

Virtual Reality Final Project Report

Racing Game with Gesture Recognition

R06922065 郭芷玲

R06922097 鄭雅文

B03902037 陳郁棋

ABSTRACT

利用 Unity 搭建類似跑跑卡丁車的遊戲場景，包含跑道、指示牌、各種障礙物以及不同特效的驚喜箱，並將角色從原本的汽車改為摩托車，控制玩家角色移動的方式從原本的電腦鍵盤改為 Leap Motion 偵測到的玩家手勢。我們設定玩家有加速、減速、右轉、左轉等多種手勢組合，而玩家的目的是要讓摩托車通過我們設計的種種考驗，成功繞一圈回到終點線。藉由這些手勢的使用，讓玩家能夠更融入遊戲場景，體驗遊戲中的各種樂趣。

INTRODUCTION

本次作業主要是利用 Unity 以及 Leap Motion 裝置來完成一款老少咸宜且有趣好玩的遊戲。從大家耳熟能詳的跑跑卡丁車遊戲為出發點，我們利用 Unity 繪製了跑道場景並設立了指示方向的告示牌以及多種障礙物，例如：人行道、樹叢、噴水池、上下坡、其他路上的車輛、火箭、飛機、竹籬笆、會走動的人、不會動的人等等，並在行經道路多處放有驚喜箱，一旦玩家碰到就會有暫停、原地旋轉、跳躍、遮蔽視野、加速等等的效果，替遊戲增添了趣味性。

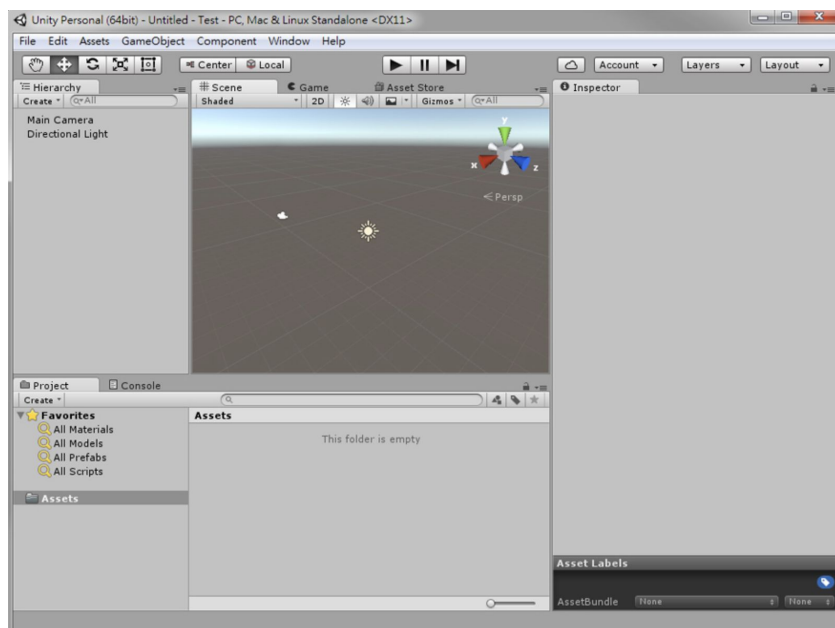
原本設定遊戲角色的移動方向是由電腦鍵盤的上下左右來控制，後來為增加遊戲的特色以及新穎性，我們決定結合 Leap Motion 偵測手勢的功能，將角色移動方向改為由使用者的各種手勢來控制，並且我們將原本玩家角色的跑跑卡丁車改為摩托車，期望能結合日常生活中大家騎乘摩托車時對龍頭握把的各種手勢，藉由玩家在 Leap Motion 前比畫的手勢，來控制摩托車的前進、後退、左轉、右轉，更能增加玩家身臨其境的感覺。其中我們設定玩家右手握拳代表加速、右手張開代表減速、左手握拳代表右轉、左手張開代表左轉。

