

# Pygame

## 第四組

110207315

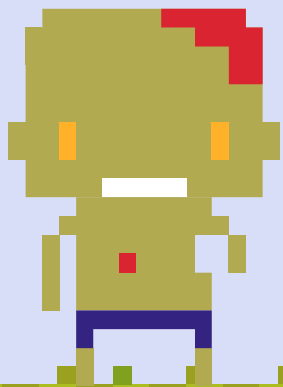
地政一 蔣喬同

110502003

阿語一 陳士穎

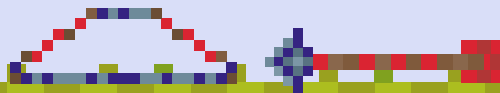
110502006

阿語一 江采育





# INTRODUCTION





# INTRODUCTION

Pygame是跨平台Python模組，專為電子遊戲設計，包含圖像、聲音。

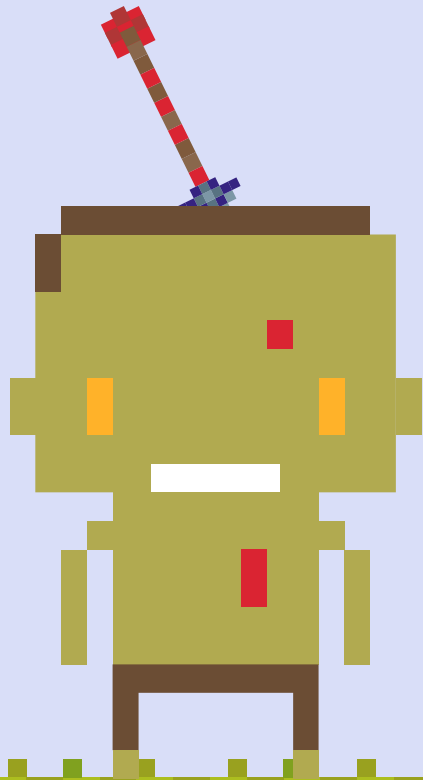
建立在SDL基礎上，允許即時電子遊戲研發而無需被低階語言，如C語言或是更低階的組合語言束縛。

基於這樣的設想，所有需要的遊戲功能和理念都（主要是圖像方面）完全簡化遊戲邏輯本身，所有的資源結構都能由高階語言提供，Python本身就是個高階程式語言。

SDL (Simple DirectMedia Layer) :

