

# Virtual Reality Haptic Interactions

## (虛擬實境與觸覺回饋互動)

Lecturer: Ray  
Week 10 (11/13)

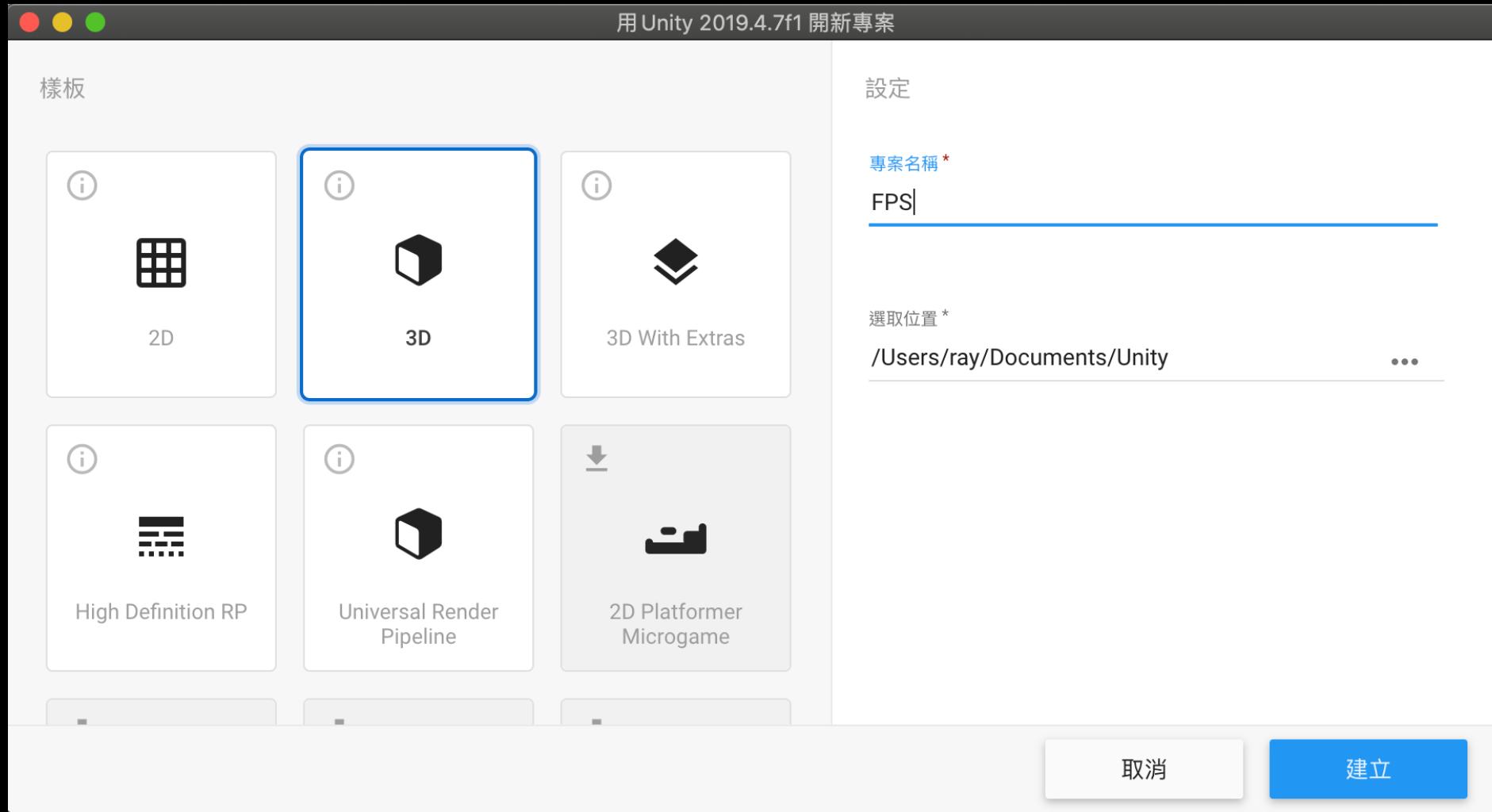
# 本周目標

製作一個基本的第一人稱射擊遊戲

