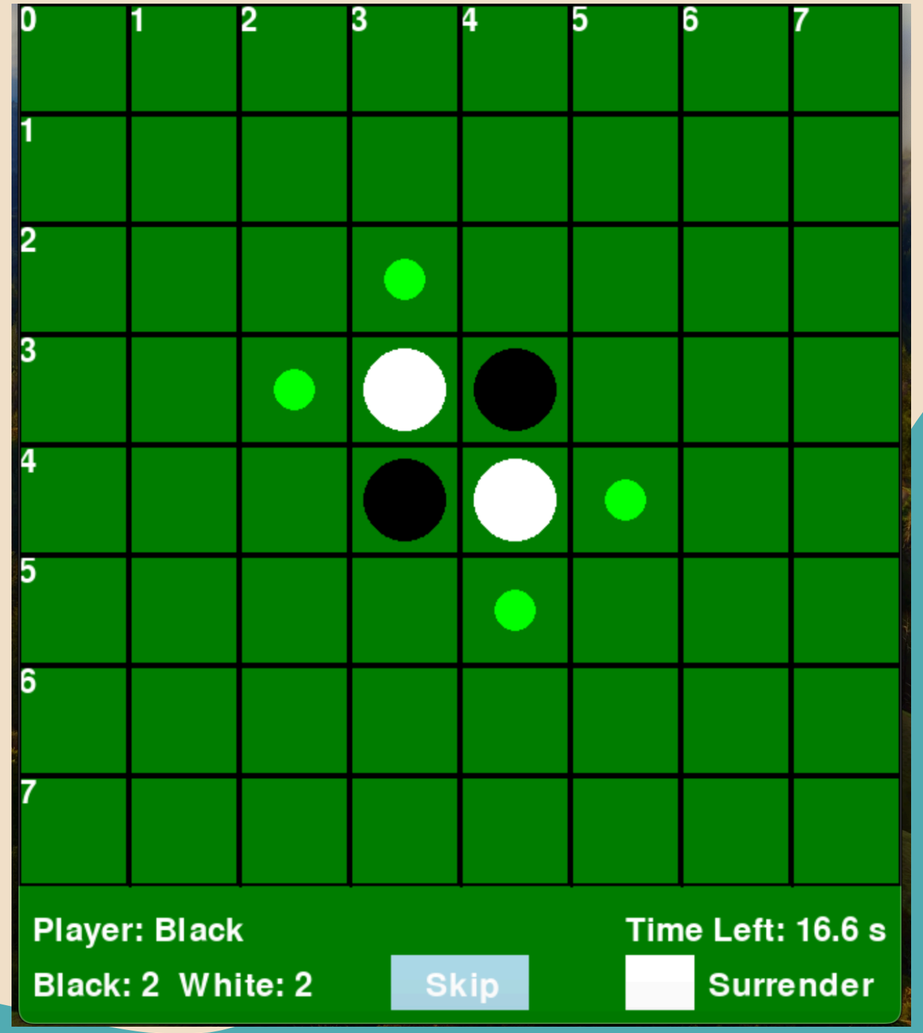
黑白棋的規則介紹

合法移動:

- 棋子只能放在水平、垂直或對角線方向上，且必須夾住對方的棋子，夾住的對方棋子將被翻轉成自己的顏色。
- 如果無地方可下，另一方可繼續下

勝利條件:

- 無子可下時，場上剩餘棋子數量較多者勝
- 若棋子數量相同，則平手。



黑白棋的規則介紹

- 用說的，不如實際玩玩看
- 打開“reversi.py”檔案，選擇 player v.s. player 來和你的組員對戰看看吧!

