SiNGULAR

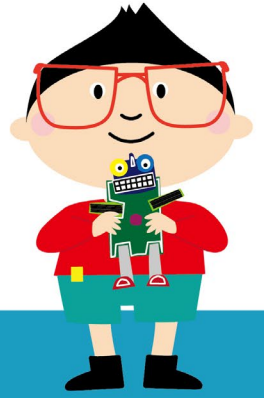
奇點創意

程式創客教室

機器人 / AI人工智慧 / 程式語言

Join Singular!
Be a super inventor!

AnimeGames14

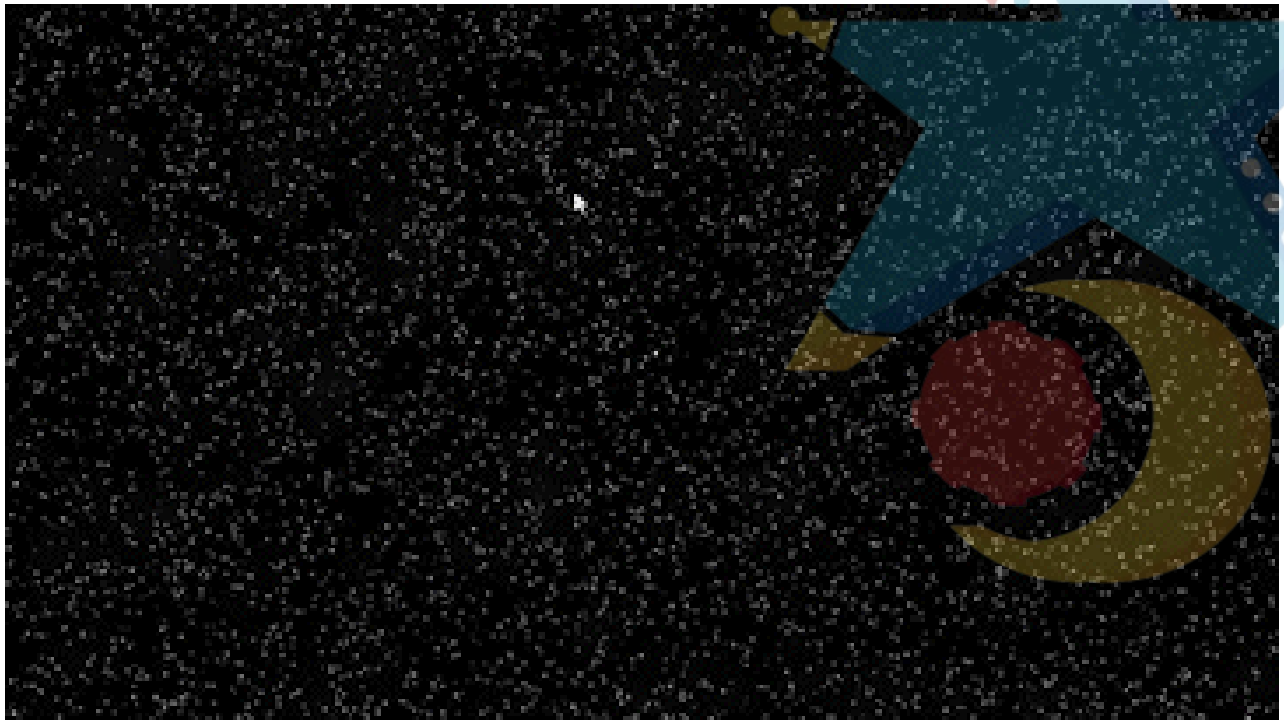


Maker + Coder = Singular Super Inventor

飛機大戰

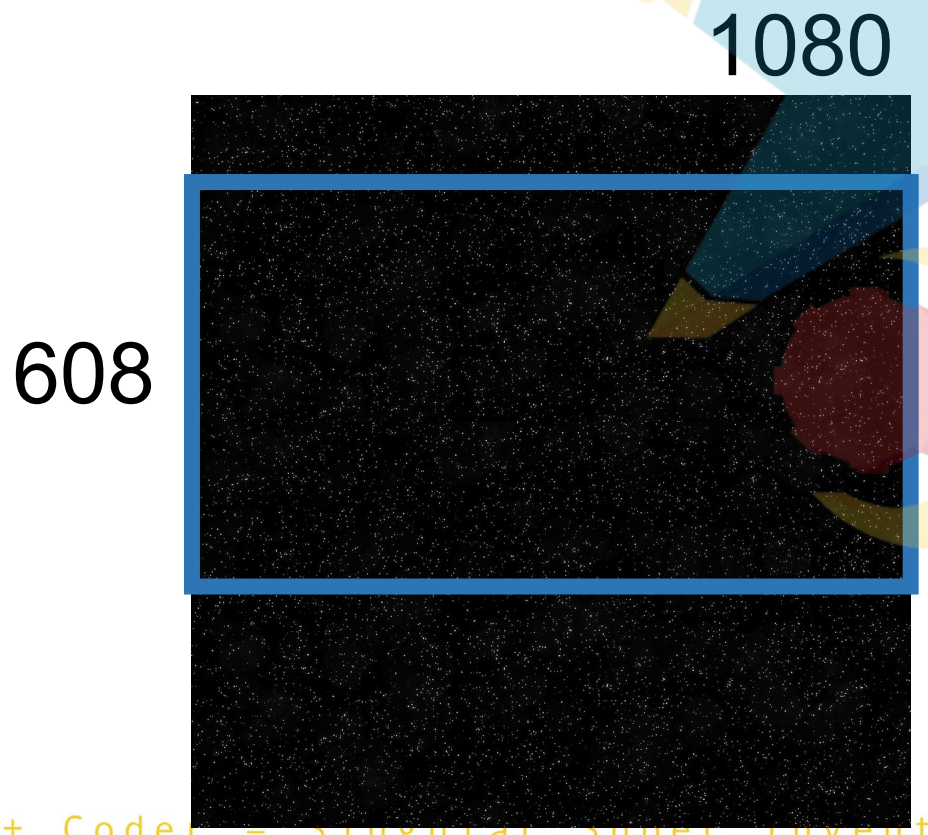
任務1

- 捲動背景



捲動背景