以下來簡單介紹一下我們這次作業中所會用到的 Unity 軟體、Leap Motion 裝置。

● Unity

Unity 是一套跨平台的遊戲引擎於 2012 年 11 月 15 日推出，隔一段時間會發布新版本，可開發執行於 PC、Mac OS 單機遊戲，或是 iOS、Android 手機或平板電腦的遊戲。Unity 也可開發線上遊戲，只需在網頁瀏覽器安裝外掛程式後即可執行。Unity 也可

用於開發 PS3、XBox360、Wii 遊戲主機上的遊戲。圖一展示了 Unity 的操作介面，由多個不同功能的視窗組成，使用者可以調整視窗的大小、位置以及版面配置。



[圖一] Unity 操作介面

- **Leap Motion**

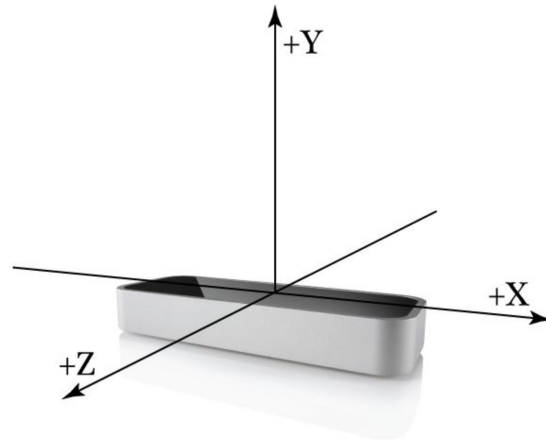
Leap Motion 裝置利用手掌和手指動作來進行輸入，無需手部接觸或者輕觸。它使用進階的動作感應專利技術進行人機互動，Leap Motion 裝置實體如圖二所示。



[圖二] Leap Motion 裝置實體

Leap Motion 可以清楚地分辨五隻手指頭、關節、手掌、手正反面，辨識的原理為 Leap Motion 根據內置的兩個攝像頭從不同角度捕捉的畫面，重建出手掌在真實世界三維空間的運動信息。其檢測的範圍大約在裝置上方 25 毫米到 600 毫米之間，檢測的空間大約為一個倒四棱錐體。首先，Leap Motion 會建立一個直角座標系，座標的原點是裝置的

中心，座標的 X 軸平行於裝置，指向屏幕右方；Y 軸指向上方；Z 軸指向背離屏幕的方向，單位毫米，相對的方位如圖三所示。在使用過程中，Leap Motion 會定期的發送關於手的運動信息，每份這樣的信息稱為一個 frame，而 frame 所包含的資訊有：所有手掌、手指的列表和信息以及手持工具（細的、筆直的、比手指長的東西，例如一枝筆）的列表及信息。Leap Motion 可以根據每個 frame 對比前一個 frame 所檢測到的數據，產生物體運動的訊息，並將其展示在電腦螢幕上，如圖四所示。



[圖三] Leap Motion 直角坐標系相對位置示意圖



[圖四] Leap Motion 實際偵測手勢的結果

IMPLEMENTATION

● 遊戲場景設計

在遊戲一開始時玩家需按下畫面中央的「PLAY」按鈕來開始遊戲，如圖五所示。進入到遊戲畫面後，畫面中間我們有設定三秒的時間讓玩家做遊戲開始的準備，並在畫面正中央以黃色文字顯示三秒的倒數計時，如圖六所示。在畫面左上角顯示玩家此次遊戲的

闖關時間，左下角的小螢幕為 Leap Motion 實際偵測到手勢的示意圖，此部分如圖七所示。



[圖五] 玩家按下 PLAY 開始遊戲



[圖六] 畫面中間倒數計時開始遊戲



[圖七] 左上角為遊戲時間。左下角為 Leap Motion 實際偵測到的手勢。前方場景為上坡，玩家需加到足夠的速度才能成功到達斜坡頂端。

除了一般跑道的搭建，在一些跑道的旁邊我們設計了人行道，玩家的摩托車角色將無法騎上人行道，而會被人行道所阻擋，這個設計使遊戲更貼近現實生活中騎乘摩托車的場景。除了平面跑道的搭建，我們也設計了一些上下坡的路徑並且考慮真實世界中的地心引力，玩家需加到足夠速度才能成功到達斜坡頂端，下坡時也需要注意為加速所導致的暴衝，場景如圖七所示。我們也設計了指示方向的告示牌以及許多障礙物，例如：樹叢、噴水池、其他路上的車輛、火箭、飛機、竹籬笆、會走動的人、不會動的人等等，如圖八到圖十所示。



