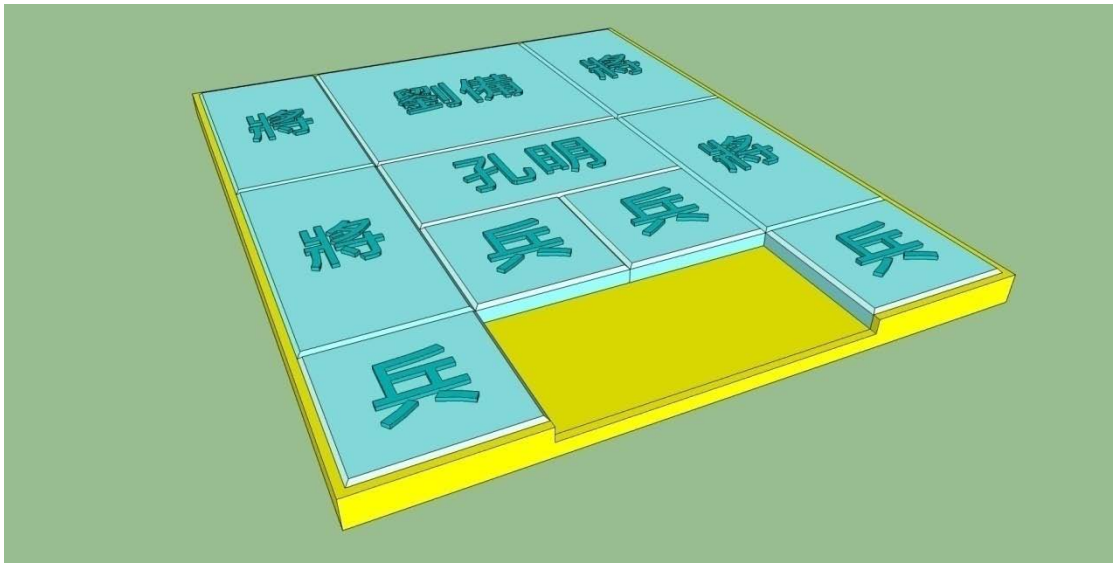


孔明棋

一、前言

很久很久以前，玩過一種版面樣式很單純的棋子，記得就是稱作『孔明棋』。棋面上只有十塊棋子，四個兵、四個將、一塊孔明及一塊劉備。兵是一小塊正方形棋子，將都是兩個兵縱向疊起來的長方形棋子，孔明則是兩個兵橫向連結起來的長方形橫躺棋子，劉備則是四個兵拼起來的大正方形棋子。整個版面只剩兩個兵大小的位置沒有擺放棋子，是可以移動的空間。棋子擺放如下圖，只准用手去推移棋子，不准把任何一塊棋子拿起來，最後要把劉備由最裡邊一步步移動到最前面的出口，才算成功。



很可惜當時原版的棋子遺失了，後來雖也曾用木塊做過一次但不夠滑溜不易推移也沒再保存下來。有一天在市面上終於看到類似產品，一看傻眼了，棋子大小運作方式雖完全一樣，可是意義卻大不同，重點在於棋子的名稱。



劉備還是劉備，孔明變成趙子龍，四個武將的名稱變成關公、曹操、張飛及孔明。聰明的你應該已經看出我傻眼在那裡了。年輕時第一次看到孔明棋就佩服中華文化自古以來的智慧，貴為君王雖權力大可是要移動需花費很多功夫安排相當不容易，武將跟文官基本思考與做法本來就常有衝突，小兵才可以隨意移動幾乎不受任何干擾。所以這個孔明棋完全發揮出這種矛盾與境界，當你在玩這個棋子時一定能充分體會到它設計的用心。

既然沒找到正確產品，因之前曾經去自我學習一下 Python 程式軟體，想說是不是可以自己試著去寫一個『孔明棋』的遊戲來玩玩。正好一位老同事在七月份網購多買了一本「Python 入門課」就送給我，看到這本書裡邊有提到 Canvas 繪圖教學，也許可以現買現賣的利用來撰寫簡易的遊戲。