Pygame 黑白棋程式碼講解

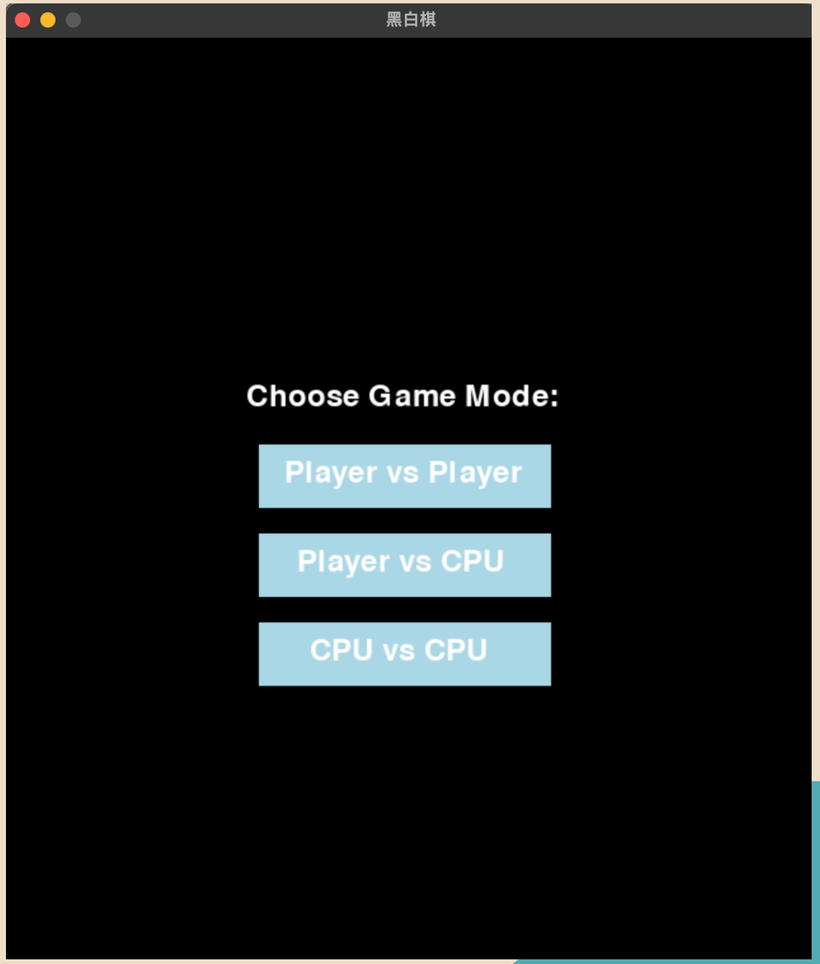
- 同學們玩完後，思考一下，如果是你在撰寫這個黑白棋遊戲的程式碼時，你覺得程式碼要實現哪些功能?

