




PYGAME

2020資訊之芽Python班

Rilak@2020/05/24



特別感謝

還抄的不太好

- 曾千育的投影片，這份投影片基本是照抄的
- Pygame's Tutorial

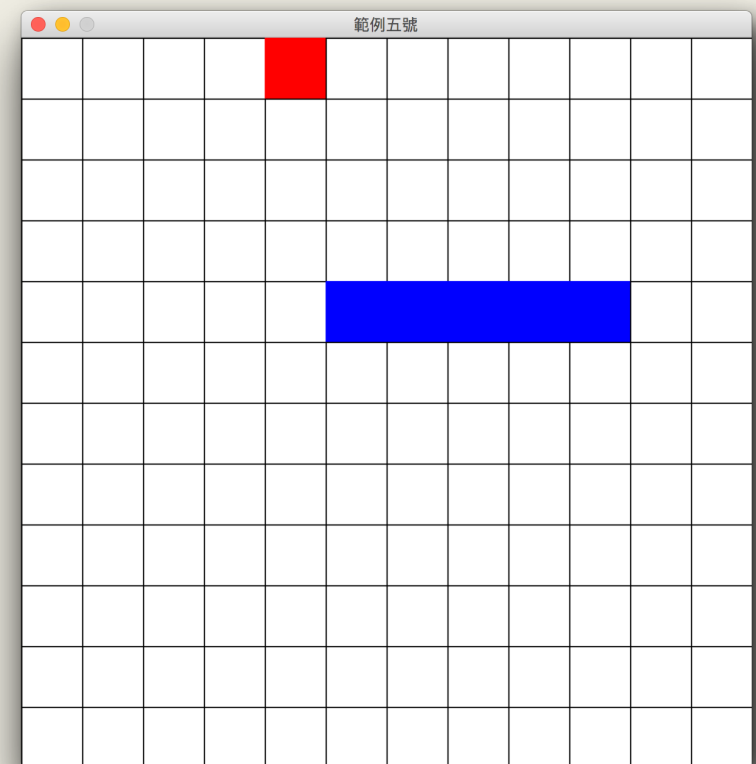
Rilak

- 周忠毅
- 資芽老人友情支援
- iCHEF 雲端工程師



Our Goal

- 建立一個遊戲視窗
- 在上面畫畫
- 處理鍵盤事件
- 實作:貪食蛇



Outline

- 介紹 Intro
- 視窗 Display
- 事件 Event
- 鍵盤 Key
- 繪圖 Draw
- 影像 Image
- 字型 Font
- 顯示型態 Surface

INTRO

基本介紹

What?

- SDL : Simple DirectMedia Layer
 - 開放原始碼的跨平台多媒體函式庫
 - C語言
 - 遊戲、模擬器、媒體播放器等等
- Pygame
 - 建立在SDL基礎之上
 - 專為遊戲設計
 - <https://www.pygame.org/docs/>

How?

- `cmd: pip install pygame`
- `import pygame`
- `import pygame as pg`
- `from pygame import ...`

Pygame Modules

- 視窗 display
- 字型 font
- 影像 image
- 顯示型態 surface
- 繪圖 draw
- 游標 cursors
- 滑鼠 mouse
- 鍵盤 key
- 事件 event
- 時間 time

DISPLAY

顯示

example1.py

先看程式碼猜個意思

初始化

```
1 import pygame
2
3 pygame.init()
```

建立視窗

```
7 window_size = (640,480)
8 screen = pygame.display.set_mode(window_size)
```

新增一個背景
把背景填紅
把背景畫在視窗上

```
18 background = pygame.Surface(screen.get_size())
19 background.fill(RED)
20 background = background.convert()
21 screen.blit(background, (0, 0))
```