[圖八] 跑道中有其他車輛當作障礙物



[圖九] 指示方向告示牌。放置不會動的人當作障礙物。
道路中不特定放置驚喜箱。



[圖十] 畫面中的人物在遊戲過程中會不斷的走動，玩家需巧妙地閃過以避免撞上。

除了上述障礙物的設置，我們另外還有在道路多處放有驚喜箱，一旦玩家碰到就會有暫停、原地旋轉、跳躍、模糊視野、加速等等效果，增添遊戲的趣味性，驚喜箱示意圖如圖九所示。最後當玩家成功繞行跑到一圈抵達最後的終點（也是最初的起點）時，畫面中間會顯示「Finish!」代表遊戲結束並成功完成遊戲，如圖十一。從圖十二遊戲俯瞰圖中可以看出我們設計的跑道路線，圖中右側黑白相間的區域為遊戲起點同時也是終點。



[圖十一] 玩家成功抵達終點



[圖十二] 遊戲場景俯瞰圖

● Leap Motion 手勢選擇：Extended Finger Detector

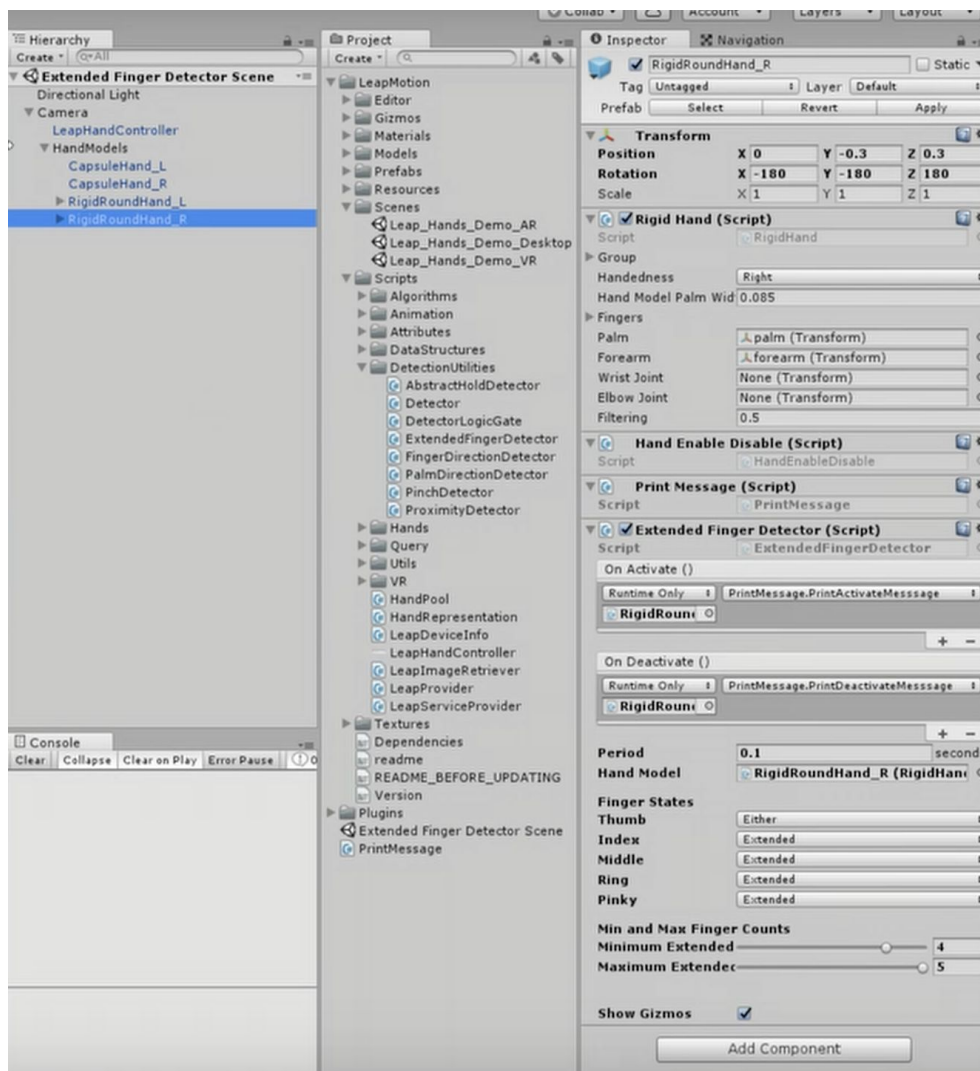
原先理想的手勢選擇為類似現實生活中騎乘機車握住握把的手勢，並利用手腕的下壓代表加速、向上代表減速以及雙手手腕的向左或向右來控制角色方向，然而在找尋網路上有關 Leap Motion 偵測手勢的相關介紹之後，礙於 Leap Motion 本身對於手腕的偵測資訊較少，而對手掌和手指張開、握起的資訊以及教學較多，也較容易偵測，因此我們決定更改我們原先對於手勢的設定，利用 Leap Motion 內建偵測手掌張開、握起的 Extended Finger Detector，將右手握拳設定成加速、右手張開設定為減速、左手握拳設定為右轉、左手張開設定左轉。