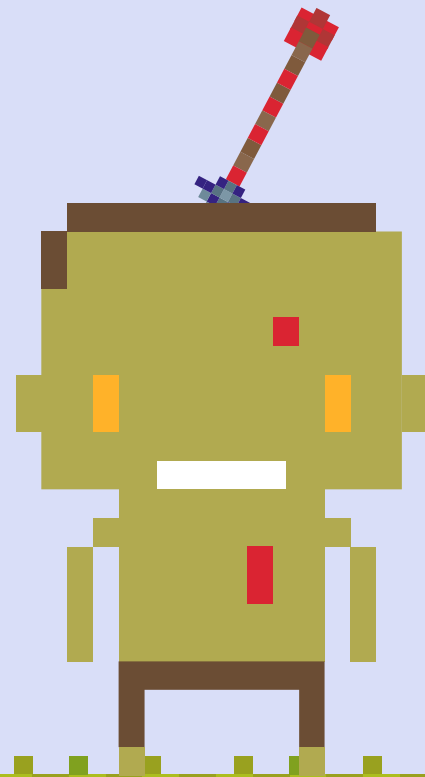
一套開放原始碼的跨平台多媒體函式庫，以C語言撰寫

資料來源：<https://zh.wikipedia.org/zh-tw/Pygame>





# PROJECT INTRODUCTION





# Project introduction



## 遊戲目標

擊敗怪物，最後將以遊戲時間、對怪物造成的傷害  
及自身所受的傷害為評分標準。

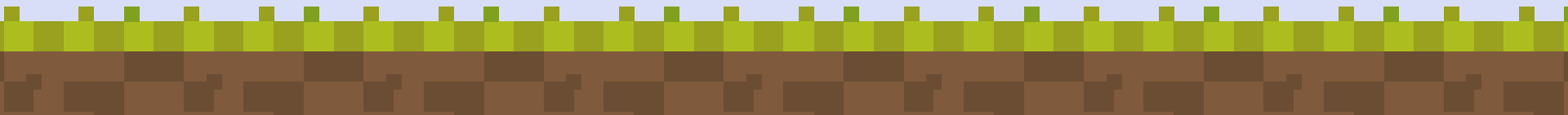




# Project introduction

## 操作方法：

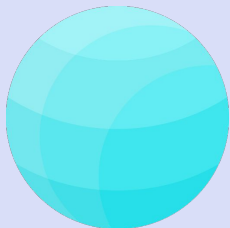
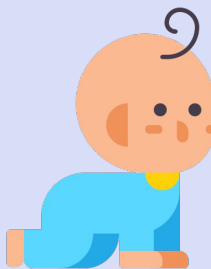
1. 遊戲主選單按下P開始遊戲，按Q退出遊戲
2. WASD鍵：上下左右移動
3. IJKL鍵：視角（射擊方向改變）
4. 空白鍵：發射子彈
5. 遊戲過程中或結束後可按e鍵返回主畫面

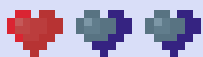




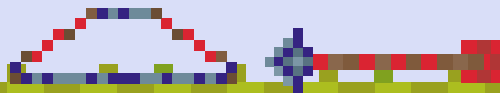
# Project introduction

遊戲物件：





**REFERENCE**








# Pygame documentation





pygame documentation

★ **pygame 4000 book update** ★ Download pygame book, plus example code here.

Pygame Home || [Help](#) [Contents](#) || [Reference](#) [Index](#)

**Most useful stuff:** [Color](#) | [display](#) | [draw](#) | [event](#) | [font](#) | [image](#) | [key](#) | [locals](#) | [mixer](#) | [mouse](#) | [Rect](#) | [Surface](#) | [time](#) | [music](#) | [pygame](#)

**Advanced stuff:** [cursors](#) | [joystick](#) | [mask](#) | [sprite](#) | [transform](#) | [BufferProxy](#) | [freetype](#) | [gfxdraw](#) | [midi](#) | [PixelArray](#) | [pixelcopy](#) | [sndarray](#) | [surfarray](#) | [math](#)

**Other:** [camera](#) | [controller](#) | [examples](#) | [fastevent](#) | [scrap](#) | [tests](#) | [touch](#) | [version](#)

## Pygame Front Page

### Documents

[Readme](#)  
Basic information about pygame: what it is, who is involved, and where to find it.

[Install](#)  
Steps needed to compile pygame on several platforms. Also help on finding and installing prebuilt binaries for your system.

[File Path Function Arguments](#)  
How pygame handles file system paths.

[LGPL License](#)  
This is the license pygame is distributed under. It provides for pygame to be distributed with open source and commercial software. Generally, if pygame is not changed, it can be used with any type of program.

### Tutorials

[Introduction to Pygame](#)  
An introduction to the basics of pygame. This is written for users of Python and appeared in volume two of the Py magazine.

<https://www.pygame.org/docs/index.html>





# Opengameart.org

**OPENGAMEART.ORG**

OpenID Username or Password **Log In** Register

Home Browse Submit Art Collect Forums FAQ Leaderboards Donate

Follow us:   

**CHAT WITH US!**  
**Discord:** OpenGameArt  
<https://discord.gg/yDaQ4NcCux>  
**IRC:** #OpenGameArt on  
<https://freegamedev.net/irc/#opengameart>

**ACTIVE FORUM TOPICS - (VIEW MORE)**

- New PC, PS3 and Xbox360 Game - Knights of al-Aqsa Mosque 21 hours 13 min ago by MedicineStorm
- Building a Library of Images for Everyone 3 days 23 hours ago by Eric Matyas
- Free sound effects for game developers 1 week 2 days ago by Commander
- Summer 2022 Art Challenge - Platformer 1 week 2 days ago by Tsothan Grove
- Bump Requests 1 week 2 days ago by MedicineStorm
- [For Hire] Pixel artist open for commissions 1 week 5 days ago by gillymew
- Sharing My Music and Sound FX - Over 2000 Tracks 2 weeks 14 min ago by Eric Matyas
- Vagabond enters in alpha 2 weeks 42 min ago by pvigier

