

# 2013

## Fog of War 使用説明



Kyoku

2013/12/3

# 目錄

---

一、安裝.....	1
1. 下載 .....	1
2. 匯入 .....	1
二、內建範例.....	2
1. 開啟範例 .....	2
2. 修正範例的旋轉視窗錯誤 .....	2
3. 修正「Horizontal2」和「Vertical2」的錯誤.....	4
4. 修正 Layer 遺失.....	6
三、在自己的場景中使用 Fog of War.....	9
1. 開新場景 .....	9
2. 建立 Fog 物件.....	10
3. 建立角色 .....	12
4. 建立迷霧攝影機 .....	16
5. 調整 Fog-of-War 使其覆蓋在地面上.....	18
6. 修正 Mesh 錯誤 .....	19
7. 執行及調整 .....	22
版權聲名.....	24

# 一、安裝

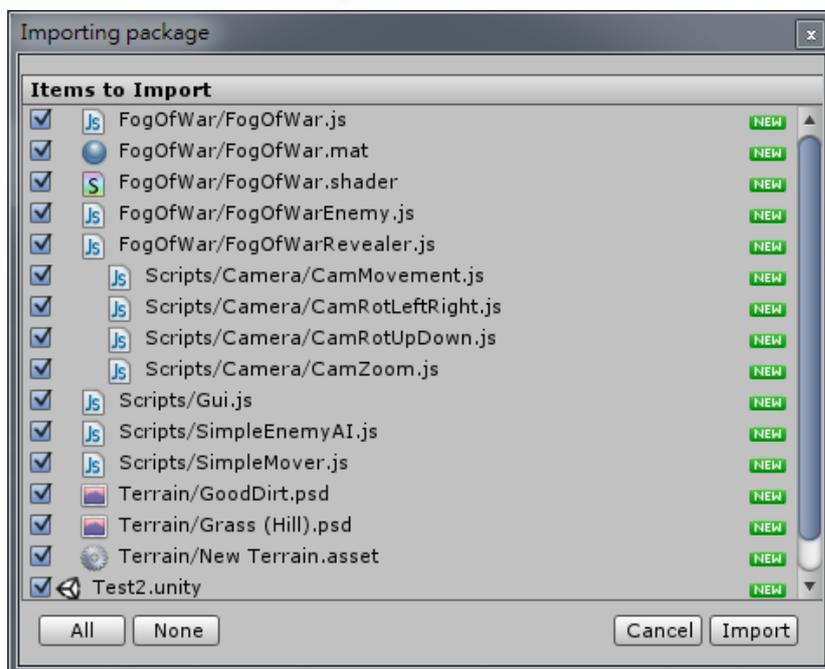
## 1. 下載

首先先到以下網址下載「FogOfWarTest-Revision7.unitypackage」

<http://anomalousunderdog.webs.com/UnityProjectUploads/>

## 2. 匯入

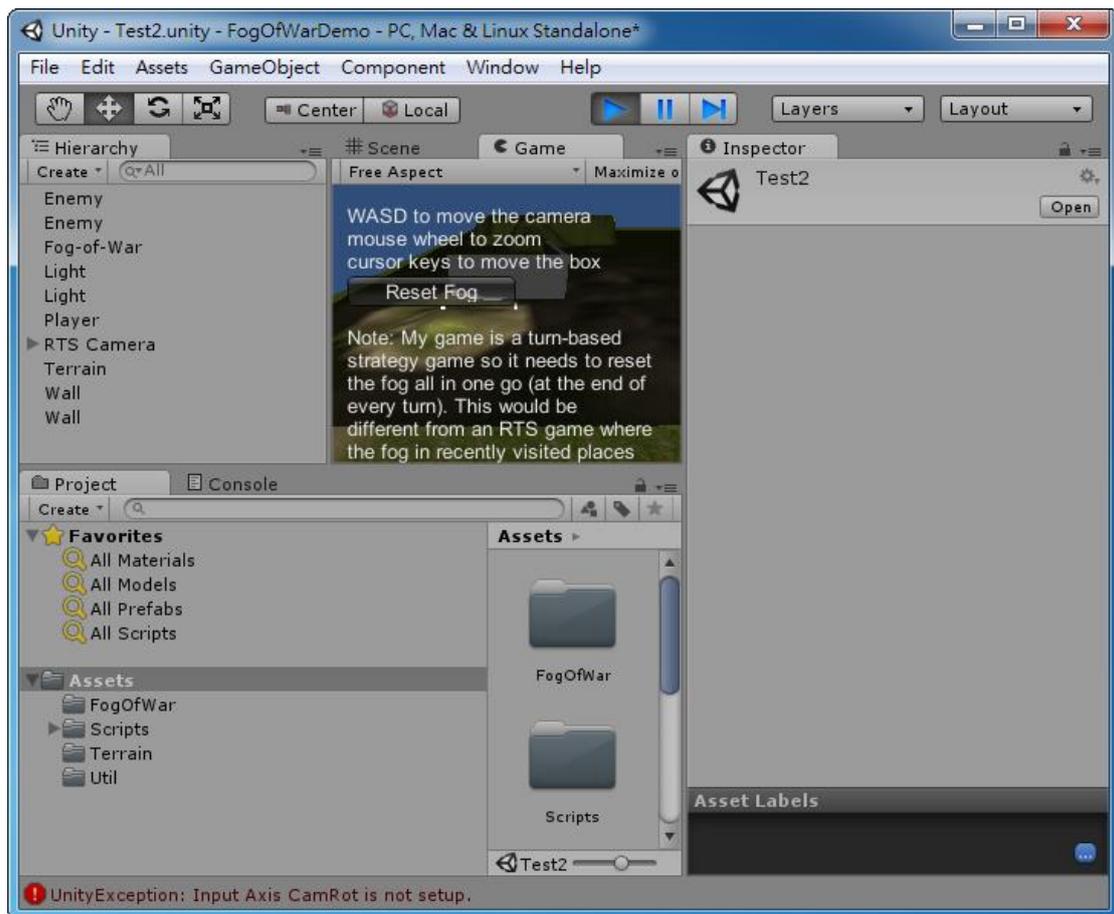
開一個新專案並匯入 FogOfWar 資源包，選擇全部匯入



## 二、內建範例

### 1. 開啟範例

裡面有一個 Test2 的 Scene，打開這個範例，但此範例執行後會產生錯誤。



### 2. 修正範例的旋轉視窗錯誤

會發生錯誤是因為「Project settings」沒有定義「CamRot」事件，我們把這個 key 換掉即可，開啟「CamRotLeftRight.js」檔，修改程式碼，把原本的 CamRot 註解掉，改成「c」

```

var speed = 100;

function Update ()
{
    //if (Input.GetAxis("CamRot") > 0)
    if( Input.GetKey('c') )
    {
        rot = Vector3(0, Input.GetAxis("Mouse X"), 0);
        transform.Rotate(rot * Time.deltaTime * speed);
    }
}

```

開啟「CamRotUpDown.js」，一樣把原本的 CamRot 註解掉，改成「c」。

```

var speed = 100;
var upperLimit = 90;
var lowerLimit = 10;

function Update ()
{
    var my = Input.GetAxis("Mouse Y");
    //if ((Input.GetAxis("CamRot") > 0))
    if( Input.GetKey('c') )
    {
        if (((my < 0) && (transform.localEulerAngles.x <= upperLimit)) ||
            ((my > 0) && (transform.localEulerAngles.x >= lowerLimit)))
        {
            rot = Vector3(-my, 0, 0);
            transform.Rotate(rot * Time.deltaTime * speed);
        }
    }
}

```

現在我們可以按住 c 加上滑鼠移動來旋轉攝影機了。

## 3. 修正「Horizontal2」和「Vertical2」的錯誤

原本的範例有定義 Horizontal2 和 Vertical2 兩個輸入事件，這個也會造成錯誤，這是角色移動用的變數，我們把他改成「直接按方向移動角色，Shift 加方向移動攝影機」，先開啟「SimpleMover.js」，修改程式碼如下。

```
function FixedUpdate()
{
    if( !Input.GetKey(KeyCode.LeftShift) && !Input.GetKey(KeyCode.RightShift) ){
        var controller : CharacterController = GetComponent(CharacterController);
        if (controller.isGrounded)
        {
            // We are grounded, so recalculate
            // move direction directly from axes
            //moveDirection = Vector3(Input.GetAxis("Horizontal2"), 0,
            // Input.GetAxis("Vertical2"));

            moveDirection = Vector3(Input.GetAxis("Horizontal"), 0,
                Input.GetAxis("Vertical"));

            moveDirection = transform.TransformDirection(moveDirection);
            moveDirection *= speed;

            if (Input.GetButton ("Jump"))
            {
                moveDirection.y = jumpSpeed;
            }
        }
    }
}
```

```

    }

    }

    // Apply gravity
    moveDirection.y -= gravity * Time.deltaTime;

    // Move the controller
    controller.Move(moveDirection * Time.deltaTime);
}
}

