<資訊書報專題-學習使用 Unity 編寫遊戲之心得>

0410009 羅宇呈　0410017 何智興

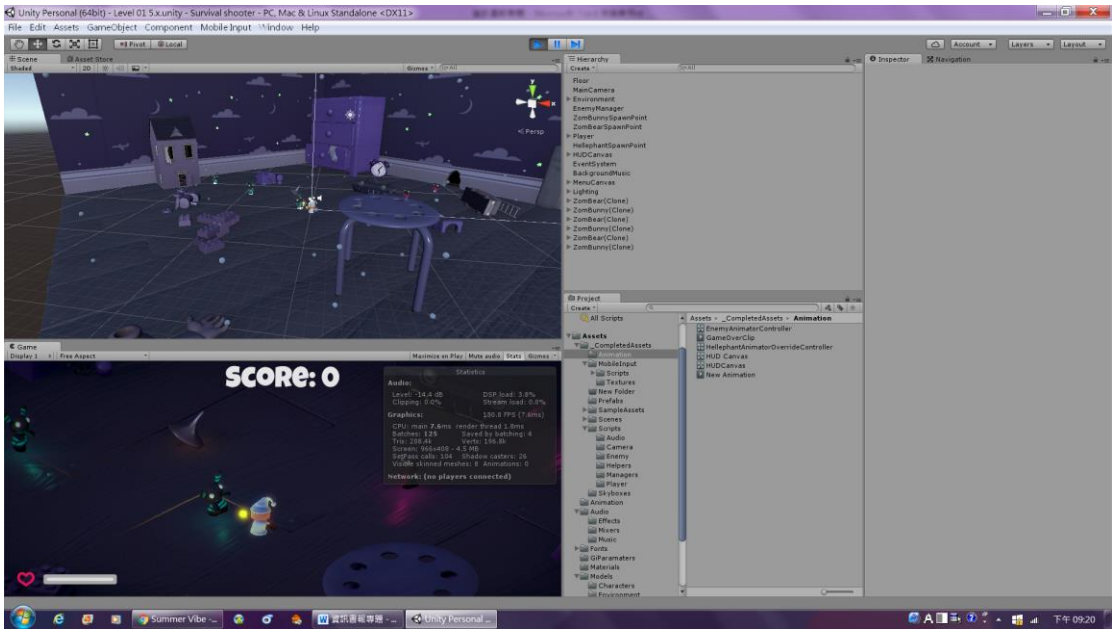
1. 研究動機

近幾年中，各種遊戲不斷的出現，而看到這麼多逼真、好玩的遊戲，讓我們也想更加深入的了解，甚至做出一個遊戲，因此決定以 UNITY 作為我們的專題內容。電子遊戲的發展歷程從一開始第一期的遊戲機主要是用手柄控制電視螢幕上的光點裝置，每部機器也只能玩一種遊戲。再到第二期的遊戲機，開始可以透過卡匣來更換遊戲，使得遊戲開始多元化。到現在的遊戲類型更加多樣化、網路化、可攜帶化。

最近十分流行在全球引起旋風的 POKOMON 就是結合 AR 擴增實境的技術，計算攝影機的影像加上圖像的技術，把虛擬世界套用在現實世界進行互動。虛擬實境分為三大內容：虛擬物和現實的結合、即時互動和三維空間的概念，將虛擬和現實加以結合。

2. 遊戲的介紹以及研究過程

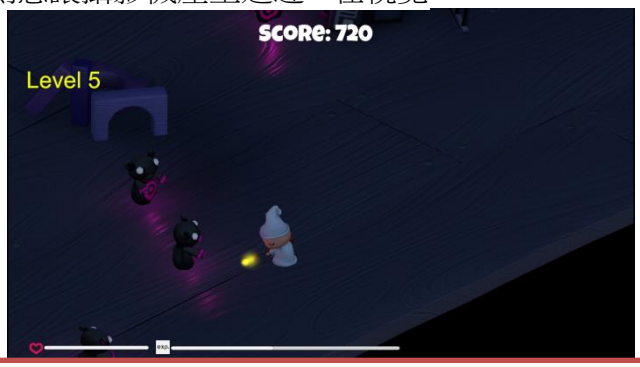
我們這次研究的是 Survival　Shooter，是 Unity 官方網站推薦認識 Unity 這套軟體的遊戲之一，遊戲內容是一個穿著睡衣的小男孩，手拿機槍抵抗來自四面八方的玩偶殭屍，並在擊倒敵人後獲得分數，我們跟隨教程逐步完成這個遊戲，並加入了自己的元素、設定。



