



Python應用實作專題

歡迎來到我的音樂世界

資管系一 112403003 陳譽心





目錄

專題製作日程

第04頁

專題製作動機

第06頁

簡介我的第一個遊戲：猜歌達人

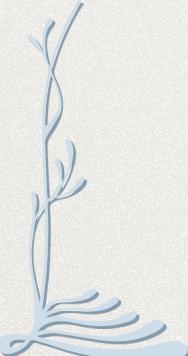
第07頁

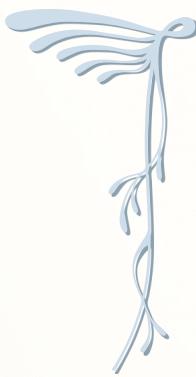
簡介猜歌達人（修改版）

第12頁

簡介我的第二個遊戲：音樂世界

第15頁





遊戲測試

第20頁

困難與解決

第23頁

GPT的使用

第25頁

收穫與成長

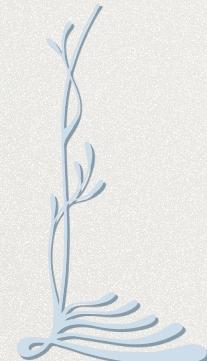
第26頁

心得與反思

第27頁

未來展望

第28頁



專題製作日程

原先與GPT討論後，我決定用Pygame製作遊戲專題：猜歌達人，但製作3週後遇到程式邏輯太過複雜、介面設計過於繁複等問題。因此，我設計了一個功能相同但將遊戲呈現簡化的版本，並開始進行第二個遊戲專題，也是我此次期末專題的重點：音樂世界，耗時約3週。於第16週，我製作專題報告，並完成此次的期末專題。



第8~10週

第11週

第12~15週

第16週



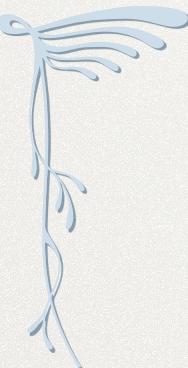
與GPT討論並製作第一個遊戲：猜歌達人

完成「猜歌達人」功能完整、但改變畫面呈現的版本

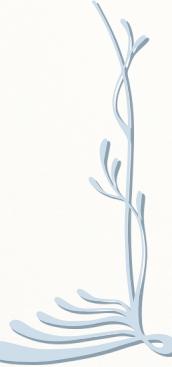
與GPT討論並完成製作第二個遊戲：音樂世界

完成專題報告





專題製作動機



對音樂的喜愛

從小學開始，我陸續接觸鋼琴、長笛等樂器，並對音樂產生濃厚的興趣。因此，不論是第一個遊戲專題：猜歌達人，或是「音樂世界」，我想設計跟音樂相關的遊戲



對設計遊戲的憧憬

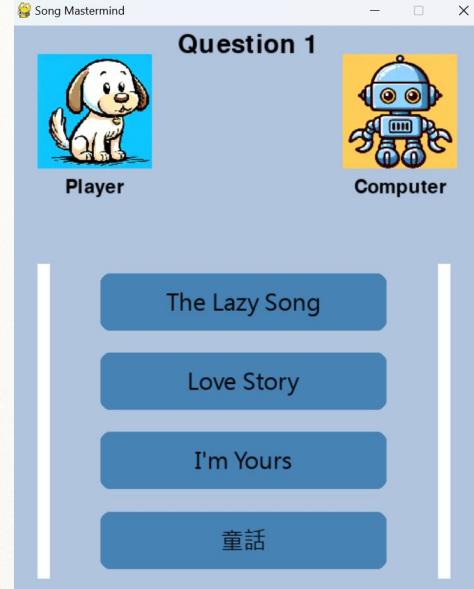
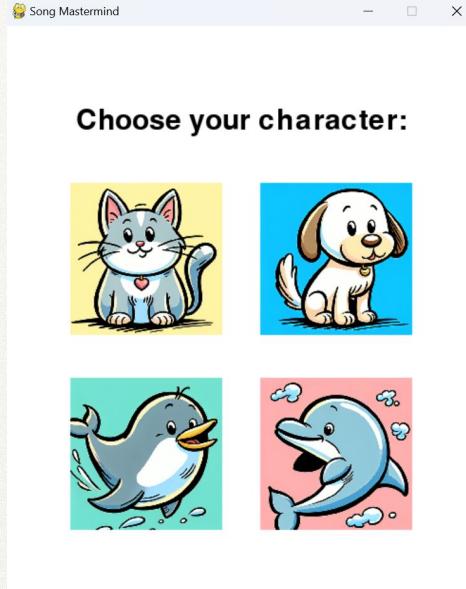
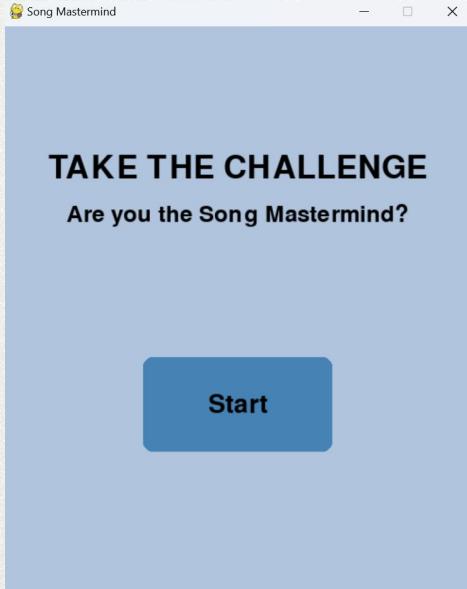
在高中初步接觸程式後，我便夢想能設計出一套遊戲程式，我認為能做出能跑、能動、能受大家喜愛的遊戲是一件很酷的事。因此，在發想主題時，我便立刻決定製作遊戲相關專題