**RECENT COMMENTS - (VIEW MORE)**

- Re: Destiny by Xom Adept
- Re: 24 cute fighting game voice recordings by kamacheng

**LEGAL NOTICE REGARDING NFTs:**

**WARNING:** Taking art from OpenGameArt.org to be sold as NFTs? You may be committing FRAUD.

Visit this link for legal details: <https://opengameart.org/content/warning-taking-art-from-opengameartorg-t...>

**Note of caution to NFT purchasers or those interested in trading NFTs: You could be getting scammed!**

Please visit this link for more information: <https://opengameart.org/content/note-of-caution-to-nft-purchasers-or-tho...>

**POPULAR THIS WEEK - (VIEW MORE)**

<b>DESTINY</b> 	<b>LOW POLY MANA...</b> 	<b>SPRITE EFFECTS ...</b> 	<b>WASTELAND CAR...</b> 
<b>FOOL'S PHILOSO...</b> 	<b>THE GRAND PLAN</b> 	<b>DIZZY RACING</b> 	<b>SHIELD AND SWO...</b> 

**LATEST ART - (VIEW MORE)**

**POPULAR THIS MONTH**

16x16 Assorted RPG Icons	21
Puny Characters	14
MiniWorld Sprites	10
Nordic Tileset	9
Little Epic Journey	8
It Takes A Hero	7
Too Strong	6
Together One Last Time - by ATMANAN	6
Red Engines Blue Wings	6
Graffiti Constructor	6

**FAVORITE SUBMITTERS THIS MONTH**

Shade	35
Alexandr Zhelanov	24
Sevarihk	19
ATMANAN	12
Tsothan Grove	10
DavidR	8
Zane Little Music	8
CraftPix.net 2D Game Assets	7
AntumDeluge	7
Umplix	7

**POPULAR ART COLLECTIONS - (VIEW MORE)**

- Epic fantasy music
- OGA 16x16 JRPG Sprites & Tiles
- Liberated Pixel Cup
- Space Game Starter Set
- Sideview pixel art RPG enemy sprites
- Platformer/Sidescroller Tiles and Backgrounds
- User Interface