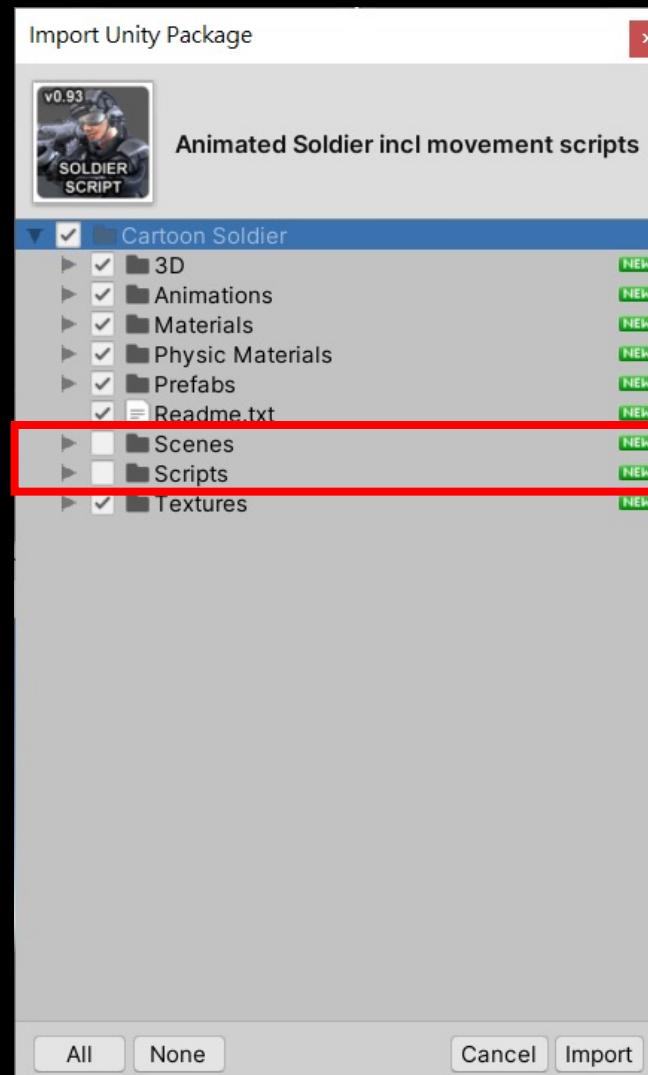
## Download assets:

- <https://bit.ly/3P43vDI>
- <https://bit.ly/3Fa6Yw3>
- <https://bit.ly/3Bg0333>
- <https://bit.ly/3BdRiqv>
- <https://bit.ly/3VA0IEF>

