



# 2D 雙人對戰射擊遊戲

HeroMarine

---



# Outline

01. 遊戲內容

03. 程式碼實現

02. 遊戲操作、Demo



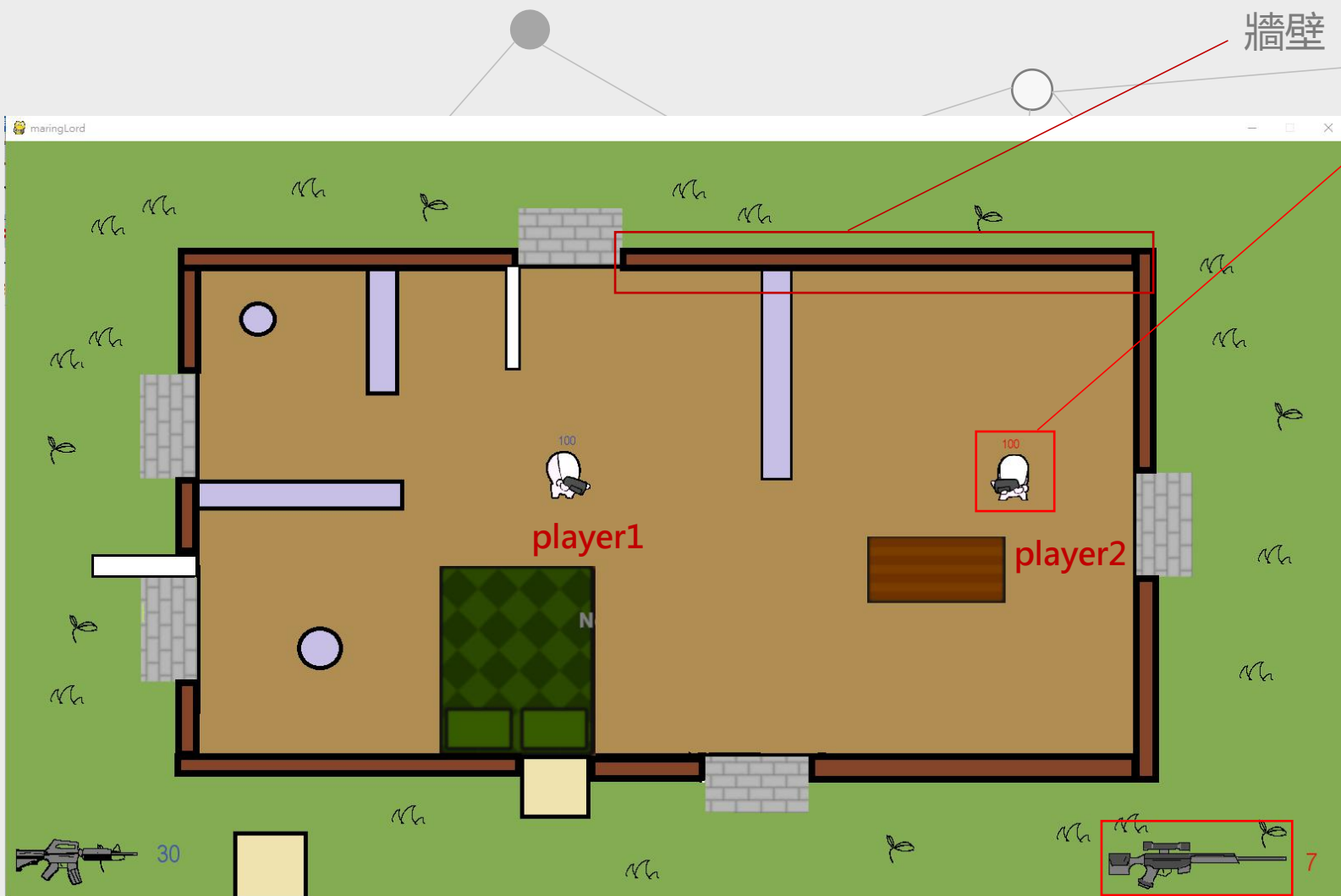
# 01

## 遊 戲 內 容

點選src下的main.py以開啟遊戲

# 遊戲畫面

## 雙人2D射擊遊戲



牆壁

人物

## 規則

雙方互相射擊，誰先將對方擊殺則獲勝

Player1:

- 1.機關槍
- 2.單發間隔短
- 3.殺傷力較低
- 4.有瞬跑功能

Player2:

- 1.狙擊槍
- 2.單發傷害高
- 3.冷卻較久



02

遊 戲 操 作

---

鍵盤輸入、demo

# 遊戲操作



wasd上下左右



player1



J射擊、F換彈

子彈數30、每發傷害  
15、最短間隔0.2s、  
換彈時間3s



H瞬跑

短暫爆衝



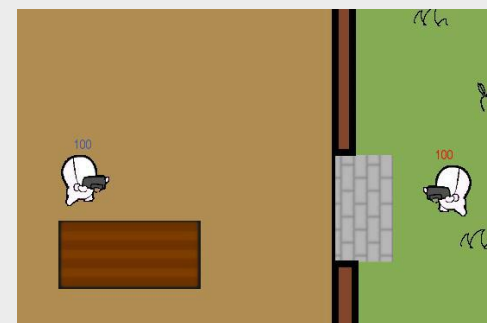
方向鍵上下左右

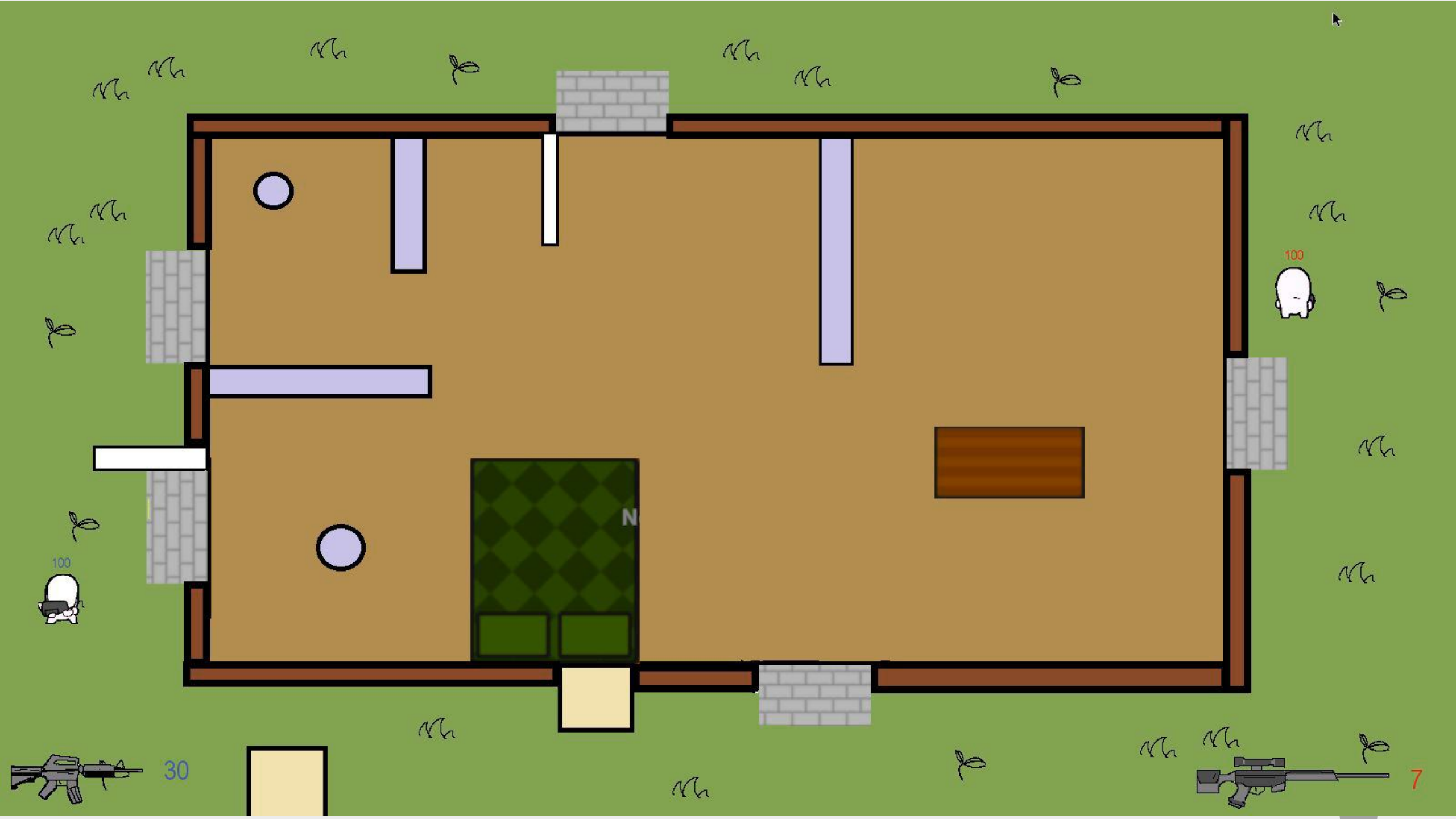


鍵盤右側0射擊  
鍵盤右側1換彈



子彈數 7、每發傷害50、間隔2s、換  
彈時間4s





# 03

## 程式碼實現

Pygame、小精靈、印圖片、音效、時序



# Pygame模組

- 1.設計用來開發電子遊戲的跨平臺Python模組
- 2.適合設計2D遊戲
- 3.使用簡單

## Pygame.Sprite

```
class MySoldier(pygame.sprite.Sprite):  
    def __init__(self, playerNumber):
```

小精靈，可以當成是遊戲內的物件。  
例如槍兵、牆壁、子彈等

## Pygame.rect

```
self.rect = pygame.Rect(left , top , width , height)