設定標題

```
24 title = "範例一號"
25 pygame.display.set_caption(title)
```

更新畫面

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

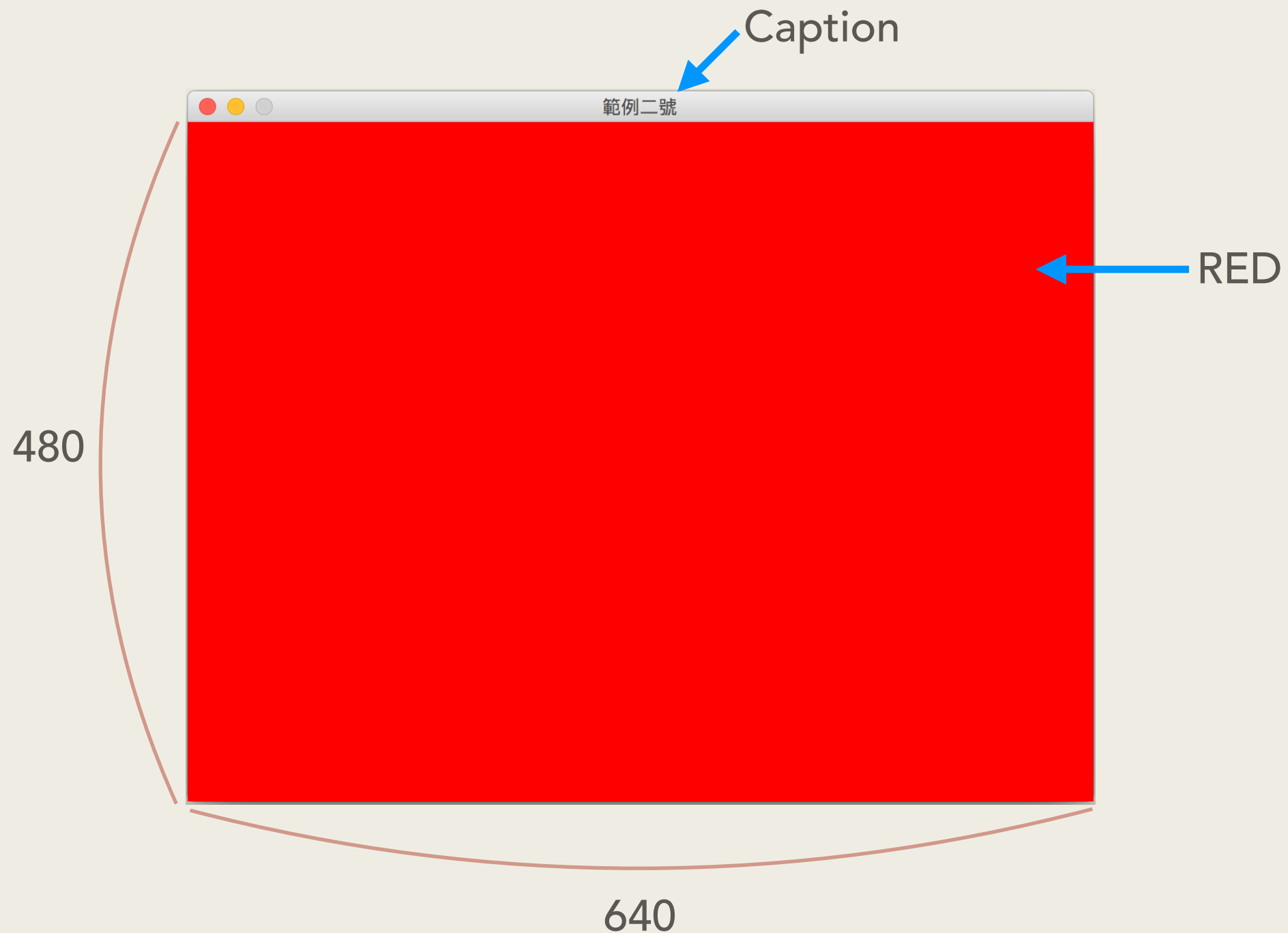
執行 example1.py

- 你應該會發現視窗跳出來之後馬上就關閉了

把這裡的註解拿掉

```
23 # 試試看編輯title
24 title = "範例一號"
25 pygame.display.set_caption(title)
26
27 # 更新畫面
28 pygame.display.flip()
29
30 '''
31 running = True
32 while running:
33     pass
34 '''
```

