



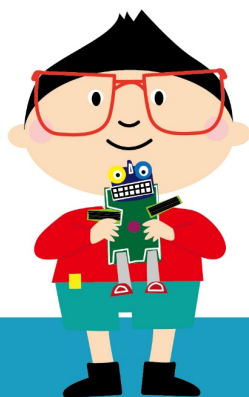
# SINGULAR 奇點創意

## 程式創客教室

機器人 / AI人工智慧 / 程式語言

*Join Singular!  
Be a super inventor!*

# AnimeGames15



Maker + Coder =

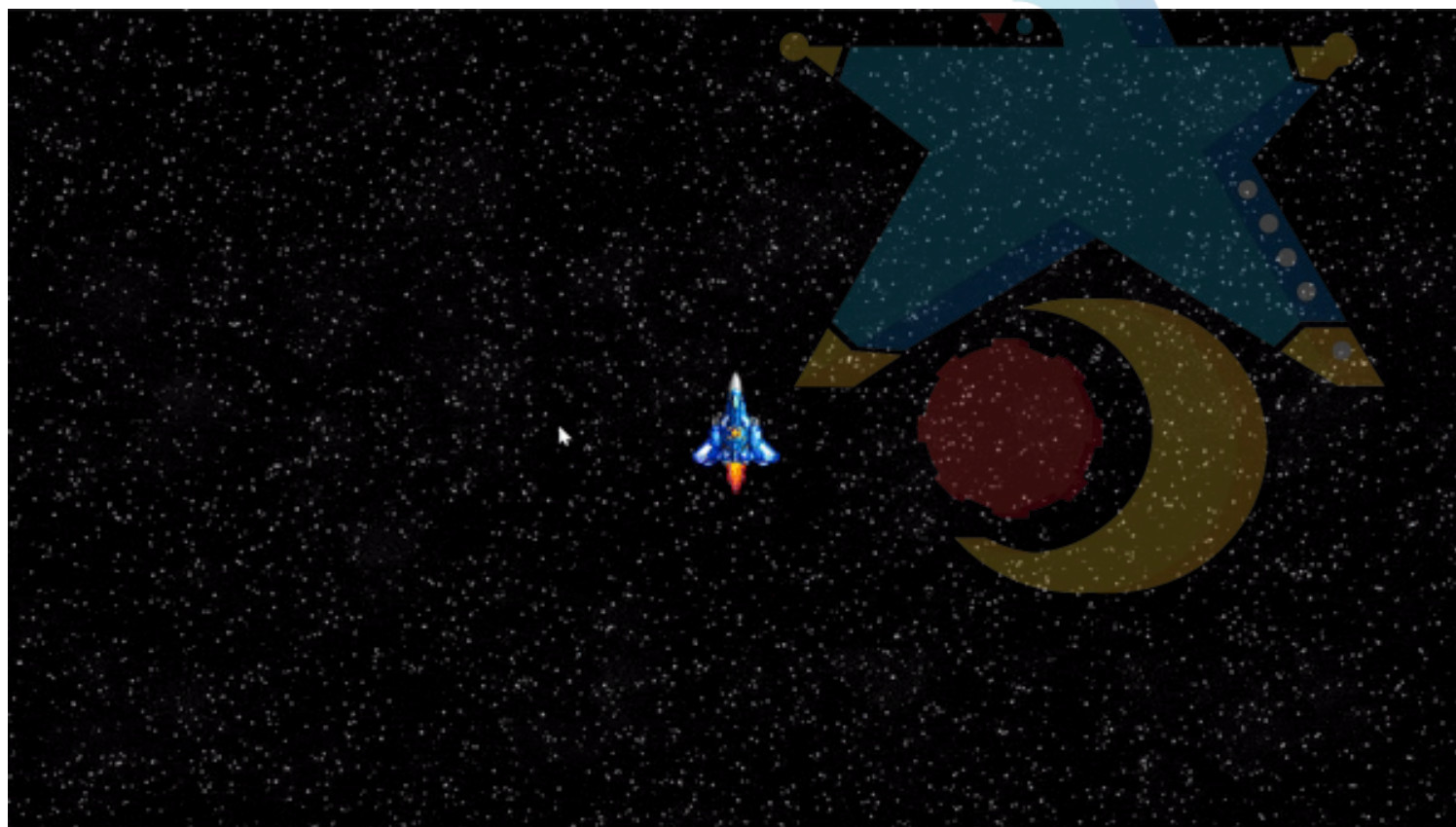


Singular Super Inventor



# 任務

- 單發子彈



Maker + Coder = Singular Super Inventor



SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 新增子彈物件

#####物件類別#####

```
class Missile:
    def __init__(self, x, y, image, shift):
        """初始化飛彈"""
        self.x = x
        self.y = y
        self.image = image
        self.active = False
        self.shift = shift
```

# 新增子彈物件指令

#####物件類別#####

```
class Missile:
```

```
    def __init__(self, x, y, image, shift):
```

```
        """初始化飛彈"""
```

```
        ...省略...
```

```
    def launch(self, x, y):
```

```
        """發射飛彈"""
```

```
        if not self.active:
```

```
            self.x = x
```

```
            self.y = y
```

```
            self.active = True
```



# 新增子彈物件指令

#####物件類別#####

```
class Missile:
```

```
    ..省略..
```

```
    def launch(self, x, y):
```

```
        """發射飛彈"""
```

```
        ..省略..
```

```
    def move(self):
```

```
        """移動飛彈"""
```

```
        if self.active:
```

```
            self.y -= self.shift
```

```
            if self.y < 0:
```

```