```

意思是「若沒有按住 Shift 則移動角色」。

開啟「CamMovement.js」修改程式碼如下。

```

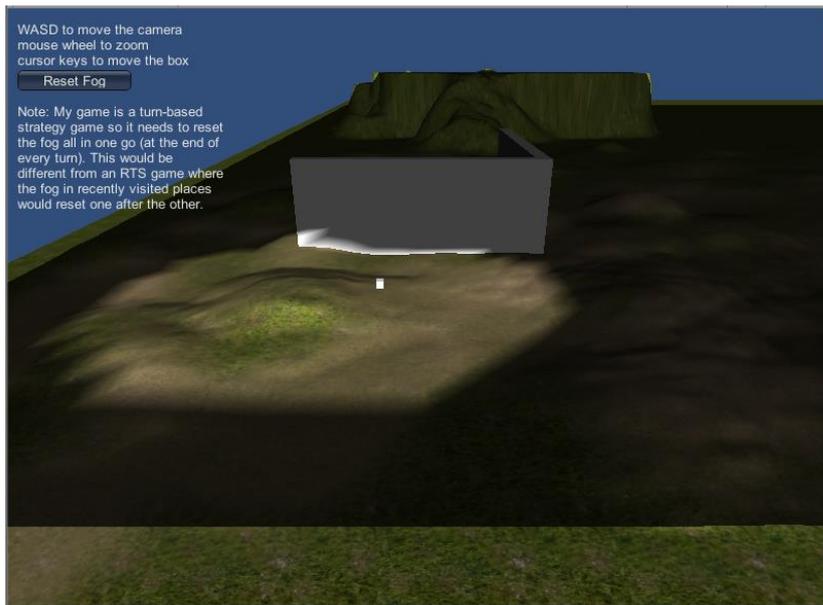
function FixedUpdate()
{
    if( Input.GetKey(KeyCode.LeftShift) || Input.GetKey(KeyCode.RightShift))
    {
        moveDirection = Vector3(Input.GetAxis("Horizontal"), 0,
            Input.GetAxis("Vertical"));
        moveDirection = transform.TransformDirection(moveDirection);
        moveDirection *= speed;

        // Move the camera
        transform.position += moveDirection * Time.deltaTime;
    }
}

```

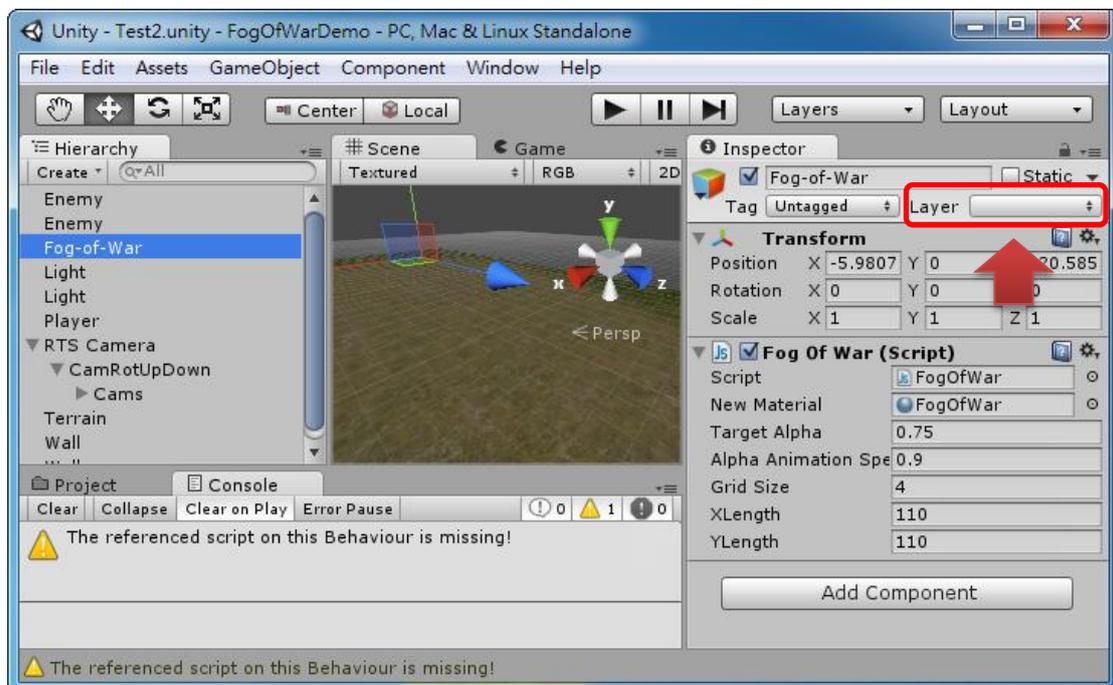
意思是若有按住 Shift 才移動攝影機。

重新執行就不會發生錯誤了，戰爭迷霧也跑得出來。



## 4. 修正 Layer 遺失

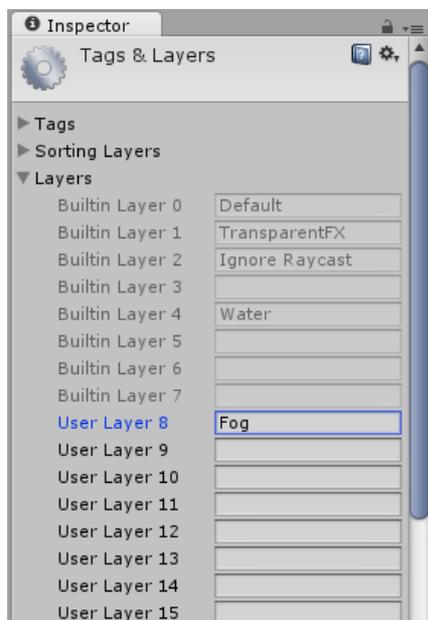
其實這個範例還有一個問題，他的迷霧用的 Layer 不見了，我們選取畫面中「Fog-of-War」物件，可以看到 Layer 變成空的。



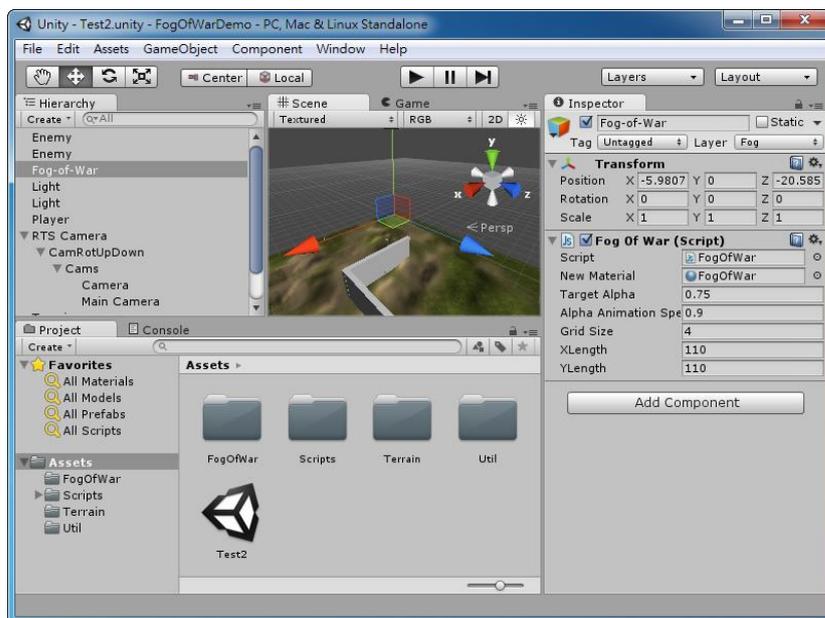
雖然執行起來沒有問題，那是因為 Layer 雖然遺失了但其實 Unity 有記著 Layer