上圖是 Unity 的操作頁面，左下角則是實際的遊戲畫面，我們除了利用 Unity 所提供的部分素材(Assets)，也自己試著研究延伸，為這遊戲增加了一點特色，以下開始介紹我們的專案。

<遊戲的示範過程>

本遊戲是一個第三人稱射擊遊戲，我們讓攝影機隨著角色移動，因此角色將會大致保持於畫面中央，而有時會略為偏移，是刻意讓攝影機產生延遲，在視覺上有較為柔和的效果。



◎遊戲畫面，顯示等級、分數、血量以及經驗值

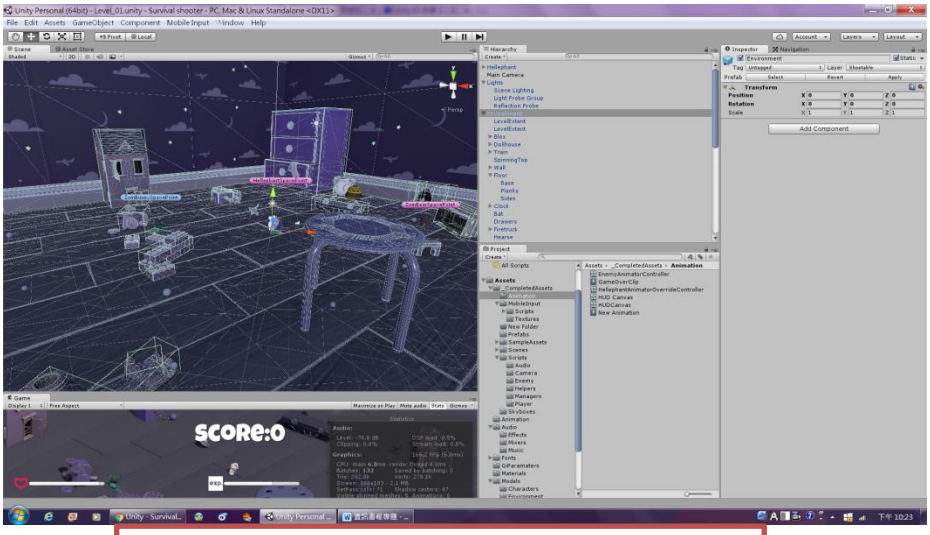
操作的部分，以鍵盤上 wsad 四個鍵分別控制上下左右，而滑鼠左鍵以及 ctrl 控制射擊；以玩家為中心，游標的方向即是玩家面對的方向。

玩家透過射擊並擊倒不斷重生的敵人取得分數以及經驗值，隨著經驗值的增加，玩家的血量、攻擊力、移動速度都將增加；相對的，敵人也將隨著玩家等級的提升而增加攻擊力、血量，在這場毫無止境的戰鬥中，究竟能夠存活多久?

(1) 系統環境設置

一個遊戲的最起初，都需要環境設置，也就是 scene，首先是 Unity 官網所提供的這個場景，可以看出來是一間房間，散落著家具和玩具等等，這是美術的部分，但在給予這些物件以前，他們都只是一個個物件，披著各自外表的 material，首先我們必須要給定一個平面作為地板，我們稱之為 Floor，而 Floor 是一個 Y=0 的 Quad，決定好這一個基準平面以後，我們才能接著決定所有物件的 situation，而就像一般常理所想，重力應該向下，所以既然決定要讓 Y=0，那麼重力的方向當然沿著 y 軸，也因此物件應該要停在地板上，故物件的最底部 Y 座標也應當為 0。