簡介我的第一個遊戲： 猜歌達人

發想自知識王，玩家進入遊戲後可以選擇角色，後進行遊戲。

遊戲規則：與電腦對戰猜歌，作答速度與正確度皆會影響比分，最後會顯示勝負及分數，完成一次遊戲。



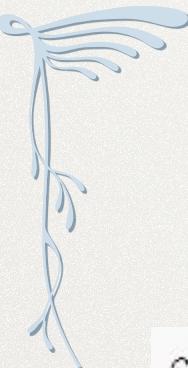


遊戲起始畫面

角色選擇畫面

遊戲畫面

*註:僅完成到遊戲畫面，玩家無法作答及遊玩



程式碼特點 - 1

```
current_dir_music = os.path.dirname(__file__)  
music_path = os.path.join(current_dir_music, "music")
```

在與程式同一路徑下建立一個資料夾存放要播放的歌曲，若之後要增加或減少歌曲可以直接在資料夾進行新增及刪減，十分方便。此外，還能針對不同群體的玩家提供不一樣的歌曲。





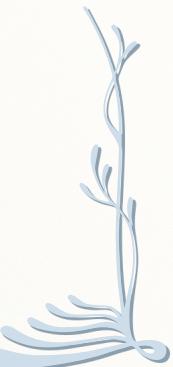
程式碼特點 - 2

```
def start_screen():
    screen.fill(LIGHT_STEEL_BLUE)

    font = pygame.font.Font(None, 40)
    lines = [
        {"text": "TAKE THE CHALLENGE", "font_size": 50, "position": (250, 150)},
        {"text": "Are you the Song Mastermind?", "font_size": 35, "position": (250, 200)}
    ]

    for line in lines:
        text_surface = pygame.font.Font(None, line["font_size"]).render(line["text"], True, BLACK)
        text_rect = text_surface.get_rect(center=line["position"])
        screen.blit(text_surface, text_rect)

    pygame.draw.rect(screen, STEEL_BLUE, button_rect, border_radius=RADIUS)
    text_surface = font.render(button_text, True, BLACK)
    text_rect = text_surface.get_rect(center=button_rect.center)
    screen.blit(text_surface, text_rect)
```

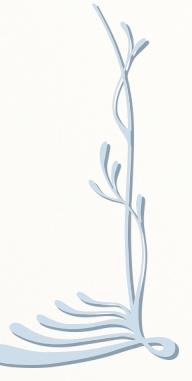


將每個畫面的指令寫在不同的**function**, 於遊戲迴圈可以直呼叫所需的畫面, 讓程式碼更好閱讀, 也方便反覆使用。



我遇到的困難

遊戲迴圈以我現在能力而言太過複雜，當玩家未作答時該如何處理？我該如何保證電腦答題的品質？若讓電腦隨機選擇，每次僅有 $1/4$ 的機率回答正確。此外，我要確保玩家越快回答時能獲得越高的分數，以及分數條相對應地增加。此遊戲有太多我可能未預期的例外需要處理，我認為在短短的9週是我無法達成的。因此，我決定製作一個簡化的遊戲版本，並開始進行第二個遊戲專題。



但我認為，這3週的專題製作並不是在浪費時間，這是我第一次接觸Pygame，在製作過程中我更加熟練相關設定及操作，也對Python更上手，幫助我在製作第二個專題時更順利。