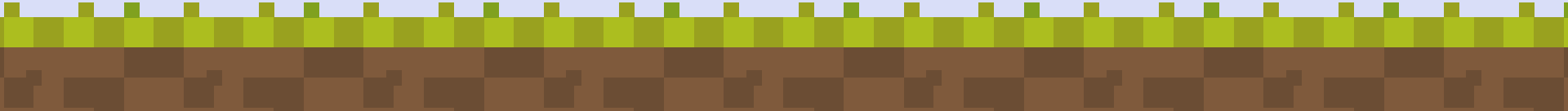


# 套件使用

`pygame`：遊戲主要使用的套件，提供遊戲運作的功能

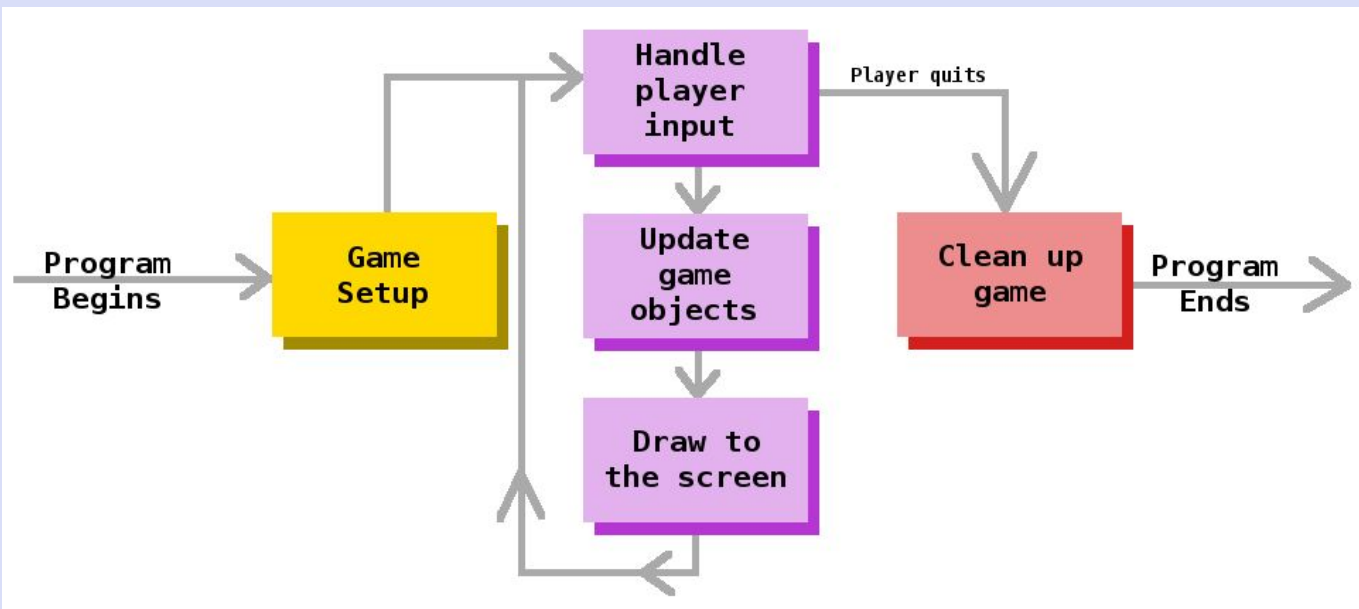
`math`：提供較複雜的運算，如`sin`、`cos`、`pi`等

`random`：提供隨機的數值，讓遊戲內容更多變





# 遊戲迴圈





# 數學運用

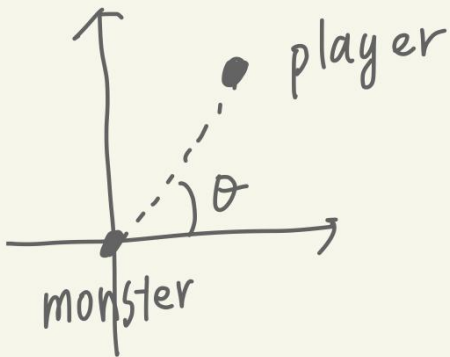


該如何讓怪物自動瞄準玩家？





# 數學運用



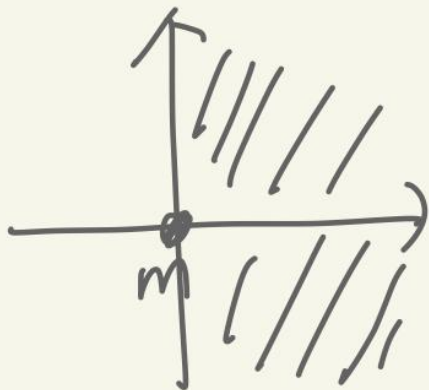
$$m = \frac{\Delta y}{\Delta x} = \tan \theta$$

$$\theta = \arctan m$$



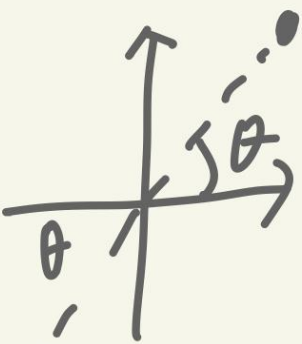
# 數學運用

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$





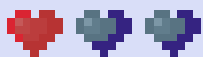
## 數學運用



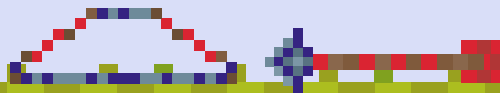
$P(r \cos \theta, r \sin \theta)$

$$P'(r \cos(180 + \theta), r \sin(180 + \theta))$$
$$= (-r \cos \theta, -r \sin \theta)$$





# PROGRAMMING





# 初始化

```
import pygame
import random
import math
```

```
#initiatize the pygame
pygame.init()

#setting
FPS = 60
NAME = 'BABY VS MONSTERS!'
HEIGHT = 800
WIDTH = 600
GAMEMODE = 1
SCORE = 0
START = 0 #game_state
STATE = 0 #player's state
MOUSEX = pygame.mouse.get_pos()[0]
MOUSEY = pygame.mouse.get_pos()[1]
TICK1 = 300
TICK2 = random.randrange(500, 1000)
TICKM = random.randrange(10, 60)
TICKS = 500
POISONNUM = 0
is_ADDED = False
```