```

設定小精靈的大小以及在圖布上的位置

## screen.blit

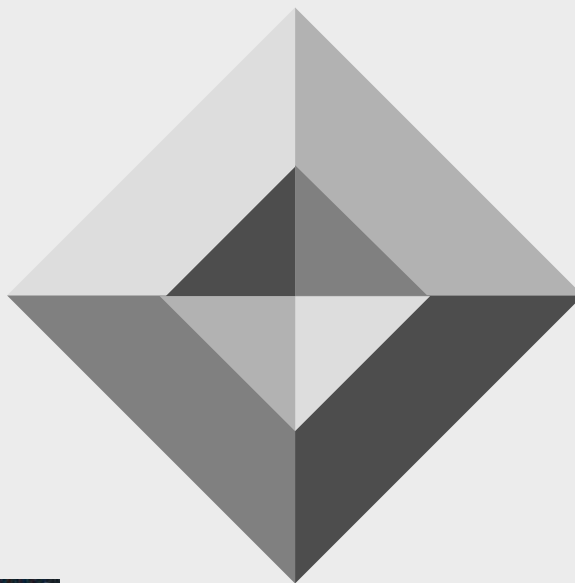
```
screen.blit(clip1_image , (10 , 830 ))
```

將圖片印在圖布上

## pygame.mixer

```
pygame.mixer.init()  
sniper_fire.play()
```

處理音樂、音效



# 程式流程

## 印出圖布上

將這一個LOOP改變的圖案、小精靈位置更新

## 碰撞判定

比如說子彈撞到人會消失，敵人血條減少、  
人物撞到牆壁會停止且不能穿牆

```
clock.tick(144)
```

每秒跑loop144次

## 移動各小精靈的位置以及圖片

依據上下左右，射擊，跑步動畫等更新圖片

及小精靈的位置



## 設定小精靈及背景音樂

設定marine、wall、bullet等小精靈。以及  
載入圖片、聲音等

```
pygame.init()
pygame.mixer.init()
resolution = 1600 , 900
screen = pygame.display.set_mode(resolution)
clip1_image = pygame.image.load(r"..\\image\\clip1.png").convert_alpha()
```