簡介猜歌達人(修改版)

基本功能與猜歌達人相同，但沒有遊戲畫面呈現，並移除與電腦競賽功能。

遊戲開始會撥放歌曲並提供選項，玩家從選項中選擇對應的歌曲，最後會顯示回答是否正確。



遊戲畫面 (作答正確時)

選項： ['Bad Day', '烏梅子醬', '等你下課', "I'm Yours"]

請選擇正確的歌名：Bad Day

答對了！

遊戲畫面 (作答錯誤時)

選項： ['你的情歌', '烏梅子醬', '不遺憾', "I'm Yours"]

請選擇正確的歌名：烏梅子醬

答錯了！正確答案是： 你的情歌



程式碼特點

```
song_name = song_file.split('.')[0]
```

可以直接從資料夾中的歌曲檔案名稱取得歌名，並在挑選播放歌曲的同時隨機挑選另外三個不同的歌名作為選項，省去額外編寫生成選項程式碼的麻煩。



簡介我的第二個遊戲： 音樂世界

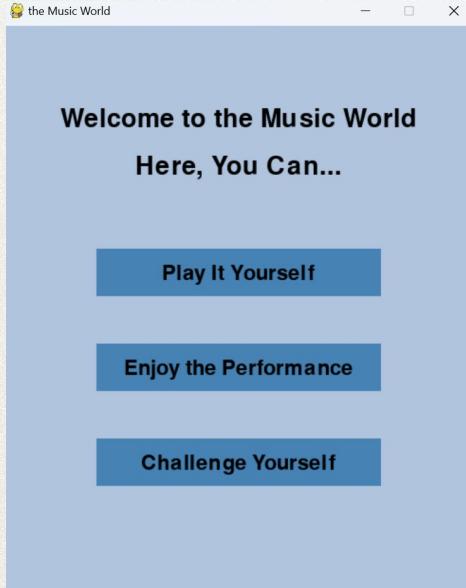
遊戲開始後，有三種遊玩音樂的模式可以選擇：自行創作、欣賞演奏以及聽音辨音符。

若選擇自行創作，玩家可以按鍵盤上的數字鍵來發出對應的音符聲音。

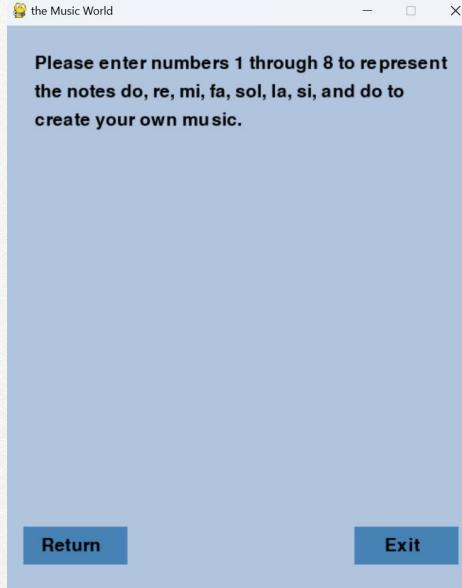
若選擇欣賞演奏，有三首歌曲可供玩家選擇及聆聽。

若選擇聽音辨音符，會隨機出**3**道題目，玩家要根據聽到的聲音回答對應的音符，遊戲結束會公布答對的題數。

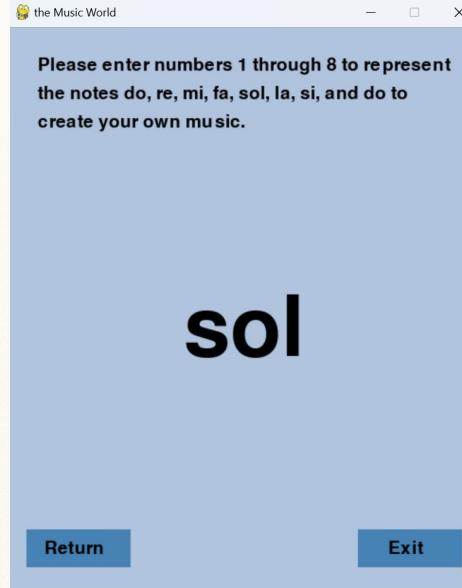




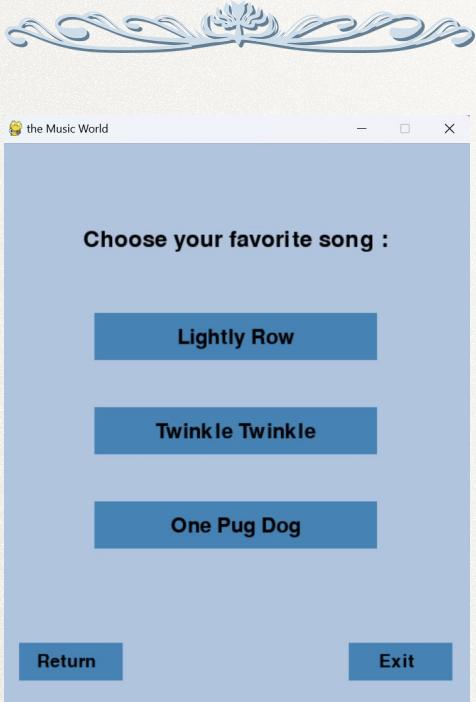
遊戲起始畫面



Play It Yourself 畫面



按下數字鍵會顯示對應的
音符及發出對應的聲音



Enjoy the Performance 畫面

歌曲選擇的小巧思

- 由於我製作這套音樂程式非真正的鋼琴樂器，音域不夠廣，因此我選擇的三首歌曲都是兒歌，音域皆只有一個八度，廣為人知又好上口
- 我挑選的歌曲有大家熟知的小蜜蜂，但英譯為**Lightly Row**，希望給玩家帶來驚喜的感覺，原來一個不熟悉的歌名點進去竟是熟悉的曲子

*後來讓同學遊玩時他們真的滿驚喜的！



the Music World

Press the play button to challenge yourself!
You'll hear three sets of five notes each.
Enter the corresponding notes, and good luck!
(Enter numbers 1 through 8 to represent
the notes do, re, mi, fa, sol, la, si, and do)
After the question is played,
please wait a second before entering your answer.

Correct

Answer: 6 1 6 6 5

Return

Play

Exit



the Music World

Press the play button to challenge yourself!
You'll hear three sets of five notes each.
Enter the corresponding notes, and good luck!
(Enter numbers 1 through 8 to represent
the notes do, re, mi, fa, sol, la, si, and do)
