

# Unity 遊戲專題實作-動作冒險類型遊戲

指導老師: 陳兆南 博士

組員:106021381 蘇家緯 106021016 呂永環 106021149 陳彥龍

## 摘要:

我們這組畢業專題是使用 Unity 2019.3.15 版本來進行遊戲設計，而我們本次的製作的內容是 3D 的第三人稱視角射擊遊戲，我們想往未來感的方向去製作，因為人類逐漸在邁向高科技化，不久的將來我們就會需要依靠到機器人來幫助我們，所以我們就設計了用機器人來對付怪物，可能哪天地球面臨到危險就需要靠機器人幫助我們，因此我們以這個方向來研究製作。

製作的內容包括了使用了 2D UI 製作我們進入遊戲前的界面設計，玩家角色的動作主要從 Unity 中的 Animator 設計，運用人家已經將骨架設定完成後或者是自己運用 Unity 賦予模型骨架再去商店找共用的動作加上一些簡易的程式和給予動作體剛體和碰撞器之後我們就可以讓模型動起來並且和場景作出互動，在我們的專題則是有建築和傳送門等等都可這些以透過程式完成碰撞事件，座標改變，還有設計敵人的對玩家做出一些互動反應，像是敵人會朝著玩家的方向前進，一定距離內會攻擊玩家，或作出一些反應。

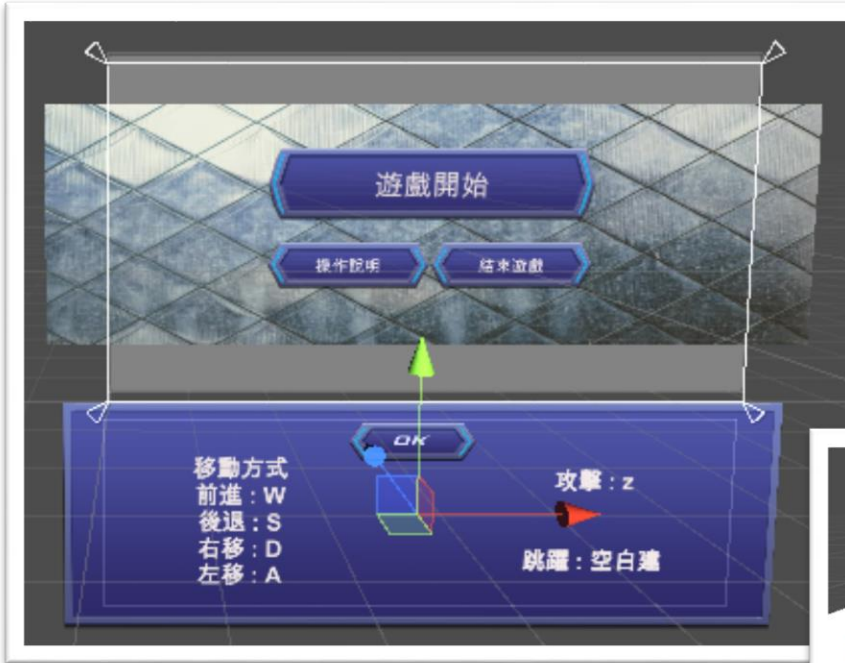
每個人對遊戲的設計概念都不同，也從網路上面看到大家不同的想法，同一種東西其實就有 3.4 種作法，可以製作出屬於自己的冒險故事。

## 前言:

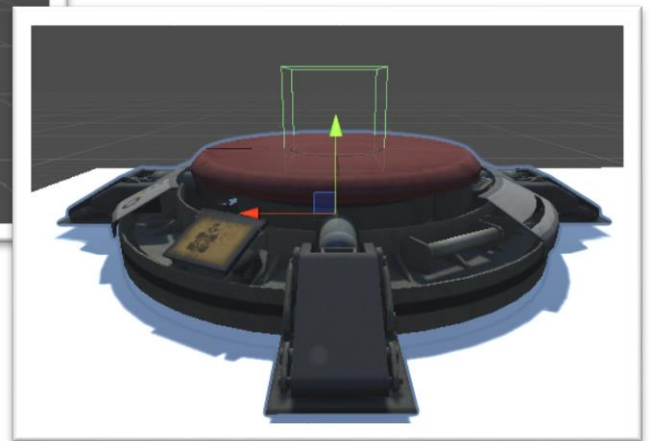
本次我們選擇 Unity 來製作的動機是因為平常生活中，我們常常接觸到的遊戲，無論是手機或者電腦，有很大一部分都出至於 Unity 這個遊戲引擎，以目前來說的市場算大的，Unity 是個能做出跨平台 2D/3D 遊戲引擎，不僅可以開發電腦主機遊戲，也能開發像是 PS 系列那種家用主機遊戲，還有現在當紅的手機遊戲也能輕鬆的製作出來，我們也對遊戲製作有很大興趣，平時也習歡一起討論遊戲相關的話題，所以就決定用 Unity 來設計一款遊戲，也因為本身研究 Unity 的時間也是最多的，所以從中學習有關遊戲製作的問題，進而轉化成自己未來所需要的能力。

