

分組：第 11 組

B10703022 曾繁宸 B10703023 吳芃寬 B10703049 柯宥圻

B10703051 謝智祐 B10703113 曾順偉 B07605050 廖致豪

主題：

本次我們這組的程式專案是一個跟台灣大學相關的小遊戲，以管中閔校長為主角，讓他能夠在校園內四處穿梭，並且找到和學校相關的問題，在時間之內，校長正確回答越多的問題，可以獲得的分數越高。剛開始討論的時候，我們希望做出的是多主軸、可以有扮演各種角色的遊戲，藉由自己的選擇發展養成角色，並且間接性的決定故事的走向和結局，有點類似第 49 組的「為美好的台大生活獻上祝福」。但由於時間緣故，我們只先做好這一個部分，我們預計未來可以再增加更多有趣的功能，譬如說當回答一個問題，各種參數值(體力、智力、情緒…)都會有相對應的改變等，但是本次的報告，我們會聚焦在我們已經達成的功能上，做解釋、釐清。

使用模組：

1. Pygame

PyGame 是一種 Python 用來製作遊戲的模組，它能让開發者可以用更簡單的方式加入文字、圖案、聲音等元素，並且進行事件處理來開發遊戲。

Pygame 內擁有許多非常好用的核心元素，讓我們在開發的過程中，可以不用撰寫過多的程式，而是直接去網路上蒐集資料，並且加以運用。以下列舉一些在程式內有用到的函式。

(1). Pygame.event.get()

此函式可以偵測玩家按下了那些按鍵，其中包含鍵盤以及滑鼠的按鍵，我們可以任意地去規劃哪一些按鍵具有相對應的功能，像是最主要的移動，按 Q 離開正在回答的題目，以及按下介面右上角的離開按鈕時，程式對應到的終端畫面，是非常好用的功能。

(2). Pygame.sprite.Sprite

Sprite 的中文意思是精靈，顧名思義，他就像是一個精靈一樣，在他的協助下，我們的各式各樣的物件(畫面上可以看到的所有東西)，就可以依靠 Sprite 來顯示。以我們的遊戲當例子，我們可以看到地圖上有各式各樣的發光指標，引領玩家到該處回答問題，而當玩家回答完畢之後，發光的指標就會熄滅，而

且不能夠再度遊玩，第一次碰到她的時候會進入遊戲，但是一旦玩過之後，再經過他時都不會有相對應的事件產生了。而玩家操控的管中閔校長也是 Sprite 內的一環。

(3). Pygame.draw.rect()

這一個函式也是非常重要的功能，她掌管了我們各式各樣的按鈕、玩家以及座標的出現位置，如果只有把一個東西定位的話，pygame 並沒有辦法直接將其顯示出來，而使用 Pygame.draw.rect() 的話，就可以讓他在應該顯示出來的地方顯示出來。舉例來說，當我們接觸到發光的座標時，問題及可以按下的按鈕就會以 Pygame.draw.rect(self.__screen, (255,255,255), options) 的型態顯示出來，而當我們按下了回答，就會出現答案以及解釋等等，這些都可以靠這個函數給「畫」出來

2. OS

OS 最主要的功用就是在讀取照片及影片上，雖然我們在確認他所處的回全時，也是用 OS 的功能，但是最後我們使用到的，只有用到 path.join 而已，像是 os.path.join("image", str(Game.__position)+".jpg")，這裡面的 Game.__position，就是玩家所在的位置，舉例來說，如果我們身處的位置在椰林大道，那麼我們就會在 image 資料夾裡面找到命名為「椰林大道」的照片，並且設為背景。類似的還有音效功能。

3. Json

JSON 全名為 JavaScript Object Notation，即 JavaScript 對象標記法，雖然早在 2001 年就提出，但是在近年來，格式都沒有做太大的改變，也就是 2000 年代寫 json 格式，即使經過作業系統的更迭，仍然可以妥善的被利用。原先我們將資料透過多個 function 建立起所有問題的呼叫，但這樣會讓檔案變得複雜且難以理解，因此，我們重新設計問題存放的設計，原先想透過老師上課所教過的 Excel 儲存我們的問題，但在使用的時候我們發現並不是那麼的方便，不論是在資料的閱讀或是分割上，因此我們使用 json 檔結構化我們的資料，不僅容易閱讀與修改，在使用上我們也只需要透過 import json 的方式即可輕鬆地使用資料了。此外，我們透過給定參數的方式將我們需要的部分回傳出來，可避免重複呼叫造成的時間浪費，也可避免每次都將所有資料傳出