                self.active = False
```



# 新增子彈物件指令

```
#####物件類別#####
```

```
class Missile:
```

```
    ...省略...
```

```
def move(self):
```

```
    """移動飛彈"""
```

```
    ...省略...
```

```
def draw(self, screen):
```

```
    """繪製飛彈"""
```

```
    if self.active:
```

```
        screen.blit(self.image, (self.x, self.y))
```





# 載入飛彈圖片

#####載入圖片#####

...省略...

# 載入飛船火焰

```
img_burn =  
pygame.image.load("image/starship_burner.png")
```

# 載入飛彈圖片

```
img_weapon = pygame.image.load("image/bullet.png")
```

# 召喚子彈物件

```
#####飛彈設定#####  
msl_wh = img_weapon.get_width() / 2 # 飛彈寬度一半  
msl_hh = img_weapon.get_height() / 2 # 飛彈高度一半  
msl_shift = 30 # 飛彈速度  
missile = Missile(0, 0, img_weapon, 30) # 建立飛彈物件
```





# 按鍵偵測

```
#####主程式#####
```

```
while True:
```

```
    clock.tick(30)
```

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

```
        ...省略...
```

```
        elif event.key == K_ESCAPE:
```

```
            screen = pygame.display.set_mode(bg_size)
```

```
        if event.key == K_SPACE:
```

```
            missile.launch(ss_x - msl_wh, ss_y - msl_hh)
```

```
roll_bg() # 捲動背景
```

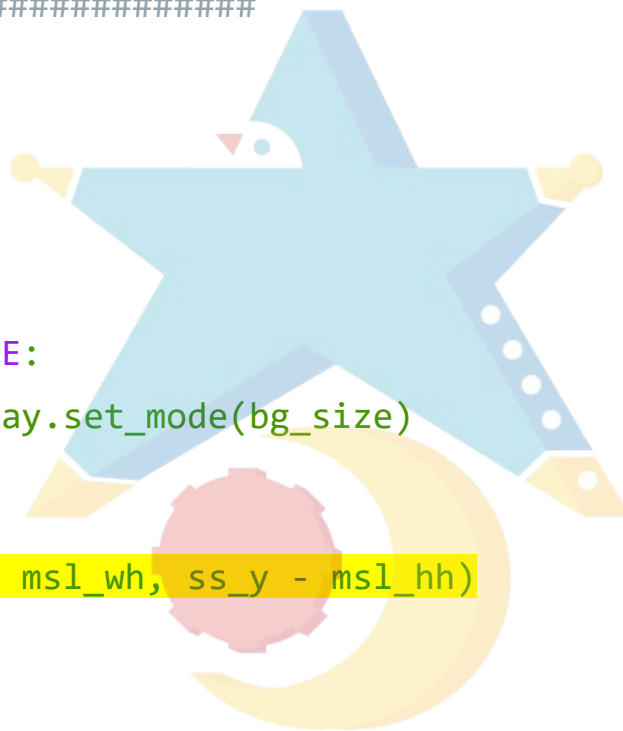
```
move_starship() # 飛船移動
```

```
missile.move()
```

```
missile.draw(screen)
```

```
pygame.display.update()
```