緊接著是設定邊界，且透過 Unity 的重力引擎讓所有物體可以站在此平面上，而這些物件，已經由 Unity 提供 mesh fliter 還有 mesh renderer，如同下方的圖，每一個物件都有一個骨架，而 mesh fliter 是骨架，mesh renderer 則讓這些骨架進行渲染，最後以 3D 的模樣呈現在遊戲畫面中。此外還有光線攝影機等，讓物體能夠擁有光影效果，而這些物件已經以 prefab 的形式完成，除了根據不同的形狀設置了 collider，也擁有各自的 material(外觀)，shader(顏色)，position(除了 Y=0，剩下就看怎麼擺)，。

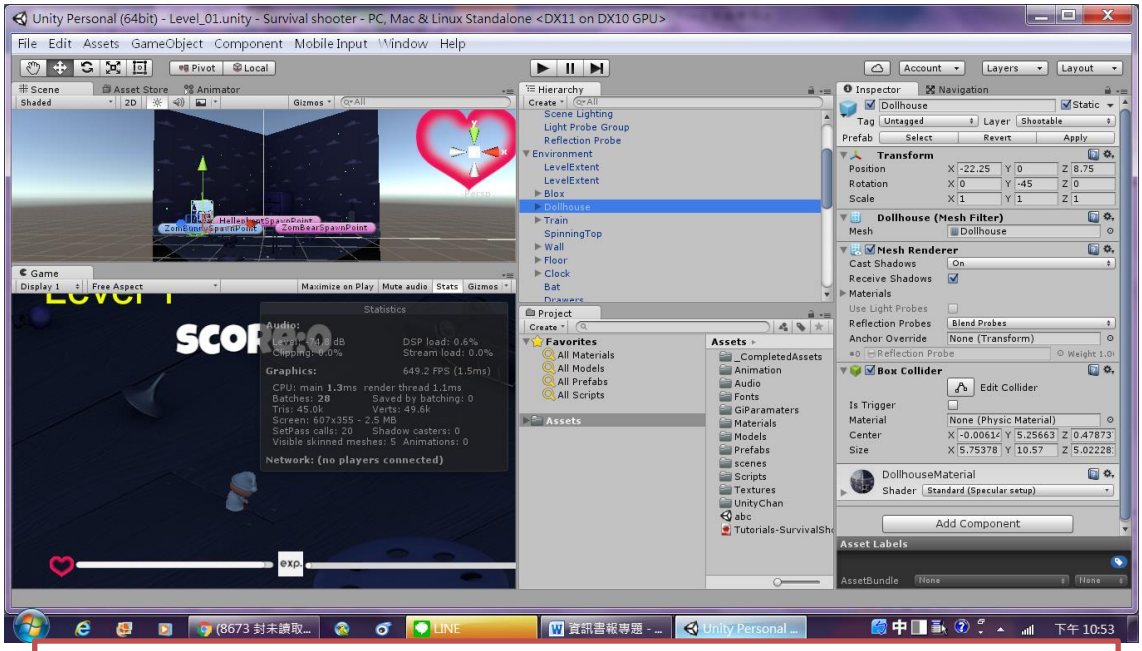


◎圖中每個網格狀的東西都是一個物件

實際來舉一個例子:

娃娃屋(doll house)是一個 gameobject，如果想要在我們的 scene 中加入這個東西，首先要決定他的位置、大小、方向，也就是 transform，決定好擺哪裡以後，接著就是外觀，如果有了骨架(mesh fliter)，那麼就能用 mesh renderer 將它以 3D 的模樣呈現，然後披上外表(material)、上色(shader)，那麼物件的外表就大致完成，最後一步是要設定娃娃屋的特性，我們希望娃娃屋靜止不動，但是理論上任何東西不應該穿越它，因此我們需要讓娃娃屋可以偵測碰撞(collider)，而娃娃屋是方形的，因此理應選擇 box collider，至此，這個 gameobject 已經可以放入我們的 scene。

當然，像 material、mesh fliter 等美術成分較高的東西，我們很難自行製作，因此妥善利用 asset store 是一個解決方法，可以在裡面找到許多資源，再導入自己的專案，就能讓自己的娃娃屋更加精緻。

