

中華大學生物資訊學系系統開發專題報告

人工智慧賽車遊戲 App

Artificial Intelligence Racing Game

專題組員:杜孟軒、方彥翔、林成翰

專題編號:PROJ2019-BIOINFO-105002

指導老師:吳哲賢老師

1. 摘要

現代的人們幾乎人手一支手機，而手遊自然是大家都有可能接觸到的東西，於是我們有了製作一款手遊的構想，而這次目的就是學習設計一款簡易的手遊 APP，從而摸透一些基礎功能。

2. 簡介

遊戲產業一直是很蓬勃發展的。它在生活中隨處可見，最常見的就是電腦、手機、平板等等各式各樣的設備上，遊戲都佔有一席之地，與我們生活

密不可分，而隨著遊戲產業越來越發達，不少的投資者或玩家也更願意投資遊戲產業或購入遊戲產品來表示支持。

根據市調公司 Newzoo 研究顯示，2019 年全球手遊收入高達 685 億美元，占全球遊戲市場規模的 45%。Newzoo

預估，手遊市場規模在 2022 年將成長至 964 億美元，霸佔半數遊戲市場，是眾多遊戲業者都不能錯過的戰場，由此可見手遊市場的龐大

這次使用的 UNITY 也是在網路上

蠻多人推薦的平台，很多知名手遊也都是透過這個平台產出，經過討論之後決定做賽車類型的遊戲玩法

3. 專題進行方式

(一) 遊戲的模式與構想

一場賽車遊戲最重要的就是車子與軌道，缺一不可，然後我們決定的形式為跑完兩圈才算結束，並且會有另一名 AI 一起參與比賽，看誰最先到達終點，到達終點後也會有秒數作為參考。

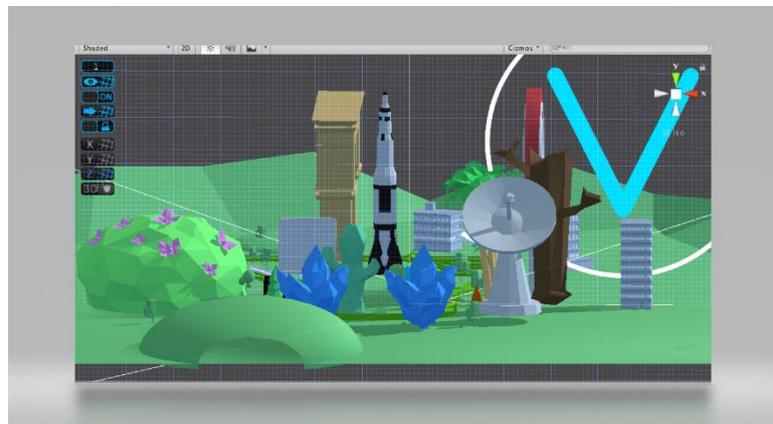
(二) 基本軌道與車手設置

賽車的設置就不多描述，基本的軌道就是一個大橢圓圈為基準，經過設計之後出來的成果如下圖。



(三) 場景佈置

除了賽車和軌道外，布置場景也是很重要的，讓玩家玩起來也比較賞心悅目，這邊的佈景裝飾物如下圖，中間的灰色方塊可以縮放所有佈置物，拖動每個 X, Y 和 Z 箭頭去縮放佈置物的大小。



(四) 終點線觸發設定

再層次結構上搜索 StartFinishLine 架設在場景中的終點線，然後在點選 GameObject > 3D Object > Cube，這就是觸發器設定，經過設定之後在經過終點線之後會產生五彩紙屑的特效

(五) 加速帶設置

我們有在軌道上放置一些加速帶如圖，



賽車經過此加速帶
會暫時得到速度的

提升，放置加速帶的主要目的是幫助

玩家在跟 AI 比賽時可以適時提升速度，

也增加獲勝機會。

(六) 碰撞彈力效果設置

在層次結構中找到 Kart 的 GameObject

隨後在裏頭找到 Bounce Effect

Prefabs 這個選項，將此效果拖到

KART 身上，就會有彈性的碰撞效果出

現。

(七) 賽車 AI 設置¹

這裡就是主要的部份，賽車 AI 的實現

有許多種方式，像是各式各樣的尋路

演算法，或是最近很流行的深度學習

等等，都是不錯的選擇，而我們最終

是決定使用尋路演算法中的

A*pathfinding algorithm。

接下來以下會說明兩者之間的差別以

及說明為何我們會選擇以

A*pathfinding 來做我們的 AI。

介紹：

● 深度學習：透過一代又一代的樣本

生產與篩選，找出合適的 Input。

● A*pathfinding：A*pathfinding

演算法，它通過計算公式 $F = G + H$

來計算最佳(短)路徑，其中，F 代表結

果總長、G 代表從起始點到此點的路

徑長、H 代表由此點到終點的直

線距離(不考慮障礙物)，此演算法的

好處在於因為它有一個評估函數在，

因此他會優先計算最適合的點，而不

需要檢查所有的點。

1

深度學習:

● 優點: 在一些複雜卻仍存在特定規則的遊戲來說，我們可以利用它來找出完美破解一個遊戲的 Input，以賽車遊戲來說，假設有特殊的機構或是地形(例如會讓賽車減速的泥巴地或是踩上去可以直接傳送的捷徑)，深度學習都可以把這些規則找出來進而加快完成賽事的時間。

● 缺點: 由深度學習訓練出來的 Input 一旦賽道改變就必需重新訓練一次，換言之如果有多個地圖就要一直重複訓練。

A*pathfinding:

● 優點: 不需要做太多的設置就可以適應多張地圖，並且 Unity 有提供優化過的 A*pathfinding 尋路功能。

● 缺點: 無法將除了路徑長度以外的

遊戲要素納入計算，所以如果遊戲要素過於複雜，則無法用這個演算法做出完美的 AI。

所以最終討論以及比較之後決定使用 A*pathfinding 作為人工智慧賽車 AI 的尋路功能。

(八) 賽程設置

由於設定上是兩圈才結算為一場比賽，所以我們在終點線設置一個 trigger，在透過腳本計算主角穿過 trigger 的次數，因為是兩圈為一場，所以總共是觸發兩次就結算並且放出秒數成績



(九) 設置虛擬鍵使其在手機運行

畢竟要讓它在手機裡跑，我們設置了

上下左右鍵，也就是按下按鈕會呼叫角色的一個腳本內的函數，而該函數負責控制角色的移動。

。



(十) 輸出成手機 APK 檔案

首先下載好 SDK 還有 JDK 檔之後，就可以輸出成 Android apk，這時就可以先把檔案移進手機裡面試玩了。

4. 主要成果

這次的 APP 總和成果為賽車遊戲配上人工智慧 AI，遊戲玩法為和 AI 比賽誰最先到達終點完成比賽，初始介面可看見遊戲分為三個難度，簡單-適中-困難，難度越高 AI 的速度也就

越快，賽道中也有設置加速帶幫助玩家更順利進行比賽，比賽結束後會及時跳出秒數畫面可檢視自己的成績。

5. 評估與展望

遊戲的發展是沒有範圍的，我們畢竟是這次專題才開始接觸遊戲 APP 的設計，在知識資本人力時間都有限的情況下，跟外面大型遊戲的公司作品來比，自然是小巫見大巫，所以這次我們設計這個人工智慧賽車 APP 的共同理念就是，雖然遊戲內容及外表看似簡單，但至少要讓它是一款能夠完整遊玩的遊戲，而不是玩到一半毫無頭緒沒有任何結果，最後也希望遊玩此遊戲的人能順利打敗 AI，通過我們設置的三個難度。也希望未來有機會的話，我們能學習到更多設計遊戲 APP 的知識。

6. 結語

這次的專題其實一開始對我們來說困難重重，雖然在之前有學過一些基礎的程式設計，但是遊戲 APP 的設計我們從來沒有接觸過，所以光是查資料的前置動作就花了不少時間，這次選擇了 UNITY 除了它的內建功能很齊全之外，他的教學網站心得也不少，這對沒有任何基礎的我們來說是一大幫助。而在我們決定設計人工賽車遊戲 APP 之前，我們也換過好幾次遊戲主題，都是一開始構想很多，實際開始設計之後才發現問題重重，並不是那麼容易可以去執行的，好在最後也有定案下來，而在開始設計 APP 之後，才發現在手遊裡很多看似簡單的動作，背後實際上卻是很多的設計與步驟才能完成，也才了解到設計一款遊戲 APP 並不是那麼容易的事，背後龐大的團隊所花的時間人力與資金，不是

一般人可以去想像的，最後順利完成了作品，也真的是學到了不少經驗。

7. 銘謝

感謝吳哲賢老師在製作專題時的各種教導，每週按時推我們進度，在我們猶豫不決不知道該如何定主題的時候給我們意見，也感謝各位組員的配合，在上課打工練校隊之餘還要抽時間來做專題，真的辛苦大家也辛苦老師了。

8. 參考文獻

- 1.A*(star)Pathfinding:
<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>
- 2.UNITY 學習網：
<https://unity.com/learn>
- 3.A*(star)Pathfinding：

<https://riptutorial.com/zh->

<TW/algorithm/topic/8787/a-->

[%E5%B0%8B%E8%B7%AF%E7%AE](#)

[%97%E6%B3%95](#)

4.UNITY 學 習 筆 記 6 :

<https://kendevlog.wordpress.com/2017/0>

<9/24/unity%E5%AD%B8%E7%BF%92>

[%E7%AD%86%E8%A8%986-prefab-](#)

[%E4%BD%BF%E7%94%A8%E9%A0](#)

[%88%E7%9F%A5/](#)