- 利用兩張一樣的背景，利用背景在視窗中位移，制造出捲動的效果。



製作流程

- 新增視窗
- 載入背景圖片
- 新增`roll_bg`背景更新指令
- 指令內會讓背景進行上下滾動並顯示在視窗上面
- 新增主程式

匯入模組

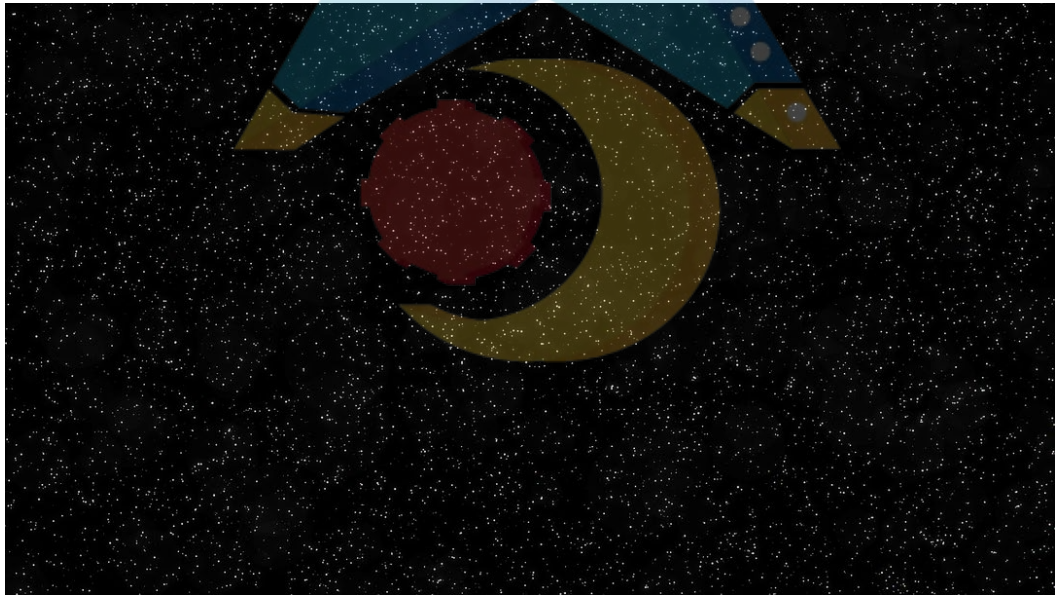
```
#############################載入套件#############################
import pygame
import sys
import os
from pygame.locals import *
```

