
tags: TA, Course, NCTU

PyGame

PyGame 介紹

PyGame 是一個跨平台的 Python 模組，提供了圖像、音頻的支援，讓使用者可以利用 PyGame 來快速開發遊戲，且便於移植到其他平台。

安裝 PyGame

在使用 PyGame 之前，我們必須要先安裝它，可以利用 pip 來進行 Python 模組的安裝：

```
$ pip install pygame
```

遊戲是怎麼產生的？

- 其實電腦是利用快速切換圖片來讓我們誤以為它是連續的影像
 - 切換圖片的速度就被稱為幀率，常用單位是 fps (frame per second)，代表一秒會刷新幾次。
- 我們想要動態的遊戲畫面，我們就需要一直去刷新遊戲畫面，所以情理之上，我們可能會寫一個 function 來做畫面的渲染，即輸出那一瞬間的遊戲畫面，之後我們再利用可能定時器的方式，來不斷的呼叫這個畫面渲染的 function
 - 至於畫面要怎麼渲染，我們就會用 PyGame 提供的方式來把圖片堆砌起來
- 遊戲視窗會去捕捉使用者的操作，最常見的就是鍵盤操作和滑鼠操作
 - 使用者的操作會被紀錄起來，並且被稱之為一個事件 (event)，例如一個 MouseDown Event 然後可能會有參數 (x, y) 意味著使用者在視窗的 (x, y) 的位置按下了滑鼠。

一個簡單的範例程式

```
import pygame # 引入 pygame 模組

pygame.init() # 初始化 pygame 模組

window = pygame.display.set_mode((800,450))
```

```
pygame.display.set_caption("這是一隻 PyGame 的範例程式")
window.fill((255, 255, 255))

pygame.display.flip() # 顯示 window

while True:
    # 捕捉 pygame 接受到的事件，並且針對關閉視窗的事件做處理
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
```

這是一隻很簡單的程式，他會：

1. 創建一個新的視窗
 - 設置大小為 800x450
 - 設置標題為「這是一隻 PyGame 的範例程式」
 - 用白色來填充視窗
2. 顯示這個視窗
3. 不斷的捕捉事件
 - 如果是關閉視窗的事件，就結束程式

此篇教學文會涵蓋到的部分

我們會介紹 PyGame 提供的以下功能：

- `pygame.font`：可以幫助我們輸出一段文字在視窗上。
- `pygame.event`：可以幫助我們捕捉、處理使用者觸發的事件。
- `pygame.image`：可以幫助我們輸出一張圖片在視窗上。
- `pygame.mixer`：可以幫助我們播放出聲音。

PyGame 教學

p1.font.sample1 用預設字體輸出文字

```
import pygame

pygame.init()
```

```

window = pygame.display.set_mode((800,450))
pygame.display.set_caption("PyGame - Font Sample Code")
window.fill((255, 255, 255))

##### 重點開始 #####

default_font = pygame.font.SysFont(None, 40)
text1 = default_font.render('Hello World!', True, (0, 0, 0))
window.blit(text1, (10, 10))

##### ----- #####

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)

```

這份 code 和前一份長的差不多，只是中間多出一段文字渲染的程式碼：

- 用 `pygame.font.SysFont` 來抓系統字體：
 - 我們用 `SysFont(name, size, bold=False, italic=False)` 來引入一個字體，之後才會用這個字體去渲染文字
 - 其中 `name` 表示字體的名稱，`None` 代表預設字體；`size` 表示大小；`bold` 表示是否粗體；`italic` 表示是否斜體。
- 用 `font.render` 來渲染文字：
 - `font.render(text, antialias, color, background=None)` 會使用該字體來去渲染 `text` 這段文字
 - 其中 `antialias` 代表是否開啟抗鋸齒；顏色用 RGB 表示；`background` 代表背景色，預設是透明。
- 之後我們還要把這段文字在視窗渲染出來：
 - `window.blit(text1, (10, 10))` 代表在 window 上面的 (10, 10) 這個位置渲染我們剛剛宣告出的 `text1` 這段文字。

效果大概長這樣：



PyGame

Hello World!