的層編號，但總是不容易閱讀專案，我們還是將他修正一下比較好。

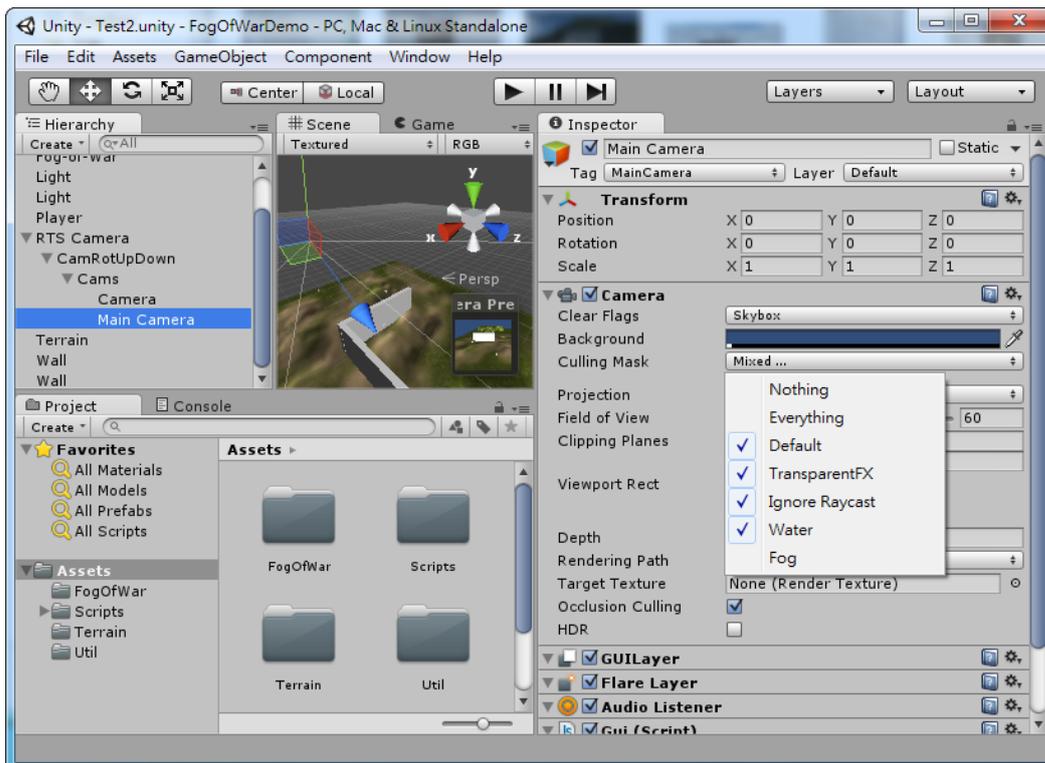
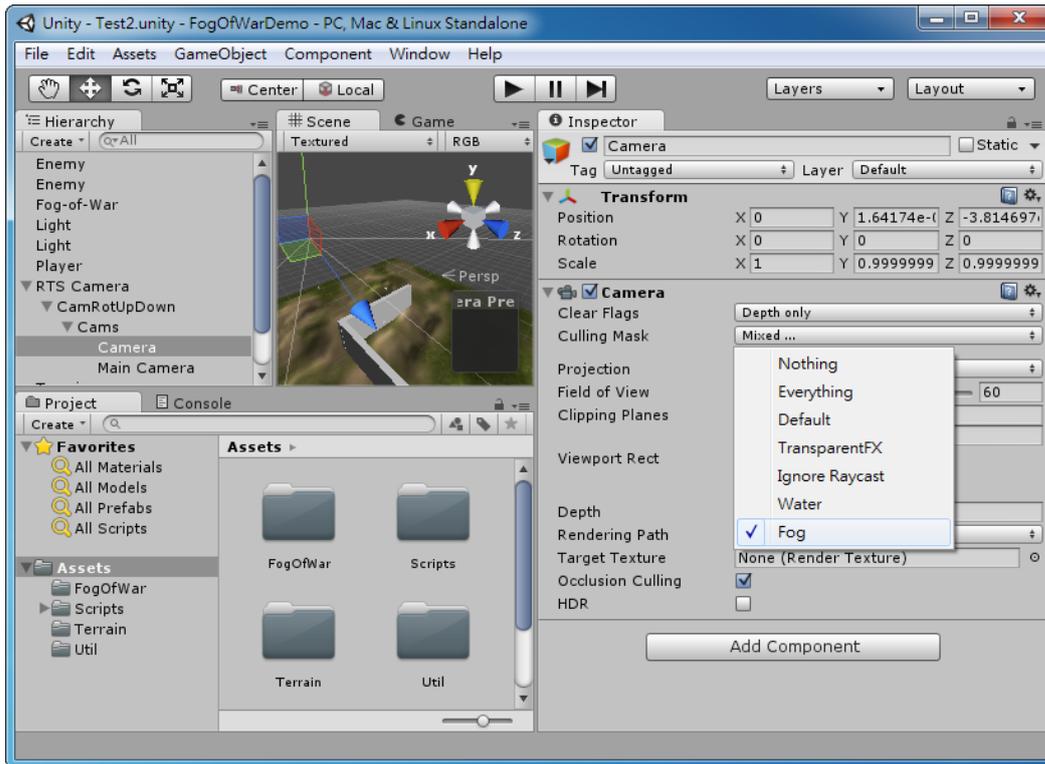
請開啟「Add Layer...」在「User Layer 8」加入「Fog」



重新選取 Fog-of-War，自動選到 Fog 這個 Layer 了。



選到「Camera」及「Main Camera」的 Culling Mask，顯示也正常了。

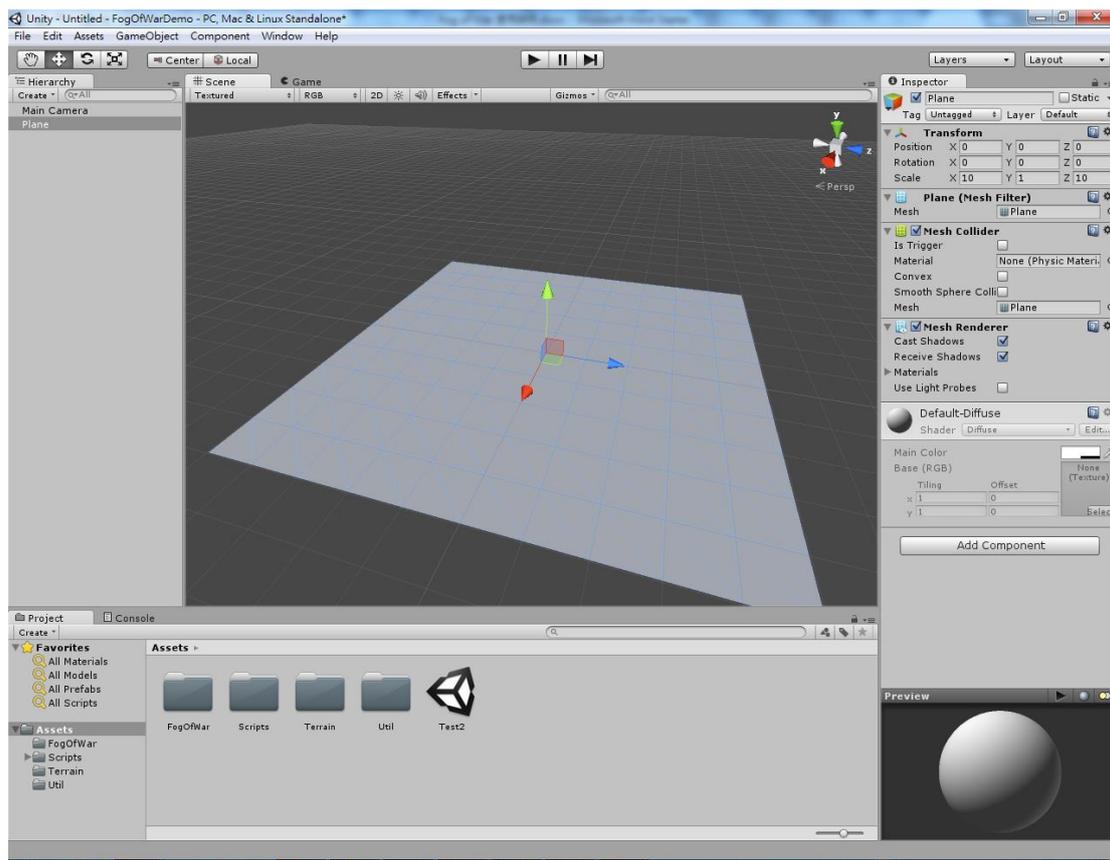


# 三、在自己的場景中使用 Fog of War

雖然成功的開啟了範例文件，但若沒辦法應用在自己的場景那一點意義都沒有，現在就來介紹如何將 FogofWar 用在自己的場景上。

## 1. 開新場景

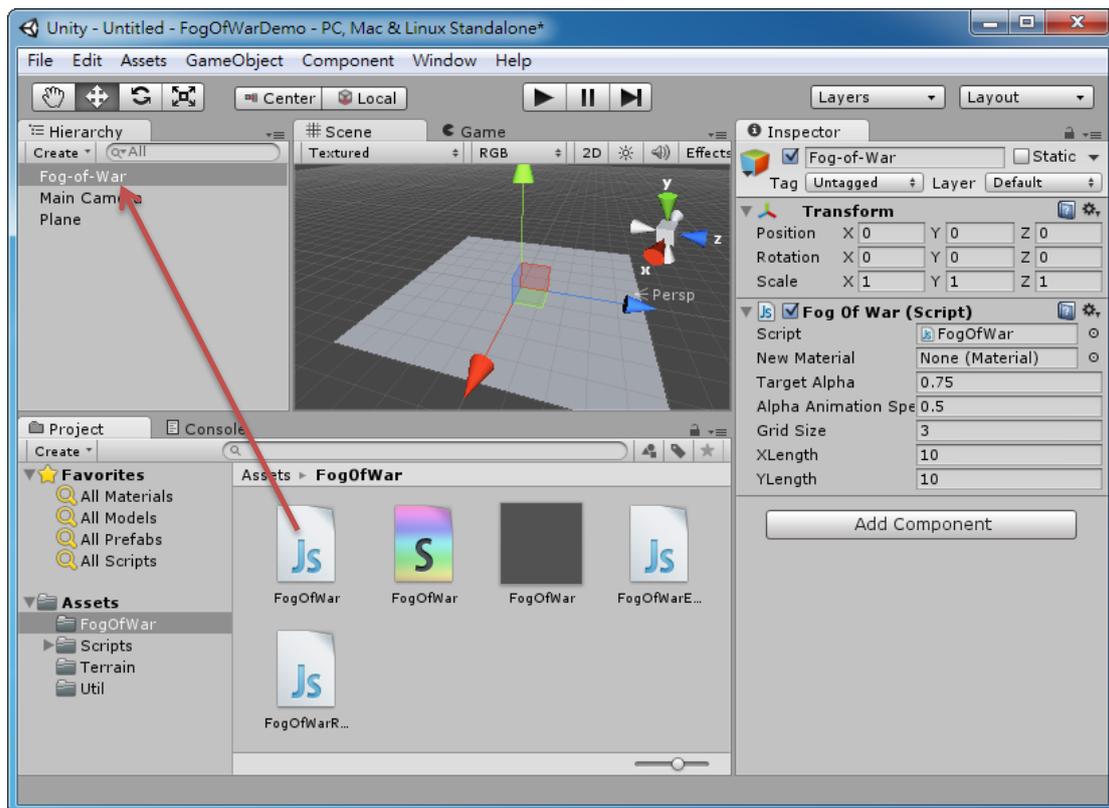
重新開啟一個空的場景，並建立一個 Plane 當作行走的地面，然後將 Plane 的 Scale 設為(10,1,10)。