初始化

```
############################初始化設定#############################
os.chdir(sys.path[0])
pygame.init()
clock = pygame.time.Clock()
```

載入圖片

```
##############################載入圖片##############################
#  載入背景圖片
img_bg = pygame.image.load("image/space.png")
```



遊戲視窗設定

```
############################遊戲視窗設定#############################
bg_x = img_bg.get_width()  # 背景圖片寬度
bg_y = img_bg.get_height()  # 背景圖片高度
bg_size = (bg_x, bg_y)  # 背景圖片大小
pygame.display.set_caption("Galaxy Lancer")
screen = pygame.display.set_mode(bg_size)
roll_y = 0
```

捲動背景

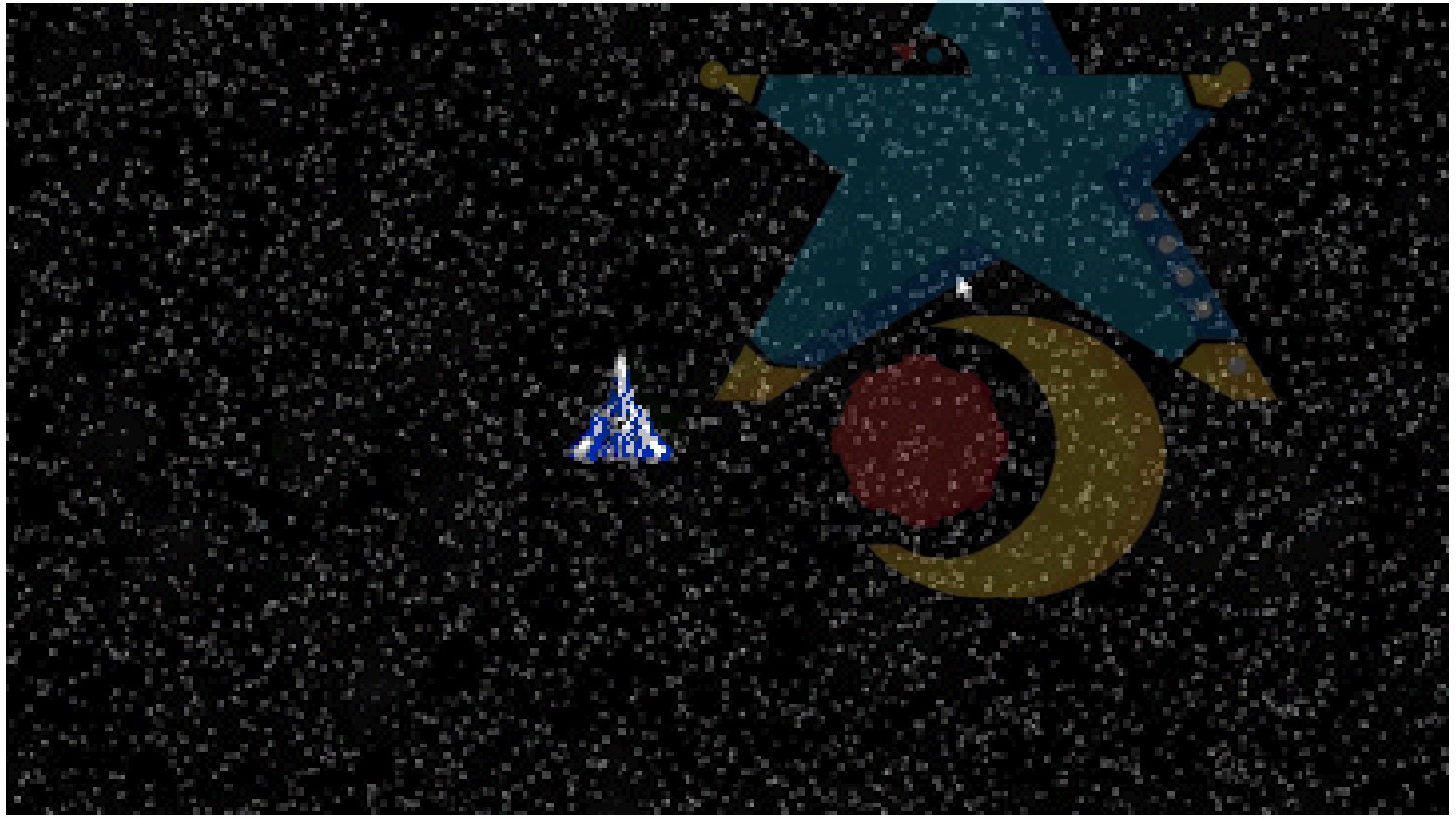
```
############################定義函式區############################
def roll_bg():
    """捲動背景"""
    global roll_y
    roll_y = (roll_y + 20) % bg_y   # 背景捲動
    screen.blit(img_bg, [0, roll_y - bg_y])   # 上半部
    screen.blit(img_bg, [0, roll_y])   # 下半部
```

