



手勢追蹤小遊戲

作者 吳彥瑾

作品名稱：手勢追蹤小遊戲

一、說明

本專題使用 anaconda3.9 以上版本、mediapipe、opencv 製作出貪吃蛇小遊戲、桌上撞球的互動小遊戲。

套件安裝準備：

Jupyter 本身是一個 Python 的編輯環境，如果直接安裝 mediapipe 可能會導致運作時互相衝突，因此需要先安裝 mediapipe 的虛擬環境，在上面安裝 mediapipe 後就能正常運行。

1. 請到 anaconda 官方網站下載軟體，到桌面開啟 cmd 執行 `conda create --name mediapipe python=3.9`。
2. 輸入 `conda activate mediapipe` 或 `pip install mediapipe` 指令安裝套件。
3. 輸入 `pip install tensorflow` 指令安裝套件。
4. 輸入 `pip install opencv-python` 指令安裝套件。
5. 輸入 `pip install cvzone` 指令安裝套件。

二、相關文章

使用 MediaPipe 進行手掌的偵測，再透過 OpenCV 讀取攝影鏡頭影像進行辨識，在手掌與每隻手指標記骨架。

文章：

MediaPipe Hands 利用多個模型協同工作，可以偵測手掌模型，返回手掌與每隻手指精確的 3D 關鍵點，MediaPipe Hand 除了可以偵測清晰的手掌形狀與動作，更可以判斷出被少部分被遮蔽的手指形狀和動作，再清晰的畫面下，針對手掌判斷的精準度可達 95.7%。

Mediapipe 偵測手掌後，會在手掌與手指上產生 21 個具有 x、y、z 座標的節點，透過包含立體深度的節點，就能在 3D 場景中做出多種不同的應用，下圖標示出每個節點的順序和位置

如果同時出現兩隻手，採用交錯偵測（短時間內偵測兩次，一次偵測一隻手），最後仍然維持 21 個點的數據，如果只希望偵測一隻手，可設定 `max_num_hands=1`

參考網址：

https://developers.google.com/mediapipe/solutions/vision/hand_landmarker



三、 實作

1. 貪吃蛇小遊戲：

用手控制蛇頭，指尖碰觸到的食物獲得一分，食物就會隨機切換位置，蛇身就變長。指尖停止移動或撞到蛇身會顯示遊戲結束與成績或按 R 從頭開始。(按 ESC 鍵結束畫面視窗)

程式碼：

```
import math
import random
import cv2
import cvzone
import numpy as np
from cvzone.HandTrackingModule import HandDetector

cap = cv2.VideoCapture(0) # 電腦的攝影機
#設置畫面的大小
cap.set(3,1280)
cap.set(4,720)

detector = HandDetector(detectionCon=0.8, maxHands=1)

class SnakeGameClass:
    def __init__(self, pathFood):
        self.points = [] # 蛇身上所有的點
        self.lengths = [] # 不同點之間距離
        self.currentLength = 0 # 當前蛇的長度
        self.allowedLength = 150 # 最大允許長度
        self.previousHead = 0, 0 # 先前蛇的頭部
        self.imgFood = cv2.imread(pathFood, cv2.IMREAD_UNCHANGED)
        self.hFood, self.wFood, _ = self.imgFood.shape
        self.foodPoint = 0, 0
        self.randomFoodLocation()
        self.score = 0
        self.gameOver = False

    def randomFoodLocation(self):
        self.foodPoint = random.randint(100, 1000), random.randint(100, 600)

    def update(self, imgMain, currentHead): # 實例方法
```

```

# 遊戲結束，打印腳本
if self.gameOver:
    cvzone.putTextRect(imgMain, "Game Over", [370, 350],
                        scale=7, thickness=5, offset=20)
    cvzone.putTextRect(imgMain, f'Your Score:{self.score}', [300, 500],
                        scale=7, thickness=5, offset=20)
else:
    px, py = self.previousHead
    cx, cy = currentHead
    self.points.append([cx, cy]) # 蛇的點列表節點
    distance = math.hypot(cx - px, cy - py) # 兩點之間的距離
    self.lengths.append(distance) # 蛇的距離列表內容
    self.currentLength += distance
    self.previousHead = cx, cy

# 長度縮小
if self.currentLength > self.allowedLength:
    for i, length in enumerate(self.lengths):
        self.currentLength -= length
        self.lengths.pop(i)
        self.points.pop(i)
        if self.currentLength < self.allowedLength:
            break
# 檢查貪吃蛇是否碰觸到食物
rx, ry = self.foodPoint
if rx - self.wFood // 2 < cx < rx + self.wFood // 2 and \
    ry - self.hFood // 2 < cy < ry + self.hFood // 2:
    self.randomFoodLocation()
    self.allowedLength += 50
    self.score += 1
    print(self.score)

# 使用線條繪製貪吃蛇
if self.points:
    for i, point in enumerate(self.points):
        if i != 0:
            cv2.line(imgMain, self.points[i - 1], self.points[i], (128, 255,
255), 20)