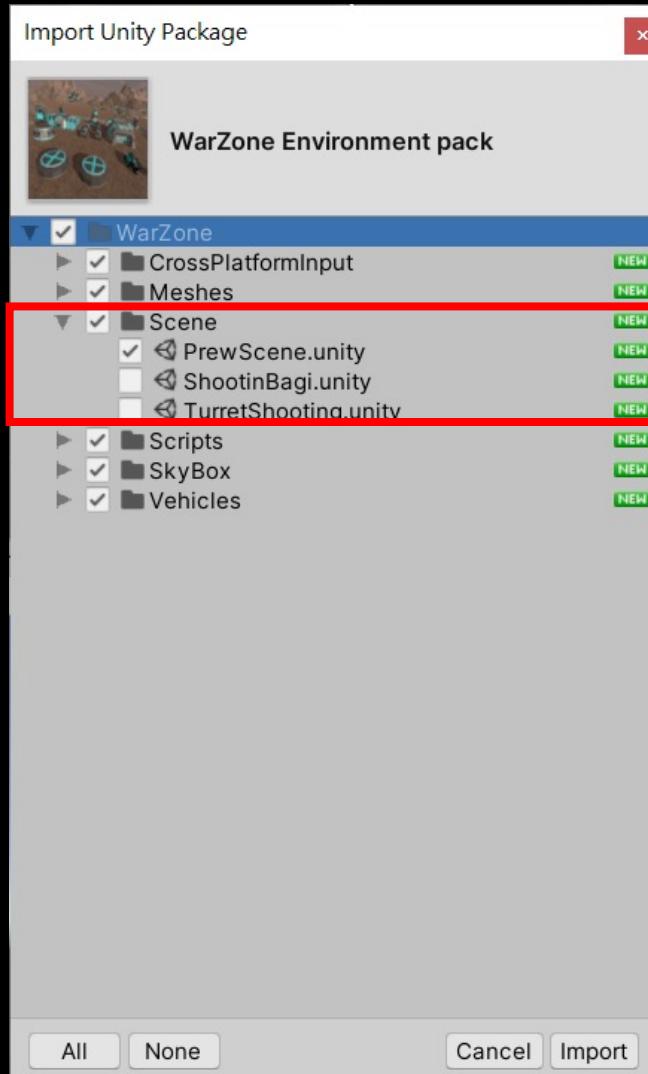
# 建立3D專案



# 主角不需要Scenes跟Scripts



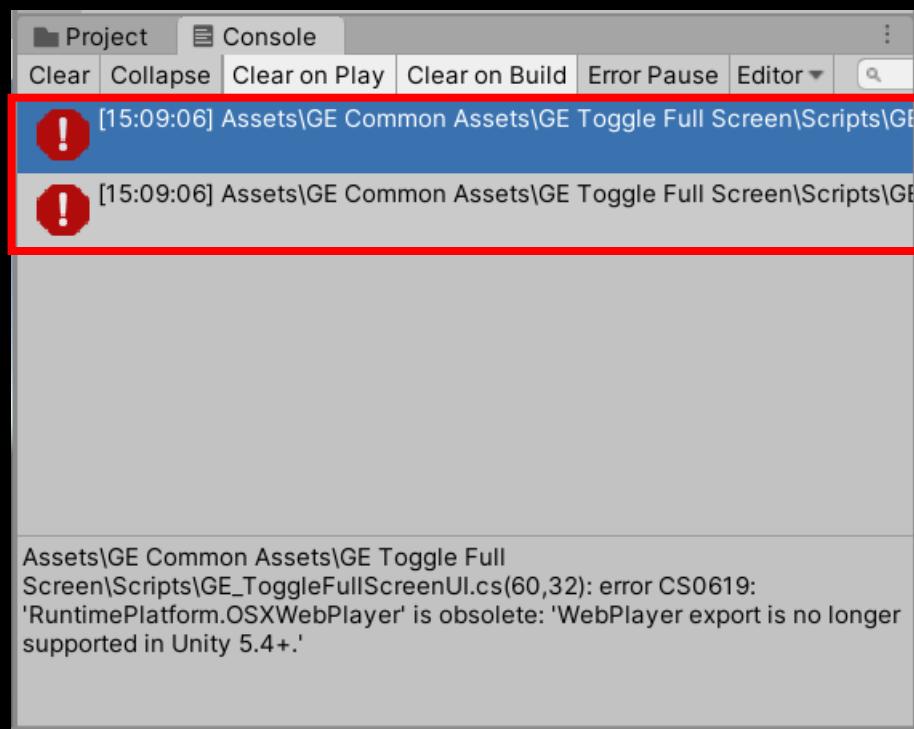
# 場景只需要PrewScene



# 特效全都要



# 雙擊Error Message去解決一下錯誤



# 刪掉報錯的地方

```
// Show Toogle Full Screen button when player is Unity Web, WebGL, Standalone
if (Application.platform == RuntimePlatform.WebGLPlayer ||
    Application.platform == RuntimePlatform.WindowsPlayer ||
    Application.platform == RuntimePlatform.OSXPlayer ||
    Application.platform == RuntimePlatform.LinuxPlayer)
{
    this.gameObject.SetActive (true);
}

// Show Toogle Full Screen button when player is Unity Web, WebGL, Standalone
if (Application.platform == RuntimePlatform.OSXWebPlayer ||
    Application.platform == RuntimePlatform.WindowsWebPlayer ||
    Application.platform == RuntimePlatform.WebGLPlayer ||
    Application.platform == RuntimePlatform.WindowsPlayer ||
    Application.platform == RuntimePlatform.OSXPlayer ||
    Application.platform == RuntimePlatform.LinuxPlayer)
{
    this.gameObject.SetActive (true);
}
// Other cases hide the button
else
{
    this.gameObject.SetActive (false);
}
```