Maker + Coder = Singular Super Inventor



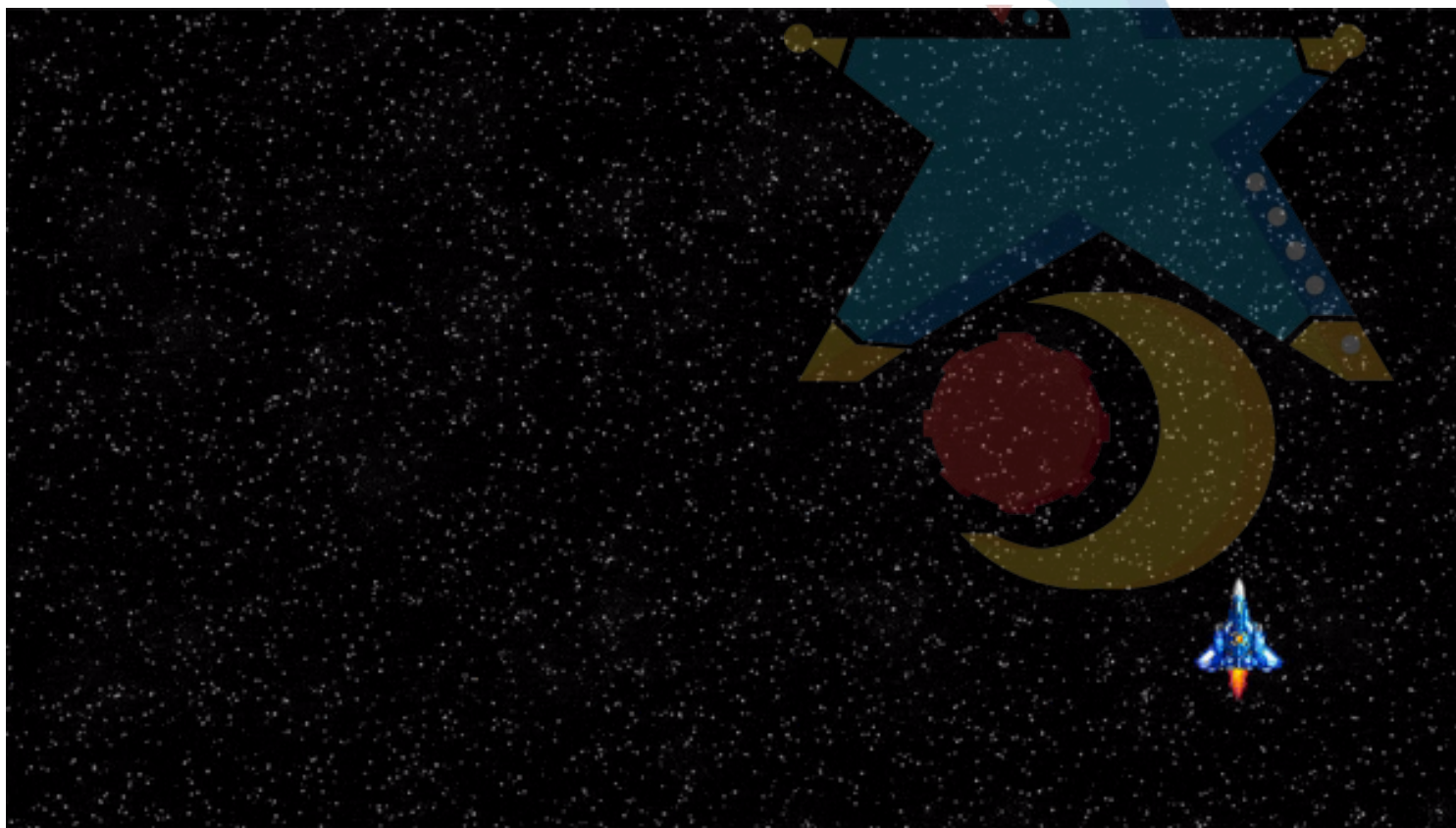
SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 任務

