

作業二

姓名: 王佳君

學號: 00557149

日期: 2019.06.09

一、想法

由老師提供的程式碼為基底，做出隨機產生上下左右指令(右上方紅色字樣)，且指令出現後隨即消失。遊戲者須遵照此指令做出相應方向之動作，若完成則可獲得一分，且得到新指令；若未完成，則指令不變，直到完成。

二、演算步驟、重要程式片段說明

1.

```
def fn(img):  
    digit = random.randint(1, 4)  
  
    return digit
```

利用此函式隨機產生出 1~4 中之一整數，其各代表"右左下上"指令

2.

```
cv2.putText(img, 'SCORE:', (300, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)  
cv2.putText(img, str(score), (520, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)
```

在畫面顯示出 SCORE，且不停更新

3.

```
if left/2 > right and left/2 > down and left/2 > up:  
    cv2.putText(imgf, 'Right', (0, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)  
    if digit == 1:  
        score += 1  
        digit = fn(img)  
elif right > left and right > down and right > up:  
    cv2.putText(imgf, 'Left', (0, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)  
    if digit == 2:  
        score += 1  
        digit = fn(img)  
elif down/2 > left and down/2 > right and down/2 > up:  
    cv2.putText(imgf, 'Down', (0, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)  
    if digit == 3:  
        score += 1  
        digit = fn(img)  
elif up/2 > left and up/2 > right and up/2 > down:  
    cv2.putText(imgf, 'Up', (0, 50), cv2.FONT_HERSHEY_PLAIN, 4, (255, 255, 255), 3)  
    if digit == 4:  
        score += 1  
        digit = fn(img)
```

判斷所做動作是否符合指令，若符合則分數加一，且產生新指令

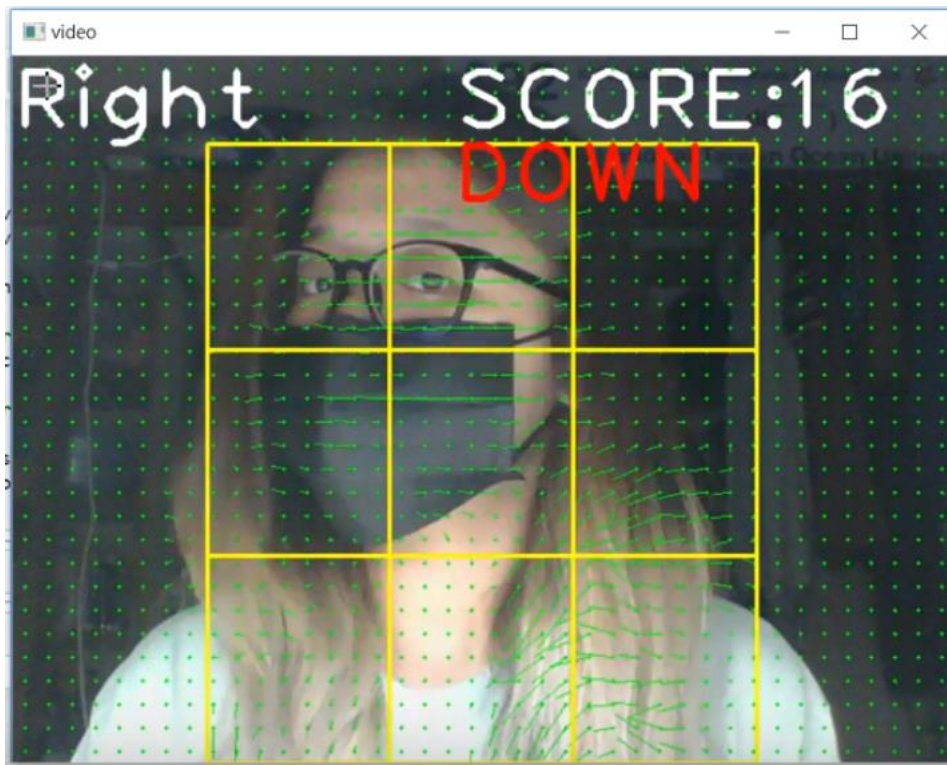
4.

```
if digit == 1:
    cv2.putText(img, 'RIGHT', (300, 100), cv2.FONT_HERSHEY_PLAIN, 4, (0, 0, 255), 3)
elif digit == 2:
    cv2.putText(img, 'LEFT', (300, 100), cv2.FONT_HERSHEY_PLAIN, 4, (0, 0, 255), 3)
elif digit == 3:
    cv2.putText(img, 'DOWN', (300, 100), cv2.FONT_HERSHEY_PLAIN, 4, (0, 0, 255), 3)
elif digit == 4:
    cv2.putText(img, 'DOWN', (300, 100), cv2.FONT_HERSHEY_PLAIN, 4, (0, 0, 255), 3)
```

判斷隨機產生出的數字，在畫面上顯示出指令字樣

三、結果

Youtube 連結: <https://youtu.be/ZSOMJY0NBT4>



四、結論

原先之遊戲設定為：每隔 1~3 秒(按得分狀況調秒數)，不論是否完成指令皆更新指令。但因為不會用 python 中的 thread，嘗試多種寫法仍失敗，無法將主要程式動作和停頓 1~3 秒才產生指令這兩件事合理的處理，故改變遊戲玩法為現在此種。

五、主要參考文獻

<https://segmentfault.com/q/1010000009706708>

<https://bbs.csdn.net/topics/390770601>

http://violin-tao.blogspot.com/2017/05/python3_26.html

<https://codertw.com/%E7%A8%B%E5%BC%8F%E8%AA%9E%E8%A8%80/367012/>

<https://medium.com/@alan81920/python-%E7%9A%84-concurrency-%E5%92%8C-p-arallelization-efeddc30c4c>

<http://www.runoob.com/python/att-time-clock.html>

http://jeremybai.github.io/blog/2015/06/10/time_vs_clock

<https://chtseng.wordpress.com/2017/06/02/%E5%81%B5%E6%B8%AC%E7%A7%BB%E5%8B%95%E4%B8%AD%E7%9A%84%E7%89%A9%E9%AB%94%E4%B8%A6%E5%8F%96%E5%BE%97%E5%85%B6%E5%BD%B1%E5%83%8F/>