

人機互動課程

-- 期末報告展示 --

組員：

資料碩一 王柏仁

廖宇凡

唐英哲

題目：

人體 Flappy Bird 互動遊戲

目錄

- 1) 內容簡介
- 2) 組員分工
- 3) 實作細節
- 4) 介面及成果展示
- 5) 問題與討論
- 6) 總結
- 7) 程式碼及參考資料

1) 內容簡介

隨著科技發展、軟硬體技術的進步，有許多人機互動的裝置一一興起，從以前的打字機、終端機，到現今的電腦、智慧音箱，腦機介面...等等。此外，也隨著技術（API）的開放，有許多人自行利用這些工具來開發程式以用在不同情境，如遊戲、智慧助理...等。其中在互動遊戲領域，有許多人利用 Kinect、VR 眼鏡...等裝置開發出許多產品，因此。在本此期末專案中，我們利用 Kinect Version 1 裝置結合前幾年掀起熱潮的熱門遊戲：Flappy Bird，開發出簡易版的人體 Flappy Bird。

在我們的遊戲中，目標是盡可能讓小鳥穿越越多水管以獲取更多分數，規則部分則需利用揮動雙臂來開始遊戲、稍大幅度的甩動雙臂當成是「鳥揮動其翅膀」來通過兩管之間，而在遊戲途中，則可以抬起右腳當成是救命工具，在快要撞上水管前，使用此技能讓小鳥瞬間移動到安全區域，但此技能只有五次使用機會而已。

2) 組員分工

介面開發及遊戲設計部分主要為碩一廖宇凡同學，王柏仁與唐英哲同學在這部分則進行輔助幫忙，協助除錯（Debug）；在讀取 Kinect 資料、資料運算則由王柏仁與唐英哲同學負責。而至於演算法、遊戲規則設計則為三人一起討論合作討論出來

3) 實作細節

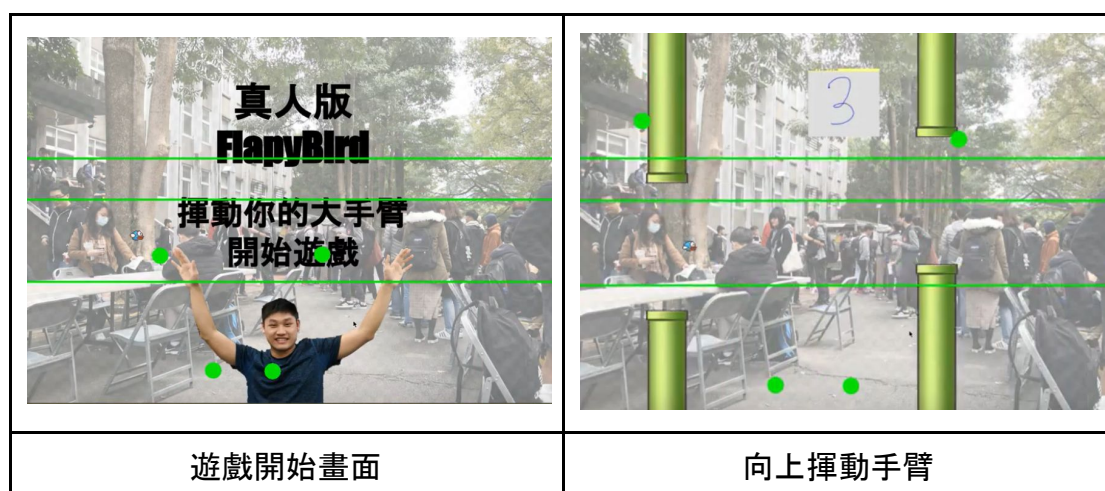
整體開發環境部分，我們使用 Python 作為開發語言，在環境為 32 bit 2.7 下執行程式，並在 Windows 系統下測試、開發。而程式部分我們則大致上分為兩部分：遊戲介面開發、讀取 Kinect 資訊並進行演算法設計

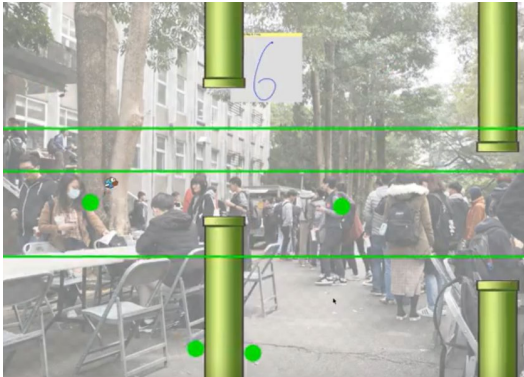
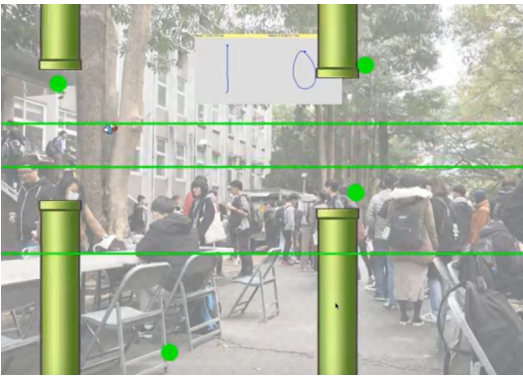
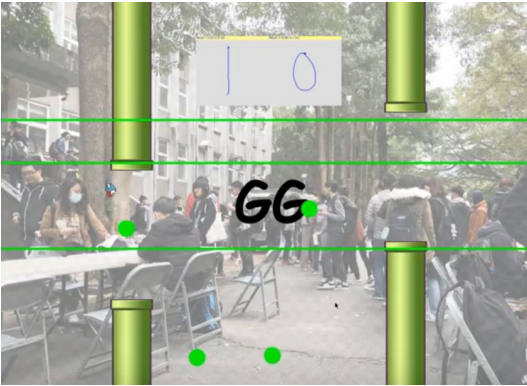
在「遊戲介面」部分，我們利用 PyGame 套件讓遊戲畫面轉場，不僅更新畫面使介面不斷出現水管，同時也利用此套件讓小鳥出現飛上、飛下及墜落等效果，並在界面上更新計分板分數。