再次執行 example1.py



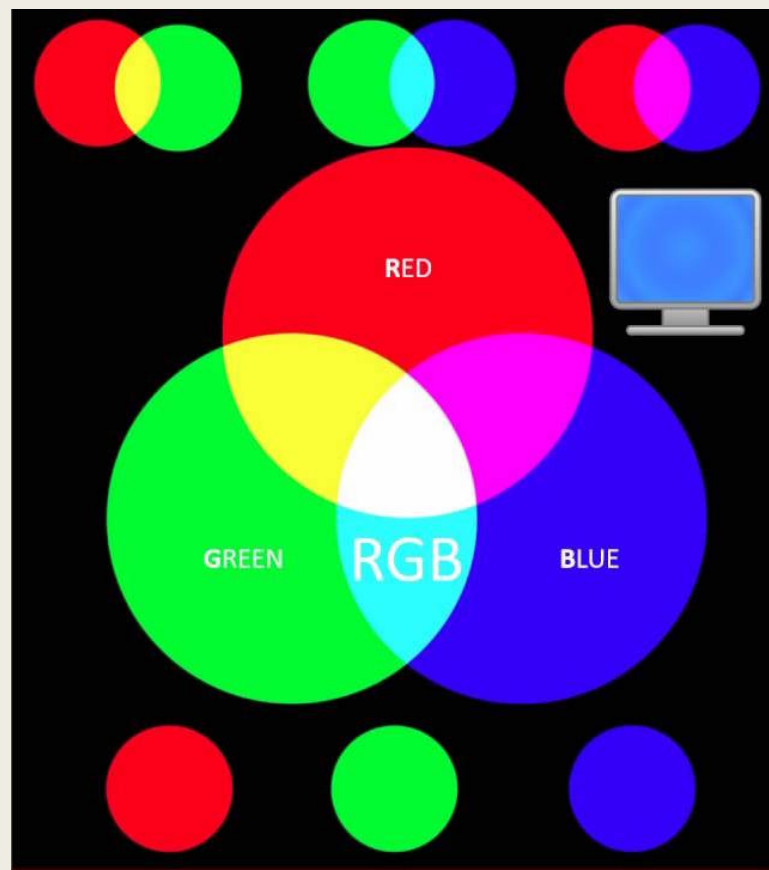
建立視窗與標題

- `pygame.init()`
- `screen =`
`pygame.display.set_mode(resolution,`
`flags, depth)`
 - `resolution` : 解析度
 - `flags` : 特殊設定
 - `depth` : 色彩位元數
- `pygame.display.set_caption(title)`
 - `title` : 標題

調整顏色

- `screen.fill(color)`
 - `color` : 顏色
 - 由 `r, g, b` 組成
 - 分別是 0~255 的整數

常見的顏色



$= (255, 0, 0)$

$= (0, 255, 0)$

$= (0, 0, 255)$

$= (255, 255, 255)$

$= (0, 0, 0)$

$= (192, 192, 192)$

- <https://www.webucator.com/blog/2015/03/python-color-constants-module/>

更新畫面

- `pygame.display.flip()`
 - 更新全部整個視窗
- `pygame.display.update(rectangle_list)`
 - 更新部分視窗
 - 若無指定，則效果相當於`flip()`

關閉視窗

- `pygame.quit()`

EVENT

事件

example2~3.py

執行 example2.py

- 按左上角的 x 或 ESC 都可以關閉視窗了。

取得發生的事件

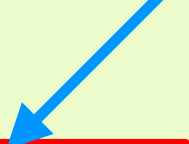
```
25 is_running = True
26 while is_running:
27     for event in pygame.event.get():
28         if event.type == pygame.QUIT:
29             is_running = False
30         elif event.type == pygame.KEYDOWN:
31             if event.key == pygame.K_ESCAPE:
32                 is_running = False
```

判斷事件類別

如果是KEYDOWN事件
判斷是不是按下了 ESCAPE

Event Type

KEYDOWN事件才有KEY



| | |
|-----------------|-------------------|
| QUIT | none |
| ACTIVEEVENT | gain, state |
| KEYDOWN | unicode, key, mod |
| KEYUP | key, mod |
| MOUSEMOTION | pos, rel, buttons |
| MOUSEBUTTONUP | pos, button |
| MOUSEBUTTONDOWN | pos, button |
| JOYAXISMOTION | joy, axis, value |
| JOYBALLMOTION | joy, ball, rel |
| JOYHATMOTION | joy, hat, value |
| JOYBUTTONUP | joy, button |
| JOYBUTTONDOWN | joy, button |
| VIDEORESIZE | size, w, h |
| VIDEOEXPOSE | none |
| USEREVENT | code |

