Lab 3: Python GUI Programming Report

學號：110511254　姓名：徐煜絨

1. 若要設計一個電腦玩家（以下稱電腦）與你（以下稱玩家）對抗，你會如何設計，請簡述你的演算法

假設電腦會是 player 2。首先 import random，如果玩家的第一手下在正中間，則電腦下的第一手會是隨機位置。反之，若玩家第一手不下在中間，電腦就要以中間那格為優先考量，如右圖。這裡會新增 place_r 和 place_c 決定電腦要下的位置：

	X(step1)	
O(step1)		

```
place_r = print(random.randint(0, 2))
place_c = print(random.randint(0, 2))
```

random 決定要下在哪裡僅限於第一手，而後是靠電腦判斷自己和玩家的互動，讓電腦往容易獲勝的方向前進。以下擷取自 handle_click 函式：

```
else:
    board[row][col] = 2
    button.config(text='X')
    current_player = 1
```

若是改成電腦，應該把 else 中的程式碼改為：

```
board[place_r][place_c] = 2
button.config(text='X')
current_player = 1
```

在玩家下好第二步後，電腦應該判斷玩家是否能在下一步就完成連線，用 if-else 的方式確認玩家第一（位置 a）、第二（位置 b）步下的格子的關係，若在可以連線的方向上，如同一行、同一列、同條對角線上，則電腦第二步要下在 a 和 b 的連線上，範例如右圖。

X(step2)	X(step1)	
	O(step1)	
		O(step2)

若 a 和 b 不在可以連線的方向上，則設計電腦下一步會下在四個角落。原因是如果玩家第一步下在中間，可以保證 a 和 b 可以再經過一手就成功連線，那就會像上一段所描述。如果玩家第一手不下在正中間，a 和 b 又不在可以連線的方向上，表示電腦佔到了正中間的格子。

X(step2)		O(step1)
O(step2)	X(step1)	

玩家下完第三手後（位置 c），如果 a, b, c 任兩點有辦法連成路徑上沒有 X 的線，且數量有兩條，那玩家一定獲勝（狀況一）；如果 a, b, c 有

辦法連成路徑上沒有 X 的線，且數量是一，則電腦會去阻擋另一條線完整形成（狀況二）；其他時候，如果已經下好的兩個 X 可以連線，則電腦第三手就讓這條成功出現，如果連線被 O 干擾，電腦會另外找和前兩個 X 有機會連線的格子（狀況三）。

	X(step1)	O(step3)
X(step2)	O(step1)	O(step2)

狀況一：電腦必輸

O(step1)	O(step2)	X(step2)
X(step3)	X(step1)	
O(step3)		

狀況二：電腦去擋

O(step1)	O(step2)	X(step2)
	X(step1)	
X(step3)		O(step3)

狀況三：電腦找機會

另外還有一種狀況如右圖，在玩家與電腦都下完前三步，且沒有任何一方有贏的跡象，之後電腦基本上都會是以防守方向下棋，以免玩家獲勝。補充：右圖中，三個 O 都無法連線，前兩個 X 也無法連線，因此第三個 X 會以隨機方式產生，接著再判斷和之前 X 的連線上會不會有 O，有的話繼續找下一個可以使用的格子（如果找不到沒有的，則隨機放置 X）。

O(step1)	X(step2)	O(step2)
	X(step1)	
X(step3)	O(step3)	

但 X 連線上有 O(step2)，所以繼續找

↓

O(step1)	X(step2)	O(step2)
X(step3)	X(step1)	
	O(step3)	

此時 X(step1, 3)連線上沒有 O，成功！

接下來就是重複前兩段敘操作，電腦藉由判斷 O 是否會連線、是否要防守、如何進攻，以決定下一個 X 要放的位子。

2. 請貼上自己的程式碼並附上註解

a. import

```
import tkinter as tk  #python的GUI工具包
from tkinter import messagebox  #警示視窗
```

b. main

```
if __name__ == '__main__':  #避免其他檔案執行時執行到不必要指令，參考Reference 1

    #create main window
    window = tk.Tk()  #一個叫window的新視窗
    window.title("Lab3 Tic-Tac-Toe")  #視窗title

    #create game board
    create_board()

    #Initialize variables
    board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
    current_player = 1
```

```
    #將Tkinter物件放入等待迴圈，讓window不斷重新整理
    window.mainloop()
```

c. create game board

```
#create board
def create_board():
    for i in range(3):  #長邊寬邊各有三格(3*3)
        for j in range(3):
            #相關表格設定
            button = tk.Button(window, text='', font=("Arial", 50),
                               height=2, width=6,
                               command=lambda row=i, col=j: handle_click(row, col))
            #grid適在畫九宮格使用，nsew指貼齊上下左右
            button.grid(row=i, column=j, sticky="nsew")
```

d. deal with the players’ clicks

```
#Handle button clicks
def handle_click(row, col):
    global current_player  #判斷是幾號玩家動作
    global board  #需要對board中被點選到的格子做改變

    #check with button has been clicked and change player
    #change text on botton to show 'O' or 'X'
    #把被選到的格子稱作button
    button = window.grid_slaves(row=row, column = col)[0]
    if board[row][col] == 0:  #表示那格尚未被選過
        if current_player == 1:
            board[row][col] = 1  #表示玩家1選到
            button.config(text='O')  #顯示"O"
            current_player = 2  #換成玩家2選格子
        else:
            board[row][col] = 2
            button.config(text='X')
            current_player = 1
        check_winner()  #檢查是否出現贏家或平手
```

