



---

# 人造引力腳本使用 手冊

---

紀曲峰

# 目錄

---

概要.....	1
匯入腳本.....	1
建立星球.....	2
建立角色.....	4
套用移動腳本.....	8
套用粒子.....	11
建立粒子物件.....	11
建立發射器.....	13
版權聲明.....	16

## 概要

此人造引力的 **Script** 是我參考網路上的腳本改寫的，原腳本只提供基本引力功能，因此我重新改寫並加入控制及粒子功能，以下是我參考的原腳本網址。

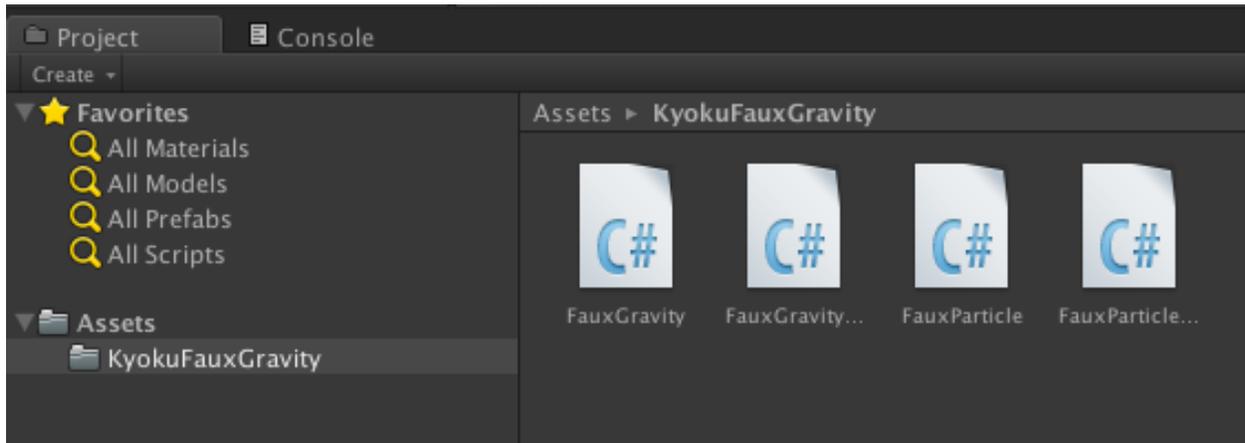
<http://forum.unity3d.com/threads/8873-Faux-Gravity-making-my-brain-spin-Help!>

我改寫是以 **C#**重新寫過，由於改寫的內容差很多，因此本手冊並不適用原來的腳本，但我的演算法是改寫自原文件之演算法，因此原理是一樣的，你可以從下列網址下載到我改寫後的程式碼及使用手冊，若將來有改版也可以從下列網址下載到。