## 專題內容:

我們主要分為介面，場景，角色，首先介面使用 2D UI Plane

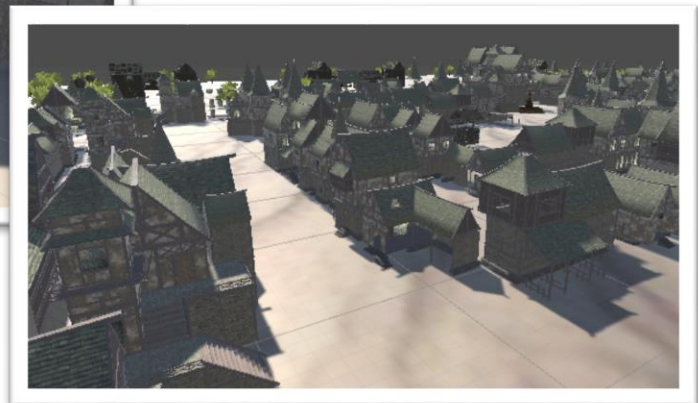
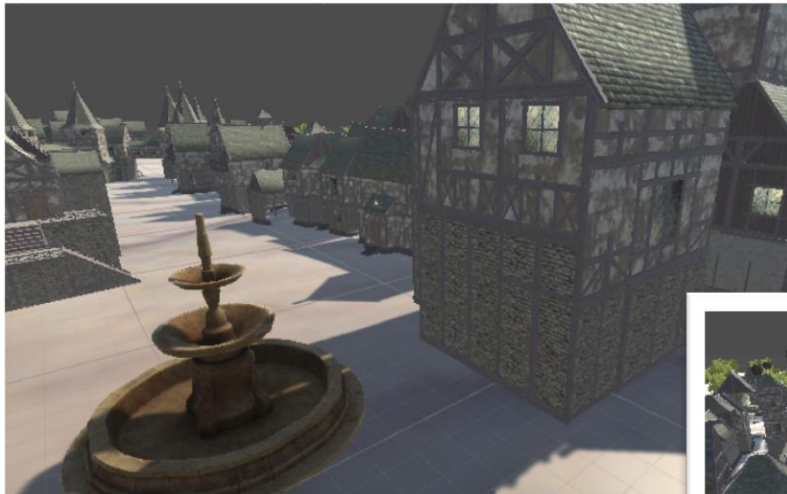


簡單的設計幾個 button，在創建另外一個 plane 之後運用動畫做出簡單的位移再用 Animator 帶入，就可以作出類似彈跳視窗的感覺，讓我們的操作說明不用再額外製作一個場景



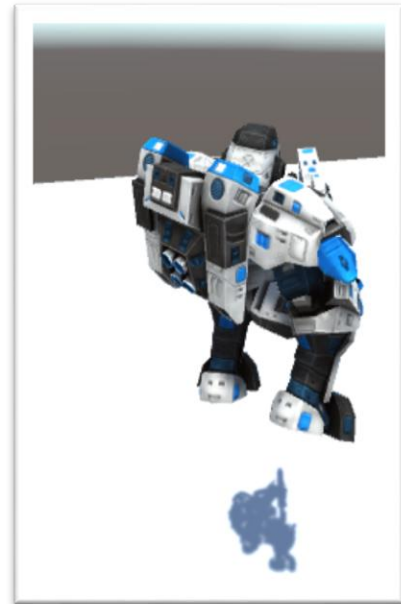
運用 Trigger 或者 Collision 函式去生成碰撞條件，當碰撞發生的時候我們就可以執行函式，而我們的傳送門就是以簡單的碰撞後改變 GameObject 的座標，也就是我們操控的角色的座標。

而場景我們只要負責給他 Collider 角色在碰撞的時候就不會發生穿過房子的問題，在給的時候要記得不是給在 GameObject 上，而是分別給每個接觸面的物件都要有自己的 Collider





接著是角色的部分，運用一些簡單的程式，控制角色基本移動，滑鼠轉動，角色正面永遠跟隨滑鼠轉動的方向來達成第三人稱視角的製作，再帶入動畫我們就可以完成一隻基本會動的人物，接著再寫函式看自己碰撞時需要觸發什麼就好了，而動畫的部分就是設定一個變數 Int，當我們按下按鈕後就會改變變數來傳達我們需要的動作是什麼。



當然我們製作怪物也是同理的，只要會做出角色剩下的就難不倒，只是怪物我們可以製作偵測某個 GameObject 的座標，就可以做出怪物會自動攻擊玩家的感覺了。

## 成果:



我們就可以做出簡易的冒險遊戲，怪物會自動攻擊玩家，有血條有記分板。



## 結論:

製作的一開始確實遇到一些問題，在想要以什麼形式呈現，加上我們自己製作素材的能力不夠強，在有限的素材能其實要把遊戲做到流暢是有點困難的，像是角色的動作量不夠，看起來有點僵硬的感覺，也製作不出那種連擊的感覺，如果使用 blendtree 來說的話動作量太少，可能需要自己去製作更多的動作，來讓動作變得更流暢生動才是。

再來就是我們一開始本來想把地圖做的很大，但是做完後發覺執行起來其實已經有些卡頓的問題出現，可能也需要自己去對建築等等場景進行畫素降低的優化，不然如果一款遊戲吃的效能過重也無法吸引到玩家的目光，再來就是物理問題，研究了很久仍然無法製作出很自然的感覺，偶爾遇到一些稜角的地方仍然是會卡住，需要再做更多細微的調整，所以後來才做了如成果第一張圖，比較簡易的場地來呈現我們專題所製作的東西，否則地圖吃的效能太高，跑起來幾次電腦確實吃不消，這就是我們這次覺得可惜的地方。

其實製作起來好像都差不多，主要還是看遊戲性希望怎麼表現，當你掌握了部分的觀念就能夠舉一反三，製作出自己喜歡的東西或是更深入的研究，2D 或者 3D 我覺得部分的觀念都是能夠共通使用的，程式碼的部分也是，儘管以前學的是 JAVA 換成寫 C# 初期確實會覺得不習慣，畢竟又是一個陌生的語言，但久而久之仍然是會發現其實邏輯上是差不多的，只要掌握好觀念也是能夠把程式寫好。

以上就是我們這次製作專題的心得和覺得自己需要解決的問題。