# 初始化

```
#loading_image
KIDSIIMAGE = pygame.image.load('baby-boy.png')
MONSTER1 = pygame.image.load('monster.png')
MONSTER1_HURTED = pygame.image.load('monster_hurted.png')
POISON = pygame.image.load('poison.png')
THREE = pygame.image.load('three.png')
TWO = pygame.image.load('2.png')
ONE = pygame.image.load('1.png')
BALL = pygame.image.load('fitness-ball.png')
FIREBALL = pygame.image.load('fireball.png')
HEALTHPACK = pygame.image.load('heart pixel art 48x48.png')
BACKGROUND = pygame.image.load('background.jpg')
BACKGROUND = pygame.transform.scale(BACKGROUND, (800, 500))
BACKGROUND_DARK = pygame.image.load('background_dark.jpg')
BACKGROUND_DARK = pygame.transform.scale(BACKGROUND_DARK, (800, 500))
ICON = pygame.image.load('monster.png')
MENU = pygame.image.load('play button.png')
BGSOUND = pygame.mixer.Sound('Lull.mp3')
SHOOTINGSOUND = pygame.mixer.Sound('8bit_bomb_explosion.wav')
```

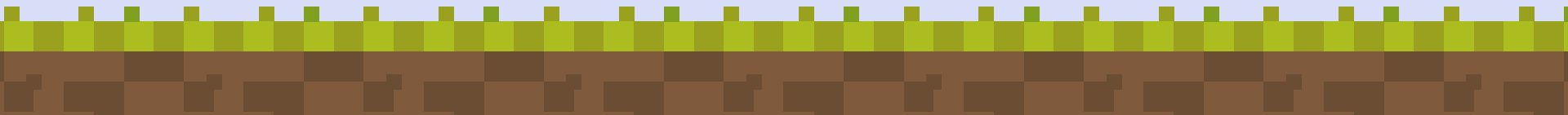


# 初始化

```
#loading fonts
FONT = pygame.font.match_font('arial')

clock = pygame.time.Clock()

#icon setting
screen = pygame.display.set_mode((HEIGHT, WIDTH)) #the size of your screen
pygame.display.set_caption(NAME)#title
pygame.display.set_icon(ICON)
```





# Class player 1

```
class Player(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(KIDSIMAGE, (70, 70))
        self.rect = self.image.get_rect()
        self.rect.x = 200
        self.rect.y = 200
        self.speedx = 8
        self.speedy = 8
        self.rect.centerx = 800/2
        self.rect.bottom = 600-10
        self.face = 0
        self.isrotated = 0
        self.health = 100
```



## Class player 2

```
def update(self):
    self.faces()
    key_pressed = pygame.key.get_pressed()#return a boolean that whether the key is pressed
    if key_pressed[pygame.K_d]:
        self.rect.x += self.speedx
    if key_pressed[pygame.K_a]:
        self.rect.x -= self.speedx
    if key_pressed[pygame.K_s]:
        self.rect.y += self.speedy
    if key_pressed[pygame.K_w]:
        self.rect.y -= self.speedy
    if self.rect.right > 800:
        self.rect.right = 800
    if self.rect.left < 0:
        self.rect.left = 0
    if self.rect.top < 0:
        self.rect.top = 0
    if self.rect.bottom > 600:
        self.rect.bottom = 600
```



# Class player 3

```
def shoot(self, direction):
    global STATE
    if STATE == 0:
        bullet = Bullet(self.rect.centerx, self.rect.top, direction, STATE)
    elif STATE == 1:
        bullet = Bullet(self.rect.centerx, self.rect.top, direction, STATE)
    bullets.add(bullet)
    all_sprites.add(bullet)
    SHOOTINGSOUND.play()

def faces(self):
    if(self.isrotated == 1):
        self.image = pygame.transform.flip(self.image, True, False)
        self.isrotated = 0

def getHurted(self, damage):
    global SCORE, START
    self.health -= damage
    if(self.health <= 0):
        self.kill()
        START = 3
    if(START == 2):
        SCORE -=1
```