<http://www.digiart.com.tw/unity.aspx>

## 匯入腳本

首先請先開啟新專案，然後直接匯入下載後的腳本，匯入後可以看到 4 個檔案。



4 個腳本都是 Open Source 的，分別作用如下。

FauxGravity .cs: 提供基本引力功能

FauxGravityController .cs: 提供幾種角色控制套件，你不一定需要用此腳本，可以自行撰寫角色控制功能

FauxParticle.cs: 人造引力的粒子物件

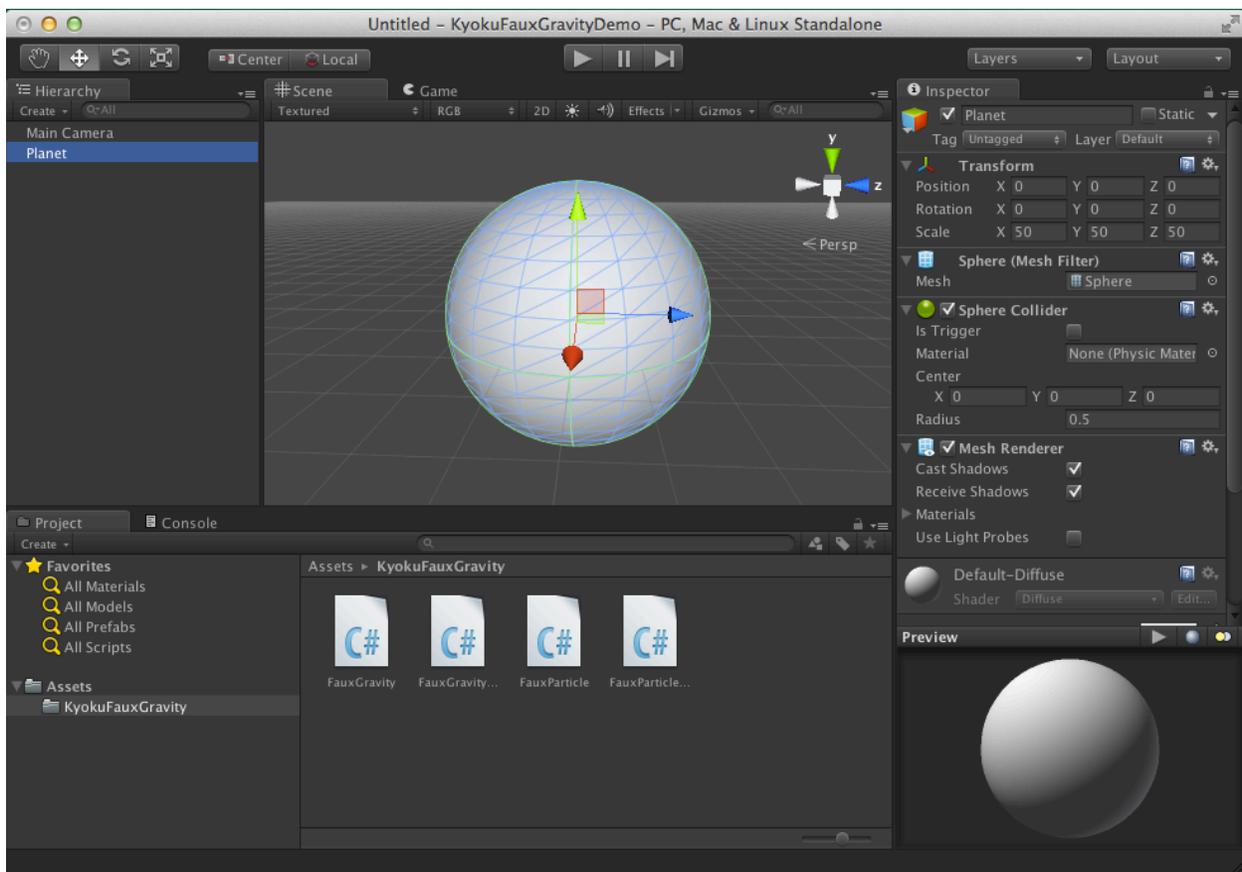
FauxParticleEmitter.cs: 人造引力粒子發射器

## 建立星球

現在建立一顆球體當作星球，你也可以自行匯入球體來使用。

GameObject > Create Other > Sphere

建好之後將此球體更名為 Planet，並更改大小為 Scale(50, 50, 50)。



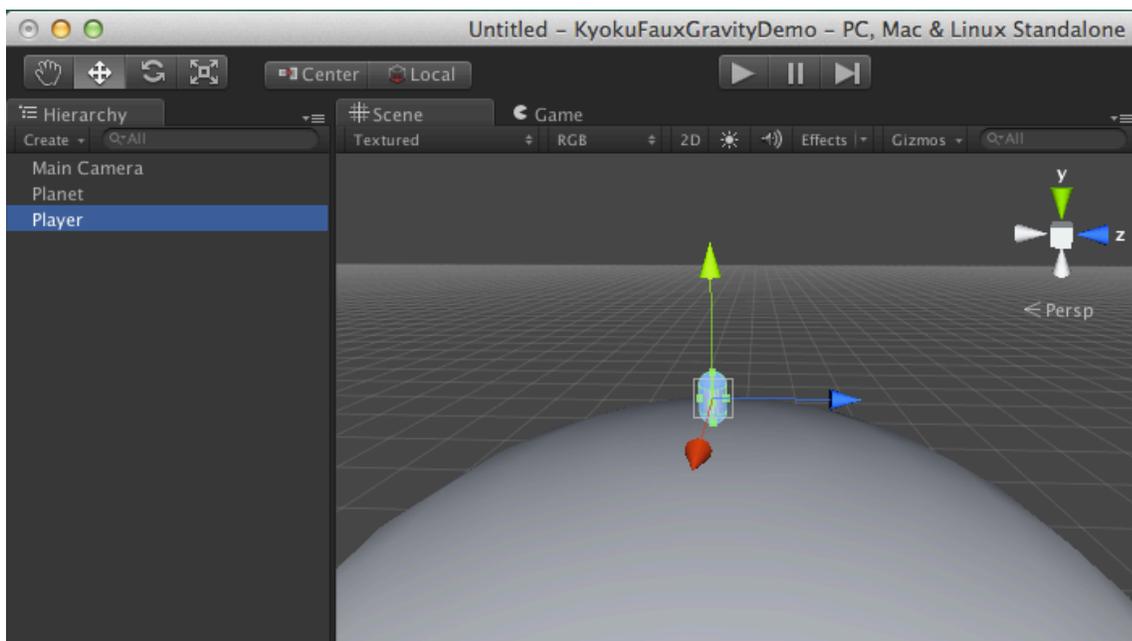
作為星球的物件一定要帶 **Collider** 的屬性，若你匯入自己的模型請自行建立 **Collider**，一般來說星球都會有地形，可建立 **Mesh Collider**。