p1.font.sample2 用其他字體輸出文字

實際上除了預設字體外，我們完全可以上網找自己喜歡的字體，並且在自己的遊戲當中使用該字體，而 PyGame 也支援我們使用非系統字體來進行文字渲染：

```
import pygame

pygame.init()

window = pygame.display.set_mode((800,450))
pygame.display.set_caption("PyGame - Font Sample Code")
window.fill((255, 255, 255))

default_font = pygame.font.SysFont(None, 40, False, True)
text1 = default_font.render('Hello World!', True, (0, 0, 0))
window.blit(text1, (10, 10))

##### 重點開始 #####

another_font = pygame.font.Font("GenRyuMinTW-Regular.ttf", 40)
text2 = another_font.render('你好世界!', True, (0, 0, 0))
window.blit(text2, (10, 80))

##### ----- #####

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
```

這份 code 也和前一份長的差不多，只是中間又多出一段文字渲染的程式碼：

- 我們用 `pygame.font.Font` 來引入不同的字體，這邊我引入了 `GenRyuMinTW-Regular.ttf` 這個文件，他是一個中文字體的字體檔。
- 我們用類似的方式把他渲染出來。

效果圖如下：

Hello World!

你好世界！

p2.event.sample1 捕捉滑鼠按下事件

```
import pygame

pygame.init()

window = pygame.display.set_mode((800,450))
pygame.display.set_caption("PyGame - Event Sample Code")
window.fill((255, 255, 255))

default_font = pygame.font.SysFont(None, 40, False, True)
text1 = default_font.render('<3', True, (0, 0, 0))

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.MOUSEBUTTONDOWN:
            x, y = pygame.mouse.get_pos()
            window.blit(text1, (x, y))
            pygame.display.flip()
```

這份 code 也和最前一份長的差不多，只是下面多出一段捕捉事件的程式碼：

- 和捕捉 `pygame.QUIT` (按下關閉視窗的事件) 類似，我們用 `pygame.MOUSEBUTTONDOWN` 來表示滑鼠被按下的事件；
- 我們用 `pygame.mouse.get_pos()` 來取得現在滑鼠所在的位置，並且在該位置輸出一個愛心 <3。

效果圖如下：

<3

<3

<3

p2.event.sample2 捕捉鍵盤按下事件

```
import pygame

pygame.init()

window = pygame.display.set_mode((800,450))
pygame.display.set_caption("PyGame - Event Sample Code")
window.fill((255, 255, 255))

default_font = pygame.font.SysFont(None, 40)

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.KEYDOWN:
            window.fill((255, 255, 255))
            window.blit(default_font.render('You press down ' + chr(e.key), True, (0,
0, 0)), (10, 10))
            pygame.display.flip()
```

這份 code 和前一份長的很像，只是改成了偵測鍵盤事件，值得一提的事情是：

- `e.key` 回傳的是 ASCII 碼，所以如果是一般的字母、數字，可以用 `chr()` 這個函數將他們轉換成我們認識的樣子；但如果是一些特殊的字元，比如你按下 CapsLock，那程式就會報錯，**不是因為 PyGame 不支援特殊字元，而是因為我這邊想要用 `chr()` 把他輸出出來。**

效果圖如下：



You press down x

還有許多不同的 events，甚至哪怕是鍵盤滑鼠，也有分按下 (Down) 和彈起 (Up) 等不同的 event，族繁不及備載，這邊就不贅述了，有興趣的同學可以去讀 [Docs](#)。

p3.image.sample1 渲染圖片

```
import pygame

pygame.init()

window = pygame.display.set_mode((1200,500))
pygame.display.set_caption("PyGame - Image Sample Code")
window.fill((255, 255, 255))

img1 = pygame.image.load('pic1.jpg')
img2 = pygame.image.load('pic2.jpg')

window.blit(img1,(0, 0))
window.blit(img2,(800, 0))

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
```