## 檢測及處理按鍵觸發

檢測玩家按了哪些按鍵、並進行動作。例如  
按J進行射擊就會有子彈聲、射擊FLAG = 1、  
開始計算射擊冷卻等。

# 1.製作小精靈

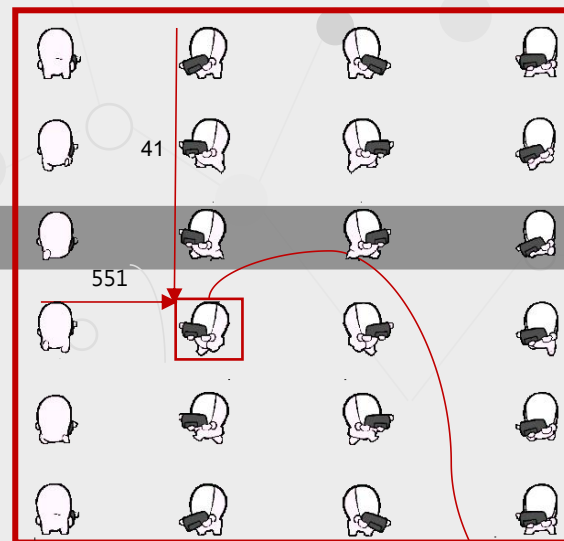
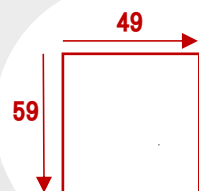
marine

## 1. 繼承

```
class MySoldier(pygame.sprite.Sprite):  
    def __init__(self, playerNumber):
```

## 2. 設定小精靈大小

```
Square_size_down = (49,59)  
self.soldier_R0 = self.soldier.subsurface((555,41),Square_size_down )  
self.rect = self.soldier_R0.get_rect()
```

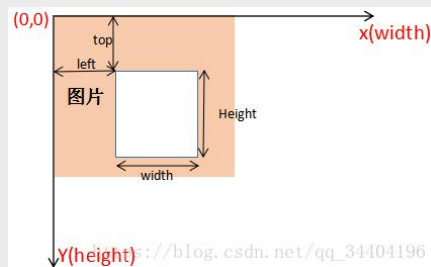


## 4. 疊起來完成



## 3. 得到子圖片

```
soldier_T1 = r"..\image\action.png"  
Square_size_down = (49,59)  
self.soldier_R0 = self.soldier_T1.subsurface((555,41),Square_size_down )
```



## 2. 檢測按鍵觸發

by `key_pressed = pygame .key.get_pressed()`

```
key_pressed = pygame .key.get_pressed()
if key_pressed[pygame.K_w]:
    movdir = 0
    running_T1 = True
    allsoldierGroup.remove(mySoldier_T1)
    if continuous_1 == movdir:
        mySoldier_T1.moveUp(1, allsoldierGroup ,bgmap.blockGroup )
    else:
        mySoldier_T1.moveUp(0, allsoldierGroup,bgmap.blockGroup)
    allsoldierGroup.add(mySoldier_T1)
```

```
if key_pressed[pygame.K_f]:
    if clip1_flag == 0:
        clip1_flag = 1
        gun_switch.play()
if key_pressed[pygame.K_h]:
    if h_delay_flag == 0:
        h_flag = 1
        h_delay_flag = 1
        mySoldier_T1.speed = 5
```

### 3.處理按鍵觸發

e.g:瞬間加速的冷卻及加速時間

h\_flag = 1代表 加速中  
h\_delay\_flag代表延遲中

```
if key_pressed[pygame.K_h]:  
    if h_delay_flag == 0:  
        h_flag = 1  
        h_delay_flag = 1  
        mySoldier_T1.speed = 5
```

#疾跑時間

```
if h_flag == 1:  
    h_clock += 1  
    if h_clock >= 30:  
        mySoldier_T1.speed = 2  
        h_flag = 0  
        h_clock = 0
```