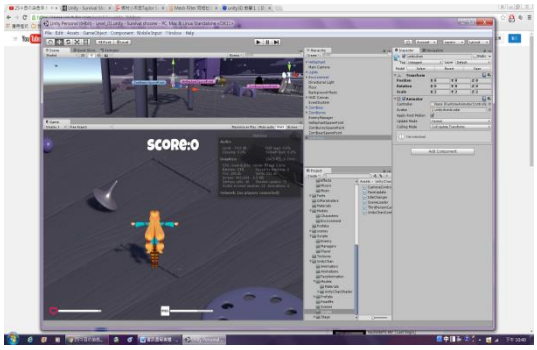


◎Inspector 內為此 gameObject 的所有性質，此處為娃娃屋(doll house)

(2)設計角色

接著就來到設計角色。本遊戲中一共有 4 種角色:主角、兔子、熊、大象。

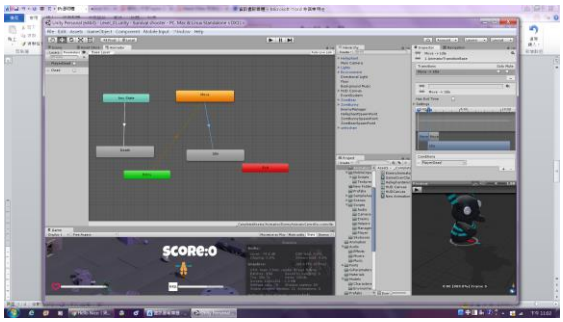
在 Hierarchy 中，有我們所有的 game object，而每一個 game object 必定會有許多特性，如位置、控制腳本、音樂、碰撞器等等，而只要在選定的 Game object 中選擇 add component，就能替這個物件打造出各種特性，而許多東西是可以共用的，舉個例子，像 3 種的人的移動腳本是相同的，創造角色時，便可以加入同一份腳本，大大省去時間。



◎免費的雙馬尾可愛小蘿莉

首先主角的美術模組是由 Unity 提供，而事實上 Unity 有所謂的 Asset Store，裏頭可以找到各式各樣的 Assets(大多收費)，從美術圖到模組，應有盡有，這就是 Unity 強大之處，利用龐大的資源可以製造出無限創意，並且內有許多玩家製作的 prefab，可以想像成是一個完整的模組，從動畫、美術、特效音樂等等都已經打包完成，只要導入專案即可。

除了美術模組，接著就是動作(Animation)，除了動畫以外，最重要的就是:甚麼時候執行這個動作?除了利用控制腳本以外，Unity 有一個很酷的狀態圖，如下:

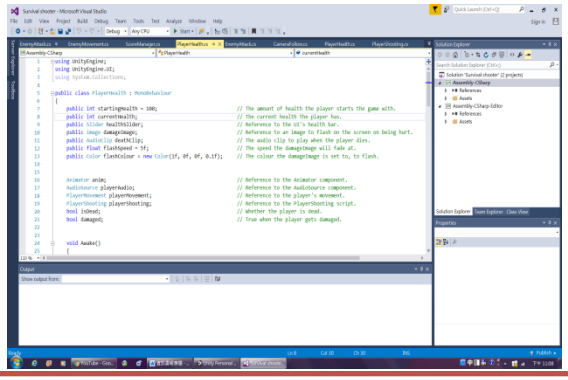


◎狀態流程圖，橘色代表 defaul state，綠色是 entry，天藍色是 any state，紅色是 exit。

圖中是 Zombear 的狀態圖，每一個方格代表一種狀態，箭頭則代表由 A 狀態轉變至 B 狀態，例如由 Move 轉變成 Idle，代表玩家死亡，因此靜止不動，那麼我們就需要一個條件，當這個條件滿足時，可以進行這個狀態轉變，也就是 Trigger，顯而易見的，只要右側的 PlayerIsDead 被觸發，那麼這個狀態轉換就會成立，透過這個概念，控制在任一時刻，所有物件的動作。

