

擴增實境音樂節奏遊戲

Augmented reality music rhythm game

學生：蘇美玉、葉昱廷、張凱閔、張瑞哲

指導教授：辛錫進

國立聯合大學 資訊工程學系

苗栗市南勢里聯大 2 號

{U0721035,U0723027,U0723033,U0824020}@o365.nuu.edu.tw

摘要

在現今科技普及，人手一機的時代，許多人平時就只坐在椅子上，使用網路娛樂、上班和人際交流，越來越少有戶外活動。我們在遊玩手機音樂節奏遊戲時，想到一些和 AR 有關的遊戲，可以使人站起來活動活動，而不是只是做坐著遊玩，所以產生了如果把 AR 技術和音樂節奏遊戲結合後會變成什麼樣的遊戲。希望能使人站起來遊玩遊戲，而非坐著。

關鍵詞：AR、音樂節奏遊戲

Abstract

Nowadays, with the growing number of high-tech users, and the trend of personal electronic device for each individual person, most people often just sit in the chair (or on the couch) to surf the Internet for entertaining, working and/or communicating with others. Thus, there are fewer and fewer outdoor activities in our daily lives. It would be better if we can play mobile musical games and at the same time are willing to stand up for indulging ourselves with AR related activities rather than just sitting and playing. This project aims to develop musical games with AR techniques to help people want to stand up and actively paly instead of just sitting and merely hand moving.

一、前言

1.1 研究背景與動機

高擴增實境(Augmented Reality)是透過各種顯示設備，利用虛擬物與現實結合，可用來傳遞訊息和與使用者即時互動。擴增實境有許多方面應用，在娛樂方面目前較常見的有 AR 卡牌、AR 書籍、AR 桌遊和繪本、變臉程式等等。在現今的擴增實境市場因為疫情，對於網路的需求日益趨多，所以有越來越多的應用程式與 AR 結合，使虛擬世界與現實世界更好的交互體驗。

AR 音樂節奏遊戲是一種利用增強現實技術開發的遊戲類型。增強現實技術可以在真實世界中把虛擬元素放置到實際環境中，讓玩家可以與這些虛擬元素互動。在音樂遊戲中，這種技術可以用來創造以音樂為基礎的遊戲體驗，比如讓玩家在跳舞或演奏樂器時移動虛擬物體。

1.2 研究目的

本研究目的在於如何將 AR 融入遊戲本身，並嘗試 AR 和遊戲本身融合後，是否有延遲和準確率。

並利用這些訊息更好的將遊戲更好的兼容和流暢率，使人們遊玩時可以活動身軀，而減少人們一直坐在椅子上的時間，使在遊玩時而非沒有運動到，促進一天內的活動時間，增加活動量。

二、內容與架構

2.1 遊戲架構圖

整體遊戲架構流程圖(如圖 1)，首先是遊戲初始畫面，點擊後進到遊戲主畫面，你可以選擇登入、註冊或玩免登入小遊戲，登入後進入遊戲選單畫面，選擇想要玩的歌曲後，進入遊戲畫面，結束後進入遊戲結算畫面，之後回遊戲選單畫面。