這份 code 和前面的字體渲染長的很像，只是改成了圖片渲染。

效果圖如下：



不過兩張卡片的大小不一似乎挺令人困擾的，如果同一份素材可以在不同的地方隨我心意縮放，似乎會更方便？

p3.image.sample2 圖片縮放

```
import pygame

pygame.init()

window = pygame.display.set_mode((1200,450))
pygame.display.set_caption("PyGame - Image Sample Code")
window.fill((255, 255, 255))

img1 = pygame.image.load('pic1.jpg')
img2 = pygame.image.load('pic2.jpg')
img1 = pygame.transform.scale(img1, (600, 450))
img2 = pygame.transform.scale(img2, (600, 450))
img1 = pygame.transform.rotate(img1, 1)

window.blit(img1,(0, 0))
window.blit(img2,(600, 0))

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
```

我們利用 `pygame.transform`

這個模組來幫助我們針對某個待渲染的東西做一些變換，例如這邊利用 `pygame.transform.scale` 來縮放一張圖片，把他改成 300x440 的大小；有利用 `pygame.transform.rotate` 來旋轉一張圖片，旋轉 1 度。

效果圖如下：



p4.mixer.sample1 點擊播放 bgm

```
import pygame

pygame.init()
pygame.mixer.init()

window = pygame.display.set_mode((800,450))
pygame.display.set_caption("PyGame - Mixer Sample Code")
window.fill((255, 255, 255))

pygame.mixer.set_num_channels(1)
audio = pygame.mixer.Sound("bgm.mp3")

pygame.display.flip()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.MOUSEBUTTONDOWN:
            if pygame.mixer.Channel(0).get_busy():
                pygame.mixer.Channel(0).stop()
            pygame.mixer.Channel(0).play(audio, loops = -1)
```

我們會利用 `pygame.mixer` 來幫助我們管理音訊：

- 首先我們利用 `pygame.mixer.init()` 來幫助我們初始化這個模組；
- 接著利用 `pygame.mixer.set_num_channels()` 來指定說我們需要多少個 channel
 - o 有多少個 channel 意味著你可以同時在多少個音軌上做操作
- 接著我們用 `pygame.mixer.Sound()` 來載入一個音訊檔
- 最後當我們要播放的時候：
 - o 我們先用 `pygame.mixer.Channel(0).get_busy()` 來取得 0 號 channel 是否在播放東西，如果在播放東西，我們就把他停止掉；
 - o 否則我們就在 0 號 channel 上面播放那則音訊，其中 `loops = -1` 意味著不要循環播放。

至此，我們就學會了開發一個遊戲最基礎的一些元件，接下來就可以利用這些元件來拼湊出一款遊戲！

範例 - 打字練習

小時候電腦課會讓孩子進行打字練習，不如自己寫一個打字練習的程式吧！

```
import pygame
from random import randint
from time import time

pygame.init()

window = pygame.display.set_mode((600,440))
pygame.display.set_caption("PyGame - Typing Practice")
window.fill((255, 255, 255))

img0 = pygame.image.load('keyboard.png')
img0 = pygame.transform.scale(img0, (600, 200))

small_font = pygame.font.Font("GenRyuMinTW-Regular.ttf", 30)
big_font = pygame.font.Font("GenRyuMinTW-Regular.ttf", 120)
txt0 = small_font.render('參考鍵盤：', True, (0, 0, 0))
txt1 = small_font.render("請按：", True, (0,0,0))

score = 0

def Loss():
    msg = big_font.render("你輸了", True, (255,0,0))
    window.blit(msg, (200, 10))
    pygame.display.flip()

    while True:
        for e in pygame.event.get():
            if e.type == pygame.QUIT:
                exit(0)

notStart = 1

#### --- 準備進入遊戲 --- ####

msg = small_font.render("任意鍵開始遊戲！", True, (0,0,0))
window.blit(msg, (150, 150))
pygame.display.flip()

while notStart:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.KEYDOWN:
            notStart = 0
```

```
#### --- 進入遊戲 --- ####
```

```
while True:
```

```
    ans = randint(0, 25) + ord('a')
```

```
    txt2 = big_font.render(chr(ans).upper(), True, (0, 0, 0))
```

```
    txt5 = big_font.render(chr(ans).upper(), True, (0, 0, 0))
```

```
    TL = time()
```

```
    while True: # 等待使用者回應
```

```
        remain = max(0, round(TL + 2 - time(), 2))
```

```
        txt3 = small_font.render("你還剩下：" + str(remain) + " 秒", True, (0, 0, 0))
```

```
        txt4 = small_font.render("你得到了：" + str(score) + " 分", True, (0, 0, 0))
```

```
        flg = 0
```

```
        window.fill((255, 255, 255))
```

```
        window.blit(img0, (0, 240))
```

```
        window.blit(txt0, (0, 200))
```

```
        window.blit(txt1, (20, 120))
```

```
        window.blit(txt2, (100, 60))
```

```
        window.blit(txt3, (200, 160))
```

```
        window.blit(txt4, (360, 200))
```

```
        pygame.display.flip()
```

```
        if remain == 0:
```

```
            Loss()
```