Event

```
28     for event in pygame.event.get():  
29         print(event)
```

```
[(base) rilak sprout $ python example2.py
```

```
pygame 1.9.6
```

```
Hello from the pygame community. https://www.pygame.org/contribute.html
```

```
<Event(4-MouseMotion {'pos': (254, 337), 'rel': (254, 337), 'buttons': (0, 0, 0), 'window': None})>
```

```
<Event(2-KeyDown {'unicode': 'Π', 'key': 160, 'mod': 0, 'scancode': 0, 'window': None})>
```

```
<Event(3-KeyUp {'key': 160, 'mod': 0, 'scancode': 0, 'window': None})>
```

```
<Event(5-MouseButtonDown {'pos': (254, 337), 'button': 1, 'window': None})>
```

```
<Event(6-MouseButtonUp {'pos': (254, 337), 'button': 1, 'window': None})>
```

```
<Event(4-MouseMotion {'pos': (254, 336), 'rel': (0, -1), 'buttons': (0, 0, 0), 'window': None})>
```

```
<Event(4-MouseMotion {'pos': (254, 335), 'rel': (0, -1), 'buttons': (0, 0, 0), 'window': None})>
```

練習： example3.py

- 一開始螢幕是白色
- 滑鼠點擊後
 - 白色就會變黑色
 - 黑色就會變白色

```
16  bg_color = WHITE
```

```
36      # 重新繪製畫面  
37      background.fill(bg_color)
```

Ans

```
if event.type == pygame.MOUSEBUTTONDOWN:  
    if bg_color == WHITE:  
        bg_color = BLACK  
    else:  
        bg_color = WHITE
```




KEY

鍵盤

example2~3.py

Event Type

| | |
|-----------------|-------------------|
| QUIT | none |
| ACTIVEEVENT | gain, state |
| KEYDOWN | unicode, key, mod |
| KEYUP | key, mod |
| MOUSEMOTION | pos, rel, buttons |
| MOUSEBUTTONUP | pos, button |
| MOUSEBUTTONDOWN | pos, button |
| JOYAXISMOTION | joy, axis, value |
| JOYBALLMOTION | joy, ball, rel |
| JOYHATMOTION | joy, hat, value |
| JOYBUTTONUP | joy, button |
| JOYBUTTONDOWN | joy, button |
| VIDEORESIZE | size, w, h |
| VIDEOEXPOSE | none |
| USEREVENT | code |

Key Type

| Key | ASCII | Common Name | K_0 | 0 | 0 |
|--------------|-------|-------------------|---------|---|-------------|
| K_BACKSPACE | \b | backspace | K_1 | 1 | 1 |
| K_TAB | \t | tab | K_2 | 2 | 2 |
| K_CLEAR | | clear | K_3 | 3 | 3 |
| K_RETURN | \r | return | K_4 | 4 | 4 |
| K_PAUSE | | pause | K_a | a | a |
| K_ESCAPE | ^[| escape | K_b | b | b |
| K_SPACE | | space | K_c | c | c |
| K_EXCLAIM | ! | exclaim | K_d | d | d |
| K_QUOTEDBL | " | quotedbl | K_e | e | e |
| K_HASH | # | hash | K_f | f | f |
| K_DOLLAR | \$ | dollar | K_g | g | g |
| K_AMPERSAND | & | ampersand | K_h | h | h |
| K_QUOTE | | quote | K_i | i | i |
| K_LEFTPAREN | (| left parenthesis | K_UP | | up arrow |
| K_RIGHTPAREN |) | right parenthesis | K_DOWN | | down arrow |
| K_ASTERISK | * | asterisk | K_RIGHT | | right arrow |
| K_PLUS | + | plus sign | K_LEFT | | left arrow |
| K_COMMA | , | comma | | | |
| K_MINUS | - | minus sign | | | |