而在「讀取 Kinect 資訊並進行演算法設計」部分，我們利用 Microsoft 釋出的 Python Tool Product -- Pykinect 套件來讀取相關要用的資訊，包括左右手腕、肚臍以及雙腳的骨架資訊。其中，我們會判斷前後 Frame 的雙手位置是否高於肚臍垂直位置上方100px 處，以及雙手位置是否低於肚臍垂直位置，以此機制使小鳥往上飛，若無甩動則會使小鳥慢慢下降；此外，也會判斷玩家的腳是否有踢高（踢到約肚臍下方的位置）來觸發「救命」機制，但同時也會使「踢腳」的使用次數減少，共五次機會。

整體來說，在開發中我們對演算法進行多次改良，對「如何資料儲存」設計了一番時間，為的就是能用最簡單的方法將讀近來的資料進行運算，並同時試著將遊戲設計成多人模式，讓大家可以一起同樂，但受迫於其他因素（下下點進行探討），使得多人模式沒有成功運作。此外，我們過程中我們也試了不同更新速度的遊戲介面、鳥的飛行速度...等動畫元素，最大原因就在於我們要找出配合「PyKinect」讀取資料的步驟以及其更新每幀的時間，讓體驗更加流暢。

4) 介面及成果展示



| | |
|---|--|
|  |  |
| <p>向下揮動手臂</p> | <p>抬腳瞬移</p> |
|  | |
| <p>遊戲結束</p> | |

5) 問題與討論

在開發過程中，我們預期加入「多位玩家」、「語音啟動遊戲」、「結合深度資訊的遊戲功能」，但在實作過程遇到一些問題，除了有技術上的困難之外，在「結合遊戲功能（實務上體驗）部分」也在嘗試後發現不太可行，以下將各別檢討：

- 多位玩家：在實作部分，Pykinect 會一次讀取六個人的骨架資訊，雖然可以用遊戲規則使得我們去追蹤哪些人在玩，但是在遊戲過程中，玩家並不會固定在單一一個點，若玩家有稍稍離開 Kinect 視線，則將追蹤不到其骨架，雖然還是同一個人，但 Kinect 依舊會視其為「其他人」。因此很難正確地追蹤。

若使用 Kinect V2 版本，則可以利用 Face 追蹤 ID，以解決此問題。

- b) 語音啟動遊戲：此部分我們曾經試過加入遊戲中，用 Pykinect Audio 套件讀取 Recongizer 所需的 XML 檔案，也參考此教學[註e]。但實作上，雖然有支援英文，在遊戲中喊出「Start」，並不一定能順利辨識出來，可能是 Recongizer 所讀取 XML 檔案沒有設定好、發音不標準、環境的因素，或是接近學期末沒有仔細研究好 Pykinect 中 Audio 套件的程式運作...等的關係，使得最終效果不理想。在嘗試之後，決定將此功能移除，選擇先讓遊戲品質提升，將此目標轉為次要目標。或許未來可以深度了解 Pykinect Audio 的運作（包括程式碼及設定檔），來讓整個辨識結果更好。

- c) 結合深度資訊的遊戲功能：

原本欲將深度資訊與「鳥」的大小結合，意即站得比較前面的人鳥會比較大隻，反之亦然。但是後來發現若這樣設計遊戲，玩家玩起來會有些不方便。但最重要的一點是，要如何判斷一個人的深度是一個很大的問題，即使有了深度資訊，但各部位的資訊不一樣，此外我們也很難去抓「人在哪」，若只用骨架資訊會太過單一，進而出現其他體驗遊戲上的問題。因此最後則將此功能移除。

6) 總結

在這次作業中，我們學到如何用 Pygame 實做遊戲介面，並瞭解 Pygame 的運作，包括 Frame 的更新、圖片 Icon 的更新、如何實作動畫...等效果。此外，我們更熟知 PyKinect 是如何與 Kinect 之間進行溝通並獲取 Kinect 上的資訊，以找到我們所需的資料並做出一個遊戲體驗不錯的成品。

7) 程式碼及參考資料：

a) 原始碼 :

<https://drive.google.com/file/d/1896cXnSdBSQ73Ehb3CJKgslqK6i1Jyzj/view?usp=sharing>

b) Pygame : <https://www.pygame.org/wiki/GettingStarted>

c) Pykinect :

i) <https://github.com/microsoft/PTVS/wiki/PyKinect>

ii) <https://github.com/microsoft/PTVS/tree/master/Python/Product/PyKinect>

d) Flappy Bird : <https://github.com/sourabhv/FlapPyBird>

e) 定義自訂辨識式 :

<https://docs.microsoft.com/zh-tw/windows/uwp/design/input/define-custom-recognition-constraints>