- 多發子彈



Maker + Coder = Singular Super Inventor



SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 召喚多發子彈物件

#####飛彈設定#####

mssl\_wh = img\_weapon.get\_width() / 2 # 飛彈寬度一半

mssl\_hh = img\_weapon.get\_height() / 2 # 飛彈高度一半

mssl\_shift = 30 # 飛彈速度

MISSILE\_MAX = 10 # 最大飛彈數量

# 飛彈列表，不可以用[Missile(0, 0, img\_weapon, mssl\_shift)] \* MISSILE\_MAX

missiles = [Missile(0, 0, img\_weapon, mssl\_shift) for \_ in range(MISSILE\_MAX)]

mssl\_cooldown = 0 # 飛彈冷卻時間

mssl\_cooldown\_max = 10 # 飛彈最大冷卻時間（連發間隔）



# 更新事件偵測

```
#####主程式#####
```

```
while True:
```

```
    clock.tick(30)
```

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

```
        ...省略...
```

```
        elif event.key == K_ESCAPE:
```

```
            screen = pygame.display.set_mode(bg_size)
```

```
            if event.key == K_SPACE and msl_cooldown == 0: # 檢查冷卻時間
```

```
                for missile in missiles:
```

```
                    if not missile.active: # 尋找一個未激活的飛彈
```

```
                        missile.launch(ss_x - msl_wh, ss_y - msl_hh)
```

```
                        msl_cooldown = msl_cooldown_max # 重設冷卻時間
```

```
                    break
```

```
roll_bg() # 捲動背景
```

```
...略省...
```

Maker + Coder = Singular Super Inventor



SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 更新事件偵測

```
#####主程式#####
```

```
while True:
```

```
    clock.tick(30)
```

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

```
        ...省略...
```

```
    roll_bg() # 捲動背景
```

```
    move_starship() # 飛船移動
```

```
    msl_cooldown = max(0, msl_cooldown - 1) # 更新飛彈冷卻時間
```

```
    for missile in missiles: # 移動和繪製所有飛彈
```

```
        missile.move()
```

```
        missile.draw(screen)
```