練習 example3.py

- 一開始螢幕是白色
- 如果按r，螢幕會變紅色
- 如果按g，螢幕會變綠色
- 如果按b，螢幕會變藍色
- 如果按w，螢幕會變回白色

example3_ans.py

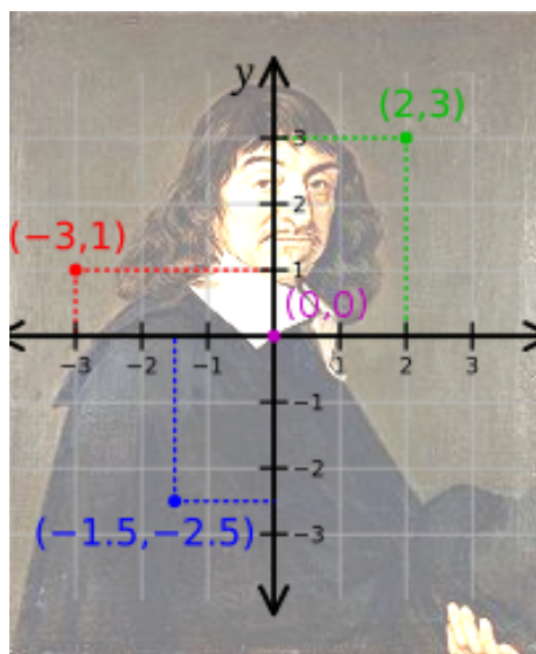
```
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_r:
        bg_color = RED
    elif event.key == pygame.K_g:
        bg_color = GREEN
    elif event.key == pygame.K_b:
        bg_color = BLUE
    elif event.key == pygame.K_w:
        bg_color = WHITE
```