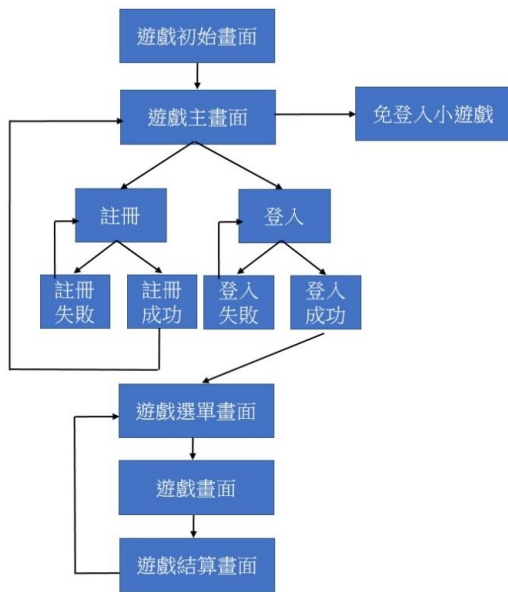


圖 1 遊戲流程架構圖

2.2 開發環境

開發環境如表一，主要使用 Unity 製作遊戲，像是遊戲介面、遊玩指令等等；vscode 和 visual studio 2022 都是寫 C# 的平台，寫 Unity 的腳本和後台指令，用以使 Unity 介面可以因為指令更換介面和動作；Spyder 為撰寫 python 的平台。C# 和 python 利用 socket 套件傳遞資料，以 python 為 server 端、C# 為 client 端，並將 server 端傳送資料到 client 端；資料庫則是使用 mysql，以 php 連接，將遊玩資料傳送到資料庫。opencv 和 tf-pose-estimation 是在使用 HumanPoseEstimation 會用到的套件，並以 python 撰寫程式碼。LMMS 是 LMMS 數字音訊工作站，主要通過製作旋律和節拍，

安排音訊片段，合成和混合音訊等方法創作音樂。

使用軟體	程式	套件	資料庫
Unity	C#		
vscode		php	mysql
visual studio 2022		socket	
Spyder	python	opencv	
		tf-pose-estimation	
LMMS			

表 1 開發環境

三、專題理論部分

3.1 手機端使用者功能

姿勢辨識的模型主要使用 tf-pose-estimation，此模型是利用 tensorflow 深度學習將人體辨識成不同節點(如圖 2[1])，並使用 OpenCV 開啟鏡頭。

tf-pose-estimation 主要基於 VGG 預訓練網絡和 caffe 格式的權重轉換，且基於基於 mobilenet 論文，使用 12 個卷積層作為特徵提取層，將人體分成 18 個節點。

tensorflow 是以不同的程式語言建立計算圖，再自行依需求設計張量運算流程，並透過訓練建構出深度學習的模型，最後即可在用戶端和執行裝置間建立連結，在不同平台上執行完成好的計算圖，流程(如圖 3 [2])。

OpenCV 是個開源電腦視覺函示庫，由 Intel 公司發起並參與開發，目的為針對電腦視覺技術，建構開源的程式庫，以提供學術界或產業界使用。在第三方開發者的協助下，目前 OpenCV 以內建許多好用的功能模組[3]。

在 tf-pose-estimation 節點會依人的動作不同而有不同辨識，但依目前的模型來說是有延遲而非毫無延遲。



圖 2 [1] 人體辨識

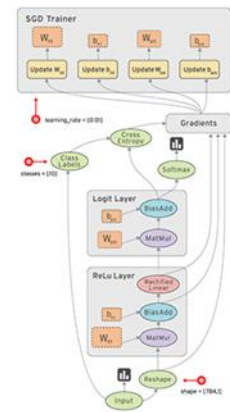


圖 3[2] 計算流程圖

四、專題實作部分

4.1 介面 UI

這使用 UGUI 的 UI 系統，做出各種介面，並用 vscode 撰寫 C# 程式，讓這些 UI 元件可以導向你想要的地方。

遊戲開始畫面(如圖 4):利用 Unity 內建功能 Animation 點擊中心製作會動標題後，進入遊戲主畫面

遊戲主畫面(如圖 5):使用按鍵導向到註冊、登入、免登入遊戲。

註冊介面:註冊按鈕按下後通過 C# 連往 php，而 php 將資料傳到資料庫。

登入介面:登入按鈕按下後通過 C# 連往 php，php 向資料庫提取資料並進行資料

核對，最後將結果傳到 C#。

遊戲選單畫面(如圖 6):畫面剛開始會出現歌曲一的名稱按左箭頭變歌曲二的名稱，再按一次變歌曲三的名稱，再按一次變回歌曲一會循環，按右箭頭則相反;