## 建立角色

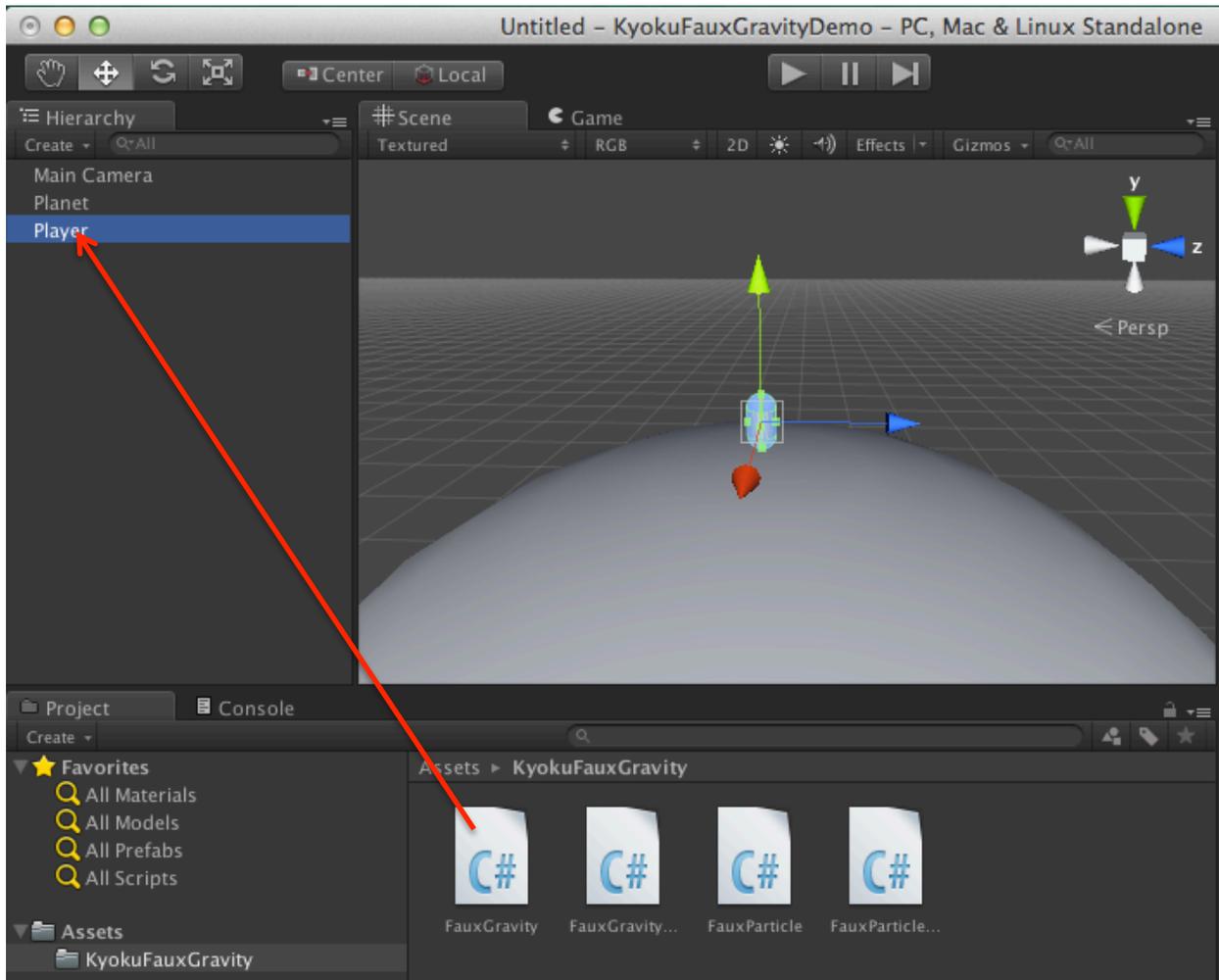
現在建立一個膠囊當作角色

GameObject > Create Other > Capsule

將角色移到星球的正上方(北極)，然後更名為 **Player**。



現在將「FauxGravity.cs」拉到 Player 身上。



FauxGravity 有幾個屬性，說明如下。

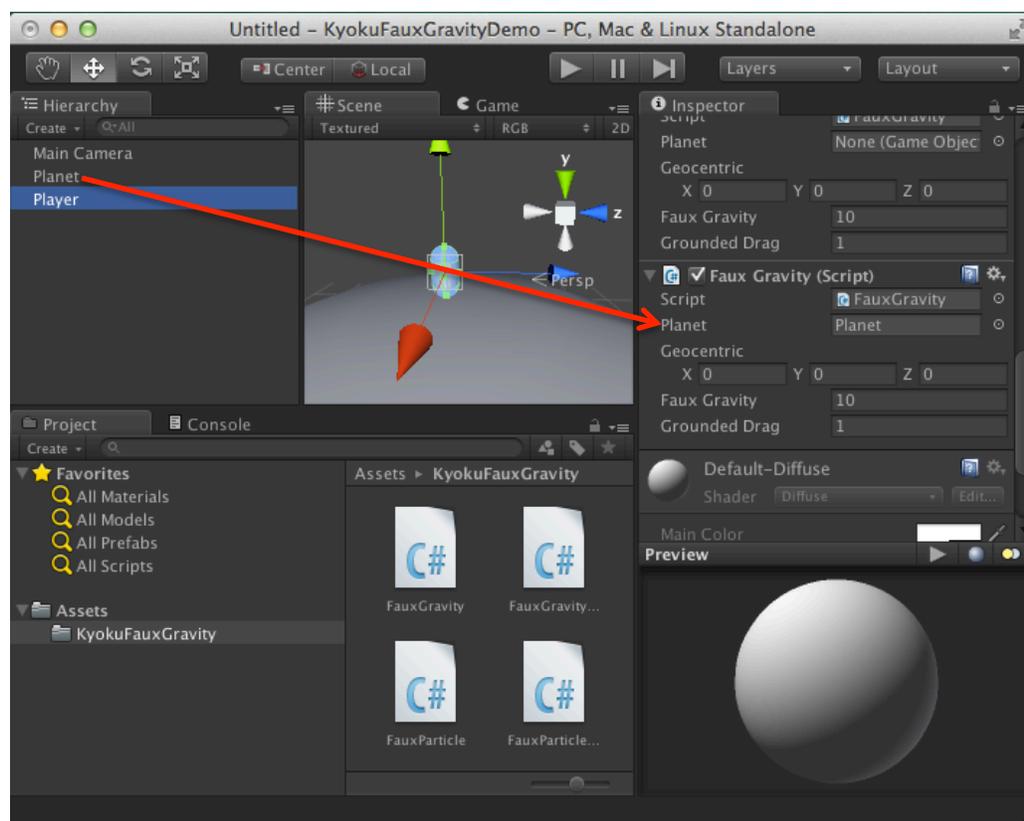
Planet：星球物件

Geocentric：星球的地心偏移量，若地心不是星球的正中心可用此值去偏移