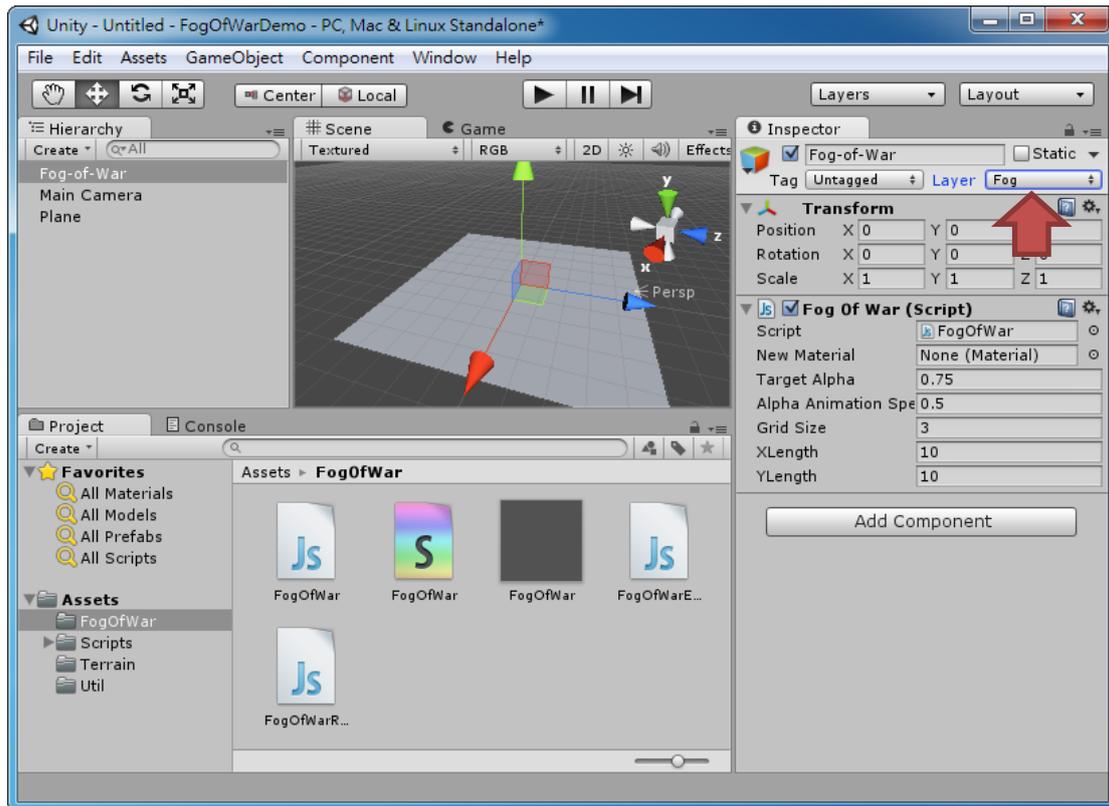
## 2. 建立 Fog 物件

新增一個 Empty 來作為 Fog 物件，執行「GameObject > Create Empty」，然後將這個 Empty 更名為「Fog-of-War」。

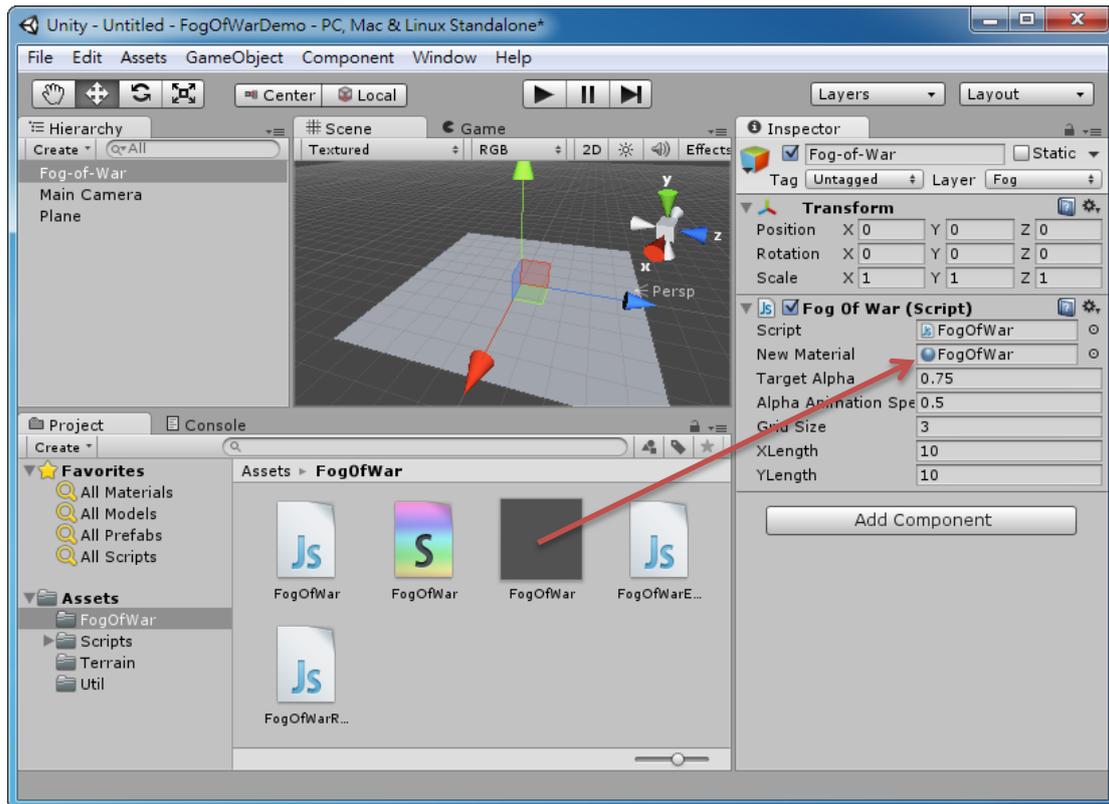
接下來將腳本「FogOfWar\FogOfWar.js」拉到 Fog-of-War 物件上。



然後建立一個 Layer，取名為「Fog」，並將 Fog-of-War 的 Layer 設為 Fog。



由於迷霧是靠不同圖層的攝影機拍攝到另一個迷霧用的遮罩 Plane 並覆蓋在上面，因此我們必需設置迷霧用的材質到「Fog-of-War」物件上，直接拖曳「FogOfWar/FogOfWar.mat」到 Fog-of-War 的腳本上。