```

```

        cv2.circle(imgMain, self.points[-1], 20, (128, 128, 255), cv2.FILLED)

    # 顯示食物
    imgMain = cvzone.overlayPNG(imgMain, self.imgFood,
                                (rx - self.wFood // 2, ry - self.hFood //
2))

    cvzone.putTextRect(imgMain, f'Your Score:{self.score}', [50, 80],
                        scale=3, thickness=5, offset=10)

    # 檢測是否碰撞
    pts = np.array(self.points[:-2], np.int32)
    pts = pts.reshape((-1, 1, 2))
    cv2.polylines(imgMain, [pts], False, (0, 0, 255), 3)
    minDist = cv2.pointPolygonTest(pts, (cx, cy), True)

    if -1 <= minDist <= 1:
        print("Hit")
        self.gameOver = True
        self.points = [] # 蛇身上所有的點
        self.lengths = [] # 不同點之間距離
        self.currentLength = 0 # 當前蛇的長度
        self.allowedLength = 150 # 最大允許長度
        self.previousHead = 0, 0 # 先前蛇的頭部
        self.randomFoodLocation()

    return imgMain

game = SnakeGameClass("snake\dd.png")
while True:
    success, img = cap.read()
    img = cv2.flip(img, 1) # 鏡像翻轉
    hands, img = detector.findHands(img, flipType=False)
    # 檢測到第一個手，標記手部位置
    if hands:
        lmList = hands[0]['lmList']
        pointIndex = lmList[8][0:2] # 只要食指指尖的 x 和 y 坐標
        img = game.update(img, pointIndex)

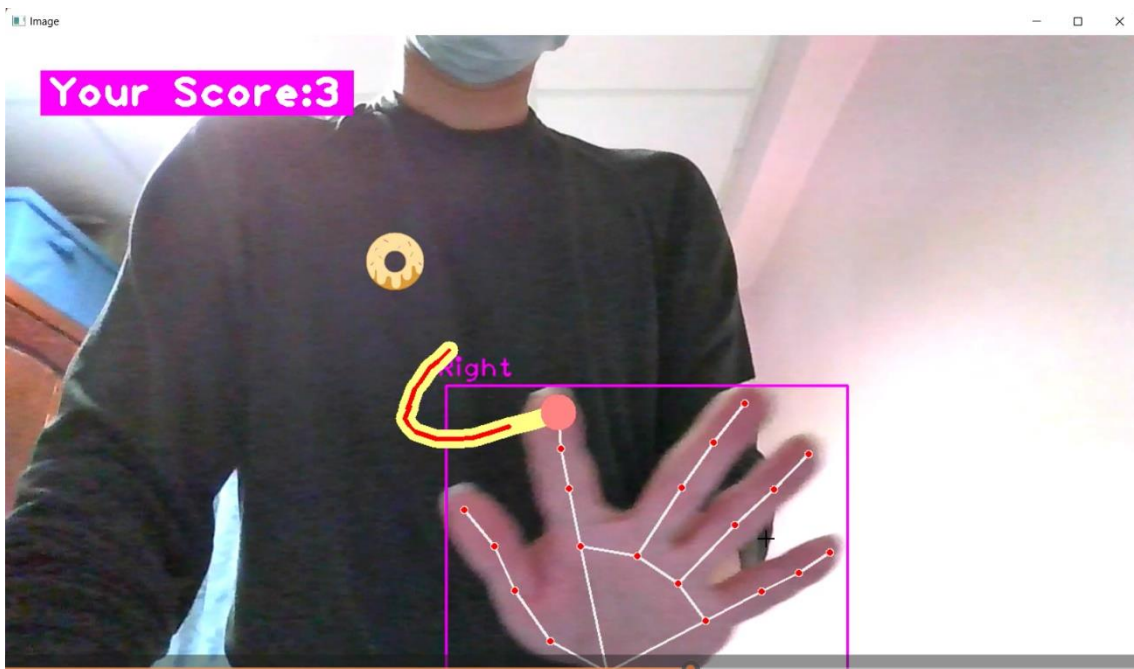
```

```

cv2.imshow("Image", img)
key = cv2.waitKey(1)
# 按下 'q'重新開始遊戲
if key == ord('q'):
    game.gameOver = False
    game.score = 0
    game.points = [] # 蛇身上所有的點
    game.lengths = [] # 不同點之間距離
    game.currentLength = 0 # 當前蛇的長度
    game.allowedLength = 150 # 最大允許長度
    game.previousHead = (0, 0) # 先前蛇的頭部
    game.randomFoodLocation()
if key & 0xFF == 27: # 按下 Esc 鍵停止
    break
#釋放頻資源
cap.release()
cv2.destroyAllWindows()