DRAW

繪圖

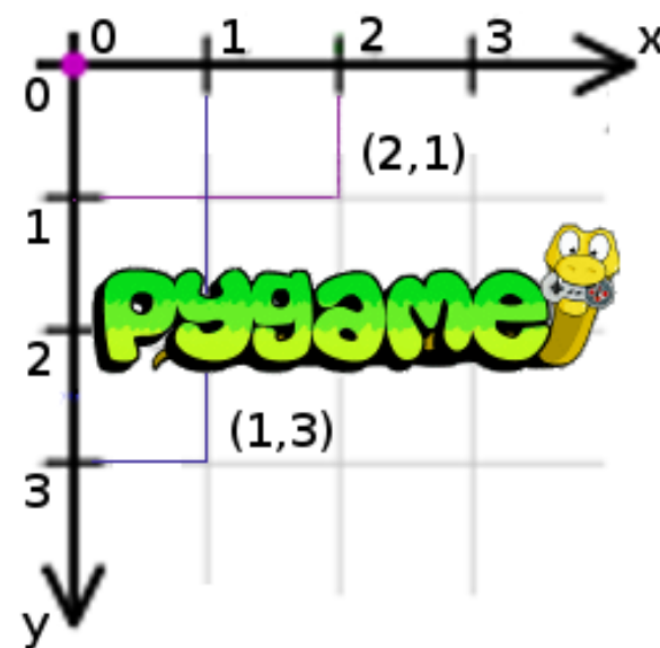
example4.py

Pygame 的座標系



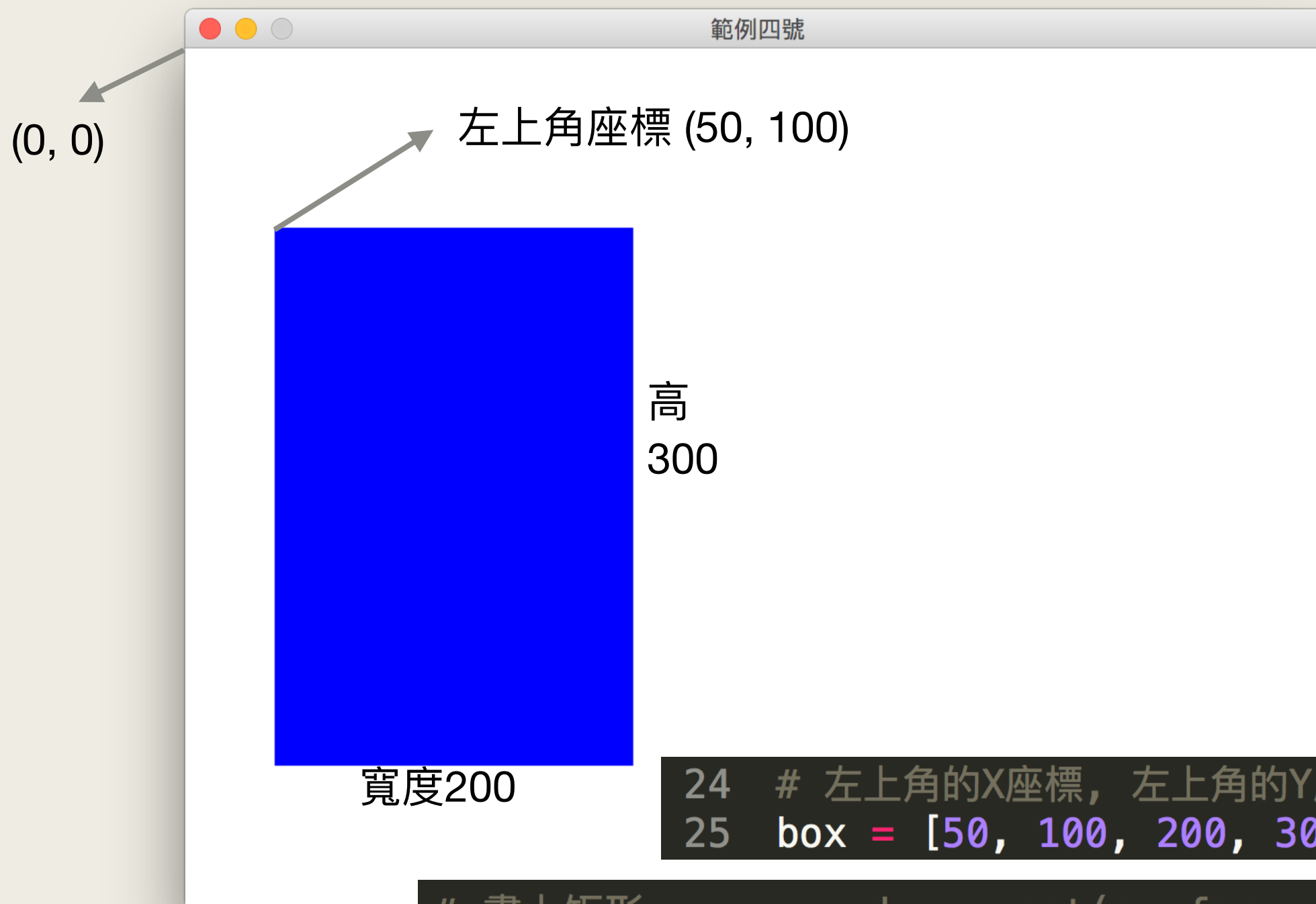
數學課的平面座標
(0, 0) 在中間

vs.



Pygame 的座標
(0, 0) 在左上角
x 和 y 只會是正的

example4.py



```
24 # 左上角的X座標，左上角的Y座標，寬度，高度  
25 box = [50, 100, 200, 300]
```

```
# 畫上矩形: pygame.draw.rect(surface, color, rect)  
pygame.draw.rect(screen, BLUE, box)
```


練習： `example4.py`

- 1. 設定 `box` 寬度和高度為 50
- 2. 按方向鍵上、下、左、右時移動方塊
 - `KEY_UP`
 - `KEY_DOWN`
 - `KEY_LEFT`
 - `KEY_RIGHT` （這個幫你寫好了）
- 3. 移動後要重新繪製畫面

example4_ans.py

```
26 # 左上角的X座標, 左上角的Y座標, 寬度, 高度
27 box = [50, 100, 50, 50]
28
29 is_running = True
30 while is_running:
31     for event in pygame.event.get():
32         if event.type == pygame.QUIT:
33             is_running = False
34         elif event.type == pygame.KEYDOWN:
35             if event.key == pygame.K_UP:
36                 box[1] -= 50
37             elif event.key == pygame.K_DOWN:
38                 box[1] += 50
39             elif event.key == pygame.K_RIGHT:
40                 box[0] += 50
41             elif event.key == pygame.K_LEFT:
42                 box[0] -= 50
43
44     # 清空畫面
45     screen.blit(background, (0, 0))
46
47     # 畫上矩形: pygame.draw.rect(surface, color, rect)
48     pygame.draw.rect(screen, BLUE, box)
```