Pygame 黑白棋程式碼結構

pygame 初始化

↓

顯示遊戲選單

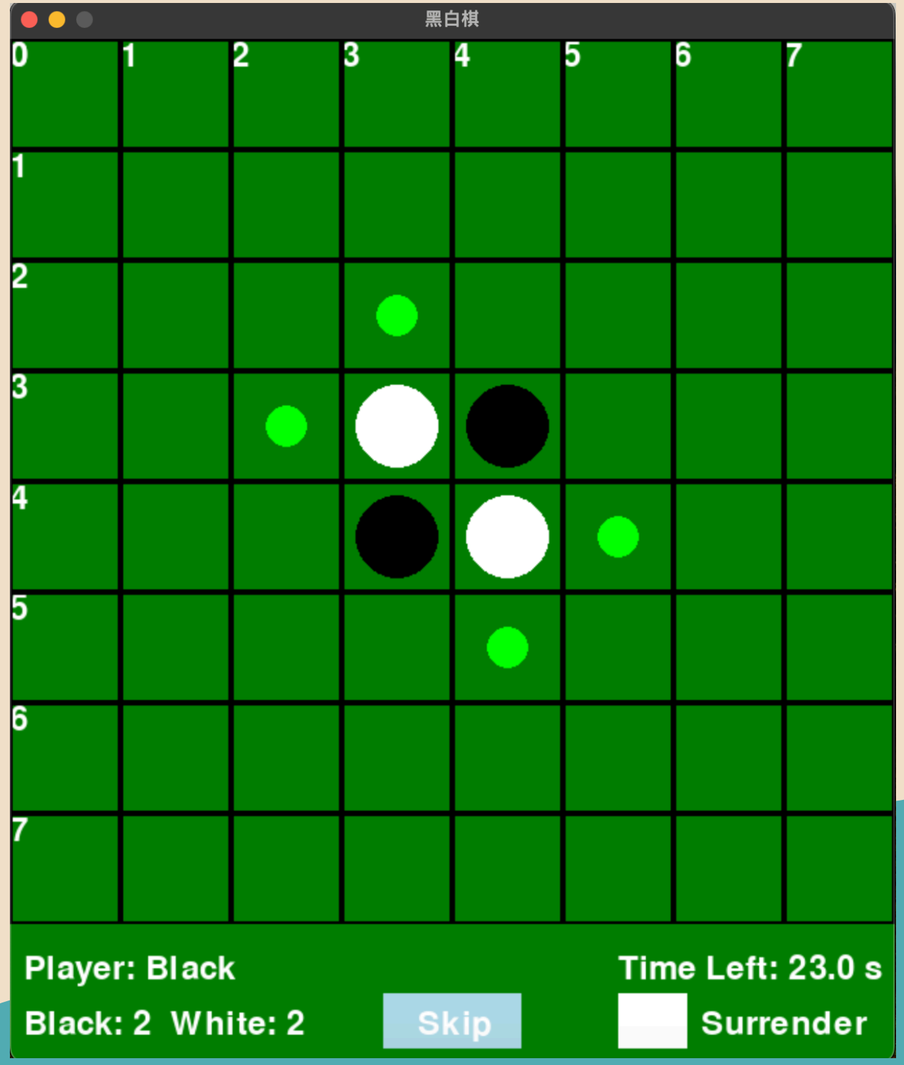


Pygame 黑白棋程式碼結構

創建初始棋盤

↓

繪製棋盤、棋子
等等所有元素



Pygame 黑白棋程式碼結構

判斷玩家點擊的位置

- 如果在棋盤內的合法位置 → 移動棋子
- 如果在跳過按鈕 → 換人
- 如果在投降按鈕內 →

Pygame 黑白棋程式碼講解

判斷勝負

if:

點擊投降

黑棋數為零

白棋數為零

無地方可下

True → 顯示勝負

False → 繼續遊戲迴圈

Pygame 黑白棋程式碼講解

- 詳細的程式碼可以查看右邊連結的網站，有解釋

點我

權重函式實作

關於電腦函式講解

前面講解完了黑白棋的架構，其實還少講了一段，那就是“computer”這部分

在 HackMD 或是 黑白棋.py 檔案中，能看到「電腦 ai_design()」和「權重計算 calculate_weight()」，這兩個就是有關電腦下棋的函式

關於電腦函式講解

電腦 ai_design()

1. 先找到哪些位置可以下
2. 使用「權重計算 calculate_weight()」計算所有可能位置的權重
3. 存下權重最大值的位置

關於電腦函式講解

權重計算 calculate_weight()

當位置是有高機率獲勝時，要讓權重值增加，反之則減少

哪些是有較高機率獲勝的位置?　ex: 佔角、
哪些是有較低機率獲勝的位置?　ex: 星位、C位...

權重函式

- 所以你們要設計的是「權重計算 calculate_weight()」這個函式的內容
- 各小組要著重在黑白棋所遭遇的情況，而讓你們的程式碼能做出應對
- 網路上有不少黑白棋的技巧，可以上網查詢
- 自己也能思考哪些位置下了之後，會更有機會獲勝

棋子設計

各小組要設計你們自己黑白棋的造型，最後一天會投票出最好看的設計

注意：繪製棋子要額外寫函式（黑棋和白棋）

```
def draw_white_pawn(screen, col, row):
```

```
def draw_black_pawn(screen, col, row):
```