(3)讓角色動起來

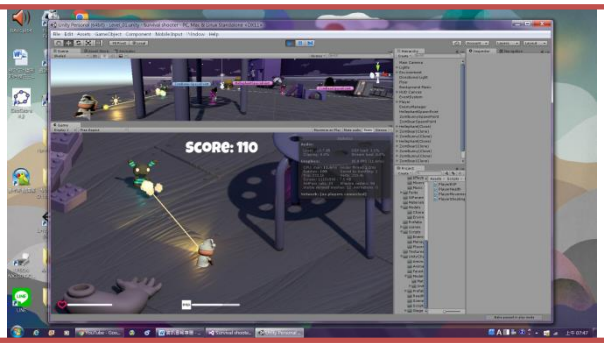
動作結束以後，接著就是腳本(Scripts)，用來控制一切數值轉變，或是動作執行，右圖為 PlayerHealth，透過 c#編寫，用來控制角色血量，受到攻擊時損血、定期更新血量、受到攻擊時血條閃爍、死亡判定等等，而腳本幾乎隨處都會用到，例如讓攝影機隨著人物移動、角色行進、攻擊、敵人自動重生、音樂控制等等。



◎我們編寫的血量腳本，和 OOP 的觀念頗為相似

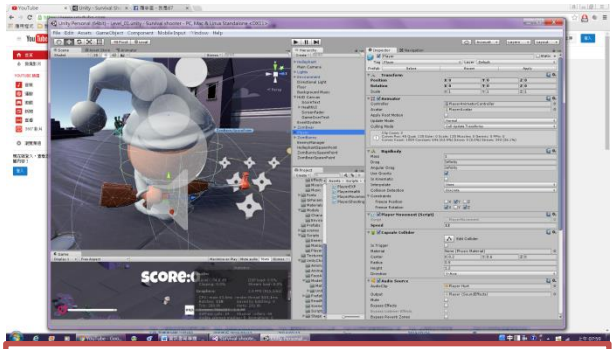
(4)在原先的遊戲上作出延伸

除了血量以外，我們又自行設計了經驗值的設定，只要擊倒敵人就能獲得 EXP，而當 EXP 量表集滿以後，角色便會升級，血量會變多，攻擊力也會變強，有機會的話，可以為角色增加一些技能等等的，這些資訊都顯示在 HUD Canvas 中，一個用來呈現遊戲相關信息的畫面，例如血量條，血量條的圖片、分數、升級畫面等等。



◎左下角的兩個量表分別為血量和經驗值

接下來是 collider(碰撞器)的設定，每一項物件要能夠互相碰撞影響，必須要有 rigid body(剛體)以及 collider 的設定，而這部分也是展現出 Unity 物理引擎強大的最佳示範。我們可以利用 Unity 內建的物理引擎，只要輸入各項數值，如質量、角轉等等，並選定想套用的物理引擎，就能讓物理展現出栩栩如生的動作。這個遊戲中，人物都是頭身 1:1 的模組，所以採用 capsule collider (膠囊碰撞器)是很好的選擇，我們可以看到，網格就是capsule collider的作用範圍，而這個範圍也可以由使用者自行更改大小，如果物件長的不像膠囊，Unity 還有提供各式各樣的碰撞器供你選擇，非常方便。