繪圖

- `pygame.draw.rect(surface, color, rect, width)`
 - `surface`：要繪圖的surface
 - `color`：顏色
 - `rect`：長方形區域(空心)
 - `width`：邊框寬度，若為零則為實心

繪圖

- `pygame.draw.line(surface, color, start_pos, end_pos, width)`
 - `surface` : 要繪圖的surface
 - `color` : 顏色
 - `start_pos` : 開始位置
 - `end_pos` : 結束位置
 - `width` : 寬度

繪圖

- `pygame.draw.lines(surface, color, closed, point_list, width)`
 - `surface`：要繪圖的surface
 - `color`：顏色
 - `closed`：最後一個點是否要連回第一個點
 - `point_list`：起點、轉折點、終點的座標
 - `width`：寬度

Q&A

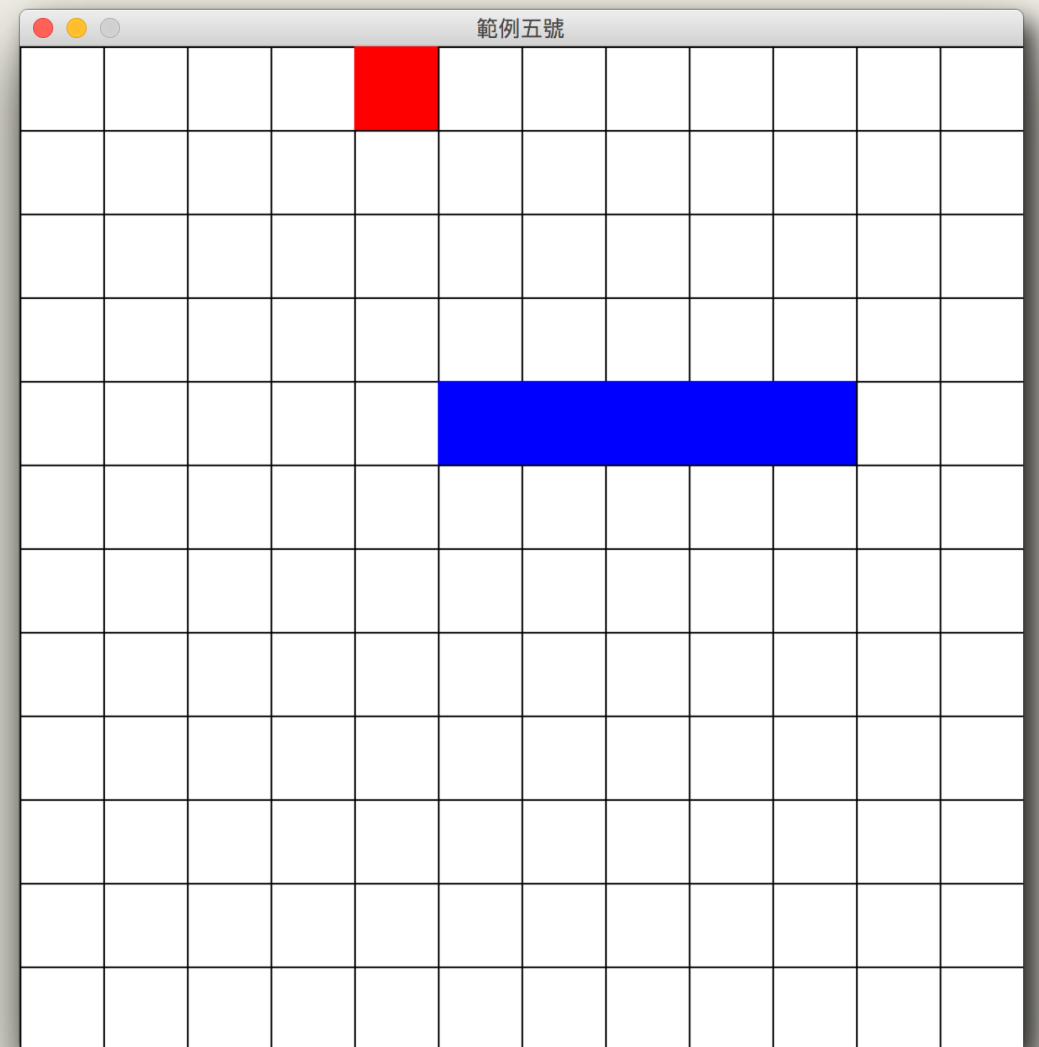
貪食蛇

會用到的語法幾乎都教了！

`example5~7.py`

再看一次遊戲畫面

- 畫背景格線
- 畫蛇蛇與食物
- 讓蛇蛇動
- 處理方向鍵
- 判斷遊戲邏輯



畫背景格線

- 因為格線不會動，所以直接畫在背景上

```
7 window_size = (600,600)
```

```
17 # 建立背景
18 background = pygame.Surface(screen.get_size())
19 background.fill(WHITE)
20 for i in range(0, 601, 50):
21     # pygame.draw.line(surface, color, start_pos, end_pos)
22     pygame.draw.line(background, BLACK, (i, 0), (i, 600))
23     pygame.draw.line(background, BLACK, (0, i), (600, i))
24 background = background.convert()
25 screen.blit(background, (0, 0))
```

畫蛇蛇與食物

- 把蛇蛇的身體看成是正方形的 `list`
- 一格一格畫上去！

```
32 class Snake():
33
34     def __init__(self, snake_color):
35         # 蛇蛇的身體！
36         # self.snake[-1] 是頭
37         self.body = [(1, 1), (1, 2), (1, 3), (2, 3), (2, 4)]
38         self.color = snake_color
39
40     def draw(self, surface):
41         for x, y in self.body:
42             pygame.draw.rect(surface, self.color, (x*50, y*50, 50, 50))
```

讓蛇蛇動

- 先用方向鍵控制蛇蛇，按一次動一格
- append 新的頭進去，pop 尾巴

```