# Class bullet

```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y, direction, type):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((10, 10))
        self.image.fill((255, 255, 0))
        self.rect = self.image.get_rect()
        self.rect.centerx = x
        self.rect.bottom = y
        self.speed = 10
        self.direction = direction
        self.type = type
        self.damage = 1

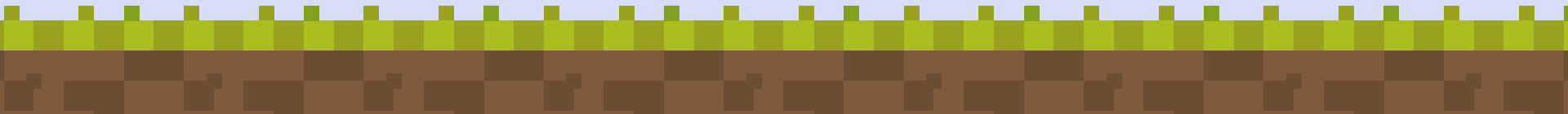
    def update(self):
        if self.type == 1:
            self.image = pygame.transform.scale(BALL, (20, 20))
            self.speed = 6
            self.damage = 10
        self.rect.y += self.speed * math.sin(self.direction)
        self.rect.x += self.speed * math.cos(self.direction)
        if self.rect.bottom < 0:
            self.kill()
```





# Class monster 1

```
class Monster(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(MONSTER1, (100, 100))
        self.rect = self.image.get_rect()
        self.rect.x = random.randrange(100, 700)
        self.rect.y = random.randrange(0, 600)
        self.v = 3
        self.direction = math.pi*2/random.randrange(1, 6)
        self.health = 50
        self.tick = random.randrange(20, 70)
        self.animation = 30
        self.is_animated = False
```





# Class monster 2

```
def update(self):
    global START
    if self.rect.top < 0 or self.rect.bottom > 600 or self.rect.left > 800 or self.rect.right < 0:
        self.v = -self.v
        self.direction = math.pi*2 / random.randrange(1, 6)
    self.rect.y += self.v*math.sin(self.direction)
    self.rect.x += self.v*math.cos(self.direction)
    self.tick -= 1
    if self.is_animated == True:
        self.image = pygame.transform.scale(MONSTER1_HURTED, (100, 100))
        self.animation -= 1
    if self.animation <= 0:
        self.is_animated = False
        self.image = pygame.transform.scale(MONSTER1, (100, 100))
        self.animation = 30

def getHurted(self, damage):
    self.health -= damage
    self.is_animated = True
    if(self.health <= 0):
        self.kill()
    # self.image = pygame.transform.scale(MONSTER1, (100, 100))
    global SCORE, START
    if(START == 2):
        SCORE += damage
```



## Class monster 3

```
def attack(self, player):  
    playerx = player.rect.x  
    playery = player.rect.y  
    monsterx = self.rect.x  
    monstery = self.rect.y  
    deltax = monsterx - playerx  
    if deltax == 0:  
        deltax = 1  
    angle = math.atan((monstery - playery)/deltax)  
    if deltax > 0:  
        direction = 0  
    elif deltax <= 0:  
        direction = 1  
    monsterbullet = MonsterBullet(self.rect.centerx, self.rect.top, angle, 1, direction)  
    monsterbullets.add(monsterbullet)  
    all_sprites.add(monsterbullet)
```



# Class monsterbullets

```
class MonsterBullet(pygame.sprite.Sprite):
    def __init__(self, x, y, angle, type, direction):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(FIREBALL, (20, 20))
        self.rect = self.image.get_rect()
        self.rect.centerx = x
        self.rect.bottom = y
        self.speed = 10
        self.angle = angle
        self.direction = direction
        self.type = type
        self.damage = 3

    def update(self):
        global START
        if START == 2:
            if self.type == 1:
                pass
            if self.direction == 0:
                self.rect.y -= self.speed * math.sin(self.angle)
                self.rect.x -= self.speed * math.cos(self.angle)
            if self.direction == 1:
                self.rect.y += self.speed * math.sin(self.angle)
                self.rect.x += self.speed * math.cos(self.angle)
            if self.rect.bottom < 0:
                self.kill()
```