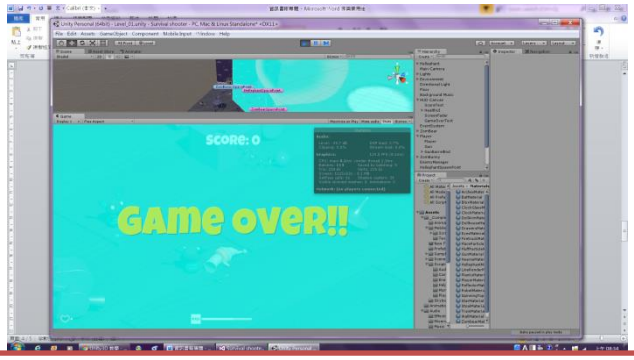


◎角色外圍的橢圓就是碰撞器範圍

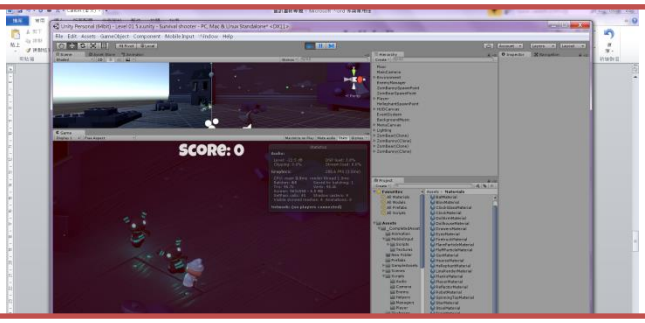
令我們驚豔的是:槍的射擊效果也是我們可以製作的!利用 Patricle system，我們可以設定發射的粒子型態、顏色、速度、大小等等，讓粒子碰撞更符合需求，當然我們也可以利用各種特效，作出像火焰槍、飛鏢之類的攻擊粒子

◎嘗試做成氣功，但比起槍好像有點中看不中用…

主要的部份已經大致講過，接著就是一些讓遊戲更精緻的小細節了:



◎角色死亡時的畫面，可以自行設定 GameOver 跳出的時間，並且調整各時間的字體大小，讓 GameOver 有”蹦出來”的感覺



◎角色受到攻擊時，螢幕跟血條都會閃爍紅色，而頻率還有顏色透明度都可以由我們控制

音效方面，除了背景音樂，我們可以加入自己想要的音效，例如槍響(氣功就改成其他的)，受到攻擊時的音效，死亡的音效等等，都讓這款遊戲更細緻。

而我們將角色設計完畢，例如腳本、動作、音效、模組等等以後，就可以將它儲存成一個 Prefab，有點像是一個包裝好的東西，只要儲存成 prefab 以後，下次就可以直接取用它，而網路上也有許多人製作的 prefab，當然我們也可以把我們的 prefab 放上網路，這樣以後想要作這個遊戲的人就可以直接載下來使用了。

最後，將我們的專案 Build&Run，就會產生一個 EXE 檔，以後只要執行這個 EXE 檔，就可以玩了，Unity 可以讓我們選擇要在甚麼平台上執行，這次我們選擇的當然是 windows 囉，完成以後，執行 EXE 檔便能開始遊戲。

我們的新增功能:

◎經驗值系統

◎一定時間後，玩家隨時間自動回復生命值

◎等級上升時撥放特殊音效

◎玩家隨等級提升，獲得額外攻擊力、生命值、移動速度，且所需經驗值增加

◎敵人隨時間增加攻擊力、生命

◎可使用 exit 離開遊戲

3.心路歷程

(1)最一開始

對於 Unity 這套軟體，我們都沒有接觸過，因此最一開始當然就是甚麼也不會，只能看著官方網站的教程逐步研究，而官方網站所提供的資源檔是較新版本，影片中所教的卻是較舊的版本，令我們傷透腦筋，此時也深刻的體會到，對這套軟體的掌握度太低，就算只有些微的差異，如幾句語法的不同，我們都需要花上許久的時間，才能勉強成功。

(2)從模仿到自己的創意

多虧了 Unity 官方詳細的教程，我們得以對這個遊戲的架構有清楚的認識，但我們卻是在加新功能的過程中，才稍微懂得運用這套軟體，因為教程上是直接告訴我們該怎麼作並解釋原因，乍看之下我們好像明白，但當我們自己做的時候卻是茫然。

(3)曾經遭遇的困難

就像我們當初討論以後，想要增加經驗值的設定，沒想到打開空白的 Script 時，不曉得要怎麼取用其他 script 的變數，或是取了卻無法改值，又或者是取到的值指到 NULL，而我們也曾經發現，我在 A 腳本宣告的變數，加了 static 以後反而無法為 B 腳本所用，系統不斷顯示: Member “xxxx”cannot be accessed with an instance reference，上網一查，原來這是 C#的特色，you can't access `static` members with instance syntax.，如果只看官方教程，我們是不會知道這件事的。