主程式

```
######################主程式#######################
while True:
    clock.tick(30)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        if event.type == KEYDOWN:
            if event.key == K_F1:
                screen = pygame.display.set_mode(bg_size, FULLSCREEN)
            elif event.key == K_ESCAPE:
                screen = pygame.display.set_mode(bg_size)

    roll_bg()  # 捲動背景

    pygame.display.update()
```

任務

- 移動飛船



製作流程

- 新增飛船圖片
- 新增飛船指令move_starship
- 在指令當中偵測使用者按下的按鍵來做出位移

載入圖片

```
##############################載入圖片##############################
# 載入背景圖片
img_bg = pygame.image.load("image/space.png")

# 載入飛船圖片
img_sship = pygame.image.load("image/fighter_M.png")
```

飛船設定

```
##############################玩家設定##############################
ss_x = bg_x / 2   # 飛船x位置
ss_y = bg_y / 2   # 飛船y位置
ss_wh = img_sship.get_width() / 2   # 飛船寬度一半
ss_hh = img_sship.get_height() / 2   # 飛船高度一半
```

飛船指令

```
########################定義函式區#########################
…省略…
def move_starship():
    """移動飛船"""
    global ss_x, ss_y, ss_wh, ss_hh

    key = pygame.key.get_pressed()

    if key[pygame.K_UP]:
        ss_y -= 20
    if key[pygame.K_DOWN]:
        ss_y += 20
    if key[pygame.K_LEFT]:
        ss_x -= 20
    if key[pygame.K_RIGHT]:
        ss_x += 20
```

移動邊界

飛船指令

```
######################定義函式區######################
…省略…
def move_starship():
    """移動飛船"""
    global ss_x, ss_y, ss_wh, ss_hh
    …省略…
    if key[pygame.K_RIGHT]:
        ss_x += 20
    if ss_y < ss_hh:  # 飛船上邊界
        ss_y = ss_hh
    if ss_y > bg_y - ss_hh:  # 飛船下邊界
        ss_y = bg_y - ss_hh
    if ss_x < ss_wh:  # 飛船左邊界
        ss_x = ss_wh
    if ss_x > bg_x - ss_wh:  # 飛船右邊界
        ss_x = bg_x - ss_wh

    screen.blit(img_sship, [ss_x - ss_wh, ss_y - ss_hh])  # 飛船本體
```

(ss_x, ss_y)

(ss_x-ss_wh, ss_y-ss_hh)

(ss_x, ss_y)