棋子設計引導

height

width

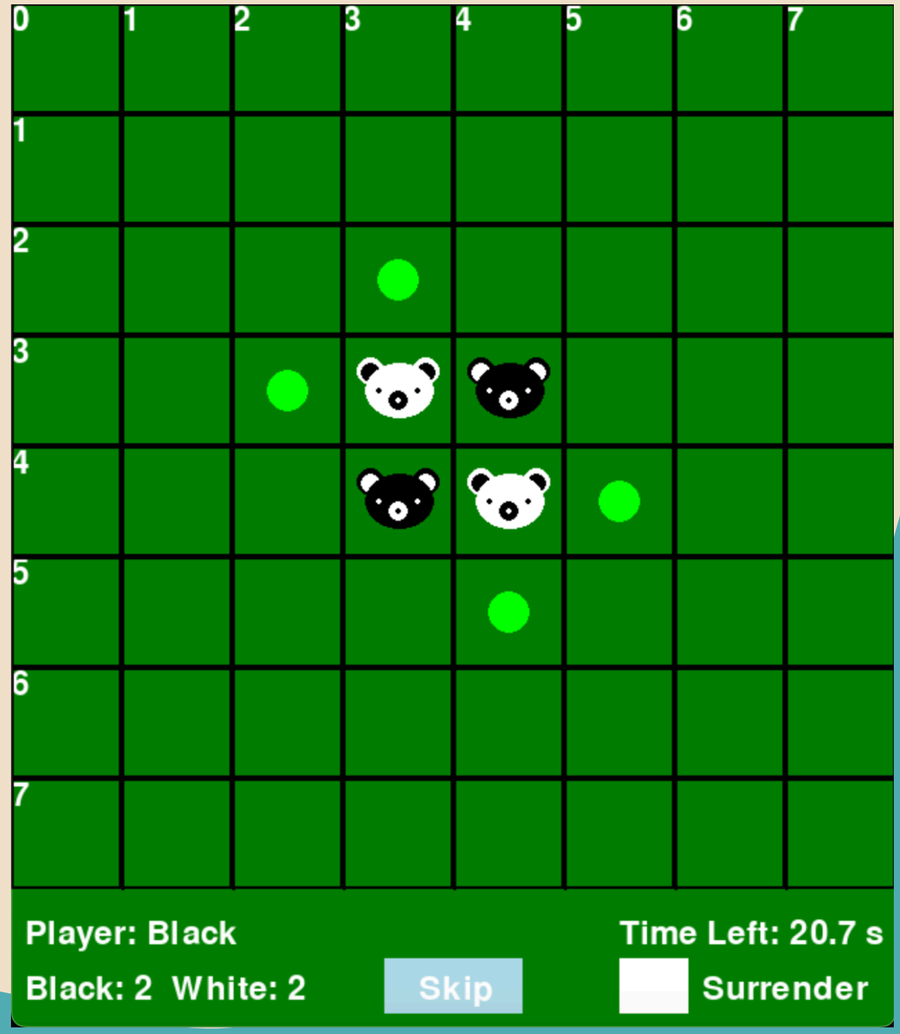
- 棋盤每一格都是依照 width x height 組出來的，所以在繪製時，可以依照這個性質更好的定位你們的圖形