來，造成大量空間的耗損，因此，我們設計 Position 的方式將我們需要的段落取出，可以大量減少使用的空間並加速程式的運算。舉例來說：

```
[{
    "question": "「他」的專長在於財務報表，桃李滿天下，被稱作日月大賢者，請問他是誰？",
    "option1": "李明賢教授",
    "option2": "陳明賢教授",
    "option3": "羅大明教授",
    "answer": "option2",
    "explanation": "因為教授如日月一般引領著學生前進，因而被管院學生尊稱為日月大賢者。"
}]
```

這裡面有 question、option 等，在我們顯示視窗的時候，這些元素的位置都已經被固定好了，所以我們要做的，就只有把 json 檔案裏面相對應的字元給顯示出來就可以了，不需要另外的去定義這些文字應該要出現的地方。

我們曾經使用但是最後沒有使用成功的模組：

1. Tkinter

我們在製作遊戲的初期到中期，都是使用 TKInter 來顯示視窗和問題，當碰觸到發光的座標時，會先有一個視窗詢問說要不要進入遊戲，接者是問題視窗以及三個按鈕，再一一的去定義按下去其中一個按鈕之後，跳出來的提示框裡面要是甚麼。然而，在使用時，tkinter 有幾個缺點，讓我們最後放棄了這個好用的程式庫：第一、因為題目長度參差不齊，所以如果我們為了每一個問題去尋找最適合的長寬比的話，該區域的程式碼會變得非常龐大且混亂，但如果我們是用程式預設的長寬比的話，又會有無法顯示、吃字的問題。第二，我沒有辦法控制 tkinter 出現的位置，他會隨機的出現在電腦螢幕上的任意位置，這樣對於玩家的操作很不利，因為他要隨時注意哪裡有出現一個小視窗，按完確定之後還要再按一下主程式畫面才能夠繼續遊玩。第三個就是 tkinter 預設介面，偏醜。最後當我們遇到關鍵性的 threading 問題後，我們才終於決定放棄這個選項，而改用現行的方法：在主畫面上暫停原本的程式運作，並且以白色

的、佔據畫面 1/3 的視窗以及按鈕來取代，這樣子的設計方式的優點是他絕對不會亂飄，而是會在固定的位置出現，並且可以統一設計問題、回答。

2. Threading

當我們在設計 tkinter 的時候，我們遇到了嚴重的錯誤：每當我們玩第二題時，就一定會在該題的第二小題完全死當，他完全無法繼續執行下去，所以也偵測不到我們按下的鍵盤、按鈕，所以只能用工作管理員強制離開。經過測試之後，我們發現是 threading 的問題：每當我們進入了一個問題的時候，他還是維持在 mainloop 裡面，代表背景的主程式以及現在正在回答的問題佔用了同一個 Mainloop，導致裡面的程式互相干擾，最後導致程式無法繼續運行下去。原本我們打算的目標，是每當我們進入回答問題的區域之後，我們就跳入另一個 Loop，而原本的 Mainloop 就先暫時暫停。但是 Python 在處理 Threading 的方面是比其他程式語言弱一些的(網路上的熱心網友的建議我們不要用 threading)。在嘗試了幾天都沒有辦法分離不同的 loop 之後，我們決定放棄這個概念，並且重新製作整個遊戲的視窗，這樣子就算是一直待在 mainloop 裡面也沒有問題了。

程式邏輯：

因為 PBC final project 是一個多人共同撰寫程式的一個專案，因此需要透過 version control 與適當的分工才能有效的將程式撰寫化繁為簡，因此，我們採用 Git 作為 version control 的工具，並以 Github 作為儲存專案的遠端空間，並透過 VScode 的擴充元件 Git graph 作為簡化操作 Git 指令的圖形化介面，以避免需要額外學習 Git 指令的時間，而在課堂中其實並沒有太多關於 version control 的教學，因此為了確保大家都可以快速理解如何進行 version control，所以我們寫了一份 document 確保大家在忘記操作時，可以有查詢指導的地方。(Link: https://hackmd.io/@Codeamon/PBC_FinalProject)

原先我們將所有的程式撰寫在通一個檔案之中，但是因為需要等待他人完工之後才能繼續撰寫，因此，我們納入了 OOP 的概念，透過將程式工作區分成不同區塊的方式，不僅可以有效地降低每個程式檔案需要處理與了解的量，也可以讓大家可以分別撰寫 function，達到同時工作的設計，此外，納入 OOP 的設計不僅可以讓程式的撰寫變得更簡便，亦可以讓每個操作的呼叫變得容易理解，像是我們的 main function 可以從原先複雜的結構簡化成幾個操作的步

驟，而在 OOP 最複雜的架構中，我們讓每個檔案皆擁有自己的權限，因此，可以更清楚我們可以使用哪些東西與方法，並透過 method get 的方式，可以避免其他人更動到不同檔案的資料，形成明確的檔案位階的設計。