Maker + Coder = Singular Super Inventor

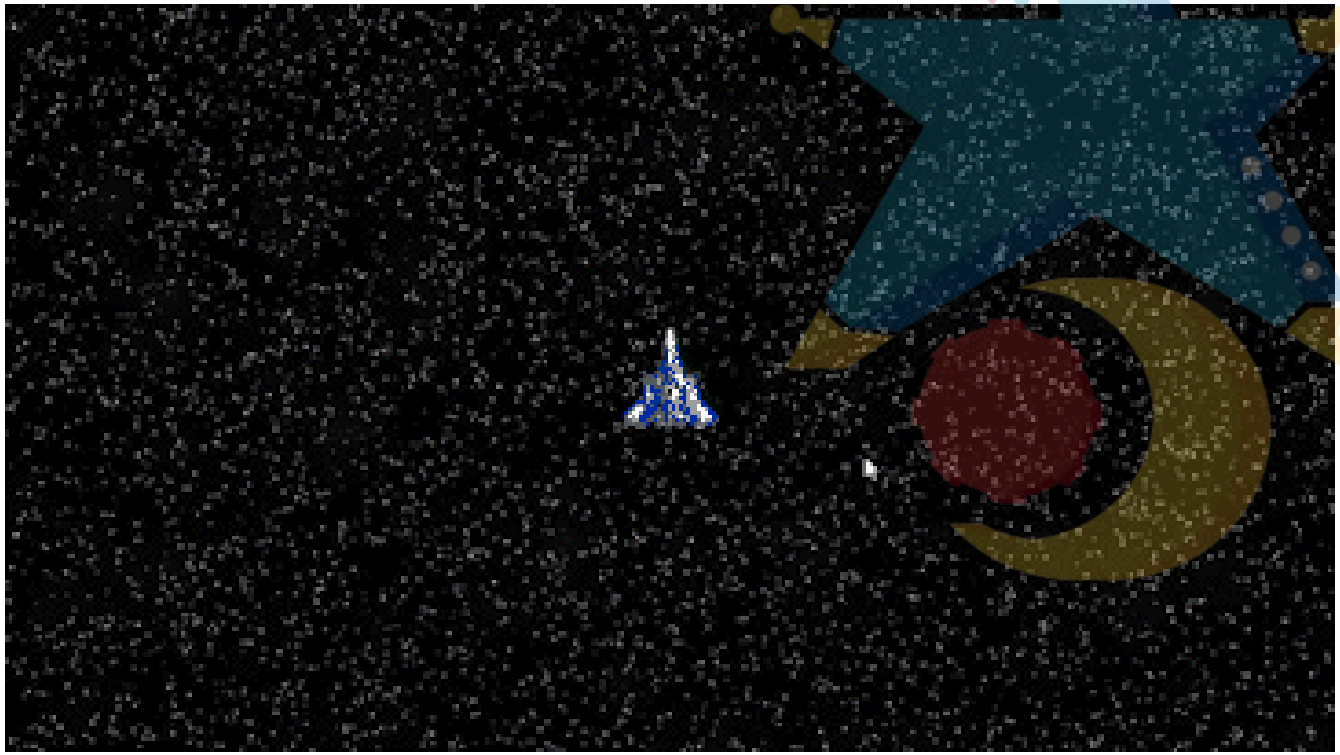
SINGULAR 奇點創意
程式創客教室
機器人 / AI人工智慧 / 程式語言

主程式更新

```
############################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

    pygame.display.update()
```

任務

- 飛船左右翻轉



製作流程

- 新增2張左與右的圖片
- 建立list將三張飛船圖片存進去，方便使用編號控制顯示
- 新增ss_img變數來設定目前飛船顯示的圖片
- 更新move_starship指令，在按下左移與右移的時候更新ss_img變數

更新圖片

```
############################載入圖片############################
# 載入背景圖片
img_bg = pygame.image.load("image/space.png")

# 載入飛船圖片
img_sship = [
    pygame.image.load("image/fighter_M.png"),
    pygame.image.load("image/fighter_L.png"),
    pygame.image.load("image/fighter_R.png"),
]
```

更新飛船基本參數

```
##############################玩家設定##############################
ss_x = bg_x / 2   # 飛船x位置
ss_y = bg_y / 2   # 飛船y位置
ss_wh = img_sship[0].get_width() / 2   # 飛船寬度一半
ss_hh = img_sship[0].get_height() / 2   # 飛船高度一半
ss_img = img_sship[0]   # 飛船圖片
```