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

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

```
                exit(0)
```

```
            if e.type == pygame.KEYDOWN:
```

```
                if ord('a') <= e.key <= ord('z'):
```

```
                    if e.key == ans:
```

```
                        score += 1
```

```
                        flg = 1
```

```
                    else:
```

```
                        Loss()
```

```
        if flg:
```

```
            break
```

效果圖如下：

你車

請按：

J

你還剩下：1

參考鍵盤：

~ ,	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8
Tab ⇄	Q	W	F	P	G	J	L	U
← Backspace	A	R	S	T	D	H	N	E
Shift ↑	Z	X	C	V	B	K	M	
Ctrl	Win Key	Alt						

Class 介紹

何為 Class ?

Class 將一大堆資料包在一起

```
class Student:
    def __init__(self, name, stuid, score1, score2):
        self.name = name
        self.id = stuid
        self.score1 = score1
        self.score2 = score2

    def get_avg(self):
        return (self.score1 + self.score2) / 2

student = Student("甲同學", 101, 95, 97)
print(student.get_avg())
```

以這個例子，我們可以創建一個 `Student` 的類別：

- `class` 中會有很多 function，這些 function 會有一個 `self` 的參數，因此 function 內可以使用這個類別其他屬性、functions (`self` 意味著自己)
- `__init__` 就是你去新增一個物件時會去呼叫的 function。

我們為什麼會使用 function ?

以一下這段程式碼為例，他會計算 S^x_y ：

```
x = int(input())
y = int(input())

ans1 = 1
for i in range(x + 1):
    ans1 *= i

ans2 = 1
for i in range(y + 1):
    ans2 *= i

print(ans1 // ans2)
```

但我們看來覺得很冗，因為他需要計算多次階乘，因此在程式碼的多處都寫著類似的程式碼，因此我們會使用 function 把他們包起來：

```
def factorial(x):
    ans = 1
```

```
    for i in range(x + 1):
        ans *= i
    return ans

x = int(input())
y = int(input())

print(factorial(x) // factorial(y))
```

如此，程式碼便變得清晰、更好維護，且出錯機率更低。

我們為什麼會使用 class？

一個最簡單的場景就是我們要把很多資料聚集起來，之後對他們進行一些處理：

```
names      = {}
deposits   = []
count      = 0

def newAccount(name):
    if name in names:
        return False
    else:
        names[name] = count
        count += 1
        deposit.append(0)
        return True

def save(name, amount):
    if name not in names or amount <= 0:
        return False
    else:
        index = names[name]
        deposits[index] += amount
        return True

def withdraw(name, amount):
    if name not in names or amount <= 0:
        return False
    else:
        index = names[name]
        if deposits[index] - amount >= 0:
            deposits[index] -= amount
            return True
        else:
            return False
```

這是一個銀行帳號系統，你可以開通新的帳號，然後來存款、提款，但這樣子的寫法很複雜，而且資料是分散的，你沒辦法從一個人的名字直接看到他的存款。

如果我們運用

class，就可以用這樣子的作法做到一樣的功能性，而且每一位用戶他的所有資料是綁定在一起的，我們可以很容易的做到擴充：

```
class account:
    def __init__(self, name):
        self.name = name
        self.deposit = 0

    def save(self, amount):
        if amount <= 0:
            return False
        self.deposit += amount
        return True

    def withdraw(self, amount):
        if amount <= 0 or self.deposit < amount:
            return False
        self.deposit -= amount
        return True
```