Fog-of-War 設置先在這裡告一段落，因為還沒有建立攝影機和角色因此看不出效果，先建立好這兩項再回來繼續設置。

### 3. 建立角色

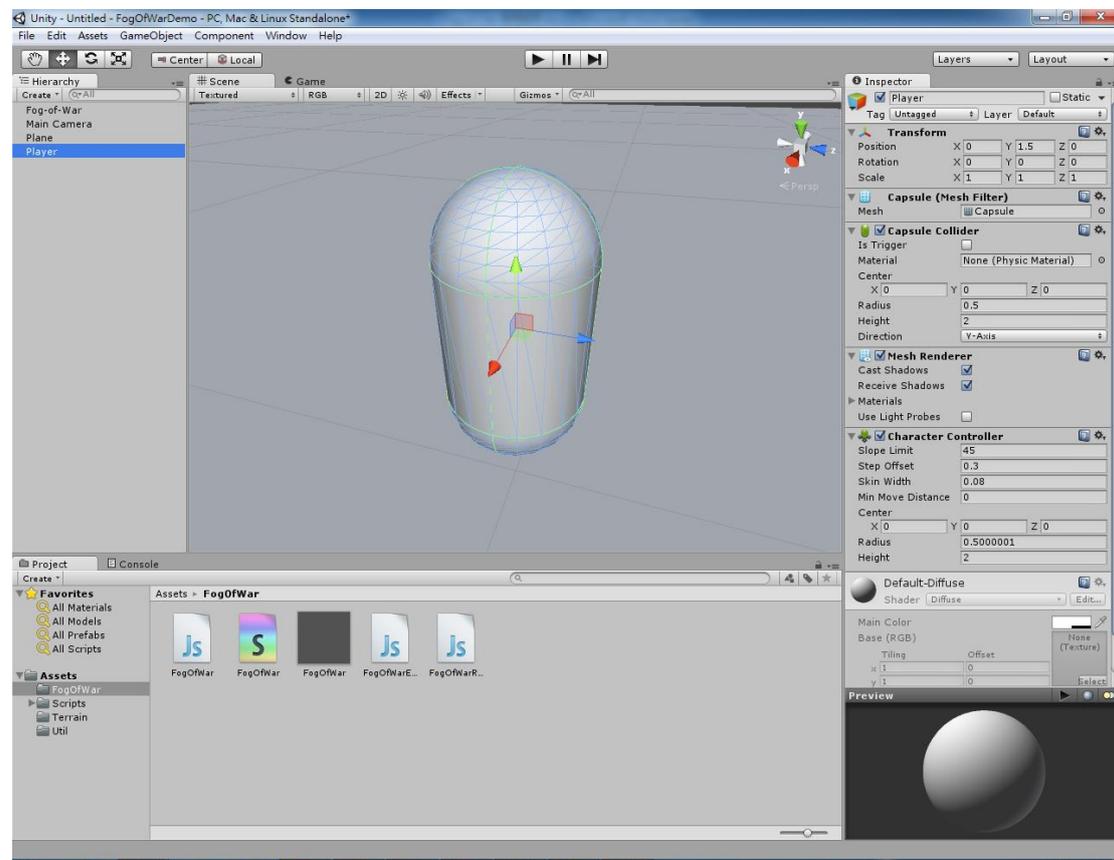
請建立一個膠囊來作為角色，執行。

GameObject > Create Other > Capsule

建立好之後將 Capsule 更名為「Player」，選取這個膠囊，建立角色控制屬性，執行。

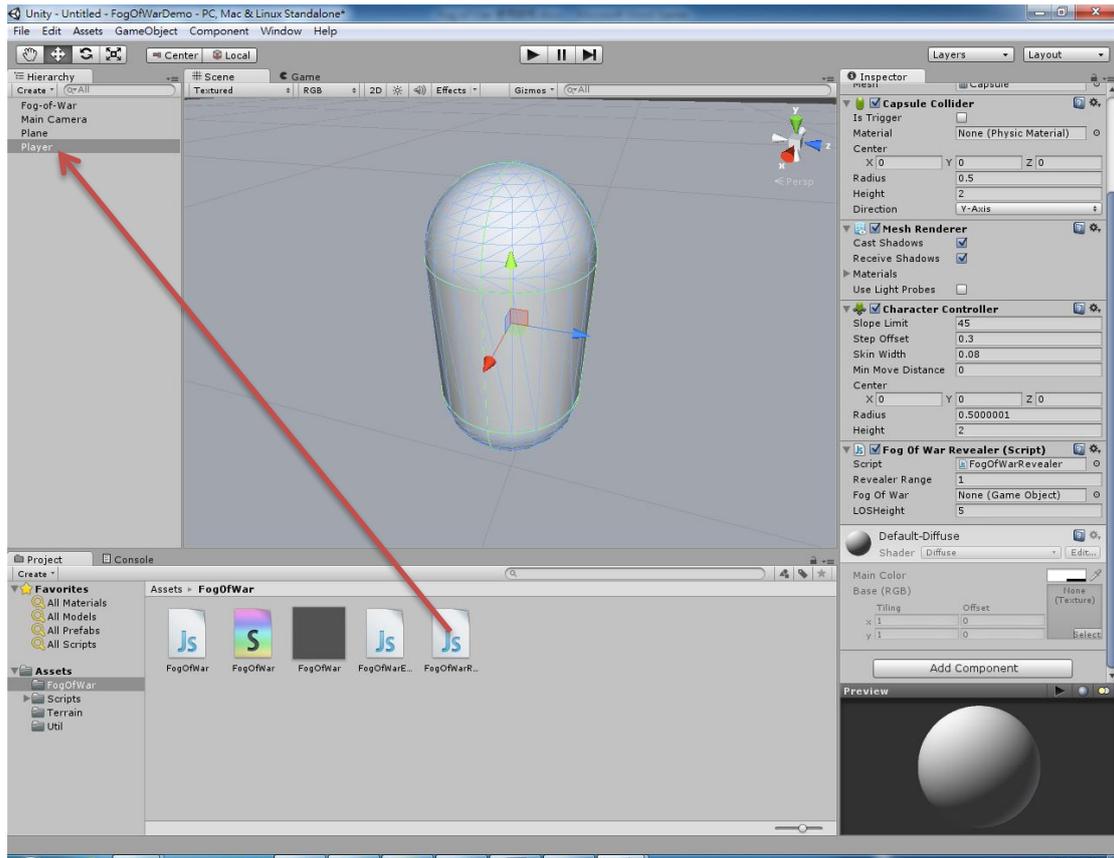
Component > Physics > Character Controller

最後將膠囊的 Position.Y 提高到 1.5，讓角色站在地表面上。



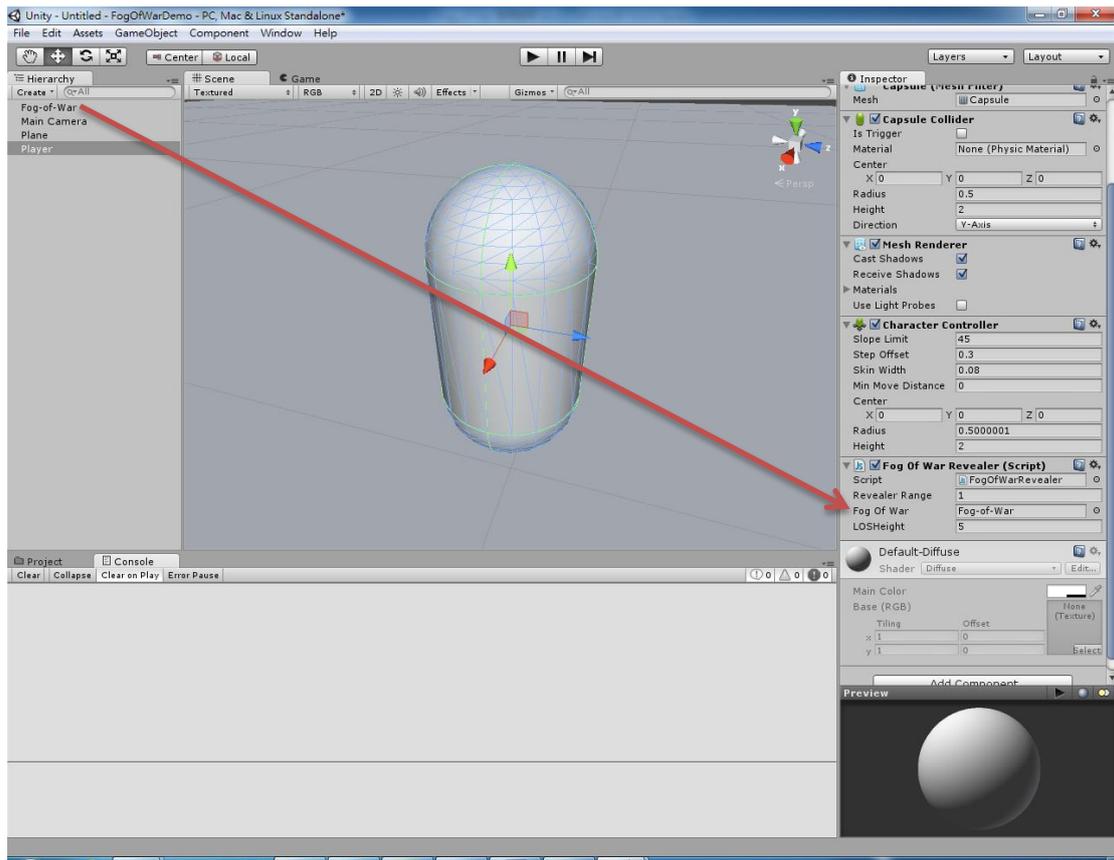
迷霧是利用角色週圍打出一圈的射線，射線擊中到迷霧遮罩的多邊型後將頂點進行著色達到透明的效果，所以要將建立射線的程式碼拉到角色身上。