更新飛船指令

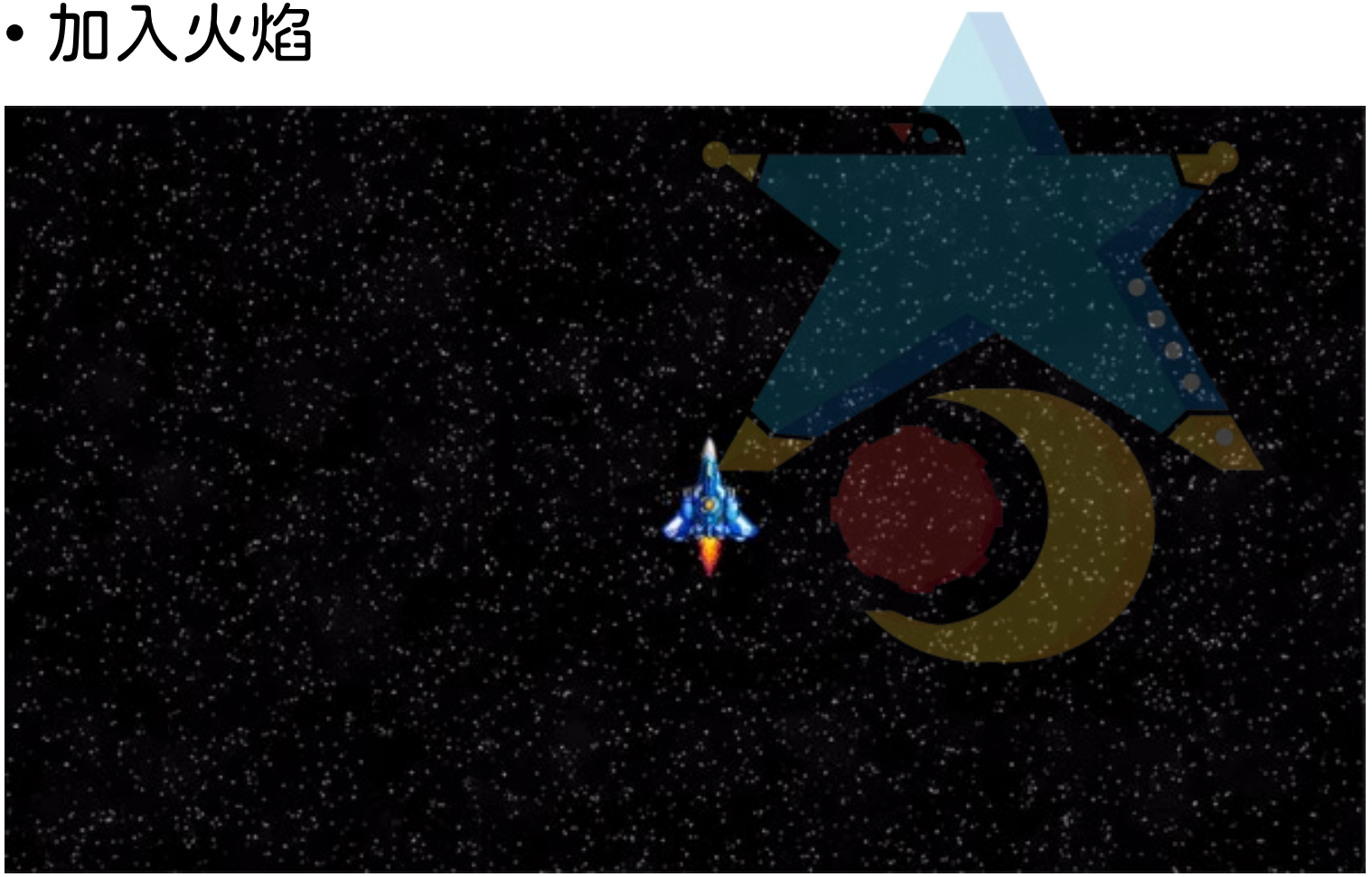
```
######################定義函式區######################
…省略…
def move_starship():
    """移動飛船"""
    global ss_x, ss_y, ss_wh, ss_hh, ss_img

    key = pygame.key.get_pressed()
    ss_img = img_sship[0]
    if key[pygame.K_UP]:
        ss_y -= 20
    if key[pygame.K_DOWN]:
        ss_y += 20
    if key[pygame.K_LEFT]:
        ss_x -= 20
        ss_img = img_sship[1]
    if key[pygame.K_RIGHT]:
        ss_x += 20
        ss_img = img_sship[2]
```