#疾跑延遲

```
if h_delay_flag == 1:  
    h_delay_clock += 1  
    if h_delay_clock >= 144*3:  
        h_delay_flag = 0  
        h_delay_clock = 0
```

30/144秒時h\_flag = 0 表示加速結束

144\*3/144秒時h\_delay\_flag = 0 表示加速冷卻結束

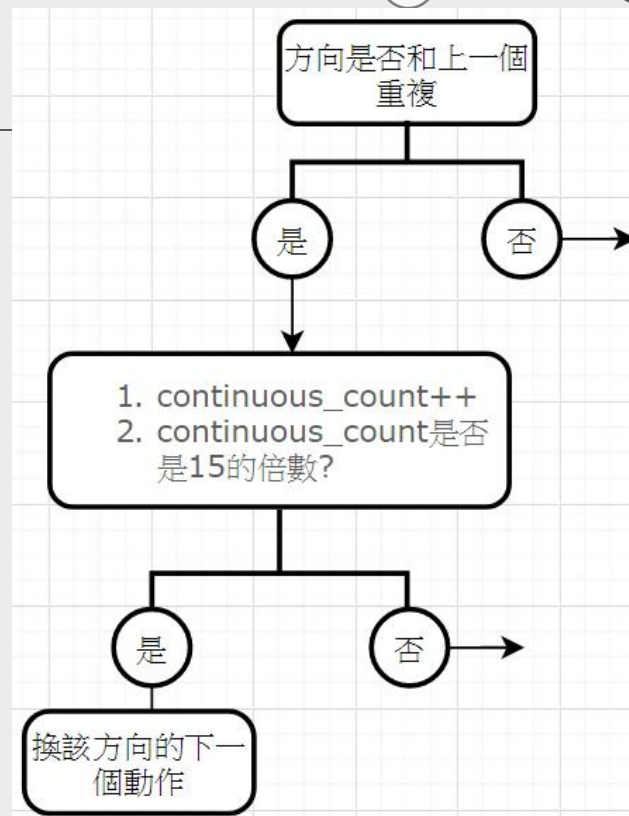
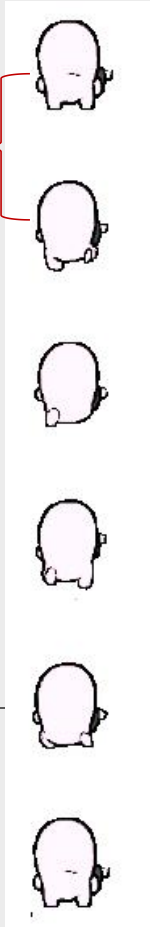
## 4.移動各小精靈的位置及圖片