After the question is played,
please wait a second before entering your answer.

Incorrect

Answer: 1 8 1 6 1

Return

Play

Exit



the Music World

Press the play button to challenge yourself!
You'll hear three sets of five notes each.
Enter the corresponding notes, and good luck!
(Enter numbers 1 through 8 to represent
the notes do, re, mi, fa, sol, la, si, and do)
After the question is played,
please wait a second before entering your answer.

You got 1 out of 3 correct!

Return

Play

Exit

Challenge Yourself 畫面
(作答正確)

作答錯誤

遊戲結束顯示答對題數

程式碼特點

```
frequency_map = {  
    '1': 261.63, # do  
    '2': 293.66, # re  
    '3': 329.63, # mi  
    '4': 349.23, # fa  
    '5': 392.00, # so  
    '6': 440.00, # la  
    '7': 493.88, # si  
    '8': 523.25 # do  
}  
  
# 創建音效  
sounds = {}  
for key, freq in frequency_map.items():  
    sample_rate = 44100  
    duration = 1.0  
    t = np.linspace(0, duration, int(sample_rate * duration), False)  
    note = np.sin(freq * 2 * np.pi * t)  
    stereo_note = np.column_stack((note, note))  
    sound = pygame.sndarray.make_sound((stereo_note * 32767).astype(np.int16))  
    sounds[key] = sound
```

用內建的方式播放音符聲音。最初我詢問**GPT**如何撰寫程式碼以播放音符的聲音，它要我上網下載素材再一個個匯入，但這樣過於麻煩且日後若想新增音域也不方便。經過後來的討論，我發現可以用如此方法取得音符聲音，創建過程不但簡單也方便。

遊戲測試

完成遊戲後，為了避免有自己看不到的盲點，我邀請了幾位同學來遊玩我的遊戲並給我一些建議和回饋。此外，我也想了解這類遊戲主題是否受到大家的喜愛，以便更好地理解玩家的需求和喜好。



以下是我邀請四位同學遊玩我的遊戲的照／影片紀錄：

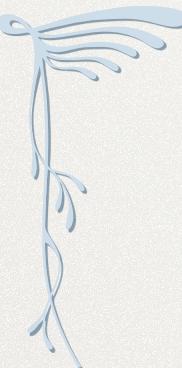
<https://drive.google.com/file/d/1bwsJjWZjTdE3Pxp0u6CWmUBDOLLIGj5l/view?usp=sharing>



https://drive.google.com/file/d/1JHHSt_zxi4DuiiO4dONxMWthkk0AQdMH/view?resourcekey