```

執行結果：



2. 桌上冰球互動小遊戲：

左右球拍可以用手上下移動、圖片是球，被球拍撞到後反彈則獲得一分，下面則次紀錄次數，球超出外界就會顯示遊戲結束與成績或按 R 從頭開始。（按 ESC 鍵結束畫面視窗）

程式碼：

```
import cv2
```

```
import cvzone
from cvzone.HandTrackingModule import HandDetector

cap = cv2.VideoCapture(0) # 電腦的攝影機
#設置畫面的大小
cap.set(3,1280)
cap.set(4,720)

imgDesk = cv2.imread('puck/desk.png')
imgBall = cv2.imread('puck/ball.png', cv2.IMREAD_UNCHANGED)
imgBlock1 = cv2.imread('puck/block1.png', cv2.IMREAD_UNCHANGED)
imgBlock2 = cv2.imread('puck/block2.png', cv2.IMREAD_UNCHANGED)

imgDesk = cv2.resize(imgDesk, dsize=(1280, 720))

imgBlock1 = cv2.resize(imgBlock1, dsize=(50,200))
imgBlock2 = cv2.resize(imgBlock2, dsize=(50,200))

detector = HandDetector(detectionCon=0.8, maxHands=2)

ballpos = [100, 100]

speedx, speedy = 10, 10

gameover = False

score = [0, 0]

while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)
    hands, img = detector.findHands(img, flipType=False)
    img = cv2.addWeighted(img, 0.4, imgDesk, 0.6, 0)

    if hands:
        for hand in hands:
            x, y, z = hand['lmList'][8]
            h1, w1 = imgBlock1.shape[0:2]
```



```

y1 = y - h1 // 2

if hand['type'] == 'Left':
    img = cvzone.overlayPNG(img, imgBlock1, (40, y1))
    if 40 < ballpos[0] < 40 + w1 and y1 < ballpos[1] < y1 + h1:
        speedx = -speedx
        score[0] += 1

if hand['type'] == 'Right':
    img = cvzone.overlayPNG(img, imgBlock2, (1180, y1))
    if 1080 < ballpos[0] < 1080 + w1 and y1 < ballpos[1] < y1 + h1:
        speedx = -speedx
        score[1] += 1
if ballpos[0] < 50 or ballpos[0] > 1150:
    gameover = True

if gameover is True:
    cvzone.putTextRect(img, "Game Over", [300, 300], scale=7, thickness=5,
offset=20)
    cvzone.putTextRect(img, f'Left:{score[0]} and Right:{score[1]}', [100,
500], scale=7, thickness=5, offset=20)

else:
    if ballpos[1] >= 600 or ballpos[1] <= 50:
        speedy = -speedy
# 球的 x 和 y 坐標
ballpos[0] = ballpos[0] + speedx
ballpos[1] = ballpos[1] + speedy
# 桌球圖片、將 imgBall 放在球桌 img 的固定坐標位置
img = cvzone.overlayPNG(img, imgBall, ballpos)
# 顯示計分板
cvzone.putTextRect(img, f'Left:{score[0]} and Right:{score[1]}', (400, 710))

cv2.imshow("Image", img)
key = cv2.waitKey(1)
# 按下 'q' 重新開始遊戲
if key == ord('q'):