eg.Marine 向上走動態圖

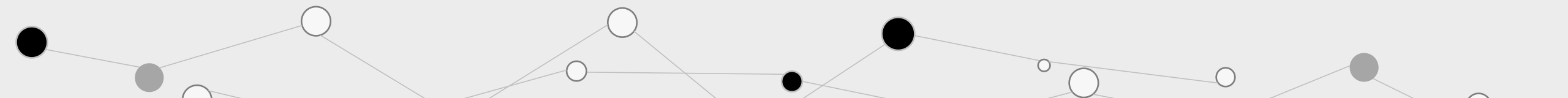
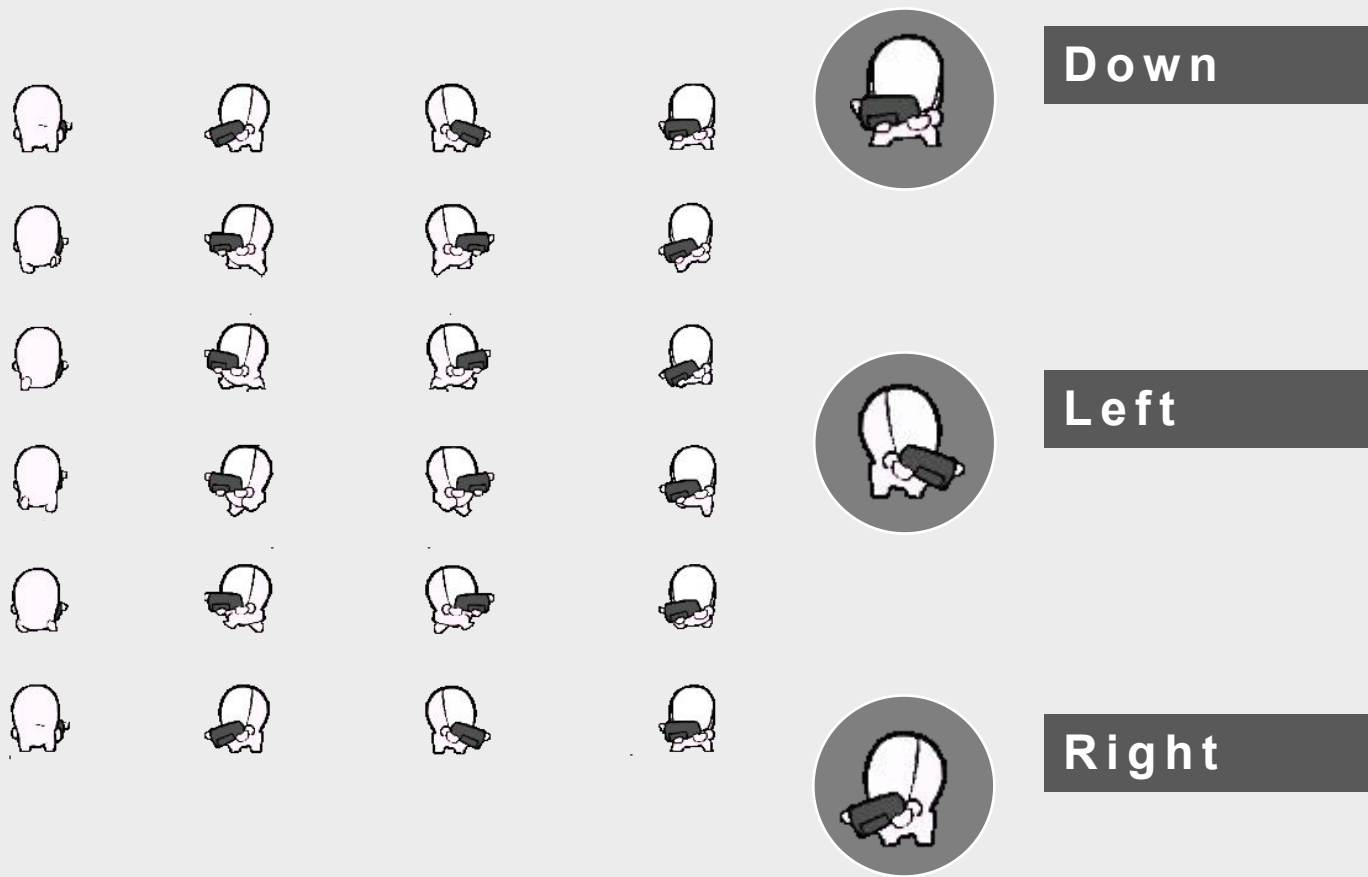
前一次移動的方向

```
if key_pressed[pygame.K_w]:  
    if continuous_1 == 0:  
        mySoldier_T1.moveUp(1, allsoldierGroup ,bgmap.blockGroup )  
    else:  
        mySoldier_T1.moveUp(0, allsoldierGroup,bgmap.blockGroup)  
  
def moveUp(self , re, allsoldierGroup , blockGroup):  
    self.rect = self.rect.move(self.speed * 0, self.speed * -1)  
    if re == 1:  
        self.continuous_count += 1  
        if self.action == 5 :  
            self.action = 0  
  
    if self.continuous_count % 15 == 0:  
        self.action += 1  
        self.continuous count == 0  
    self.soldier_R0 = self.soldier_image.subsurface((553, 41 + 100*self.action),Square_size_down )
```