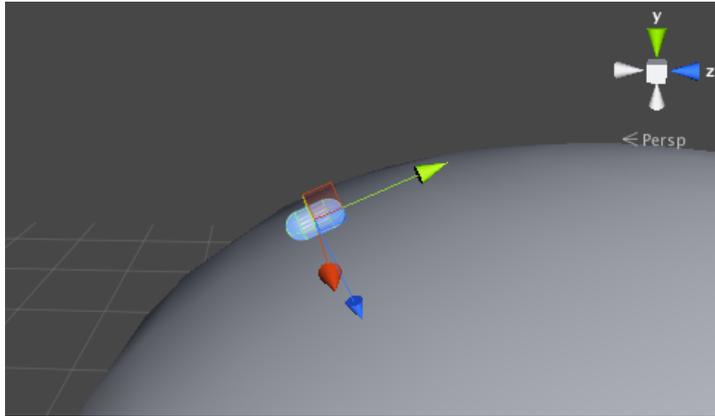
Faux Gravity：引力大小

Grounded Drag：當落到地面上時摩擦力大小

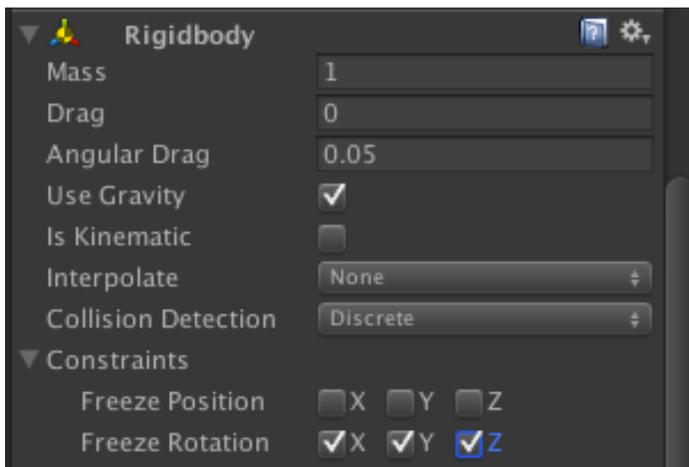
要正確執行引力必須設置星球物件讓引力知道要吸往哪個星球，選取 **Player** 後，可以看到 **Inspector** 上有一個「Planet」的屬性，請將 **Planet** 拉到這個屬性上，星球的屬性可以利用程式動態修改讓角色得以在不同星球間移動。