```



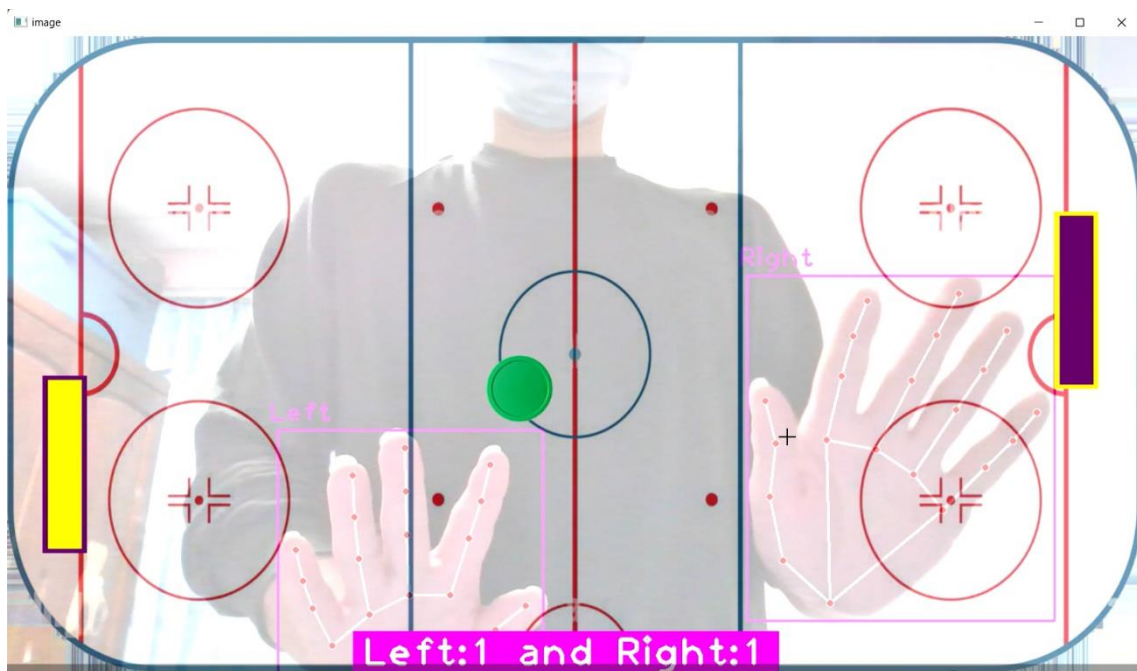
```

ballpos = [100, 100]
speedx, speedy = 10, 10
gameover = False
score = [0, 0]

if key & 0xFF == 27: # 按下 Esc 鍵停止
    break
#釋放頻資源
cap.release()
cv2.destroyAllWindows()

```

執行結果：



四、結論

本專案就到這邊，對此作品能夠帶給自己的成就，身體力行與研究學習很重要，為自己的作品展示出更好的加分。

五、參考

https://www.jb51.net/article/246585.htm#_lab2_1_1(桌上推球)、
<https://www.jb51.net/article/240489.htm>(貪吃蛇)