## Class poison

```
class Poison(pygame.sprite.Sprite):  
    def __init__(self):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.transform.scale(POISON, (50, 50))  
        self.rect = self.image.get_rect()  
        self.rect.x = random.randrange(100, 700)  
        self.rect.y = random.randrange(100, 400)  
        self.type = random.randrange(0, 3)  
  
    def update(self):  
        if self.type == 2:  
            self.image = pygame.transform.scale(HEALTHPACK, (30, 30))
```



# Class menu

```
class Menu(pygame.sprite.Sprite):  
    def __init__(self):  
        pygame.sprite.Sprite.__init__(self)  
        BGSOUND.play()  
        self.image = pygame.transform.scale(MENU, (200, 100))  
        self.rect = self.image.get_rect()  
        self.rect.x = 0  
        self.rect.y = 0  
  
    def update(self):  
        global START  
        if START == 0:  
            pass  
        elif START == 1 or START == 2:  
            self.kill()
```



# Class number

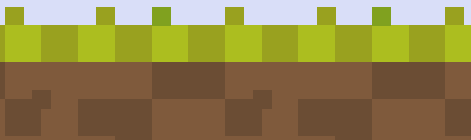
```
class Number(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(THREE, (100, 100))
        self.rect = self.image.get_rect()
        self.rect.centerx = 400
        self.rect.top = 300
        self.tick = 300

    def update(self):
        self.tick -= 1
        if self.tick <= 200:
            self.image = pygame.transform.scale(TWO, (100, 100))
        if self.tick <= 100:
            self.image = pygame.transform.scale(ONE, (100, 100))
        if self.tick < 0:
            global START
            START = 2
            self.kill()
```



# Sprites Group

```
#sprites adding
all_sprites = pygame.sprite.Group()
poisons = pygame.sprite.Group()
bullets = pygame.sprite.Group()
monsterbullets = pygame.sprite.Group()
monsters = pygame.sprite.Group()
players = pygame.sprite.Group()
player = Player()
num = Number()
menu = Menu()
menus = pygame.sprite.Group()
menus.add(menu)
nums = pygame.sprite.Group()
nums.add(num)
players.add(player)
for i in range(3):
    monster = Monster()
    all_sprites.add(monster)
    monsters.add(monster)
all_sprites.add(player)
```







## 自訂函數、簡化主程序

```
#display the scores  
def draw_text(surface, text, size, x, y):  
    font = pygame.font.Font(FONT, size)  
    text_surface = font.render(text, True, (255, 255, 255))  
    text_rect = text_surface.get_rect()  
    text_rect.centerx = x  
    text_rect.top = y  
    pygame.font.Font.set_bold(font, True)  
    surface.blit(text_surface, text_rect)
```



# 主程序

```
#Game Loop
running = True
while running:
    TICK2 -= 1
    clock.tick(FPS) #一秒中只能被執行10次
```



# 主程序：按鍵偵測

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        running = False
        # saving()
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_p and START == 0:
            START = 1
            BGSOUND.fadeout(5000)
        if event.key == pygame.K_q and START == 0:
            running = False
        if event.key == pygame.K_e and (START == 2 or START == 3):
            START = 0
            BGSOUND.play()
        if event.key == pygame.K_l and player.face != 0:
            player.face = 0
            player.isrotated = 1
        if event.key == pygame.K_i:
            player.face = math.pi*3/2
        if event.key == pygame.K_j and player.face != math.pi:
            player.face = math.pi
            player.isrotated = 1
        if event.key == pygame.K_k:
            player.face = math.pi/2
        if event.key == pygame.K_SPACE:
            player.shoot(player.face)
```



# 主程序：遊戲狀態

```
if(START == 0):  
    if len(menus) == 0:  
        menu = Menu()  
        menus.add(menu)  
    menus.update()  
elif(START == 1):  
    if len(nums) == 0:  
        print(True)  
        num = Number()  
        nums.add(num)  
    nums.update()  
elif(START == 2):  
    all_sprites.update()  
    if len(monsters) == 0:  
        START = 3  
        all_sprites = pygame.sprite.Group()  
elif(START == 3):  
    pass
```



## 主程序：物件碰撞