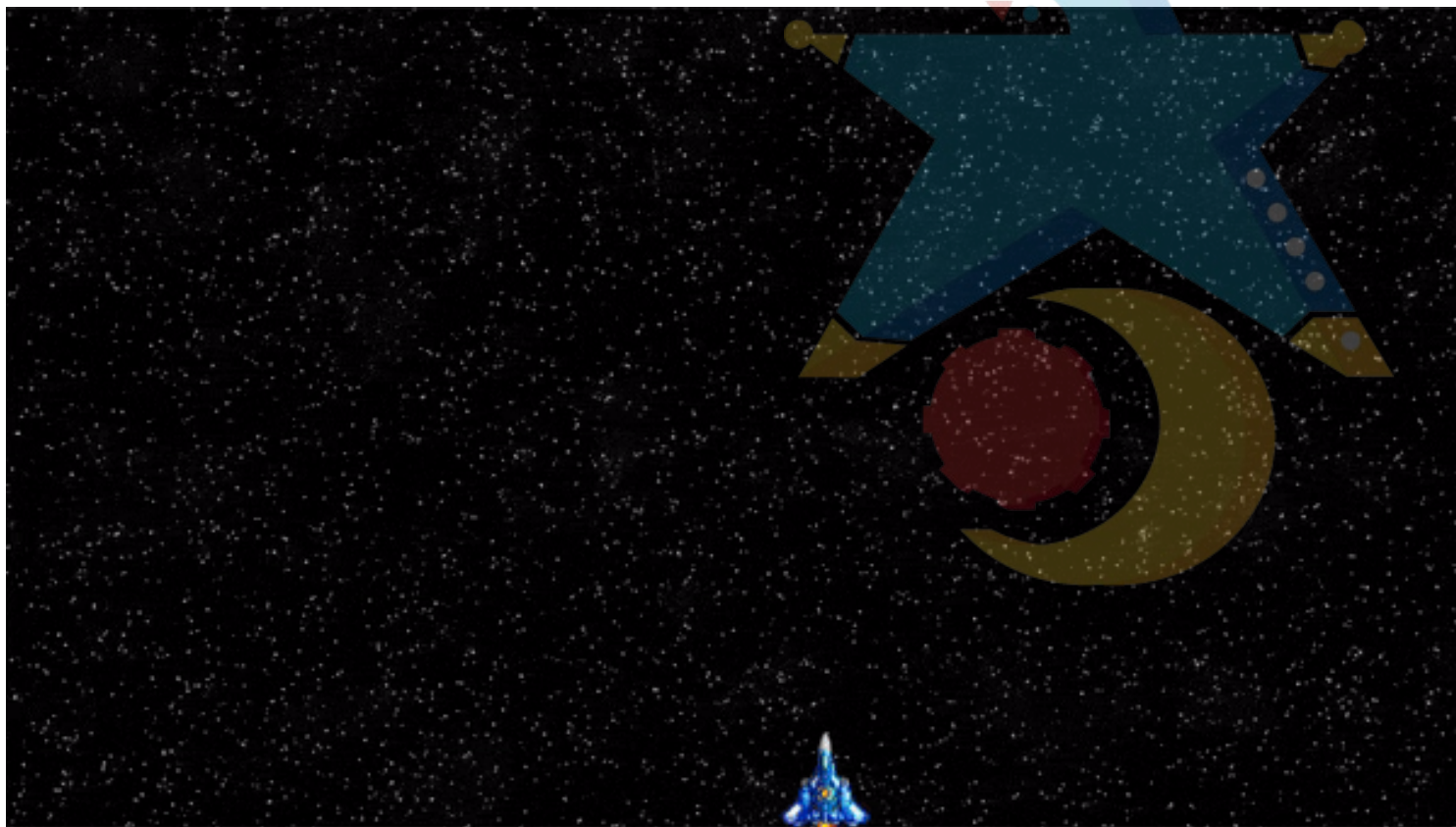
```
    pygame.display.update()
```





# 任務

- 新增敵人物件



Maker + Coder = Singular Super Inventor

# 新增敵人物件

#####物件類別#####

...省略...

```
class Enemy:
```

```
    def __init__(self, x, y, image, shift):
```

```
        """初始化敵機"""
```

```
        self.x = x
```

```
        self.y = y
```

```
        self.image = image
```

```
        self.active = True
```

```
        self.shift = shift
```

```
        self.wh = image.get_width() // 2
```

```
        self.hh = image.get_height() // 2
```





# 新增敵人物件指令

#####物件類別#####

...省略...

class Enemy:

def \_\_init\_\_(self, x, y, image, shift):

"""初始化敵機"""

...省略...

def move(self):

"""移動敵機"""

if self.active:

self.y += self.shift

if self.y > bg\_y: # 當敵機移動到屏幕底部時，標記為非活躍

self.active = False



# 新增敵人物件指令

```
#####物件類別#####
```

```
...省略...
```

```
class Enemy:
```

```
    ...省略...
```

```
    def move(self):
```

```
        """移動敵機"""
```

```
        ...省略...
```

```
    def draw(self, screen):
```

```
        """繪製敵機"""
```

```
        if self.active:
```

```
            screen.blit(self.image, (self.x - self.wh, self.y - self.hh))
```



# 新增敵人物件指令

#####物件類別#####

...省略...

class Enemy:

...省略...

def draw(self, screen):

"""繪製敵機"""

...省略...

def reset(self, x, y, image, shift):

"""初始化敵機"""

self.x = x

self.y = y

self.image = image

self.active = True

self.shift = shift

self.wh = image.get\_width() // 2

self.hh = image.get\_height() // 2



# 載入敵人圖片

#####載入圖片#####

...省略...

```
img_weapon = pygame.image.load("image/bullet.png")
```

```
# 載入敵機圖片
```

```
img_enemy = pygame.image.load("image/enemy1.png")
```

