絕世武功\_由你創造

Unity簡介：

Unity3D是一個用於建立諸如三維視訊遊戲、建築視覺化、即時三維動畫等類型互動內容的綜合型創作工具。Unity類似於[Director](https://zh.wikipedia.org/wiki/Adobe_Director)，[Blender](https://zh.wikipedia.org/wiki/Game_Blender)，[Virtools](https://zh.wikipedia.org/w/index.php?title=Virtools&action=edit&redlink=1)或[Torque Game Builder](https://zh.wikipedia.org/w/index.php?title=Torque_Game_Builder&action=edit&redlink=1)等利用互動的圖型化開發環境為首要方式的軟體其編輯器執行在[Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows) 和[Mac OS X](https://zh.wikipedia.org/wiki/Mac_OS_X)下，可發行遊戲至[Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows)、[Wii](https://zh.wikipedia.org/wiki/Wii)、[OSX](https://zh.wikipedia.org/wiki/OSX)或[iOS](https://zh.wikipedia.org/wiki/IOS)平台。也可以利用Unity web player外掛模組發行網頁遊戲，支援[Mac](https://zh.wikipedia.org/wiki/Apple_Macintosh) 和[Windows](https://zh.wikipedia.org/wiki/Microsoft_Windows)的網頁瀏覽。它的網頁播放器也被Mac widgets所支援。

程式設計師可用[JavaScript](https://zh.wikipedia.org/wiki/JavaScript)、[C#](https://zh.wikipedia.org/wiki/C_Sharp_(programming_language))或[Boo](https://zh.wikipedia.org/wiki/BOO)加以編寫。

1. 程式基本概念(C/C++/C#)：
2. 資料型別(data type)：

顧名思義，就是一個資料是屬於哪個類別，我們在宣告一個變數的時候，都必須指明它是屬於哪一個類別的，舉出幾個常見的資料型別: int (整數integer)、float (單精度浮點數)、double (雙精度浮點數)、char (字元)

1. 變數：

類似數學中的代數(每個代數有自己獨特的型別)，可以依自己喜好取名，盡量不要隨便亂取，方便以後閱讀輕鬆，有些規則：

|  |  |
| --- | --- |
| 規則 | 錯誤範例 |
| 可由英文字母大小寫。底線。數字組成 | home work(不可有空格) |
| 名稱不可使用保留字集符號常數 | int , &hello , if |
| 所取的英文字母大小寫是不同的 | Hello 不等於  hello |
| 不可使用中文變數名稱 | 附中\_data |
| 第一個符號不可使用數字 | 2cir , 6\_good |

1. 算數運算子與關係運算子的差異

簡單來說，關係運算子是用來判斷，而算數運算子是用來運算的。譬如(a = 5)是指讓a 的值變成5，而(a == 5)則是用來表示a 跟 5的關係**真假值**，若a真的等於5，這個式子即為true(真)，反之則為false(假)。

一些常見的關係運算子譬如：

==(相等), >(大於), <(小於), >=(大於等於), <=(小於等於), !=(不等於)。

一些常見的算數運算子：

+(加), -(減), \*(乘), /(除), %(取餘數)

注意： a = 5　（切記’＝’左邊只能放變數）　a就是5  
而 a = a + 5 則是表示，我把a+5這個值，存到a)

**算術運算子**

算術運算子包含加、減、乘、除、取餘數，這些需要兩個運算元 (operand) 構成運算式 (expression) ，如下列表

|  |  |  |
| --- | --- | --- |
| **運算子** | **功能** | **範例** |
| + | 加 | a + b |
| - | 減 | a - b |
| \* | 乘 | a \* b |
| / | 除 | a / b |
| % | 取餘數 | a % b |

另外有遞增及遞減運算子，這是單元運算子 (unary operator) ，也就是說只需要一個運算元即構成運算式，但在運算元有前後的差別

|  |  |  |
| --- | --- | --- |
| **運算子** | **功能** | **範例** |
| ++ | 前遞增 | ++a |
| ++ | 後遞增 | a++ |
| -- | 前遞減 | --a |
| -- | 後遞減 | a-- |

**關係運算子**

關係運算子都需要兩個運算元，包含小於、小於等於、大於、大於等於，另外有兩個測試相等性的運算子，運算結果會是 **bool** 型態的數值，不是 **true** 就是 **false**

|  |  |  |
| --- | --- | --- |
| **運算子** | **功能** | **範例** |
| < | 小於 | a < b |
| <= | 小於等於 | a <= b |
| > | 大於 | a > b |
| >= | 大於等於 | a >= b |
| == | 相等 | a == b |
| != | 不等 | a != b |

**邏輯運算子**

邏輯運算子包含邏輯非，施用於單一運算元，其他邏輯且、或都需要兩個運算元

|  |  |  |
| --- | --- | --- |
| **運算子** | **功能** | **範例** |
| ! | 非 | !a |
| && | 且 | a && b |
| || | 或 | A || b |