若玩家將左手從 Leap Motion 上方移開，那麼遊戲中的角色將會維持等速前進或是後退，然而因為 Leap Motion 本身硬體的限制，當玩家的手再次放回裝置上面的時候，往往沒有辦法太快重新偵測到正確的手勢，而使得遊戲中的機車失控或是卡在障礙物之中，因此另一個使機車走直線的方法為玩家的左手等速地在 Leap Motion 上方張開及握起，平均控制機車向右及向左的分量。

● 連接 Unity 以及 Leap Motion 裝置

Unity 連接 Leap Motion 裝置後，在 Camera > HandModels > RigidRoundHand_L 和 RigidRoundHand_R 的 Inspector 加入 Extended Finger Detector

的 script，並將選擇相對應的 function。在下方的 Finger States 部分把 Index、Middle、Ring、Pinky 從原本的“Either”改為“Extended”，如此一來就能夠將 Leap Motion 所偵測到的手勢資訊成功地連接到 Unity，操作介面及結果如圖十三所示。



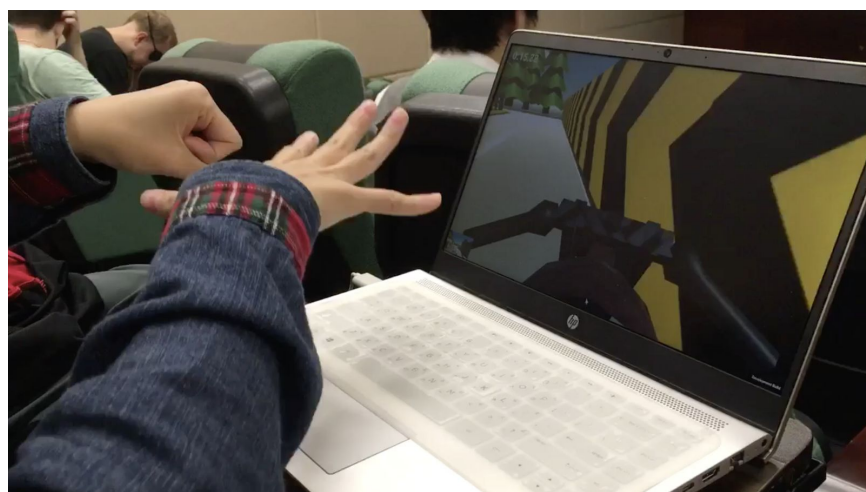
[圖十三] Unity Extended Finger Detector 操作示意圖