請將「FogOfWar/FogOfWarRevealer.js」拉到 Player 上。

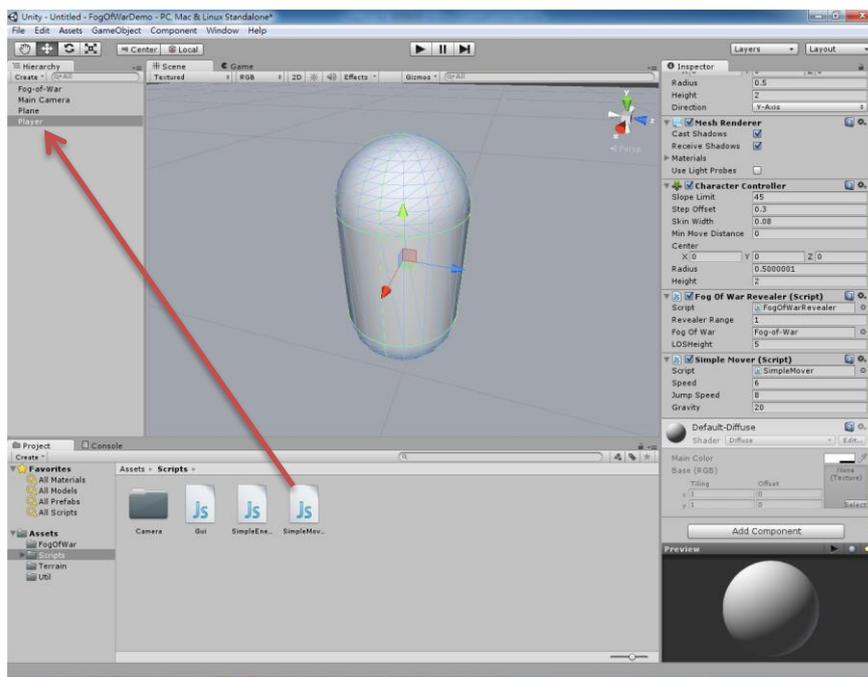


其參數「Revealer Range」可以設置迷霧消失的範圍大小。

接下來將「Fog-of-War」物件拉到 Player 的腳本「FogOfWarRevealer.js」底下的參數「Fog Of War」上。



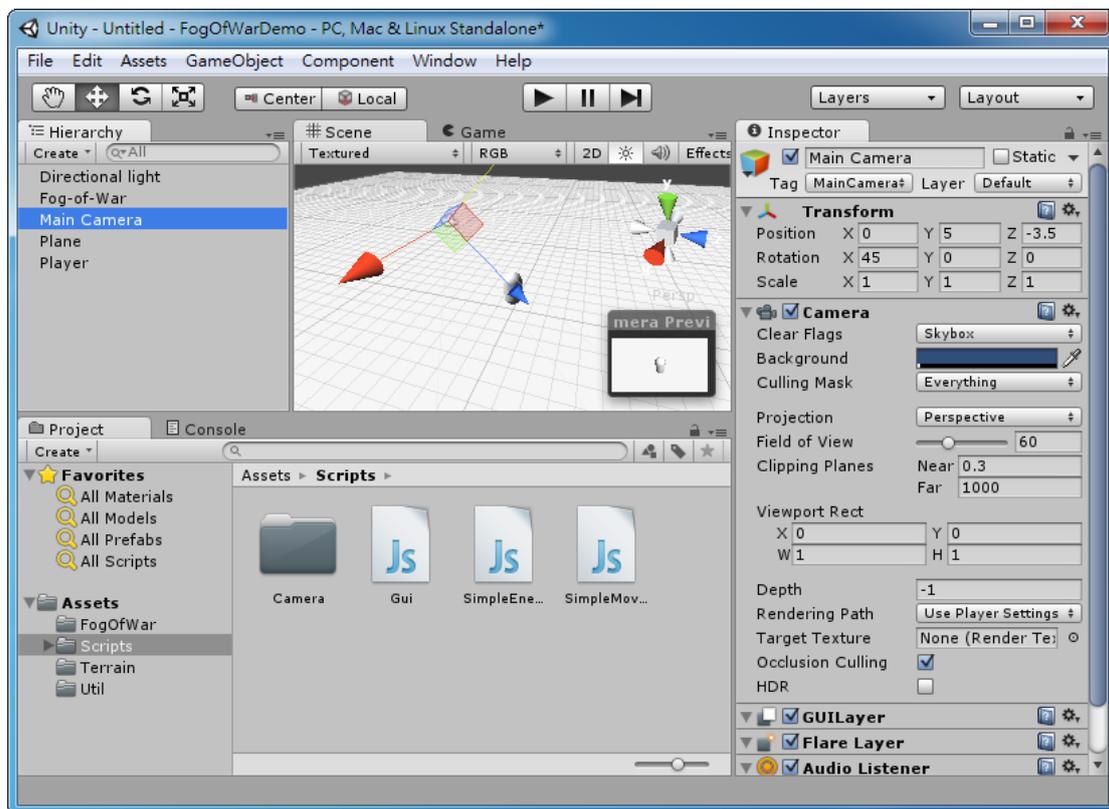
為了讓角色可以移動，將「Scripts/SimpleMover.js」拉到 Player 身上。



現在只要執行程式角色已經可以移動了，但還是會發生一些錯誤，不用在意，之後會全部修正好。

## 4. 建立迷霧攝影機

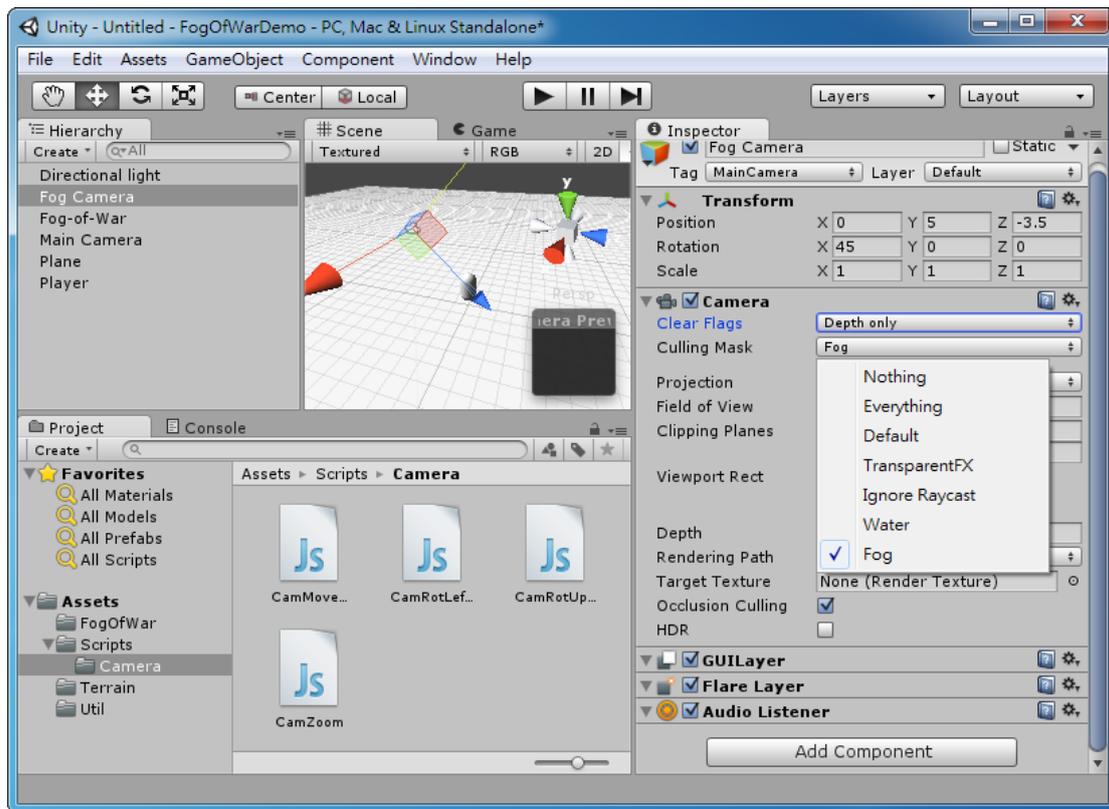
我們先調整攝影機角度，讓攝影機可以照到角色，先打一盞平行燈讓場景亮一點，然後調整攝影機，調整攝影機的角度為 Rotation 為(45,0,0)，Position 改為(0,5,-3.5)，當然，這只是參考值，可以自行調整數值。



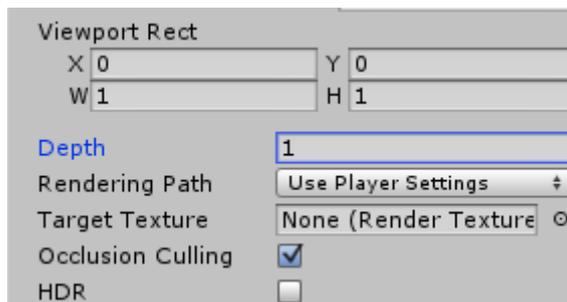
調整好之後，將「Main Camera」複製一個出來，並更名為「Fog Camera」，兩個 Camera 的位置及角度必須一樣或相近。