```
for monster in monsters:
    hits = pygame.sprite.spritecollide(monster, bullets, True)
    for hit in hits:
        monster.getHurted(hit.damage)

for player in players:
    hits = pygame.sprite.spritecollide(player, monsterbullets, True)
    for hit in hits:
        player.getHurted(hit.damage)

for player in players:
    hits = pygame.sprite.spritecollide(player, poisons, True)
    STATE = random.randrange(0, 2)
    POISONNUM -= 1
    for hit in hits:
        if hit.type == 2:
            player.health += 20
```



## 主程序：怪物攻擊時機

```
TICKM -= 1
for player in players:
    #if TICKM <= 0:
    for monster in monsters:
        if monster.tick <= 0:
            monster.attack(player)
            monster.tick = random.randrange(10, 60)
```



# 主程序

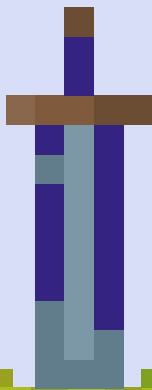
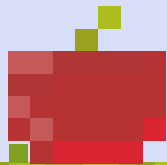
```
screen.fill((0,0,0)) #上色rgb
if TICK1 > 0:
    screen.blit(BACKGROUND, (0,100))
    if START != 0:
        TICK1 -= 1
    else:
        screen.blit(BACKGROUND_DARK, (0,100))
if TICK2 < 0 and POISONNUM < 2:
    poison = Poison()
    poisons.add(poison)
    all_sprites.add(poison)
    TICK2 = random.randrange(400,700)
    POISONNUM += 1
if START == 0:
    menus.draw(screen)
elif START == 1:
    nums.draw(screen)
elif START == 2:
    all_sprites.draw(screen)
    draw_text(screen, f'score: {str(SCORE)} health: {player.health}', 18, 400, 10)
    TICKS -= 0.1
elif START == 3:
    totalscores = int(SCORE + TICKS)
    draw_text(screen, f'Total Score: {str(totalscores)} press E to exit', 18, 400, 10)
pygame.display.update()
```

```
pygame.quit()
```

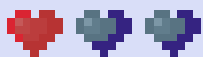


04.

DEMO

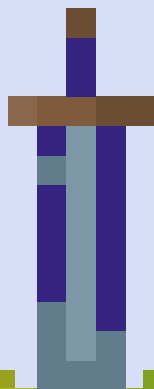
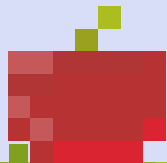






05.

CONCLUSION

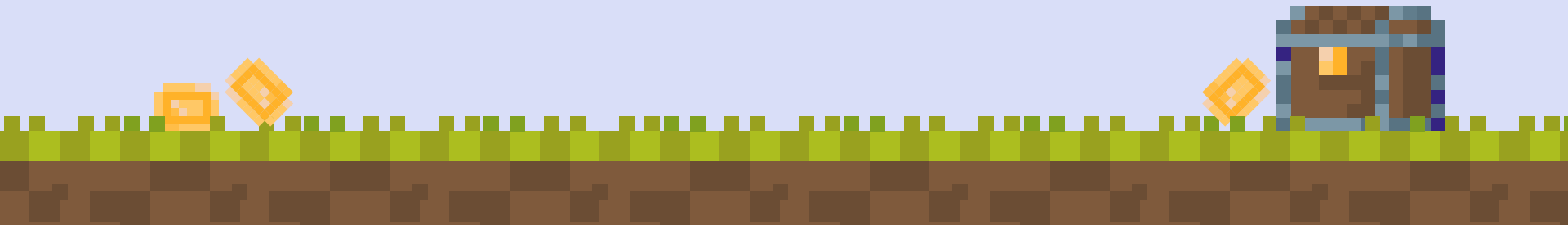




## 結論



- Class的使用，能快速的建立一個內容豐富的程式
- 遊戲的運作流程
- 事前若能製作diagram，有助於編寫程式時宏觀地了解目前掌握的部分





不要忘記，寫組別互評表。  
因為非~常好程式，簡報非~常好  
差不多一樣，🍦！ 再見~

祝各位期末歐趴 暑假愉快👍👍

Thanks for  
listening!