更新飛船指令

```
########################定義函式區########################
…省略…
def move_starship():
    """移動飛船"""
    …省略…
    if key[pygame.K_RIGHT]:
        ss_x += 20
        ss_img = img_sship[2]

    ss_hh = ss_img.get_height() / 2
    ss_wh = ss_img.get_width() / 2

    if ss_y < ss_hh:  # 飛船上邊界
        ss_y = ss_hh
    …省略…
    if ss_x > bg_x - ss_wh:  # 飛船右邊界
        ss_x = bg_x - ss_wh

    screen.blit(ss_img, [ss_x - ss_wh, ss_y - ss_hh])  # 飛船本體
```

任務

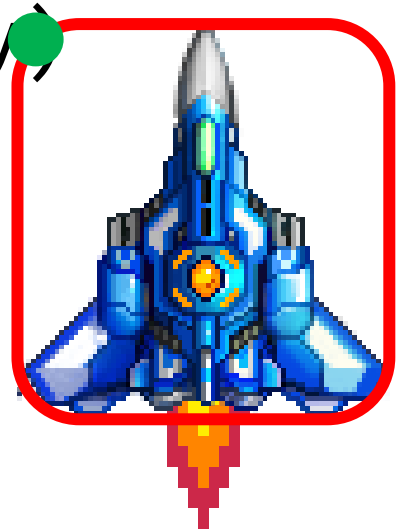
- 加入火焰

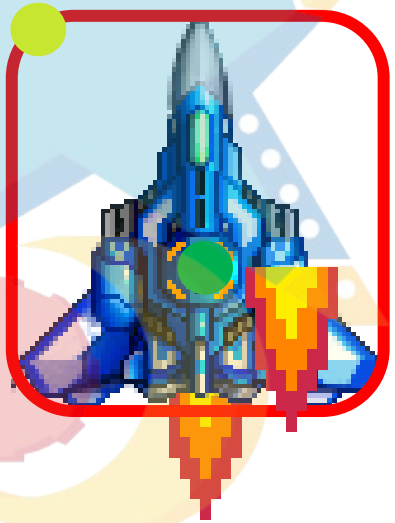


火焰坐標

(ss_x-ss_wh, ss_y-ss_hh)

(ss_x,
ss_y)





載入圖片

```
##############################載入圖片##############################
# 載入背景圖片
img_bg = pygame.image.load("image/space.png")

# 載入飛船圖片
img_sship = [
    …省略…]
# 載入飛船火焰
img_burn = pygame.image.load("image/starship_burner.png")
```

火焰初始參數

```
##############################玩家設定##############################
…省略…
ss_img = img_sship[0]   # 飛船圖片

burn_shift = 0   # 飛船火焰的位移
burn_w, burn_h = img_burn.get_rect().size   # 飛船火焰的寬度與高度
```

更新飛船指令

ss_y + burn_h + 0　　ss_y + burn_h + 2　　ss_y + burn_h + 4

```
##########################定義函式區##########################
…省略…

def move_starship():
    """移動飛船"""
    global ss_x, ss_y, ss_wh, ss_hh, ss_img, burn_shift

    …省略…

    if ss_x > bg_x - ss_wh:  # 飛船右邊界
        ss_x = bg_x - ss_wh

    burn_shift = (burn_shift + 2) % 6  # 飛船火焰的位移
    screen.blit(img_burn, [ss_x - burn_w / 2, ss_y + burn_h + burn_shift])  # 飛船火焰
    screen.blit(ss_img, [ss_x - ss_wh, ss_y - ss_hh])  # 飛船本體
```