遊戲結算畫面(如圖 7):出現最後遊戲分數並可返回遊戲選單畫面。



圖 4 遊戲開始畫面



圖 5 遊戲主畫面

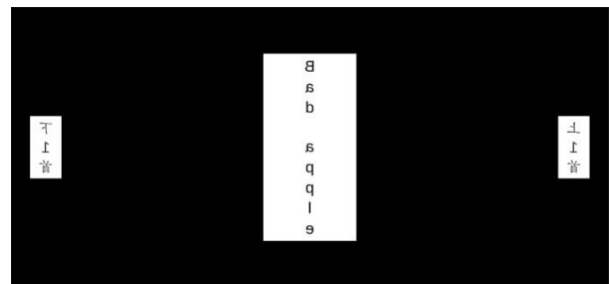


圖 6 遊戲選單畫面



圖 7 遊戲結算

4.2 遊戲介面(音樂譜面)

免登入遊戲介面(如圖 8[4]):為下落式音樂遊戲，按落下到線的對應的方向鍵，則會進行計分與連擊數，落下的方向鍵是由 C#去讀取 mdi 檔產生，而 mdi 檔是在 LMMS 數字音訊工作站(如圖 9)進行聽音樂並自行打節奏，繪製一首歌之譜面，最後匯出 mdi 檔。

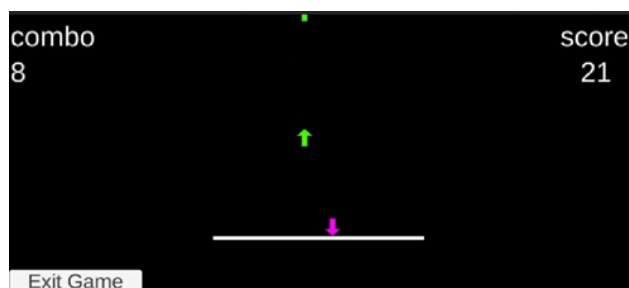


圖 8[4] 免登入遊戲介面



圖 9 LMMS 實作頁面

登入遊戲介面:進入遊戲後開始撥放指定音樂，跟隨著音樂節奏出現點擊圈圈(Hit Circle)和拖曳滑條(Slider)，當點擊圈圈出現時(如圖 10[5])，會有一個環由外到內收縮，當環與點擊圈圈重疊時點擊(如圖 11[5])，根據點擊時機判定分數，越接近越高，滑條則是必須從第一次點擊圈圈出現的地方(如圖 12[5])，跟著引導圈圈拖曳(如圖 13[5])到終點(如圖 14[5])的點擊圈圈，滑條分數為單點擊圈圈的兩倍，當你連續擊中時，會計算連擊(combo)數，擊中點擊圈圈連擊數會加一、滑條加二，連擊數會有額外加分，並隨著連擊數越高加越多，最後等音樂結束時，進入結算介面，會記錄最終分數跟以往的分數作比較。