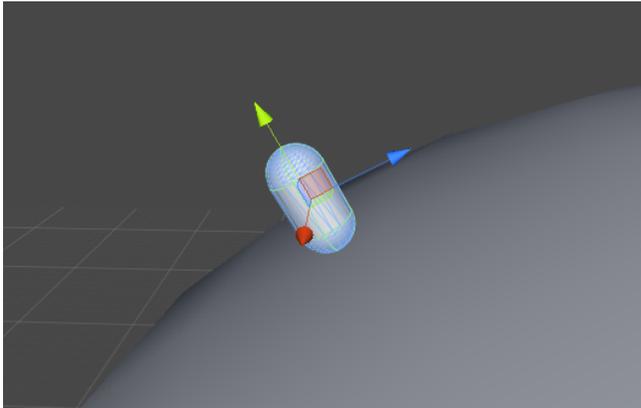
這時去執行程式，然後到 **Scene** 視窗拉動角色就可以發現角色會被自動吸到星球上了，不過會倒下去，所以我們必須鎖定角色的角度。



選取角色，到 **Inspector**，展開「**Rigidbody/Constraints**」，可以看到 **Freeze Rotation**，將三個軸向全部打勾。

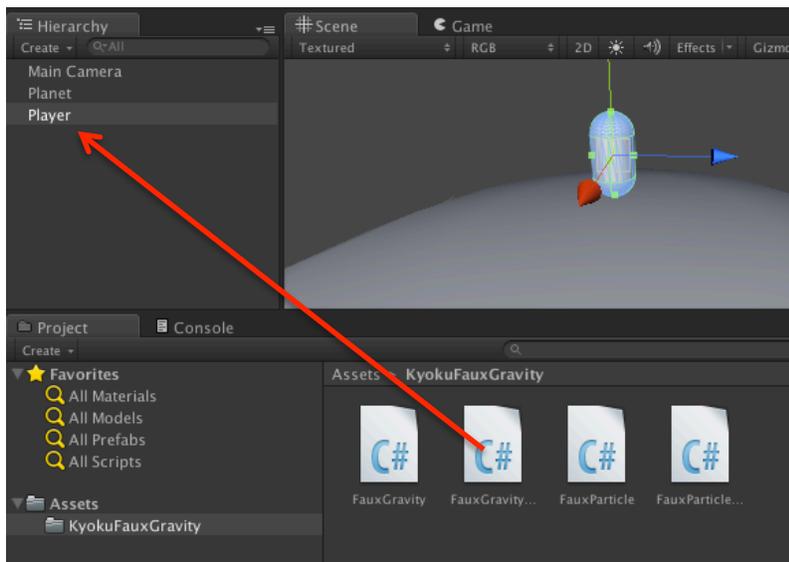


重新執行後回到 SceneView 拉動角色，這時角色已可以成功的吸在星球上了。



## 套用移動腳本

由於在星球上移動不是很容易寫，所以我也幫各位寫好了基本的移動腳本了。請直接套用「FauxGravityController.cs」到角色上。



移動腳本的屬性如下。

**Move Speed**：移動速度

**Obstacle Height**：遇到障礙物的高度，若前方障礙物高度高於此值則無法前進，請依角色的身高調整

**Obstacle Distance**：遇到前方障礙物的距離，依角色的胖瘦調整

**Can Jump**：是否可跳躍

**Jump Key Code**：跳躍的按鍵

**Jump Height**：跳躍高度

**Move Type**：移動方式

**Turn Speed**：當角色轉身時的旋轉速度

本腳本提供了幾種移動方法，可以從 **Move Type** 調整。