選取 Fog Camera，將「Clear Flags」設為 Depth Only，並將「Culling Mask」設為

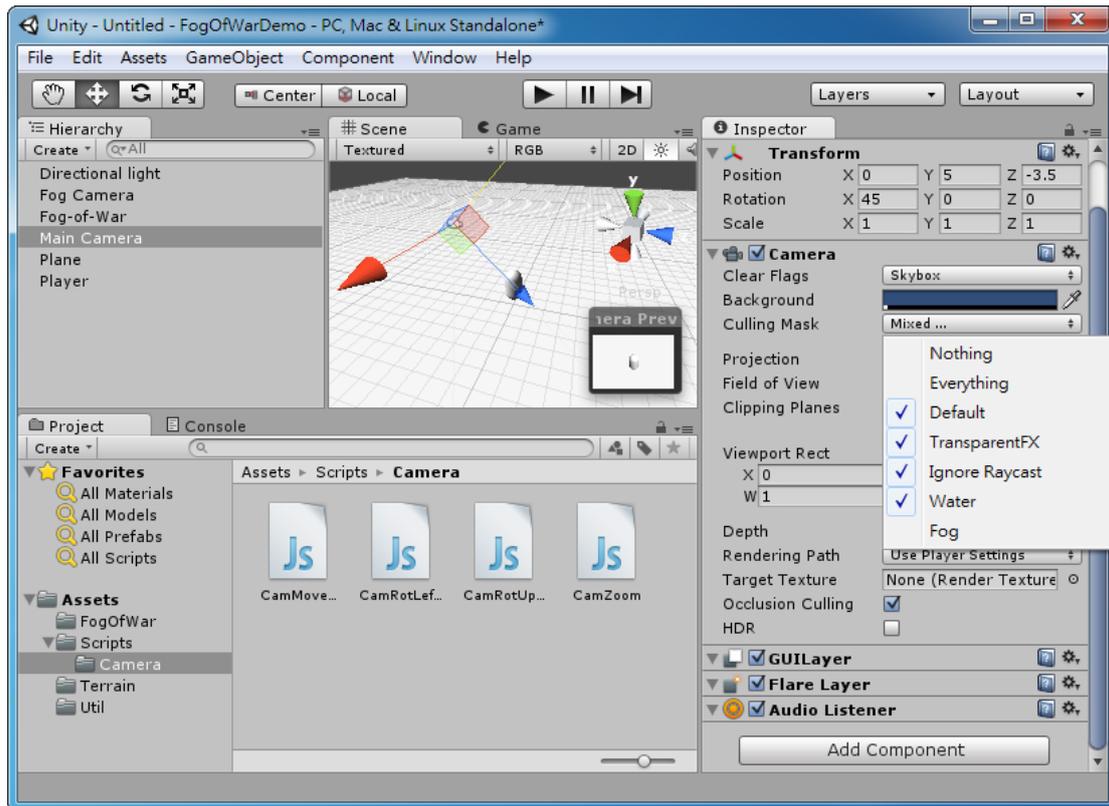
「Fog」



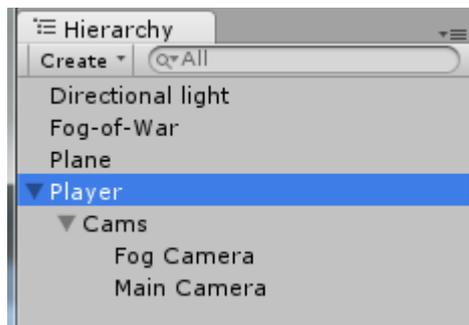
然後將 Depth 設為「1」



接下來選取 Main Camera，將「Culling Mask」的「Fog」層拿掉。



最後建立一個 Empty 取名為 Cams，將兩個攝影機都拉到此物件下，再將 Cams 拉到 Player 下，讓攝影機可以跟隨著角色移動。



## 5. 調整 Fog-of-War 使其覆蓋在地面上

現在我們的迷霧遮罩放在奇怪的位置上，沒有遮到地面，所以執行起來怪怪的，

調整一些參數使其正確的覆蓋。

選取 Fog-of-War 物件，調整 Position 為(-50,0,-50)，然後修改 Grid Size 為 4，並將 XLength 和 YLength 改為 25，若你的地面大小不一樣請自行修改調整。



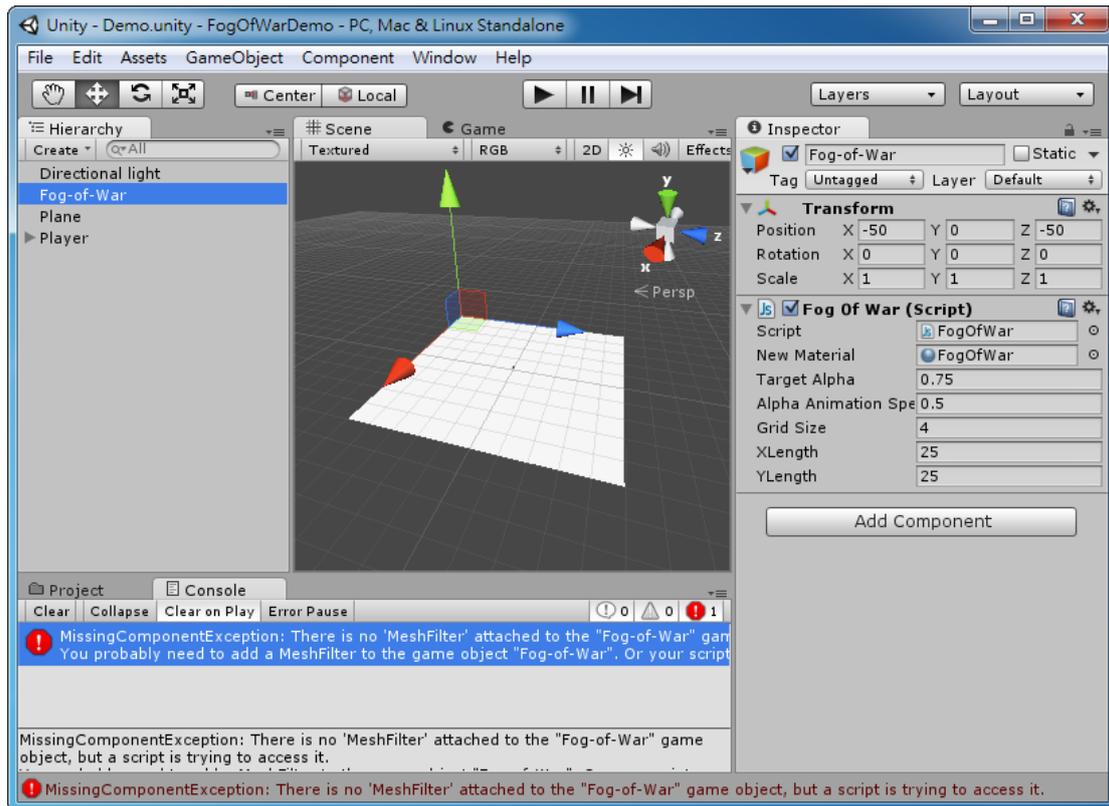
## 6. 修正 Mesh 錯誤

現在已經可以執行了，但執行之後會顯示一排錯誤：

MissingComponentException: There is no 'MeshFilter' attached to the "Fog-of-War" game object, but a script is trying to access it.

You probably need to add a MeshFilter to the game object "Fog-of-War". Or your script needs to check if the component is attached before using it.

FogOfWarRevealer.Start () (at Assets/FogOfWar/FogOfWarRevealer.js:14)



意思是 FogOfWar 要設置一個 mesh 物件，可是找不到此物件，錯誤發生在腳本「FogOfWarRevealer.js」的這裡。

```
function Start()
{
    fogOfWarMesh = fogOfWar.GetComponent(MeshFilter).mesh;
    fogOfWarScript = fogOfWar.GetComponent(FogOfWar);

    fogOfWarScript.AddRevealer(this);

    InitialReveal();
}
```

打開「FogOfWar.js」找到 MeshFilter，可以看到以下的程式碼。

```
function Start()
{
    gameObject.AddComponent("MeshFilter");
    gameObject.AddComponent("MeshRenderer");
    mesh = GetComponent(MeshFilter).mesh;

    mesh.Clear();

    .....
}
```

因為執行順序的問題，在執行到 `FogOfWarRevealer.js` 的 `Start` 時，`FogOfWar.js` 的 `Start()` 還沒有被執行到，因此 `MeshFilter` 還沒有被建立起來。

最簡單的解決方法是直接修改 `FogOfWar.js` 的 `Start`，改成 `Awake()` 以確保 `MeshFilter` 會被優先建立起來。

```
//function Start()
function Awake()
{
    gameObject.AddComponent("MeshFilter");
    gameObject.AddComponent("MeshRenderer");
    mesh = GetComponent(MeshFilter).mesh;

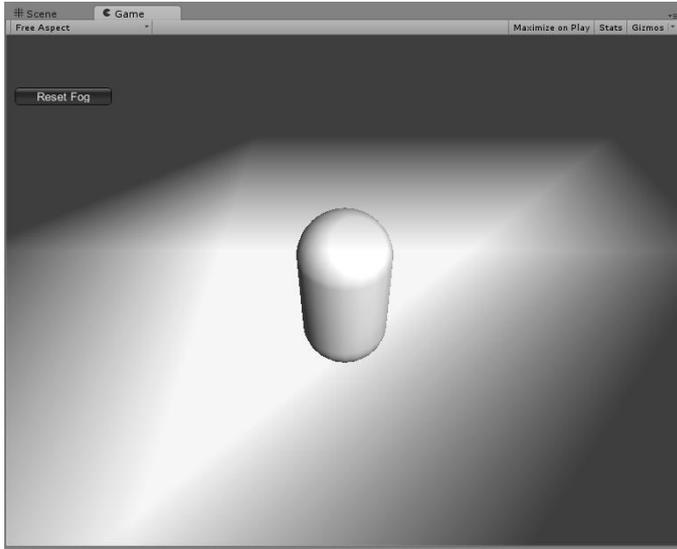
    mesh.Clear();

    .....
}
```