<https://drive.google.com/file/d/1pUcExLkf5AV6egWLGLa2xA0GWw-vrtD/view?usp=sharing>

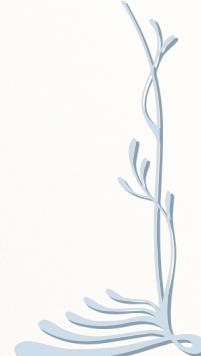


建議與回饋



- 可以做成手遊嗎超好玩
- 欸好厲害喔
- 版面整潔、清晰
- 遊戲很有趣
- 指令清楚好懂



- 如果在猜音符環節三次答題都正確要加鼓勵的話給玩家回饋
 - 猜音符環節題與題間作答時間太快反應不及
 - 撥放音符時每個音太長
 - 輸入音符時太快會出現雜音
 - 皆已修改完成
- 

困難與解決

在專題製作的過程中，我並非一路順利。雖然我是資管系的學生，但我從大一才開始真正專注於程式設計，因此對程式的撰寫還不算精熟。起初，我遇到了不少問題，甚至一度不得不更換主題。然而，最終我克服了這些挑戰，並完成專題。以下是在過程中遇到的困難與解決：



對Pygame
不熟悉

透過網路資源和GPT的幫助，我從零開始學習Pygame，為期9週的專題製作也讓我更進一步加深對Pygame的認識及掌握。

找不到程
式碼問題

在與GPT的問答中學習，讓我可以立即找到程式碼的錯誤並更正。除此之外，我學會精簡程式碼，提升程式的運行效率。

時間不足

擔心專題製作無法如期完成，我從第8週便開始進行製作。然而，由於能力因素需要更改主題，導致剩下約4週的時間完成專題，同時還需兼顧系上的課業。而我學會制定計畫並排定每日代辦事項，這讓我在做事時更有動力也更有效率。



GPT的使用

有了**GPT**的幫助，我在專題製作過程中克服了許多困難。以下是幾個主要的應用：

- 討論遊戲構思：從初步的「我該用什麼製作**Python**遊戲？」到後期「我的遊戲還能加入哪些功能？」等，**GPT**都給我很好的建議。原本「音樂世界」只有自行演奏和欣賞音樂兩個模式，**GPT**建議我可以加入聽音辨音符，而這也是最受到大家喜愛的模式。
 - 圖片生成：**GPT**能立即生成符合我需求的圖片，如在「猜歌達人」中玩家及電腦的頭像和簡報製作中符合主題的「藍色系音樂圖片」，讓整體更生動且統一。
 - 討論程式碼：不論是程式碼如何撰寫或是如何修改，它總能迅速提供我解決方法，讓我知道問題所在。
- 



收穫與成長

- 我覺得這次專題最大的收穫是學會使用 **Pygame**, 在製作此專題前我並不知道**Python**有這個模組。首先，我觀看**YouTube**的教學影片讓我對**Pygame**有初步的認識，再透過與**GPT**的對答熟悉它的運用。
- 製作過程中讓我更熟悉**Python**語法。先前對於**Python**的認識多聚焦在知識層面，透過此次實作，我更熟練**Python**如**function**等的應用。



心得與反思

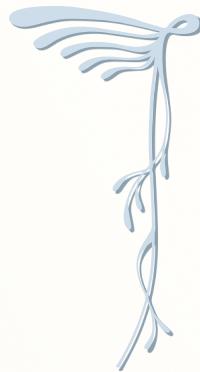
完成此專題後，我最大的感想就是，原來做遊戲並不是一件容易的事。原本以為能輔助使用**ChatGPT**，我只要有想法就好，程式的事情都交給它解決，但沒想到輔助終究是輔助，許多邏輯等問題還是要靠自己解決，比如程式碼應該放在哪個迴圈等。這次專題投入了許多時間，也讓我發現，雖然過程中遇到許多挫折，但做遊戲真的是一件有趣的事，尤其是看到別人玩自己所做的遊戲時開心的樣子。以前，當網頁上彈出擾人的遊戲廣告時，我總是急著把它們關掉。現在，我都會多看幾眼，因為我知道這背後凝聚了許多程式設計師辛勤的努力。



未來展望

雖然這次的專題完成了，但我希望自己還能就以下幾點持續進步：

- 改變畫面呈現：雖然同學一致回饋畫面呈現簡潔俐落，但我覺得太過單調也不特別吸引人，希望能將畫面修改得更立體及生動。
- 完成「猜歌達人」：未來有機會希望能將我的第一個遊戲完成，為此我需持續精進我的程式設計能力。





簡報結束 謝謝

資管系一 112403003 陳譽心