**指派運算子**

基本的指派運算子為單一個等號 = ，這是用來將等號右邊的值拷貝給給左邊的變數資料。等號也可以跟其他運算子合用，會直接將結果儲存到原變數之中，如

|  |  |  |
| --- | --- | --- |
| **運算子** | **功能** | **範例** |
| = | 指派 | a = b |
| += | 相加同時指派 | a += b |
| -= | 相減同時指派 | a -= b |
| /= | 相除同時指派 | a /= b |
| %= | 取餘數同時指派 | a %= b |

補充：a = 5　（切記＝左邊只能放變數）　a就是5  
 而 a = a + 5 則是表示，我把a+5這個值，存到a的這個位置。

!(a >= b) 就表示a<b，！就類似於集合中的所教的～，只是～在程式有

些微不同的意義

1. 函數及引數

程式中的函式其實類似於數學函數，舉f(x)為例子，f即為這個函式(函數)名字，而x即為傳入函式(函數)的引數(變數)，引數也就是這個函式的操作對象，所以回來程式的觀點，我們舉一個例子：

int example(int a, int b){…}

我們宣告一個函式的時候，必須講明函式傳回的型別，跟傳入引數的型別(\*a)，之後呼叫的時候必須傳入對應的引數。

int main(void){

…

example(5, 7);//呼叫example這個function

…

}

example是函式的名字，而這個函式必須傳入兩個int型別的變數，也就是兩個整數。

1. Public 與Private的差異

Public意指這個變數可以被其他人(除了這個class之外的人)來使用或改變，Private即此變數不能被其他人改變，只能被這個class本身所使用或改變。

1. \*unity的語法
   1. void(表示回傳的資料型態)->什麼都不需要回傳
   2. start()表示初始化(用一個東西前，要知道它的名字或是長寬高等等，才知道如何用)，Update()是每個畫面切換都會執行裡面的東西，而FixedUpdate()則是固定時間執行裡面的東西(多用於物理現象的運算時)。
   3. Vector3向量，用來儲存三維數值。
   4. GetComponent<所要用到的屬性>，而AddForce是Rigidbody中的函式(類似Rigidbody是本書，要有書才能知道其中的章節AddForce)。
2. Unity操作提示：
3. 主要分為五個視窗：hierarchy, scene, game, project, inspector

Hierarchy：主要是創物件、管理物件

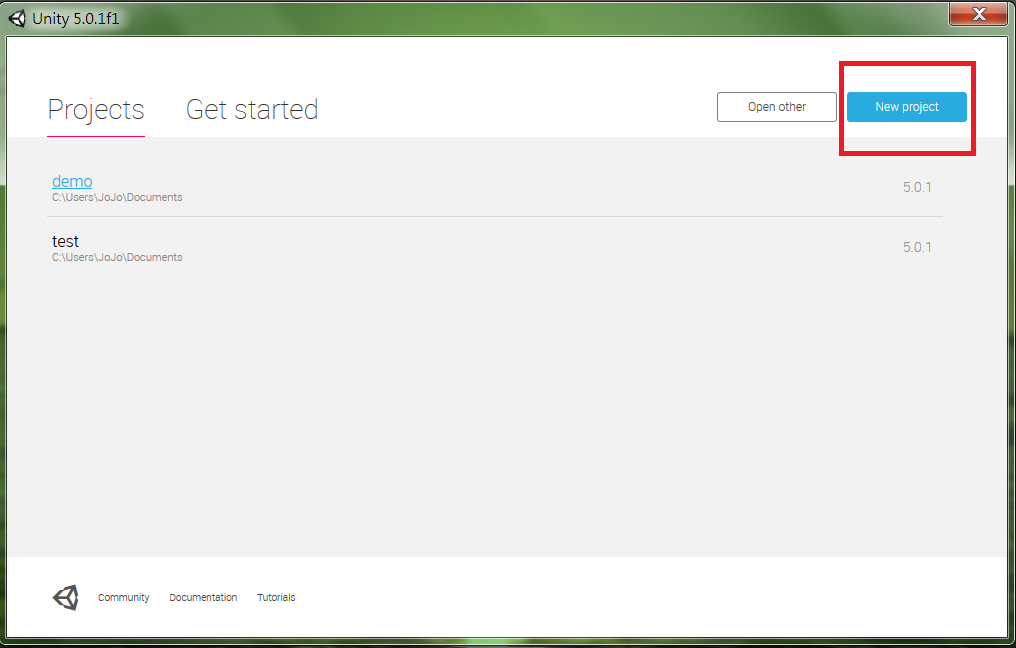
Scene：在視窗中可直接管理物件(拉曳、伸縮、旋轉等等)