- 在 draw_board() 中，找到繪畫棋子的那段程式碼，改成自己做出來的函式

注意：後面畫上的圖案會把前面已有的圖案蓋上

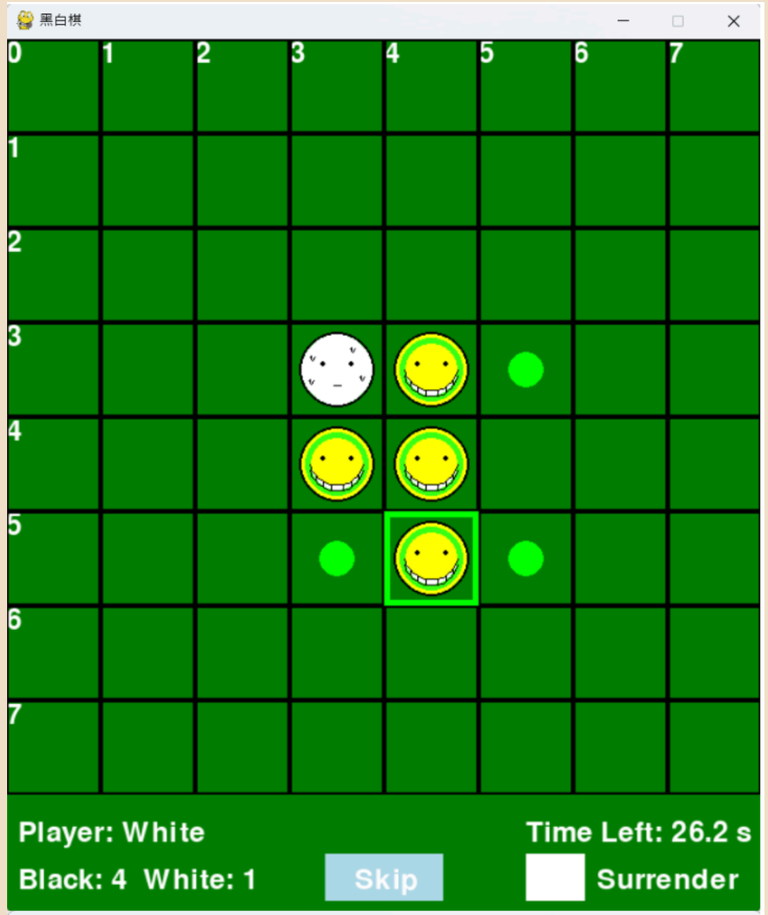
棋子設計引導

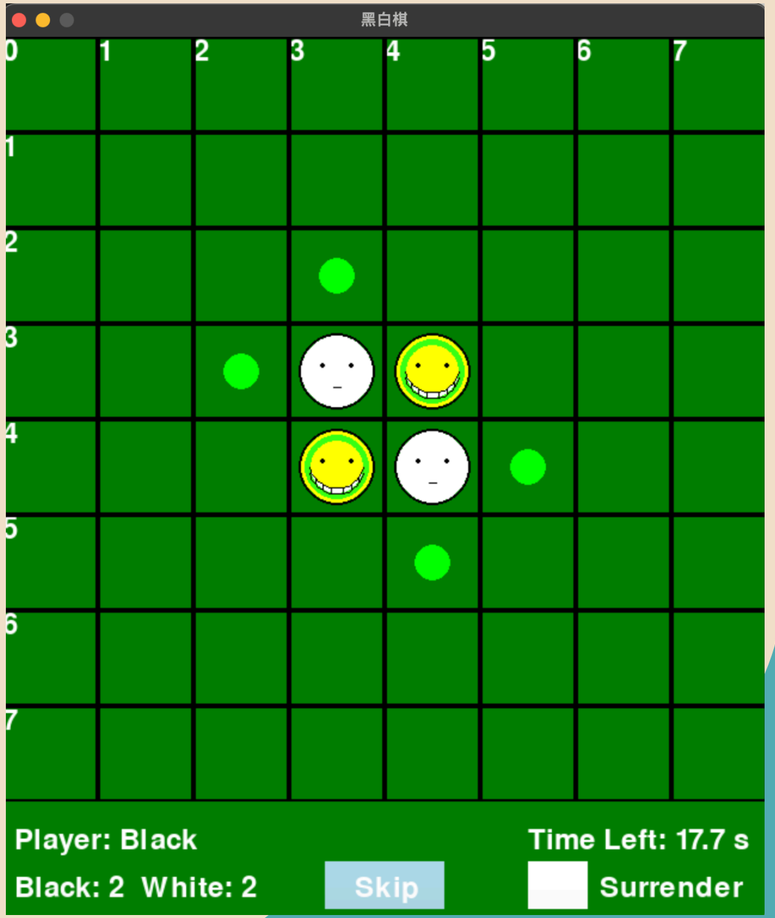
- 棋子無變化



棋子設計引導

- 棋子有變化





棋子設計

在 draw_board() 中，找到繪畫棋子的那段程式碼，改成自己做出來的函式 (建議將原本的註解掉而不是直接刪除)

```
for row in range(ROWS):
    for col in range(COLS):
        pygame.draw.rect(screen, LINE_COLOR, (col * WIDTH, row * HEIGHT, WIDTH, HEIGHT), 2)
        if board[row][col] == 'X':
            draw_black_pawn(screen, col, row)
            #pygame.draw.circle(screen, BLACK, (col * WIDTH + WIDTH // 2, row * HEIGHT + HEIGHT // 2), RADIUS)
        elif board[row][col] == 'O':
            draw_white_pawn(screen, col, row)
            #pygame.draw.circle(screen, WHITE, (col * WIDTH + WIDTH // 2, row * HEIGHT + HEIGHT // 2), RADIUS)
```

總結要做的事

1. 權重函式的實作 - 最後一天營期小組比賽
2. 黑白棋的圖案設計 - 最後一天營期投票出最好看的作品

團隊合作，能更有效地完成喔！

Thank You