二、程式撰寫歷程

1. 首先需要各棋子的影像檔，是去利用 SketchUp 來先繪製 3d 棋子。並繪製一個孔明棋的黃色外塑膠盒，順便展示出已經遺失的原版棋子樣子。
2. 用 Python 來試做孔明棋遊戲，需先繪製一個畫布，再把十個棋子的影像依預設位置擺上去。這部分只要設定好座標，處理起來比較單純。十個棋子的影像檔分別為 劉備.gif 孔明.gif 將 1.gif 將 2.gif 將 3.gif 將 4.gif 兵 1.gif 兵 2.gif 兵 3.gif 兵 4.gif。平常我較常用的是 jpg 格式，可是 Canvas 不接受 jpg 檔，只好透過 Photoshop 轉成 gif 檔案格式。這些影像檔集中放置在主程式次目錄中供程式抓取，檔案夾名稱是用 pictures。
3. 接著就是要去思考怎麼讓使用者移動棋子來玩了。在這本書中找到一個案例，是利用定義一個 moveImage(event) 的函式，來用方向鍵移動指定影像的方法如右圖程式檔案。

雖然它只是指定某一特定圖像，每按一次方向鍵，就移動 3 單位的距離。但有了這個案例就可以去依樣畫葫蘆利用來撰寫孔明棋程式中移動棋子的部分了。因為我設定的最小方塊『兵』的影像檔尺寸是 156x156，所以只要改變移動尺寸為 156 就可以每按一次就依方向移動 156 單位了。

只是因棋子有 10 顆

```
# 鍵盤控制圖片的移動
import time
from tkinter import *
def moveImage(event):# 圖片移動函式
    if event.keysym=='Up':
        canvas.move(1,0,-3)# 移動 ID 為 1 的事物，使得橫坐標加 0，縱坐標減 3
    elif event.keysym=='Down':
        canvas.move(1,0,+3)
    elif event.keysym=='Left':
        canvas.move(1,-3,0)
    elif event.keysym=='Right':
        canvas.move(1,3,0)
    tk.update() # 更新表單

tk=Tk()# 窗口
canvas=Canvas(tk,width=400,height=400)# 畫布
canvas.pack()
myImage=PhotoImage(file='cake.png')

img=canvas.create_image(200,200,image=myImage)# 載入圖片
tips=canvas.create_text(10, 50, anchor=NW,text="使用鍵盤控制移動方向")# 創建提示文字
print (img);print (tips) # 顯示圖片和提示文字的 ID
canvas.bind_all('<KeyPress-Up>',moveImage)# 綁定方向鍵 up
canvas.bind_all('<KeyPress-Down>',moveImage)
canvas.bind_all('<KeyPress-Left>',moveImage)
canvas.bind_all('<KeyPress-Right>',moveImage)
```

要指定移動那一顆，在這個案例中並沒有解答，所以必須再找解方。

- 4.上網去尋找如何指定要移動的圖樣，果然找到，不但可以用滑鼠指定要移動的圖像，還可以依滑鼠拖拉位置去移動。

[Python实现用鼠标拖动Canvas上的图形或图像使其移动\(有问题参见后边文](#)

[章\)_geng_zhaoying的博客-CSDN博客_python怎么移动图形](#) 網址提供的案例如下

有了參考資料，就在原程式中加上這部分用滑鼠來指定想要移動的棋子。而且也可以直接用滑鼠左鍵拖拉棋子。不過移動棋子，不可以如該案例用任意角度移動，只能一定是上下或左右每次移動一格(156單位距離)，所以還要去修改移動模式。包括檢測出滑鼠移動的方位角自動調整為往正左、正右移，或是往正上、正下方向移動。

```
1 import tkinter as tk
2 root = tk.Tk()
3 root.geometry('300x300')
4 cv = tk.Canvas(root, height=300, width=300, bg='silver')
5 cv.pack()
6 def StartMove(event):
7     global first_x, first_y, clickID
8     allID=cv.find_closest(event.x, event.y) #halo => 容易找到細直線
9     if len(allID) > 0:
10        clickID=allID[0]
11        first_x, first_y = event.x, event.y
12 def StopMove(event):
13     global first_x, first_y, clickID
14     cv.move(clickID, event.x-first_x, event.y-first_y)
15     clickID=-1
16 def OnMotion(event):
17     global first_x, first_y, clickID
18     if clickID!=-1:
19        cv.move(clickID, event.x-first_x, event.y-first_y)
20        first_x, first_y = event.x, event.y
21 clickID=-1
22 cv.create_line(10, 10, 50, 50, activefill='red')
23 cv.create_oval(10, 70, 100, 160, activefill='red')
24 cv.create_rectangle(50, 120, 150, 180, fill='#CCFFFF', activefill='red')
25 cv.bind("<ButtonPress-1>", StartMove) #绑定鼠标左键按下事件
26 cv.bind("<ButtonRelease-1>", StopMove) #绑定鼠标左键松开事件
27 cv.bind("<B1-Motion>", OnMotion) #绑定鼠标左键被按下时移动鼠标事件
28 root.mainloop()
29
```