此外，在課堂中，當我們開始學習 Class 類別時，曾經有提到物件導向程式設計(Object-oriented programming, OOP)的概念，它是一個具有物件概念的開發方式，能夠提高軟體的可用性、擴充性及維護性，在開發像是我們這種專案的大型應用程式時，更是被廣泛的被使用。當時，雖然我們對類別及物件等有一些基本的了解，也有在 Pdogs 上面練習過，但是因為我們撰寫的程式規模過小，所以我們並沒有意識到 class 的實用性，而覺得這是一種多餘累贅的設定，但是老師有強調過 OOP 概念的重要性，所以就默默地記下來。在一開始的時候，因為對這部分還沒有太熟悉，所以就暫時不用 OOP 邏輯來建構架構，但是當我們改成以 OOP 方式為主軸來撰寫程式時，我不僅感受到 Class 的妙用萬千，還可以讓我們可以有效地知道我們現在所處的位置，以及是哪一個部分出現了問題等，並分配工作，著手修改。

因此，我們分成四個程式，有「CMKuan_inNTU」、「CMKuan_player」、「CMKuan_event」以及「CMKuan_game」，而負責運行的 CMKuan_inNTU 只有短短的 30 行，但是我們可以在這個程式上找到各種問題和錯誤，我們可以把結果印出，就可以知道是哪裡出了問題，我們稱其為 scaffolding(架起鷹架)。該程式裡面有一個判斷是否退出的 while 區域，如果我們按下了離開鍵，就會離開該程式，否則就進入到 gameloop 迴圈，裡面包含了程式內最重要的六大面向：display_background 設定背景；display_event 設定碰到物件時的反應，display_status 設定管爺的基礎狀態，display_player 設定管爺要往哪邊看以及他在哪一個區域等，check_event 是按下按鈕後的反應，而 get_timer 則是計時器的運轉，我們可以很清楚地了解到目前我們程式的停留位置在哪裡，也可以進行相對應的維護。

遇到的問題：

1.文字顯示

在 pygame 的遊戲設計上，本身並不支援非英語系的文字顯示，因此會造成中文字與非 big5 編碼的文字無法顯示，因此我們首先將 json 檔的資料以 utf-8 的形式讀進來，再透過直接將我們需要的字形放在跟目錄下，這樣不僅可

以不必受限於各個電腦使用的不同字型與是否缺少字型的問題，同時我們也需要透過呼叫跟目錄下的字形即可顯示我們要的文字與中文字。

```
def display_text_with_position(self, text, size, centerx, centery):
    text_font = pygame.font.Font('msjh.ttf', size)
    text = text_font.render(text, True, self.__text_color, self.__text_background_color)
    text_rect = text.get_rect()
    text_rect.centerx, text_rect.centery = centerx, centery
    self.__screen.blit(text, text_rect)
```

2. 權限呼叫及工作分配

由於遊戲中的事件必須透過檢查是否碰觸到事件來觸發，因此若是透過函數檢查容易造成因為數值更動導致檢查錯誤或是錯誤觸發的情況，且檢查事件的觸發與否屬於事件本身的工作並非他人的工作，因此我們透過給適當的標記，只讓 Event 自己調控監測的工作，不僅可以避免他人物誤觸事件，同時也可以透過回傳的結果進行適當的工作。

```
@classmethod
def reach_event(cls, position, player_cx, player_cy):
    event_exit = True
    event_number = 0
    counter = 0
    for item in Event.__posMatrix[position]:
        if (player_cx > (item[0]-60)) and (player_cx < (item[0]+60)) and (player_cy > (item[1]-60)) and (player_cy < (item[1]+60)):
            if Event.__reachMatrix[position][counter] == 1:
                Event.__reachMatrix[position][counter] = 0
                event_exit = False
                event_number = counter
            counter += 1
    return event_exit, event_number
```

分工方式：

吳芃寬、謝智祐、曾順偉、曾繁宸、柯宥圻：企劃發想以及劇情編寫。

謝智祐：影片剪輯

吳芃寬、柯宥圻：地圖製作

廖致豪：程式編寫、實作測試，撰寫最終程式中的程式碼、部分報告書寫，github 教學。

柯宥圻：初步設定及邏輯判斷、程式編寫、美編、文字編輯、A1 期末報告計畫書撰寫、B1 投影片製作、口頭報告、螢幕錄製、影片繳交、C1 大部份報告書寫、D1 家庭影片製作、拍攝、繳交，.exe 檔、下載檔製作。

心得：

B10703023 吳芃寬:這個學期對我或是對於任何選修這門課程的同學而言，都是很不容易的過程。在最後的期末專案當中，要特別感謝其他同學，不論是負責程式編寫部分的柯宥圻和廖致豪，或是負責影片製作的謝智祐，以及曾繁宸和曾順偉在遊戲內容設計中悉心提供的發想和劇情的編制。也許不是每