範例 - OOX

```
import pygame
from random import randint
from time import time

pygame.init()

window = pygame.display.set_mode((390,390))
pygame.display.set_caption("PyGame - OOX")
window.fill((255, 255, 255))

imgo = pygame.transform.scale(pygame.image.load('o.png'), (128, 128))
imgx = pygame.transform.scale(pygame.image.load('x.png'), (128, 128))

class Block:
    def __init__(self, r, c):
        # 0 -> empty
        # 1 -> O
        # 2 -> X
        self.type = 0
        self.r = r
        self.c = c

    def turn(self, round_):
        if self.type:
            return False
        if round_ == 0:
            self.type = 1
        else:
            self.type = 2
        return True
```

```

def blit(self, window):
    if self.type == 1:
        window.blit(imgo, (self.r * 130, self.c * 130))
    elif self.type == 2:
        window.blit(imgx, (self.r * 130, self.c * 130))

class Game:
    def __init__(self):
        self.blocks = [[], [], []]
        self.round_ = 0
        for i in range(3):
            for j in range(3):
                self.blocks[i].append(Block(i, j))

    def click(self, x, y):
        if self.round_ == -1:
            return

        x //= 129
        y //= 129
        if self.blocks[x][y].turn(self.round_):
            self.round_ = not self.round_

    def blit(self, window):
        pygame.draw.rect(window, (0, 0, 0), (128,0,2,390))
        pygame.draw.rect(window, (0, 0, 0), (258,0,2,390))
        pygame.draw.rect(window, (0, 0, 0), (0,128,390,2))
        pygame.draw.rect(window, (0, 0, 0), (0,258,390,2))

        for r, row in enumerate(self.blocks):
            for c, blk in enumerate(row):
                blk.blit(window)

        if self.isSet():
            self.round_ = -1

        pygame.display.flip()

    def isSet(self):
        for i in range(3):
            if self.blocks[i][0].type != 0 and self.blocks[i][0].type ==
self.blocks[i][1].type and self.blocks[i][1].type == self.blocks[i][2].type:
                pygame.draw.rect(window, (255, 0, 0), (128 * i + 64,10,10,370))
                return True
            if self.blocks[0][i].type != 0 and self.blocks[0][i].type ==
self.blocks[1][i].type and self.blocks[1][i].type == self.blocks[2][i].type:
                pygame.draw.rect(window, (255, 0, 0), (10,128 * i + 64,370,10))
                return True

        for i in range(3):
            for j in range(3):
                if self.blocks[i][j].type == 0:
                    return False

```

```

        return True

#### --- 進入遊戲 --- ####

game = Game()

while True:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.MOUSEBUTTONDOWN:
            x, y = pygame.mouse.get_pos()
            game.click(x, y)

    game.blit(window)

```

- `pygame.draw.rect(window, (0, 0, 0), (128,0,2,390))`
 - 意味著從 (128, 0) 這個點繪製一個大小是 (2, 390) 的黑色矩形
- 會有一個 `game` 的 class 代表那一場 OOX 的遊戲：
 - 會有 `isSet()` function 會告訴我們遊戲是否結束
 - 會有 `blit(window)` function 會在 `window` 上進行一次螢幕繪製
 - 會有 `click(x, y)` function 會幫我們處理視窗點擊
 - 會有一個屬性 `blocks` 包含 3 * 3 個 `block`
- 會有一個 `block` 的 class 代表那是一個玩家可以點的格子：
 - 有一個 `type` 代表他的狀態
 - 0 意味著沒點過，1 意味著 1 號玩家，2 意味著 2 號玩家
 - 會有 `blit(window)` 在 `window` 上繪製他

這是一個簡單的 class

範例，可以幫助我們讓程式變的簡單明瞭。如果我們想要讓遊戲可以一直玩，不需要重開程式，我們也可以透過回收舊的 game 來做到：

```

### 前面略

#### --- 進入遊戲 --- ####

while True:

```

```
window.fill((255, 255, 255))
game = Game()

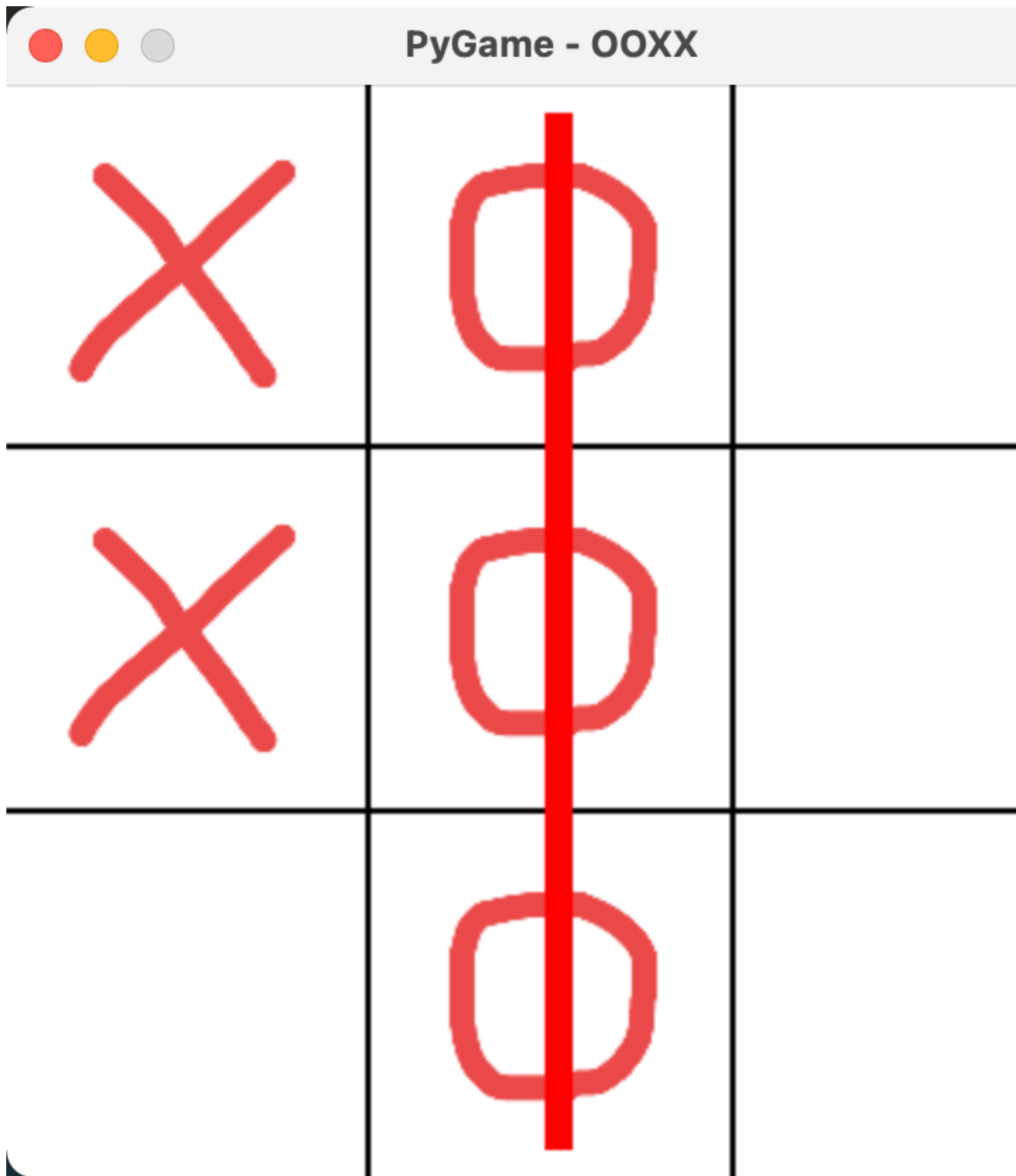
while not game.isSet():

    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.MOUSEBUTTONDOWN:
            x, y = pygame.mouse.get_pos()
            game.click(x, y)

    game.blit(window)

del game

# 點一下繼續遊戲
flg = 1
while flg:
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            exit(0)
        if e.type == pygame.MOUSEBUTTONDOWN:
            flg = 0
```