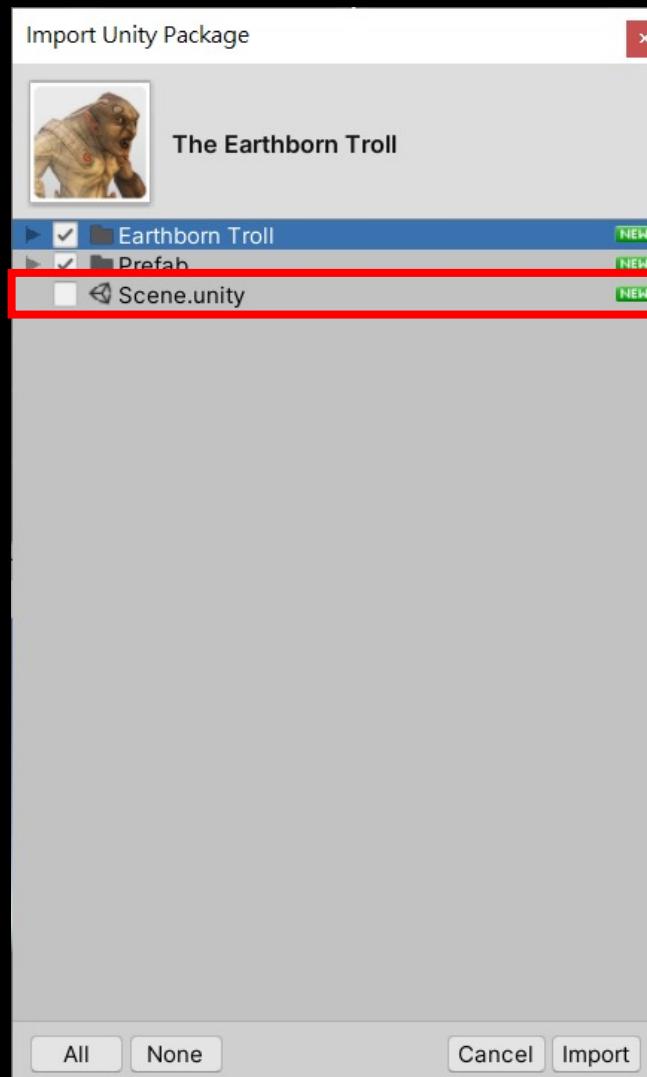
The image shows a Microsoft Visual Studio interface with two sections of C# code in a file named `GE_ToggleFullScreenUI.cs`. A red arrow points from the bottom section to the top section, indicating that the code in the bottom section should be deleted. The code is as follows:

```
// Show Toogle Full Screen button when player is Unity Web, WebGL, Standalone
if (Application.platform == RuntimePlatform.WebGLPlayer ||
    Application.platform == RuntimePlatform.WindowsPlayer ||
    Application.platform == RuntimePlatform.OSXPlayer ||
    Application.platform == RuntimePlatform.LinuxPlayer)
{
    this.gameObject.SetActive (true);
}

// Show Toogle Full Screen button when player is Unity Web, WebGL, Standalone
if (Application.platform == RuntimePlatform.OSXWebPlayer ||
    Application.platform == RuntimePlatform.WindowsWebPlayer ||
    Application.platform == RuntimePlatform.WebGLPlayer ||
    Application.platform == RuntimePlatform.WindowsPlayer ||
    Application.platform == RuntimePlatform.OSXPlayer ||
    Application.platform == RuntimePlatform.LinuxPlayer)
{
    this.gameObject.SetActive (true);
}
// Other cases hide the button
else
{
    this.gameObject.SetActive (false);
}
```

The code uses `RuntimePlatform` enum values to check the current platform and set the active state of a game object accordingly. The bottom section of the code is highlighted with a red box, and the top section is also highlighted with a red box, with a red arrow pointing from the bottom to the top.