# 設定敵人基本參數指令

#####定義函式區#####

..省略..

```
def create_enemy():
```

```
    """
```

```
    建立敵機
```

```
    return: 敵機x位置, 敵機y位置, 敵機圖片
```

```
    """
```

```
    emy_img = img_enemy
```

```
    emy_wh = emy_img.get_width() // 2 # 敵機寬度一半
```

```
    emy_x = random.randint(emy_wh, bg_x - emy_wh) # 起始x位置
```

```
    emy_y = random.randint(-bg_y, -emy_wh) # 起始y位置
```

```
    return emy_x, emy_y, emy_img
```



# 解包序列

```
numbers = [1, 2, 3]
print(numbers) # 輸出: [1, 2, 3]
print(*numbers) # 輸出: 1 2 3
```

```
def add(a, b, c):
    return a + b + c
```

```
numbers = [1, 2, 3]
result = add(*numbers)
print(result) # 輸出: 6
```

```
def get_coordinates():
    return 10, 20 # 返回一個元組 (10, 20)
```

```
print(get_coordinates()) # 輸出: (10, 20)
```

# 一般情況下，我們會使用兩個變數來接收函式返回

```
x, y = get_coordinates()
print(x) # 輸出: 10
print(y) # 輸出: 20
```

```
print(*get_coordinates())
```

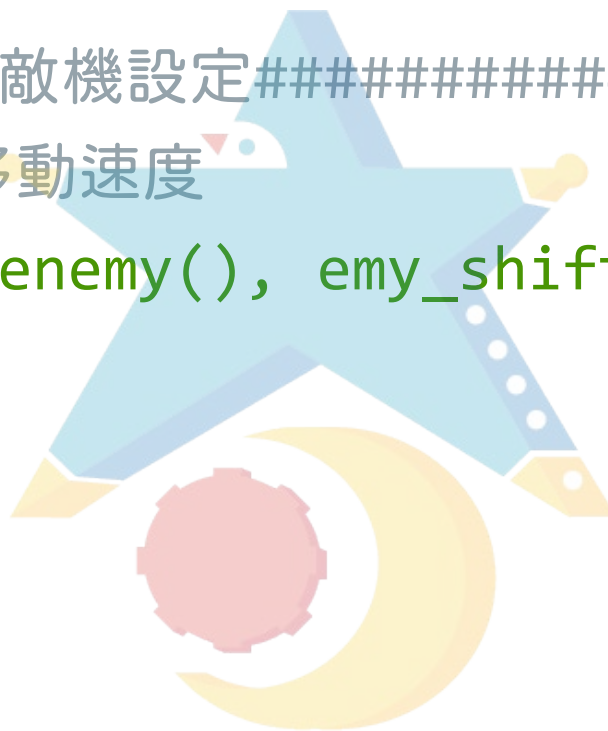


# 召喚敵人物件

#####敵機設定#####

emy\_shift = 5 # 敵機移動速度

enemy = Enemy(\*create\_enemy(), emy\_shift) # 建立敵機物件





# 更新主程式

```
#####主程式#####
```

```
while True:
```

```
    clock.tick(30)
```

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

```
        ...省略...
```

```
    ...省略...
```

```
    msl_cooldown = max(0, msl_cooldown - 1) # 更新飛彈冷卻時間
```

```
    for missile in missiles: # 移動和繪製所有飛彈
```

```
        missile.move()
```

```
        missile.draw(screen)
```

```
    enemy.move() # 移動敵機
```

```
    enemy.draw(screen) # 繪製敵機
```

```
    if not enemy.active: # 檢查敵機是否活躍
```

```
        enemy.reset(*create_enemy(), emy_shift)
```

```
    pygame.display.update()
```

Maker + Coder = Singular Super Inventor



SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 試看看

- 如果要新增敵機2呢?



Maker + Coder = Singular Super Inventor



SINGULAR  
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

# 載入圖片

#####載入圖片#####

...省略...

# 載入敵機圖片

`img_enemy = pygame.image.load("image/enemy1.png")`

`img_enemy2 = pygame.image.load("image/enemy2.png")`