個人都那麼擅長設計程式、編寫程式，但是因為我們對 python 都有深入的認識，所以在溝通上能夠明確的表達出自己的想法和交流彼此的 idea，並且每個人都完成專案一小 part，進而完成複雜又豐富的 project。也許我們的成品不到完美，還是希望我們製作的小遊戲能夠帮助大家更認識台大的一些小知識。選修這堂課實在獲得很多啟發，已經迫不及待更深入了解其他 python 的工具和用途了，希望能夠將這門課應用在之後的各種情況或是開發有實用性的工具讓更多人使用。



↑ 吳芃寬做的初版地圖

B10703022 曾繁宸：這次期末報告真的要感謝柯宥圻和廖致豪兩位大腿將我們這組報告的程式編寫部分凱瑞起來，因為我們剩下的組員真的都不會寫程式嗚嗚嗚嗚，我覺得我的能力目前大概僅止於能夠將 PDOGS 上的題目完成以及期中考考試而已，所以我必須再三感謝他們為這組所做出的貢獻，沒有他們負責將程式寫出來，我們是製作不出來那麼有趣的遊戲的。雖然說我們其他組員在程式碼這塊的貢獻幾乎是 0，但還是有負責劇情編寫以及遊戲的發想，我們起初是要做類似養成遊戲，還要分成學生、校長、助教等等，但後來發現時間不太夠所以只好專心在管爺的部分，劇情以及玩法縮減了不少，但還是有實現我們起初要讓大家更認識台大的宗旨，希望我們這一組做出來的東西能為大家帶來一點樂趣和一些台大的小知識。

B10703049 柯宥圻：這次我獲得最多的，不只是程式設計的能力，更多的是工作分配及良好的溝通的重要性，一開始我做出程式的骨架之後，一方面我沒有很確定接下來的路線要往哪個方向前進，另一方面當時的我認為，我現在做的東西解釋起來有點麻煩，根據機會成本原則，我直接繼續做會比跟他們

講還要快，沒想到一不小心就把程式設計的部分都落在自己和廖致豪學長的肩上了。如果還有機會重來的話，我希望我可以找到更多時間可以一起討論製作的方向及更多的小細節，而不是埋頭苦幹，像一隻無頭蒼蠅般的亂竄，我相信這次的作業能夠變得更好，或許也更接近一開始的時候構想出的方案，其他人辛苦做出的地圖也不會白費掉。過程中，雖然把自己腦中想法具現化的過程是很清爽的，但是大部分的時候就是工作分配不均所帶來的壓力，下次一定。

B10703113 曾順偉：在這次的期末報告中，我認為我們這組碰到最大的問題是分工問題。在最先開始的分工並未考慮到工作量的差異，導致部分人的工作負擔較重而部分人負擔較輕，而後期發現分工錯誤的問題發生時，由於沒有充分的溝通，已經難以改變原先的分工。我想我在這項專案中最大的收穫是適當分工的重要性，若能在專案過程中做出適當的分工，就可以想必能讓我們專案更順利完成。此外，這門課雖然十分困難，也曾一度讓我覺得很挫折，但卻讓我學習到了很多，除了 Python 的知識，更多的是團隊可能發生的問題，相信這門課結束後大家都會有所成長。最後我想和所有組員說你們都辛苦了。

B10703051 謝智祐：這次的期末專案對我來說是相當特別的經驗，我們試著使用 Python 來製作一個關於臺大的小遊戲，雖來因為時間因素只製作了管校長這一條路線，不過內容包含了許多學校的小知識，也達到了我們讓大家更熟悉校園的目的，也希望能讓試玩的同學會心一笑。專案進行過程中，我們遇到最大的問題是分工不均，讓少數組員承擔了大部分的工作，而到了後期則因為構思內容的同學不太了解程式碼的狀況，而無法較好的分工。也特別感謝柯宥圻同學，除了編寫程式碼、讓遊戲得以順利產出之外，也負責了簡報、錄影、報告的部分，如果缺乏他的幫助，我們勢必難以產出今天這樣的成果。總而言之，這門課給了我許多的啟發，除了了解 python 的知識之外，也讓我認識到許多團隊合作中可能出現的問題，非常感謝各位組員，也希望大家都能在過程中有所學習、成長。

B07605050 廖致豪：這次的作業讓我收穫良多，我嘗試了很多以前沒有試過的程式碼，並且首次接觸到了 Pygame 的運用，雖然在死線前都熬夜到 5、6 點才可以去睡，但是其中製作的過程是非常有趣的，一步步地從 0 到 1，過程真的非常的令人感動，希望我的其他五名財金系組員可以繼續加油。