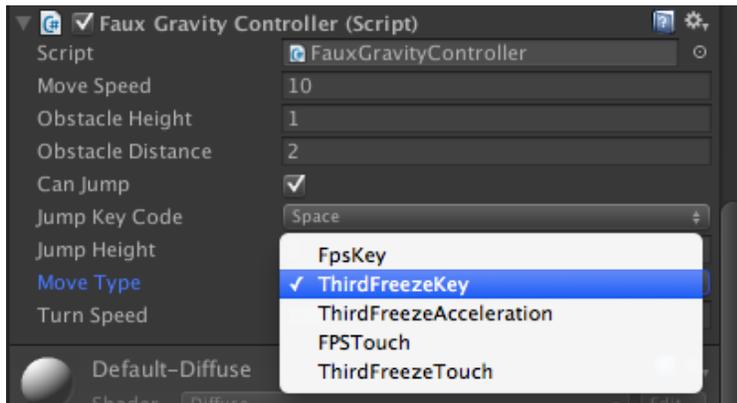
**FpsKey**：類似 FPS 的移動方式，左右轉身，上下鍵前後移動

**ThirdFreezeKey**：第三人稱移動方式

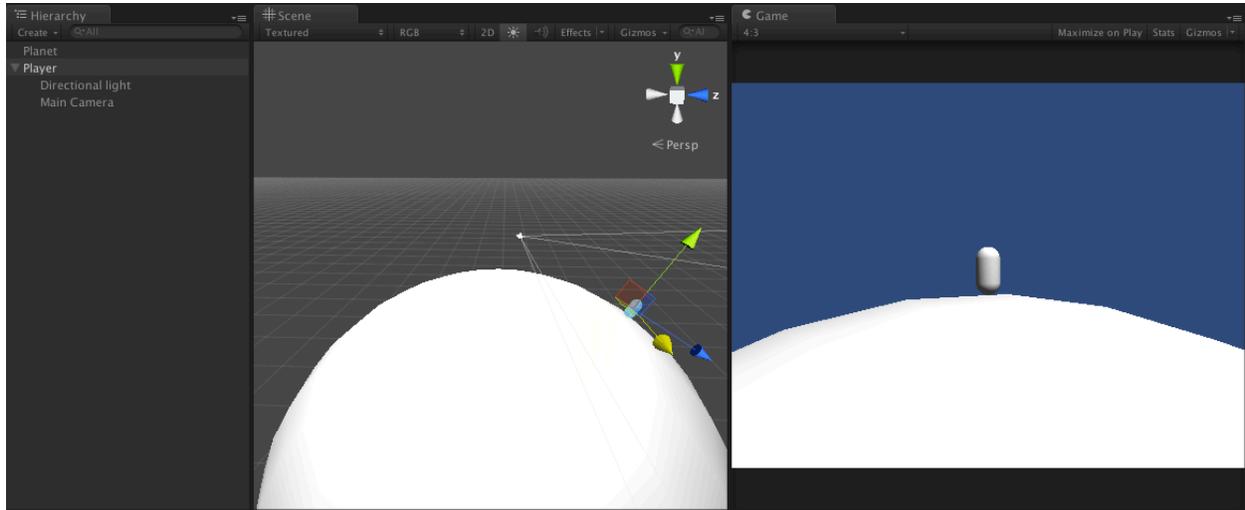
**ThirdFreezeAcceleration**：利用手機三軸控制，我只用 iPhone 測過

**FPSTouch**：FPS 移動方式，利用手機左右滑動旋轉方向，前後滑動則前進後退，我只用 iPhone 測過。

ThirdFreezeTouch: 第三人稱移動方式，利用手機滑動決定移動方向



請設置 **FpsKey** 或 **ThirdFreezeKey**，並將攝影機移到接近角色的位置再打一盞燈，然後將攝影機和燈拉到角色底下成為子物件，如此角色便可以跟著攝影機移動，然後執行遊戲，角色便可以在星球上移動了。



# 套用粒子

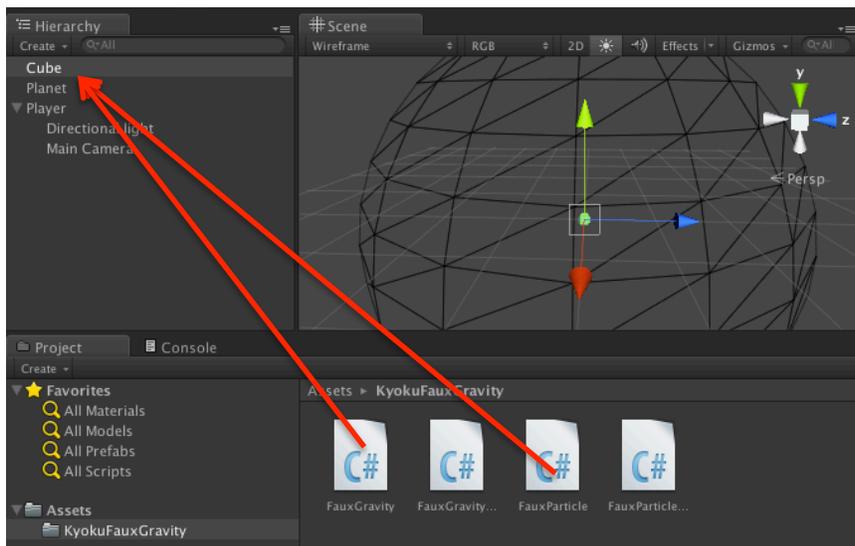
粒子系統因為要用人造引力去計算，無法使用內建的粒子，所以我自行開發了一套粒子系統，但只能使用實體模型去建立粒子，若有使用者能加以改善希望也能提供給我分享出來。