15/144



## 人物-動態圖





# 碰撞判定

eg.子彈與牆的碰撞

## 1.將牆加入block小精靈群組

```
self.wall = block( 203 , 642 ,29 , 89 )  
self.blockGroup.add(self.wall)  
  
self.wall = block( 203 , 731 ,634 , 29 )  
self.blockGroup.add(self.wall)  
  
self.wall = block( 957 , 731 ,418 , 34 )  
self.blockGroup.add(self.wall)  
  
self.wall = block( 1345 , 517 ,31 , 247 )  
self.blockGroup.add(self.wall)  
  
self.wall = block( 1345 , 127 ,28 , 269 )  
self.blockGroup.add(self.wall)
```

## 2.pygame.sprite.spritecollideany( bullet, blockGroup )

小精靈內建函式，可根據小精靈矩陣檢測是否有碰撞

```
if pygame.sprite.spritecollideany(self.bullet, self.bgmap.blockGroup):  
    self.bullet.life = False
```



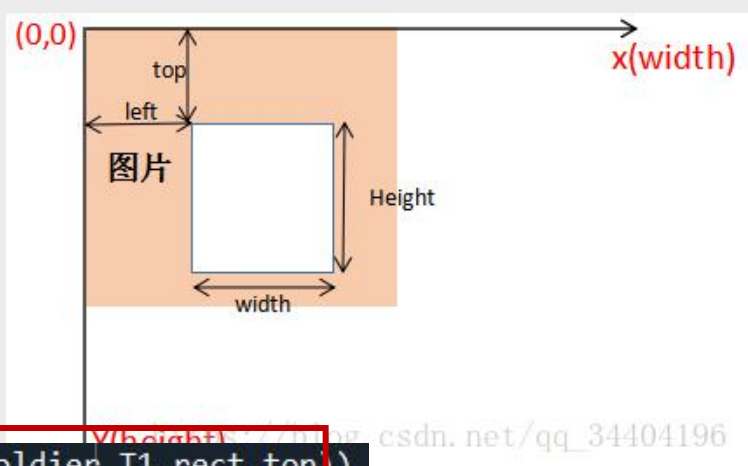
## 印在圖布上

eg. 印槍兵在圖布上

要印出的圖片

印在圖布上的位置

```
screen.blit(mySoldier_T1.soldier_R0, (mySoldier_T1.rect.left, mySoldier_T1.rect.top))
```



screen.blit 可以將小精靈、文字、圖片等印在圖布上

先載入圖片

```
clip1_image = pygame.image.load(r"..\\image\\clip1.png").convert_alpha()
```



```
screen.blit(clip1_image, (10, 830))
```

印在圖布上



謝 謝 各 位



# 1.製作小精靈

## 地圖及障礙物

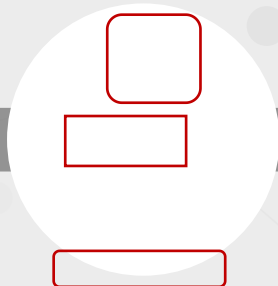
### 2.弄出障礙物小精靈的矩形以及位置

並且加入障礙物Group

不印出障礙物圖片

### 3.疊上背景圖

```
#繪製背景  
screen.blit(background_image, (0, 0))
```



### 1.製作障礙物小精靈

```
import pygame  
  
class block (pygame.sprite.Sprite):  
    def __init__(self, left , top , width , height):  
        pygame.sprite.Sprite.__init__(self)  
        self.rect = pygame.Rect(left , top , width , height)  
  
class Map():  
    def __init__(self):  
        self.blockGroup = pygame.sprite.Group()  
        #房子的外牆 由左下開始屬  
        self.wall = block( 203 , 642 ,29 , 89 )  
        self.blockGroup.add(self.wall)  
  
        self.wall = block( 203 , 731 ,634 , 29 )  
        self.blockGroup.add(self.wall)
```