e. check if someone wins the game or tie

```
#check for a winner or a tie
def check_winner():
    winner = None  #用以存取贏的方式
    global whoWin  #紀錄是誰贏
    whoWin = 0
    count = 0
```

```
    #check rows
    if (board[0][0] == board[0][1] == board[0][2] != 0):
        whoWin = board[0][0]  #看是哪一位玩家在這格留下紀錄
        winner = "1"  #贏的方式是type1
    elif (board[1][0] == board[1][1] == board[1][2] != 0):
        whoWin = board[1][0]
        winner = "2"
    elif (board[2][0] == board[2][1] == board[2][2] != 0):
        whoWin = board[2][0]
        winner = "3"

    #check columns
    if (board[0][0] == board[1][0] == board[2][0] != 0):
        whoWin = board[0][0]
        winner = "4"
    elif (board[0][1] == board[1][1] == board[2][1] != 0):
        whoWin = board[0][1]
        winner = "5"
    elif (board[0][2] == board[1][2] == board[2][2] != 0):
        whoWin = board[0][2]
        winner = "6"
```

```
    #check diagonals
    if (board[0][0] == board[1][1] == board[2][2] != 0):
        whoWin = board[0][0]
        winner = "7"
    elif (board[0][2] == board[1][1] == board[2][0] != 0):
        whoWin = board[1][1]
        winner = "8"

    #check if tie
    if whoWin==0:
        for k in range (3):
            for p in range (3):
                count = count + board[k][p]
        #當player1按過五次、player2按過四次則所有格子填滿，遊戲結束
        #1*5+2*4=13，所以若count=13就表示遊戲結束且平手
        if count==13:
            whoWin = 0
            winner = "Tie"

    if winner:  #若贏的方式有值（知道贏的type，有8種）
        declare_winner(winner)
```

f. change the color of the board according to the result

```
# Declare the winner and ask to restart the game
def declare_winner(winner):
    if winner == "Tie":  #平手則全部格子變紅色
        for r in range (3):
            for t in range (3):
                button1 = window.grid_slaves(row=r, column = t)[0]
                button1.config(bg='red')

    #依照贏的方式不同，決定要讓哪幾格變色（有成功連線的格子們）
    elif winner == "1":
        button1 = window.grid_slaves(row=0, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=0, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=0, column = 2)[0]
        button1.config(bg='blue')
    #以下太過繁瑣，附於appendix
    elif winner == "2":…

    elif winner == "3":…

    elif winner == "4":…
```

```
    elif winner == "4":…

    elif winner == "5":…

    elif winner == "6":…

    elif winner == "7":…

    elif winner == "8":…

    #ask if to continue
    answer = tk.messagebox.askyesno(title='End', message='Again?')

    if answer:  #使用者想繼續玩下一輪
        #play another round
        global board
        board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]  #格子參數初始化

        for i in range (3):
            for j in range(3):
                button = window.grid_slaves(row=i, column=j)[0]
                #讓格子的顏色變回白色
                button.config(text="", bg=default_bg_color)
                # button.config(text="", bg='white')
        global current_player
        current_player = 1  #player初始化
    else:
        window.destroy()
```

3. 心得

這次 lab 從 Android Studio 改回 Python，我希望可以學好用 Python 做 GUI 介面，因為我的專題會使用到學長姐透過 Python 所創的介面，期待能夠學習這項工具，並理解如何對其進行排版或優化。此外，回到宿舍和室友玩這次的成果，讓我頗有成就感！

Reference:

1. https://blog.castman.net/教學/2018/01/27/python-name-main.html
2. https://hn28082251.blogspot.com/2019/05/python-if-name-main.html

Appendix

```
elif winner == "1":
        button1 = window.grid_slaves(row=0, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=0, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=0, column = 2)[0]
        button1.config(bg='blue')
    #以下太過繁瑣，附於 appendix
    elif winner == "2":
        button1 = window.grid_slaves(row=1, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 2)[0]
        button1.config(bg='blue')

    elif winner == "3":
        button1 = window.grid_slaves(row=2, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 2)[0]
        button1.config(bg='blue')

    elif winner == "4":
        button1 = window.grid_slaves(row=0, column = 0)[0]
        button1.config(bg='blue')
```

```
        button1 = window.grid_slaves(row=1, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 0)[0]
        button1.config(bg='blue')

    elif winner == "5":
        button1 = window.grid_slaves(row=0, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 1)[0]
        button1.config(bg='blue')

    elif winner == "6":
        button1 = window.grid_slaves(row=0, column = 2)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 2)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 2)[0]
        button1.config(bg='blue')

    elif winner == "7":
        button1 = window.grid_slaves(row=0, column = 0)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 2)[0]
        button1.config(bg='blue')

    elif winner == "8":
        button1 = window.grid_slaves(row=0, column = 2)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=1, column = 1)[0]
        button1.config(bg='blue')
        button1 = window.grid_slaves(row=2, column = 0)[0]
        button1.config(bg='blue')
```