範例 - 接龍遊戲

```
import pygame
from random import randint, shuffle
from time import time

pygame.init()

window = pygame.display.set_mode((1600,900))
pygame.display.set_caption("PyGame - 接龍")

font = pygame.font.SysFont(None, 25)
suits = [pygame.transform.scale(pygame.image.load('flower.png'), (18, 20))]
ranks = [font.render(j, True, (0, 0, 0)) for j in [str(i) for i in range(11)] + ['J', 'Q', 'K']]

class Card:
    # 這是一張撲克牌
    def __init__(self, suit, rank):
        self.s = suit
        self.r = rank

    def blit(self, focus = 0):
        # blit 會回傳一個可以被繪製的物件
        # 可以用 window.blit(x.blit(), (x, y))
        obj = pygame.Surface((100, 140))
        obj.fill((255, 255, 255))

        if focus == 0:
            pygame.draw.rect(obj, (0, 0, 0), [0, 0, 100, 140], 3)
        else:
            pygame.draw.rect(obj, (255, 0, 0), [0, 0, 100, 140], 9)

        obj.blit(suits[self.s], (5, 5))
        obj.blit(ranks[self.r], (9, 30))
        obj.blit(ranks[self.r], (27, 7))
        return obj

class Pool:
    # 代表卡池，也就是維護剩下的卡片
    def __init__(self):
        self.cards = []

        # 初始化加入三副花色相同的牌
        for i in range(3):
            for j in range(1, 14):
                self.cards.append(Card(i, j))

        shuffle(self.cards) # 洗牌

    def remaining(self):
```

```

        return len(self.cards)

    def pop(self):

        # 把牌組中最上面的一張牌拿出去，可能是發牌，或是補牌

        if len(self.cards):
            return self.cards.pop()
        else:
            return None

class Pane:
    # 手牌
    def __init__(self, pool):
        # 初始化的時候抽 6 張牌加入手牌
        self.pool = pool
        self.hand = []
        for i in range(6):
            self.hand.append(self.pool.pop())
        self.focus = -1

    def setFocus(self, focus):
        # Focus 代表那張手牌正在被點選
        self.focus = focus
        self.focus = max(self.focus, -1)
        self.focus = min(self.focus, len(self.hand) - 1)

    def width(self):
        # 會回傳手牌物件的寬度，來判斷有沒有點擊到手牌
        return 100 + max(len(self.hand) - 1, 0) * 28

    def blit(self):
        obj = pygame.Surface((500, 140))
        obj.fill((13, 139, 58))

        pygame.draw.rect(obj, (0, 0, 0), [0, 0, 100, 140], 3)
        txt = font.render("X" + str(self.pool.remaining()), True, (0, 0, 0))

        obj.blit(txt, (65, 120))

        for i in range(len(self.hand)):
            obj.blit(self.hand[i].blit(self.focus == i), (150 + 28 * i, 0))

        return obj

class Slot():
    # 一個上面的區塊，可以讓你調整牌面
    def __init__(self, pool = None):
        # 初始化的時候抽 5 張牌
        self.cards = []
        if pool:

```

```

        for i in range(5):
            self.cards.append(pool.pop())
        self.focus = -1

    def height(self):
        # 會回傳 slot 物件的高度，來判斷有沒有點擊到 slot
        if len(self.cards) <= 1:
            return 140
        else:
            return 140 + max(0, len(self.cards) - 1) * 26

    def blit(self):
        obj = pygame.Surface((100, 600))
        obj.fill((13, 139, 58))
        pygame.draw.rect(obj, (0, 0, 0), [0, 0, 100, 140], 1)

        for i in range(len(self.cards)):
            obj.blit(self.cards[i].blit(), (0, 26 * i))

        if self.focus != -1:
            pygame.draw.rect(obj, (255, 0, 0), [0, self.focus * 26, 100,
self.height() - self.focus * 26], 6)

        return obj

    def setFocus(self, focus):
        # Focus 代表那一個 slot 的某部分牌正在被選取
        self.focus = focus
        self.focus = min(self.focus, len(self.cards) - 1)
        self.focus = max(self.focus, -1)
        for i in range(self.focus + 1, len(self.cards)):
            if self.cards[i].r != self.cards[i - 1].r + 1:
                self.focus = -1
        return

    def merge(self, segment):
        # 要有牌從手牌或是其他 slot 被移動過來
        if len(self.cards) == 0:
            self.cards = segment
            self.focus = -1
            return 1
        elif segment[0].r == self.cards[-1].r + 1:
            self.cards += segment
            self.focus = -1
            return 1
        return 0

class Game():
    # 一局遊戲
    def __init__(self):
        self.pool = Pool()
        self.slots = []

```

```

# 一局遊戲會有 8 個 slot
for i in range(4):
    self.slots.append(Slot(self.pool))
for i in range(4):
    self.slots.append(Slot())

shuffle(self.slots) # 隨機調換 slot 的位置，讓他看起來更隨機

self.pane = Pane(self.pool)
self.focus = None

def setFocus(self, area, card):
    pa = pc = na = None
    if self.focus:
        # 把舊的資料記錄起來並且清除掉
        pa, pc = self.focus
        if pa == "pane":
            self.pane.focus = -1

        for i in range(8):
            if pa == "slot" + str(i):
                self.slots[i].setFocus(-1)
            pa = i

    self.focus = [area, card]

    if area == None:
        self.focus = None
        return

    # 如果是摸手牌
    if area == "pane":
        self.pane.setFocus(card)
        self.focus[1] = self.pane.focus
        na = area
        nc = self.focus[1]

    # 如果是摸某個 slot
    for i in range(8):
        if area == "slot" + str(i):
            self.slots[i].setFocus(card)
            self.focus[1] = self.slots[i].focus
            na = i
            nc = self.focus[1]

    if pa == None:
        return
    elif type(pa) == int and type(na) == int and pa != na and pc != -1:
        # 從某個 slot 轉移到另一個 slot
        segment = self.slots[pa].cards[pc:]
        if self.slots[na].merge(segment):
            self.slots[pa].cards = self.slots[pa].cards[:pc]

```

```

        elif pa == "pane" and type(na) == int and pc != -1:
            # 從手牌轉移到某個 slot
            segment = [self.pane.hand[pc]]
            if self.slots[na].merge(segment):
                self.pane.hand[pc] = self.pool.pop()
                if self.pane.hand[pc] == None:
                    del self.pane.hand[pc]

    def blit(self):
        obj = pygame.Surface((1600,900))
        obj.fill((13, 139, 58))
        obj.blit(self.pane.blit(), (100, 700))

        for i in range(len(self.slots)):
            obj.blit(self.slots[i].blit(), (300 + i * 150, 80))

        return obj

#### --- 進入遊戲 --- ####

game = Game()

while True:

    window.blit(game.blit(), (0, 0))
    pygame.display.flip()

    for e in pygame.event.get():

        if e.type == pygame.QUIT:
            exit(0)

        if e.type == pygame.MOUSEBUTTONDOWN:
            x, y = pygame.mouse.get_pos()
            flg = 1
            # 你摸到手牌，然後要點擊手牌
            if 250 <= x <= 250 + game.pane.width() and 700 <= y <= 840:
                # pane
                x = (x - 250) // 28
                game.setFocus("pane", x)
                flg = 0

            # 摸某個 slot
            for i in range(8):
                if 300 + i * 150 <= x <= 300 + i * 150 + 100 and 80 <= y <= 80 +
game.slots[i].height():
                    game.setFocus("slot" + str(i), (y - 80) // 26)
                    flg = 0

            # 如果是無效點擊，就把舊的紀錄清除掉
            if flg:
                game.setFocus(None, None)

```



♣ 9
♣ 2
♣ 5
♣ 4
♣ 9 9

♣ K
♣ 1
♣ 8
♣ Q Q