因為程式中有兩種移動機制，所以滑鼠指定移動對象後，用鍵盤或用滑鼠都可以去移動棋子了。不過撰寫移動棋子的程式，應該還要有一個很重要的機制，也就是只有空位才可以移過去，碰到旁邊有棋子應該拒絕移動的。可是關於這一點也上網去尋找資源，可惜迄今尚未找到，自己想破頭也沒結果。所以這個自撰的程式還有很大瑕疵，只是如果能自律不要去硬撞其他棋子，還是勉強可玩一下的。

三、此自撰孔明棋玩法

目前尚未安裝可以將 PY 檔案包裝轉成 exe 執行檔的工具，要玩時只能：

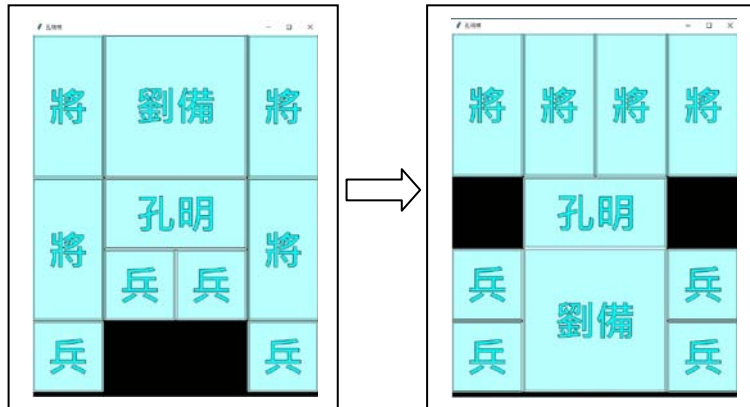
1.先安裝 python

上網應該很容易就可以找到下載 python 的網站，但這幾天上去找卻發現好像已經找不到可以安裝在 win7 系統上的了，但 win8 以上系統應該沒問題。

- 2.把後附的 Conmin_mouse.py 程式放在一檔案夾中，並建立一個次檔案夾，名稱設為 pictures，並把後附的劉備.gif 孔明.gif 將 1.gif 兵 1.gif 四個影像檔放上去，再把 將 1.gif 複製多出 2.gif 將 3.gif 將 4.gif 三個檔，把兵 1.gif 複製多出 兵 2.gif 兵 3.gif 兵 4.gif 三個檔。

- 3.去點選 Conmin_mouse.py 就會直接利用 python 來執行，顯現出孔明棋的畫面。由原始畫面用滑鼠點要走的棋子，再用滑鼠拖拉或方向鍵移位，由左圖走至右圖畫面，就算完成了。

下面是自撰孔明棋程式的文字資訊，後附 Conmin_mouse.py 檔案萬一無法下載，有興趣的人也可以自己建立程式 py 檔。



```

Conmin_mouse.py > ...
1 # 滑鼠控制玩孔明棋
2
3 import time
4 from tkinter import*
5 # 聲明用於圖片的全域變數
6 global img0,img1,img2,img3,img4,img5,img6,img7,img8,img9
7 me=0
8 def StartMove(event):
9     global first_x,first_y,clickID,me
10    allID=myCanvas.find_closest(event.x,event.y)
11    if len(allID)>0:
12        clickID=allID[0]
13        first_x,first_y=event.x,event.y
14        me=clickID # 點要移動的圖像
15 def StopMove(event):
16     global first_x,first_y,clickID
17
18
19     ddx=event.x-first_x
20     ddy=event.y-first_y
21
22     # print (clickID)
23     # py print (first_x);print(event.x);print(ddx)
24     # print (first_y);print(event.y);print(ddy)
25
26     if (ddx > 0 ) and (ddx > abs(ddy)):
27         myCanvas.move(clickID,156,0)
28     elif (ddx < 0 ) and (abs(ddx) > ddy):
29         myCanvas.move(clickID, -156,0)
30     elif (ddy > 0 ) and (ddy > abs(ddx)):
31         myCanvas.move(clickID,0,156)
32     elif (ddy < 0 ) and (abs(ddy) > ddx ):
33         myCanvas.move(clickID,0,-156)
34
35     # myCanvas.move(clickID,100,50) # 先不移動作法，再用鍵盤移動
36     clickID=-1
37
38 x=156
39 y=156
40
41
42 def moveImage(event): # 圖面鍵盤移動函式
43     if event.keysym=='Up':
44         myCanvas.move(me,0,-y) # 移動ID為 me 的事物，使得橫座標加 0，縱座標減 3
45     elif event.keysym=='Down':
46         myCanvas.move(me,0,+y)
47     elif event.keysym=='Left':