## 建立粒子物件

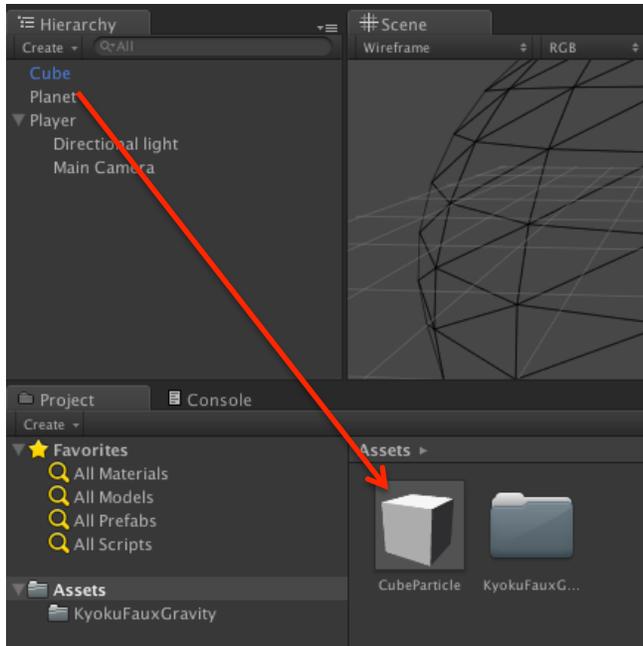
要使用粒子必須先建立粒子物件，為了要和地表碰撞因此目前只能用實體物件，我們使用 Cube 來教學，請先建立一個 Cube，注意 Position 必須在原點上，若粒子物體沒有 RigidBody 屬性會被自動添加此屬性。