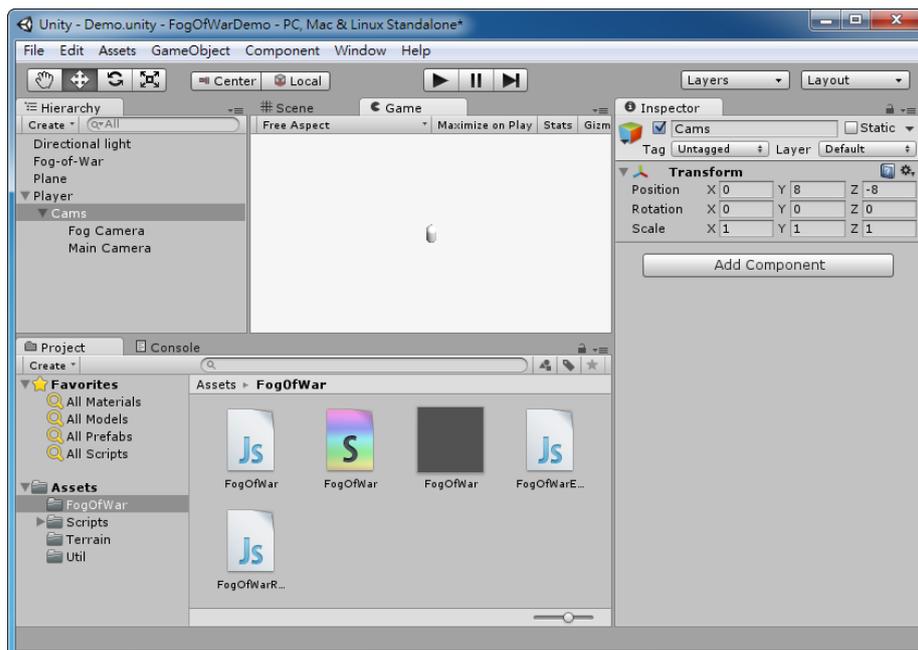
如此一來便不會再發生讀不到 `MeshFilter` 的錯誤了。

## 7. 執行及調整

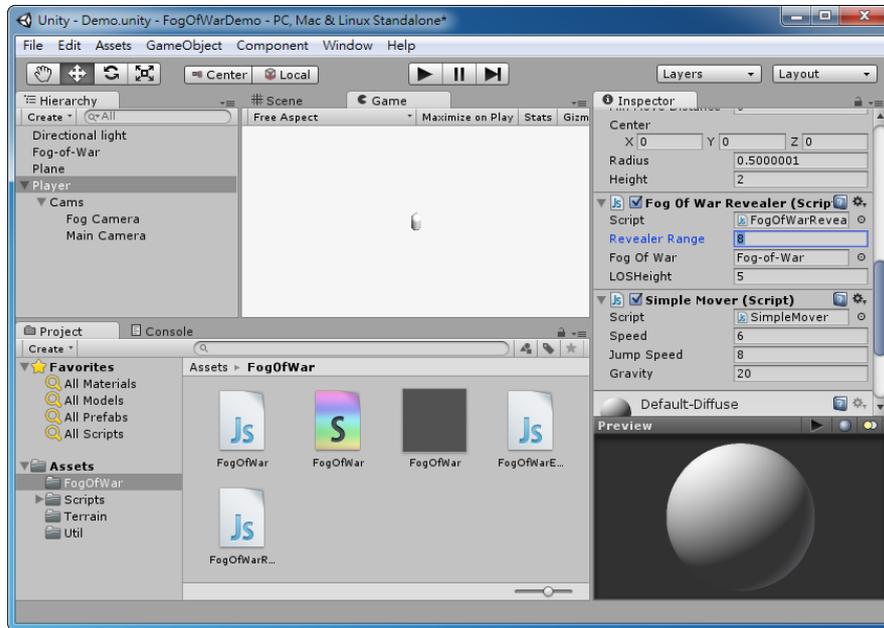
執行後覺得攝影機太近了，而且迷霧亮部的範圍也太小，迷霧解開的速度也希望可以快一點。



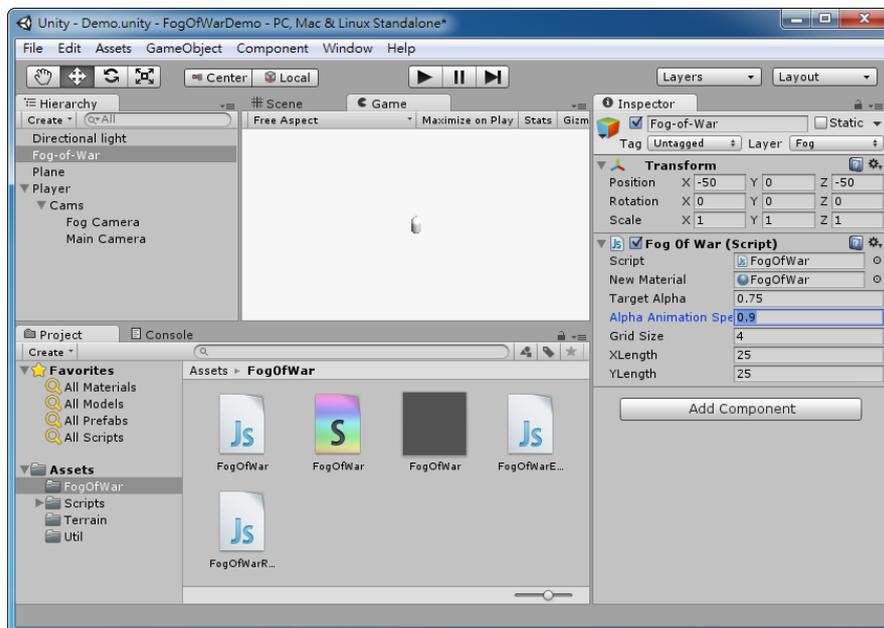
接下來做點調整，先選取「Cams」物件，將 Position 調為(0,8,-8)，將攝影機拉遠一點。



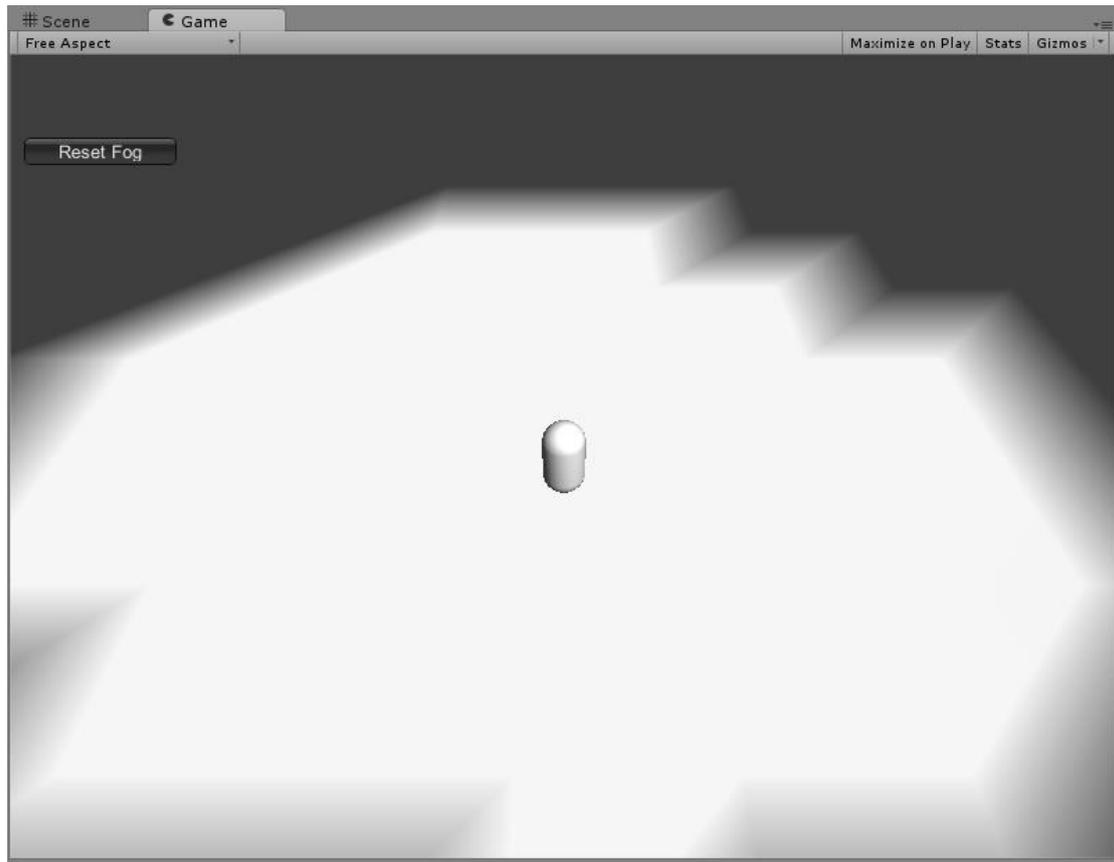
選取 Player，將 Fog Of War Revealer 的 Revealer Range 改為「8」，加大亮部的範圍。



最後選取 Fog-of-War，將 Alpha Animation Speed 改為 0.9，讓亮起來的速度變快。



重新執行後效果變得很好了。



## 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：[mkkii@ms49.hinet.net](mailto:mkkii@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。