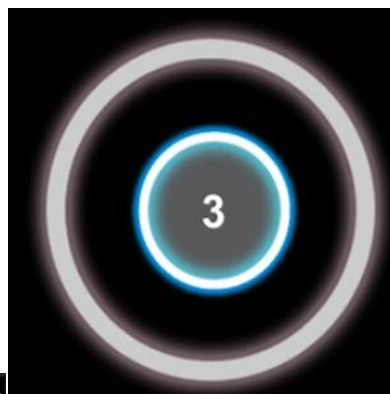


圖 10[5] 點擊圓圈

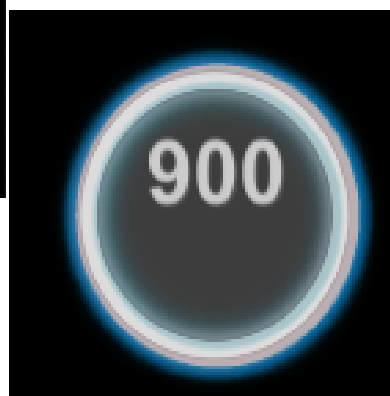


圖 11[5] 圓圈重疊點擊

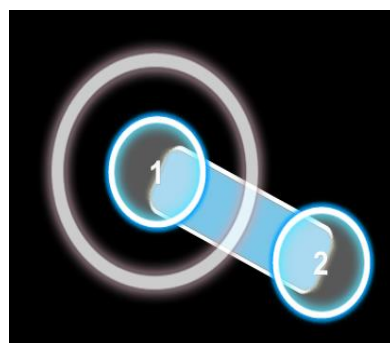


圖 12[5] 圓圈出現的地方

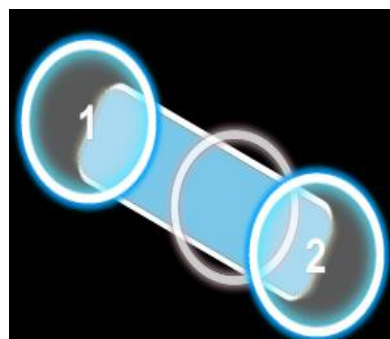


圖 13[5] 圓圈拖曳

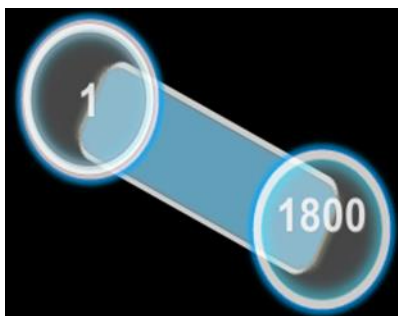


圖 14[5] 圓圈拖曳終點

4.3 遊戲介面(音樂譜面)

此遊戲主要使用 Unity 製作，所以在 AR 辨識方面使用 HumanPoseEstimation，HumanPoseEstimation[6]是基於 tf-pose-estimation 上增加到可在 Unity 上使用，利用 socket 套件將 Python 的資料傳給 C#，socket 是利用 ip 和 port 連線，並使用 TCP 或 UDP 傳輸協定，而在此研究中我們使用的是 TCP 傳輸協定。

Python 有連接到鏡頭並利用開啟 socket 的 server 端(如圖 15)，同時將資料傳輸到 server 裡，同時等待有沒有 client 端連接，此研究使用的 client 端是 Unity 腳本，因為我們需要把資料傳進遊戲本身。當 client 端連接上 server 端，利用 client 端將 server 端裡的資料讀去過來，在進行資料判讀。

```

[2022-12-12 12:31:15,609] [TfPoseEstimator-WebCam]
[DEBUG] image process+
[2022-12-12 12:31:16,258] [TfPoseEstimator-WebCam]
[DEBUG] postprocess+
[2022-12-12 12:31:16,260] [TfPoseEstimator-WebCam]
[DEBUG] (0: (534, 330), 14: (520, 322), 15: (536,
322), 16: (434, 336))
[2022-12-12 12:31:16,300] [TfPoseEstimator-WebCam]
[DEBUG] image process+
[2022-12-12 12:31:16,990] [TfPoseEstimator-WebCam]
[DEBUG] postprocess+
[2022-12-12 12:31:16,992] [TfPoseEstimator-WebCam]
[DEBUG] (0: (524, 330), 14: (508, 316), 15: (538,
320), 16: (420, 328))
[2022-12-12 12:31:17,038] [TfPoseEstimator-WebCam]
[DEBUG] image process+
[2022-12-12 12:31:17,668] [TfPoseEstimator-WebCam]
[DEBUG] postprocess+
[2022-12-12 12:31:17,669] [TfPoseEstimator-WebCam]
[DEBUG] {}
[2022-12-12 12:31:17,710] [TfPoseEstimator-WebCam]
[DEBUG] image process+
[2022-12-12 12:31:18,482] [TfPoseEstimator-WebCam]
[DEBUG] postprocess+
[2022-12-12 12:31:18,483] [TfPoseEstimator-WebCam]
[DEBUG] (0: (530, 338), 14: (518, 328), 15: (532,
322), 16: (438, 334))
[2022-12-12 12:31:18,442] [TfPoseEstimator-WebCam]
[DEBUG] image process+
[2022-12-12 12:31:19,889] [TfPoseEstimator-WebCam]
[DEBUG] postprocess+
[2022-12-12 12:31:19,890] [TfPoseEstimator-WebCam]
[DEBUG] {}