GameObject > Create Other > Cube

然後將 FauxGravity 和 FauxParticle.cs 拉到 Cube 上，因為粒子同時也需要有引力屬性才能順利落到星球上。



接下來建立一個 Prefab，取名為「CubeParticle」，並將 Cube 拉到 Prefab 上。



現在可以刪除場景中的 Cube 了。

但此時粒子對此星球來說有點太大，可以自行修改大小和顏色，我把粒子改成紅色並設成  $Scale(0.2, 0.2, 0.2)$  的大小。

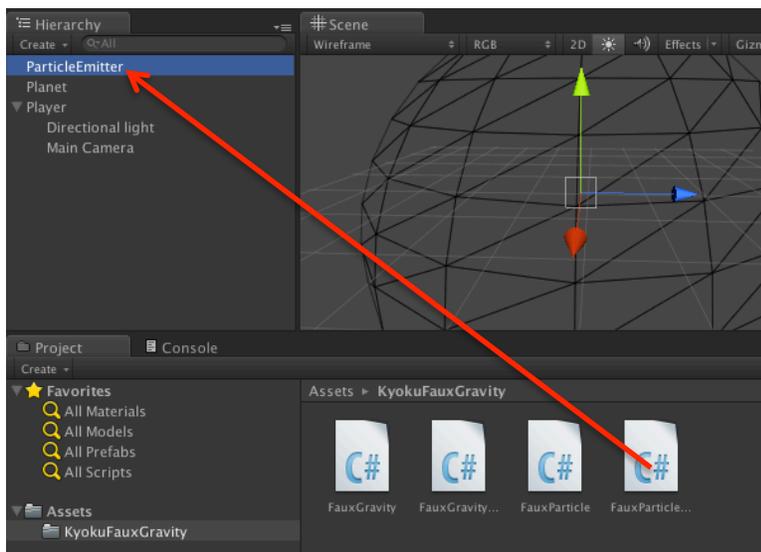


## 建立發射器

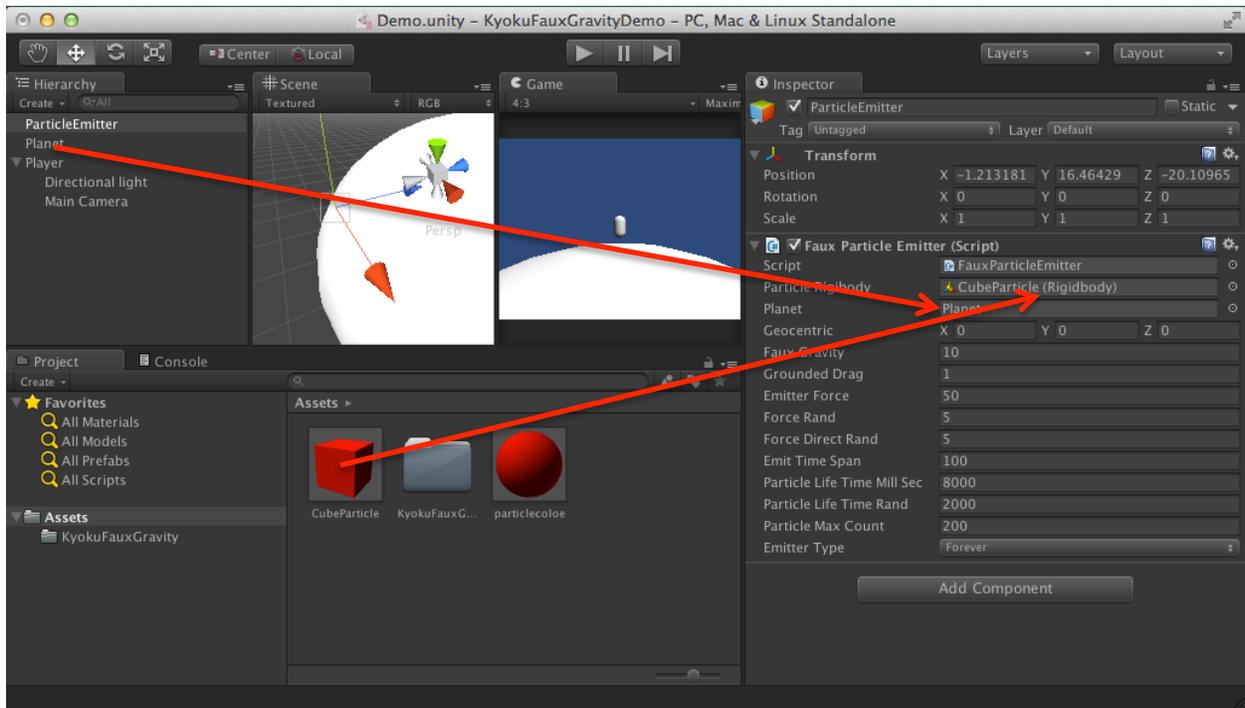
接下來建立一個 **Empty** 來當作發射器。

**GameObject > Create Empty**

將此 **Empty GameObject** 更名為「**ParticleEmitter**」，這個物件將作為粒子的發射器，接著把「**FauxParticleEmitter.cs**」拉到 **ParticleEmitter** 上。



接下來選取 **ParticleEmitter** 物件，將 **Planet** 發射器的 **Planet** 屬性，把粒子的 **Prefab** 拉到 **ParticleRigidbody** 屬性上。



然後將 Emitter Force 改為 50，將 Emit Time Span 改為 100，將 Particle Max Count 改成 200。

其他的屬性我也介紹一下。

Particle Rigibody：粒子物件

Planet：星球

Geocentric：地心偏移量

Faux Gravity：引力強度

Grounded Drag：粒子掉到地面後的摩擦力

Emitter Force：粒子發射力場強度

Force Rand：粒子發射強度亂數變化

Force Direct Rand：粒子發射方向亂數變化

Emit Time Span：每一波粒子發射時間間隔

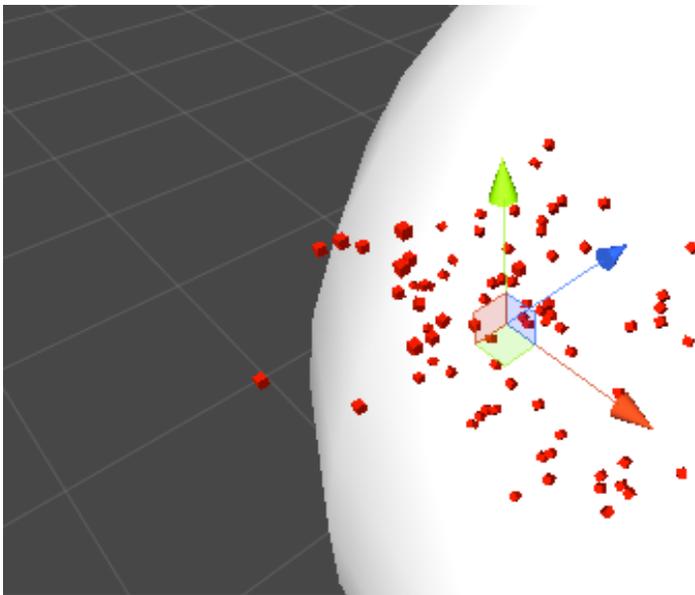
Particle Life Time Mill Sec：粒子生命長度(微秒)

Particle Life Time Rand：粒子生命亂數

Particle Max Count：在場景中的粒子最大值

Emitter：只發射一次或持續發射

移動發射器到適當位置，執行後粒子便可以發射了。



# 版權聲明

---

本教學僅供教學及自習之用，可完全免費取得，除非得到作者紀曲峰本人書面授權，禁止內於任何印刷品或任何收費行為之用，若用於教學，僅能向學員收取印刷工本費，禁止另收講義編輯費。

本教學中所有圖、文均屬作者紀曲峰及原 Unity 公司及其他相關公司所有，若要利用於任何教學或免費轉載之用，禁止刪除版權聲明及作者介紹等項目。

**作者：紀曲峰(M.K.)**

**Email： [mkii@ms49.hinet.net](mailto:mkii@ms49.hinet.net)**

**HomePage： <http://www.digiart.com.tw>**

※ 若有任何問題需要 Email 聯絡我，請在主旨註明 Photon 問題、Unity3D 問題、Maya 問題等，若只填入打招呼字句會被我直接移到垃圾郵件箱。