DEMO

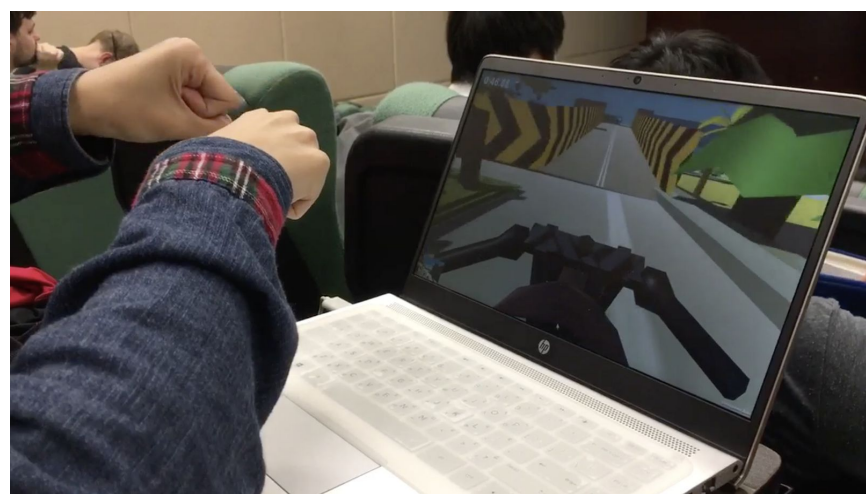
圖十四到圖十六分別為一位玩家實際玩遊戲的畫面，我們設定玩家右手握拳代表加速、右手張開代表減速、左手握拳代表右轉、左手張開代表左轉。因此圖十四中代表加速左轉，圖十五中代表減速右轉，圖十六中代表加速右轉。在摩托車被障礙物卡住而無法移動時，可以按電腦的 R 鍵重新開始遊戲。



[圖十四] 實際玩遊戲的場景。控制機車加速左轉。



[圖十五] 控制機車減速右轉



[圖十六] 控制機車加速右轉

DIFFICULTIES

Leap Motion 是我們最大的優勢同時也是劣勢，不論是手移動的幅度或是距離 Leap Motion 的遠近皆會受到裝置的限制，也容易受到干擾，在裝置感應手動作的上方，除了兩隻被感測的手之外，不能夠有其他任何物體，像是玩家的頭若太前面，位於手的上方，就會出現干擾導致辨識手的部分失敗。除此之外，Leap Motion 有時候也會偵測到錯誤的手勢，或者是需要一點緩衝的時間讓裝置辨識，無法很立即地反映在電腦螢幕上。

FUTURE WORK

這次作業有鑒於 Leap Motion 裝置偵測手勢的侷限，沒有辦法容易地按照現實生活中騎乘機車握住龍頭的手勢來模擬，未來可以對此裝置關於手勢的偵測做更進一步的研究，找到更加合適的動作，讓玩家更能夠輕鬆地玩遊戲，不用花太多心思在思考什麼動作是前後移動、什麼是控制左右轉。

再者，可以結合 Google VR Cardboard，將遊戲改成讓玩家藉由頭戴式裝置，對於遊戲場景能有更深入的感受。也可以增加多人一起競賽的機制，達成類似實際跑跑卡丁車的情境，若真的能夠達成並改善裝置偵測手勢的一些不足，我們遊戲的優勢將會是能夠讓多位玩家在賽車之餘，藉由 Leap Motion 裝置的連結讓玩家能夠身歷其境，達成 Virtual Reality 的技術以及目標。

REFERENCES

[1] Unity 的簡易介紹

<https://dotblogs.com.tw/unity/2018/01/08/unity>

[2] Leap Motion 原理

<https://www.zhihu.com/question/20252985>

[3] Leap Motion Unity Modules

<https://leapmotion.github.io/UnityModules/>

[4] Leap Motion Orion Tutorial in Unity: Extended Finger Detector

<https://www.youtube.com/watch?v=HDMHyMYNEkA&index=6&list=PLnTTrMDXCsLpjeAYUXdgcpcuGBWZSp7vyS>