而寫 Script 的時候，常常心裡有個想法卻不知道該怎麼做，例如我們希望角色可以隨著時間回復生命值，本來是打算每一次執行 update 的時候回血，沒想到 update 的速度太快，玩家的血量變的完全不會損失，於是我們改成利用紀錄時間的計數器，例如每隔 2 秒回血一次，但是這樣有點麻煩。於是我們就用白話的方式，將我們想要的方式進行搜尋，例如:"unity 重複執行動作"，然後我們就發現了 InvokeRepeating()這個函數，這對我們是一大驚喜，有了這個經驗，每當我們有甚麼想法，就會上網 google，往往能找到我們想要的函數，又或者是相近的想法，我們只要再稍微修改一些便可，除了發現內建函式庫的強大，也學會了一個解決程式問題的方法，利用這個方式，我們發現，要讓腳色在受到攻擊時閃爍白色，只要準備 2 個 material，一個全白，然後觸發後不斷交換就好了。

創作的過程中，往往是腦海裡先有一個構思，像是"我想要在攻擊時受到後座力而後退"，然後想像一下會用到甚麼 component，例如 audioclip、rigidbody、particle system 等等，然後才能利用 script 把所有的東西串起來。常常我們把 script 寫完了，試著套用到我們的 gameObject，才發現:不是開不起來，就是效果與我們所想大相逕庭，我們有想到幾種可能:

1. 我們的 component 用的不夠多，或是用錯了，有些我們想要的效果，是由許多 component 共同運作，例如 shader 與 light 等等，但我們卻只想到其中幾個，script 也要重寫。
2. Component 都對了，script 也對了，但特效的勾選，還有各項特效的數值，總是抓不到概念，這必須要透過非常大量的嘗試，"這個效果打勾會怎樣?不勾又會怎樣?"，或是"這一項調大一點或小一點如何?"，透過這樣的方法觀察畫面呈現的差異，才能做出自己想要的樣子。

當以上兩點都解決以後，我們才終於能看到比較接近"理想中模樣"的特效。

(4)解決問題的方法

以上這些例子都是在說明我們碰到問題時如何解決，而答案，除了靠自己摸索，就是網路。往往我們碰到的問題，其他人也曾經碰過，當我們把碰到的問題丟上網，例如 UNITY 顯示的錯誤資訊等等，馬上就能發現同樣有這個問題的人，而底下往往就會有這個問題的解答或解釋，當然不一定完全適用於我們的情況，但多多少少都會有幫助。

而許多網站也提供了詳盡的 Unity 教學，往往看了教學，就可以在自己的專案裡，加入一些新東西，或是把舊有的升級，當我們掌握的技術越多，能夠做出來的變化也就更多，比起看著教程製作，能夠將自己的構想實體化，才更了解 Unity。

4.各人貢獻項目

羅宇呈(60%):編寫 C#腳本

遊戲內容製作、修改

何智興(40%):提供創意構思、協助畫面優化

3. 心得

Unity 是一套免費版的功能就十分強大的軟體，讓入門者可以對製作遊戲有基礎概念，又不需要花費大量的金錢，現今智慧型手機的 App Store 中也有不少遊戲是以 Unity 開發，只要有足夠的創意跟技術，有一天你我的遊戲都能夠上架。

在製作一個遊戲的過程中，我們深刻體會到，一件看似簡單的事，背後是需要諸多考量的，就像敵人會追著玩家而來，這很合理，但到底要怎麼讓敵人繞過地圖物件呢?Unity 官方提供的解法是利用 Navigation，考量物件位置高度形狀等等，創造出一個可移動的範圍；又或者是我們當初想自行增加的經驗值設定，原本想法是只要判定的人死亡，就增加對應的經驗值，這樣簡單的一個觀念，卻花了我們一整天才找出 bug；最簡單的，光要讓人物動起來，就夠讓人傷腦筋的了，看著 Unity 官網提供的 tutorial，發自內心的佩服著這些工程師。

製作遊戲真的很有趣，我們常討論著可以增加甚麼元素，或是又在 Asset store 發現了甚麼有趣的特效，現在的我們對 Unity 只是小有認識，有朝一日，或許我們做的東西也能在 Asset Store 供人下載，更或許，我們的遊戲也能在 APP 商店上架，但在那之前，我們必須要作的，就是更加精通程式語言的能力，這次編寫 C#，我們深刻體會到自己的不足之處，空有創意卻無法付諸現實的遺憾，希望下一次我們再使用這套軟體時，可以更加得心應手。