```

```
Conmin_mouse.py > ...
48     myCanvas.move(me,-x,0)
49     elif event.keysym=='Right':
50         myCanvas.move(me,+x,0)
51     tk.update() # 更新表單
52
53 tk=Tk() # tk 是本程式 代號
54 tk.title("孔明棋") # 也可用中文替換
55 myCanvas=Canvas(tk,width=624,height=788,bg='black')
56 myCanvas.pack()
57
58 # 圖片
59 img0=PhotoImage(file='pictures\\孔明.gif')
60 img1=PhotoImage(file='pictures\\劉備.gif')
61 img2=PhotoImage(file='pictures\\將1.gif')
62 img3=PhotoImage(file='pictures\\將2.gif')
63 img4=PhotoImage(file='pictures\\將3.gif')
64 img5=PhotoImage(file='pictures\\將4.gif')
65 img6=PhotoImage(file='pictures\\兵1.gif')
66 img7=PhotoImage(file='pictures\\兵2.gif')
67 img8=PhotoImage(file='pictures\\兵3.gif')
68 img9=PhotoImage(file='pictures\\兵4.gif')
69
70 myCanvas.create_image(156,312,anchor=NW,image=img0) # 孔明
71 myCanvas.create_image(156,0,anchor=NW,image=img1) # 劉備
72 myCanvas.create_image(0,0,anchor=NW,image=img2) # 將1
73 myCanvas.create_image(468,0,anchor=NW,image=img3) # 將2
74 myCanvas.create_image(0,312,anchor=NW,image=img4) # 將3
75 myCanvas.create_image(468,312,anchor=NW,image=img5) # 將4
76 myCanvas.create_image(0,624,anchor=NW,image=img6) # 兵1
77 myCanvas.create_image(156,468,anchor=NW,image=img7) # 兵2
78 myCanvas.create_image(312,468,anchor=NW,image=img8) # 兵3
79 myCanvas.create_image(468,624,anchor=NW,image=img9) # 兵4
80
81 myCanvas.bind_all('<KeyPress -Up>',moveImage)
82 myCanvas.bind_all('<KeyPress -Down>',moveImage)
83 myCanvas.bind_all('<KeyPress -Left>',moveImage)
84 myCanvas.bind_all('<KeyPress -Right>',moveImage)
85
86 myCanvas.bind("<ButtonPress-1>",StartMove)
87 myCanvas.bind("<ButtonRelease-1>",StopMove)
88
89 tk.mainloop()
90
```

四、結語

- 1.想寫孔明棋的文章前，也先上網去想找一下畫面，結果出乎我預料之外，上去查「孔明棋」的照片，居然都是如下頁圖，是一個大圓盤，上面有 36 顆玻璃彈珠，可以跳來跳去的玩意兒。也就是叫做「孔明智慧盤」的遊戲。我文章內容的這種「孔明棋」還真不知道在網路上有沒有出現過呢。

2.首次聽到 Python 軟體是在數位社活動的課程中，後來也曾去網上找教學影片自學了一下初級課程，也嘗試把原用 Visual Basic 寫的幾個平面曲線的程式用 Python 改寫，都還可以正常運作。只是沒有繼續研究下去，很快一晃三四年過去了。既然再度接觸 Python，應該趁熱再去玩玩，別讓逐漸生鏽的腦袋癡呆掉。



- 3.寫遊戲來玩也是第一次嘗試，照講應該是弄出一個執行檔，用滑鼠左鍵點一下就能玩，可是卻還沒做到。另外移動旗子也應該碰壁時要限制不能移動，也沒能做到。所以本篇只能算是介紹這種「孔明棋」及嘗試寫遊戲的一個經驗談，要讓遊戲真的可以玩，還要後續花時間找資源去學去研究了。
- 4.遊戲總要附解答，因為步數需要一百多步，用圖片太多了，所以就去做一個影片 mp4 檔，放在附錄中供參考。步數細節不是唯一，只能算是參考解法。

附錄檔案：

- | | |
|-------------------|----------|
| 1.Conmin_mouse.py | 主程式 |
| 2.劉備.gif | 劉備棋子的影像檔 |
| 3.孔明.gif | 孔明棋子的影像檔 |
| 4.將 1.gif | 武將棋子的影像檔 |
| 5.兵 1.gif | 兵棋子的影像檔 |
| 6.孔明棋.mp4 | 參考解法小影片檔 |

雲端硬碟資料夾：

https://drive.google.com/drive/folders/1C1qOU0wL_vfIbt4jrDjzKywkhITyXBL4?usp=sharing

Youtube 影片：

<https://youtu.be/40YkaG5YDVU>