```

圖 15 socket 的 server 端

在 HumanPoseEstimation 中主要使用 tf-pose-estimation，所以在使用時為了不讓一個.py 檔不要太多程式碼，會將檔案分開，使程式碼可以更加容易地被使用者讀取和修改。

HumanPoseEstimation 中影像辨識(如圖 16)。節點各分為 16 點，節點代表位置(如圖 17[7])。而在此遊戲中，主要使用 4 和 7 兩點，在此模型中 4 和 7 的代表位置為手臂和手掌的連接點，所以在遊玩時以手腕為主。

在遊戲中我們只會使用到手腕的位置，只讀取 4 和 7 兩節點，所以在辨識中雖然會有 16 節點，但會將 4 和 7 特別讀取，進行判讀。

在遊玩時可以由兩人以上遊玩，但遊玩模式皆為一人，在遊玩時盡量不要有重疊或背景有很多人，因為會造成 AI 辨別出現問題，可能會導致判斷錯誤進而影響遊戲辨別失誤。



圖 16 HumanPoseEstimation 影像辨識

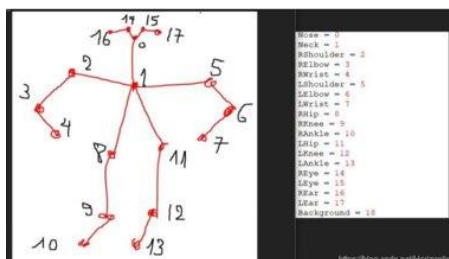


圖 17[7] 節點位置

在此研究中，在遊戲點擊時，程式碼部分是利用 ScreenToViewportPoint、WorldToScreenPoint API 轉換成 viewpoint，virwpoint 在 Unity 相機中的左下角相對位置是(0,0)，右上角則為(1,1)，轉換結果第三行和第四行的fourpoint 和 sevenpoint(如圖 18)。



圖 18 轉換結果

在整合遊戲譜面和 AR 部分時，會出現兩個 C#檔，須將 AR 部分(如圖 19)的資料傳輸到遊戲譜面的 cs 檔(如圖 20)，所以我們使用 txt 檔傳輸資料並讀取，寫入 txt 檔 (如圖 21)，讀取 txt 檔(如圖 22)。

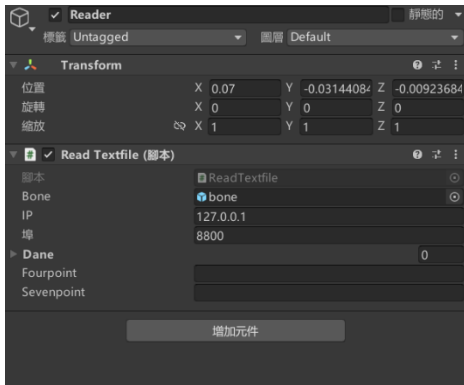


圖 19 AR 部分



圖 20 遊戲譜面的 cs 檔

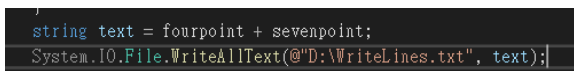


圖 21 寫 txt 檔

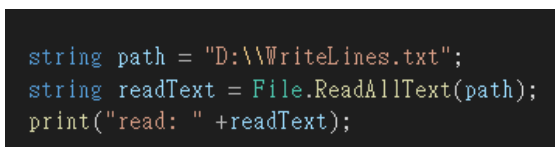


圖 22 讀取 txt 檔

在讀取完 txt 檔資料後，在遊戲過程中是直接判斷和即時讀取，txt 檔裡的資料會因為 HumanPoseEstimation 的變動而產生不同資料，同時進行比對，進而去產生分數