44     def move(self, dx, dy):
45         head = self.body[-1]
46         new_head = (head[0] + dx, head[1] + dy)
47         self.body.append(new_head)
48         self.body.pop(0)
```

```
58         elif event.type == pygame.KEYDOWN:
59             if event.key == pygame.K_UP:
60                 snake.move(0, -1)
61             elif event.key == pygame.K_DOWN:
62                 snake.move(0, 1)
63             elif event.key == pygame.K_RIGHT:
64                 snake.move(1, 0)
65             elif event.key == pygame.K_LEFT:
66                 snake.move(-1, 0)
```

處理方向鍵 – Move

- 幫蛇蛇加上方向(dir)的屬性
- 把 Move 的 dx, dy 參數改成 self.dir

```
35     def __init__(self, snake_color):
36         # 蛇蛇的身體！
37         # self.snake[-1] 是頭
38         self.body = [(1, 1), (1, 2), (1, 3), (2, 3), (2, 4)]
39         self.color = snake_color
40         self.dir = (1, 0)
```

```
53     def move(self):
54         head = self.body[-1]
55         new_head = (head[0] + self.dir[0], head[1] + self.dir[1])
56         self.body.append(new_head)
57         self.body.pop(0)
```

處理方向鍵 – 蛇蛇暴衝!?

- 因為迴圈執行太快了> <!
- `clock = pygame.time.Clock()`
- `clock.tick(3)`

處理方向鍵 – 改方向！

- 寫個 Function 來設定方向吧！
- 避免回頭，所以要判斷新的方向會不會卡到脖子(?)

```
46     def change_dir(self, dx, dy):  
47         head = self.body[-1]  
48         neck = self.body[-2]  
49         if (head[0] + dx, head[1] + dy) == neck:  
50             return  
51         self.dir = (dx, dy)
```

判斷遊戲結束

- 加個 `is_alive` 屬性
- 如果吃到自己或是出界都改成 `False`
- 遊戲主迴圈如果看到蛇蛇死掉就跳出

判斷吃到東西

- 如果頭的位置和食物一樣
- 把身體加長 (尾巴多 x 個)