Project：管理資源，把物件從hierarchy視窗拉過來便成為了預製物 (prefab)(\*h)，製作材質球跟寫腳本也是在這裡創建

Game：測試遊戲用，無法更動任何遊戲的數值內容

Inspector：控制物件的屬性及數值

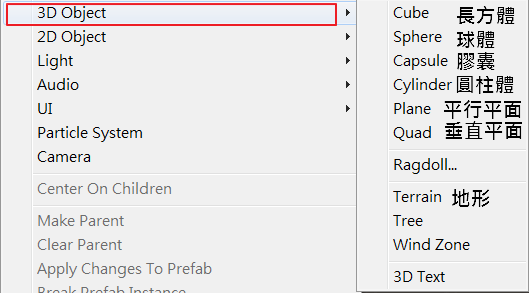
1. 在hierarchy中點選一個物件，然後**按下F**，可以直接切換到看這個物件的適當大小
2. 在scene視窗中，可以按住**右鍵**來旋轉畫面
3. 遇到多零件的物件，必須先在hierarchy創建一個空的物件(empty object)來當作類似資料夾的作用
4. 點選一個物件或者一個資源，便會出現inspector視窗，在這個視窗中可以更動物件的數值、屬性等等
5. Inspector視窗中，在物件的名字旁邊有個小方框，若不打勾則此物件無效力，也不會出現在scene中，在製作遊戲圖中可以善用以達到畫面整潔
6. Unity的各個視窗可以拉曳，可調整至自己習慣的排版方式
7. **預製物**就是把我們做好的物件變成可重複利用的資源，當一個物件成為預製物的時候，在Hierarchy視窗會變成**藍色**
8. 製作步驟
9. 創立專案



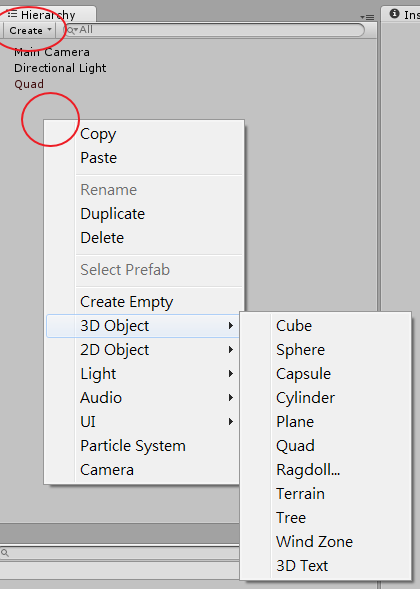
1. 製作物件

從上方的工具列創建物件





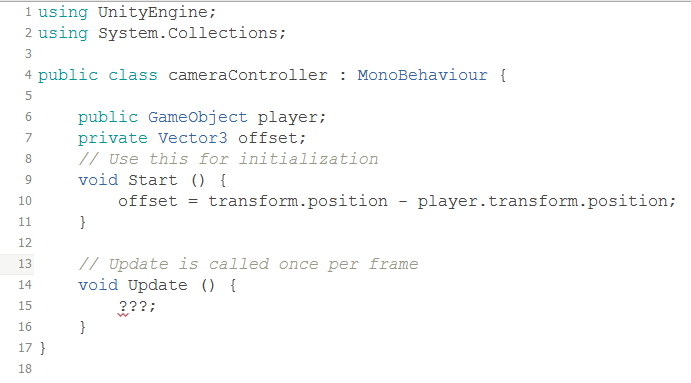
也可以在Hierarchy的空白處點右鍵創建



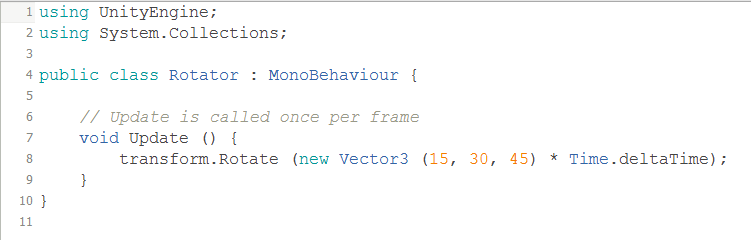
1. 寫腳本

本遊戲需要寫3個腳本：第一人稱視角控制器(cameraController)、方塊旋轉器(rotator)、讓車子移動的腳本(playerController)

cameraController:



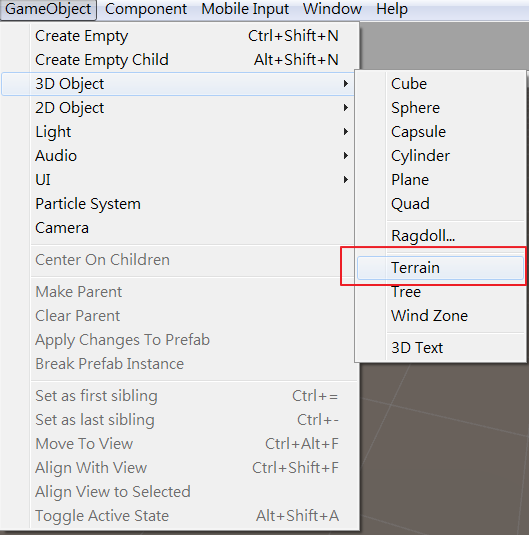
Rotator:



playerController:



1. 繪製起伏地形





第二個拉高可以固定住數值，可製作等高度的平台、梯田之類的，若要更改地形(terrain)的長寬，可以到設定(齒輪圖示)的地方更改