和連擊數。

在遊戲進行過程中，會因為電腦的 GPU、CPU、所剩記憶體和網路速度影響，而使遊戲的延遲速度有不同的影響，造成現實中和遊戲中的連結不一，造成資料的比對失敗。

4.4 資料庫

資料庫為儲存帳號密碼及遊玩分數，接收由 php 傳來之資料並儲存，使下一次使用者在登入之後可以讀取遊玩資料，而不用重玩一遍。

在資料庫中我們會以 id、username、hash、salt 儲存使用者基本資訊(如圖 23)。id 為後台自動產生代表碼，username 為使用者名稱，而 hash 為密碼 crypt 亂數編碼，此數由 salt crypt 隨機碼加上密碼而成，使得密碼安全性提高。



圖 23 使用者基本資訊

使用 username 連接 userscore 資料(如圖 24)。將每首歌的最高分數存入不同特定欄位(如圖 25)，以便增加下次遊玩此遊戲時判斷是否超過最高分數。



圖 24 username 連接 userscore 資料

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
1	username	varchar(16)	utf8mb4_general_ci	否	非				修改 刪除 更多
2	score1	int(10)		否	0				修改 刪除 更多
3	score2	int(10)		否	0				修改 刪除 更多
4	score3	int(10)		否	0				修改 刪除 更多

圖 25 每首歌的最高分數存入不同特定欄位

五、結論

在本研究中，使用 HumanPoseEstimation 進行 AR 的辨識，並利用 socket 連接 Unity 腳本，將原先在 Python 上執行資料傳輸到 Unity 腳本。使用兩個 cs 檔傳輸資料到主要遊戲介面。

在研究過程中，因為遊戲過程需要站起來活動，所以在撰寫程式中必須一直站起來或活動測試，所以無論是在遊戲過程和撰寫程式中都會像前言所說一般，會增加一天的活動量，而非一直坐在椅子上。所以在遊玩過程中的確是會增加使用者的活動次數，雖然遊戲本身需要的活動量需求較少，但也會減少一天坐在椅子上的時間。

在此研究中，因為 AR 辨識有一定的失誤率和判別失敗，所以在遊玩過程中可能會因為各種因素而造成無法點擊到或判別失敗，在未來我們希望在 AR 辨識部分可以更加地將準確率提升，將對遊戲的影響降得更低。在遊戲內容方面，我們現在的歌曲量因為 AR 辨識的部分，所以在歌曲難度上做了取捨，在未來會希望在模型準確率提升後可以將歌曲難度調高，而非將所有的歌曲難度都在差不多的水平線上。

參考文獻

[1] ZheC/Tf-Pose-Estimation. (2018, February 1).
<https://github.com/ZheC/tf-pose-estimation>

- [2] Jackjack.(2020,September 27)..Day 12|什麼是 TensorFlow (三) .
<https://ithelp.ithome.com.tw/articles/10245369>
- [3] Damienkk.(2022,September9).[Day 2]OpenCV 介紹
<https://ithelp.ithome.com.tw/articles/10288910>
- [4] Skyansam.(2021,July17). RhythmGameTutorialProject.GitHub
<https://github.com/SkyanSam/RhythmGameTutorialProject>
- [5] osu!造型
<https://osuskins.net/>
- [6] Mertcanozturk.(2020,August 11). Mertcanozturk/HumanPoseEstimationAndUnityIntegration.
<https://github.com/mertcanozturk/HumanPoseEstimationAndUnityIntegration>
- [7] Tf-Openpose 得到關節點的座標 (tf-Openpose-Estimation) . (2022, April 21).
<https://blog.csdn.net/Horizonhui/article/details/89243449>