# 怪物不需要場景檔



# 複習一下在Asset Store載素材

Assets ▾ Tools ▾ Services ▾ By Unity ▾ Industries ▾ Sell Assets Feedback FAQ

**DOTween (HOTween v2)**

Demigiant ★★★★★ 5 | 207 Reviews

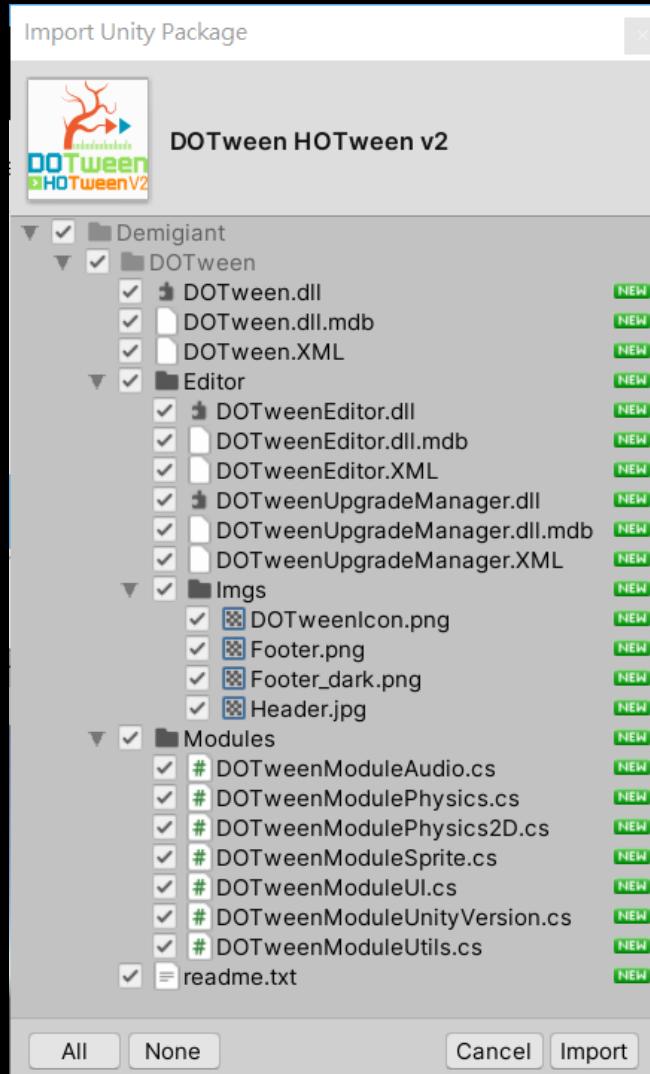
**FREE**

[Open in Unity](#)

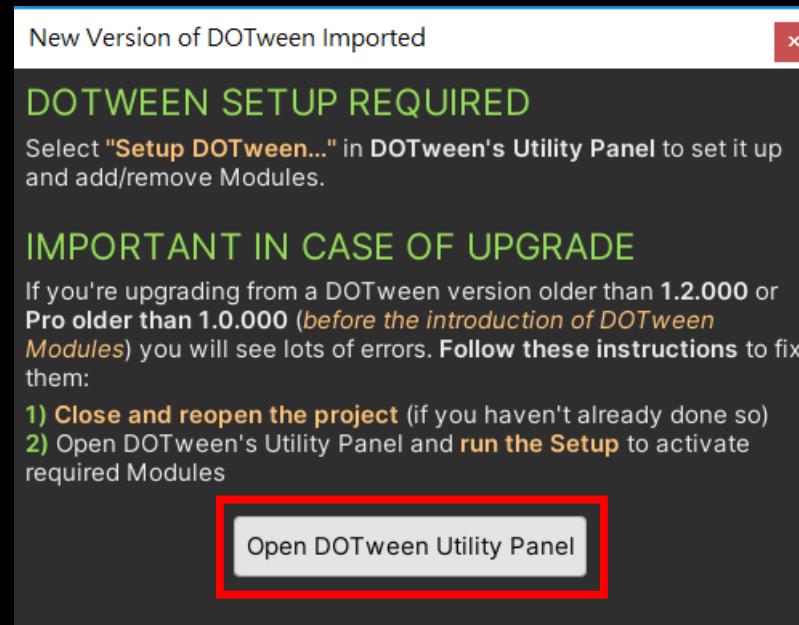
[Love](#) [Add to List](#) [Share](#)

|                        |                 |
|------------------------|-----------------|
| License                | Extension Asset |
| File size              | 306.7 KB        |
| Latest version         | 1.2.335         |
| Latest release date    | Feb 3, 2020     |
| Support Unity versions | 4.6.0 or higher |

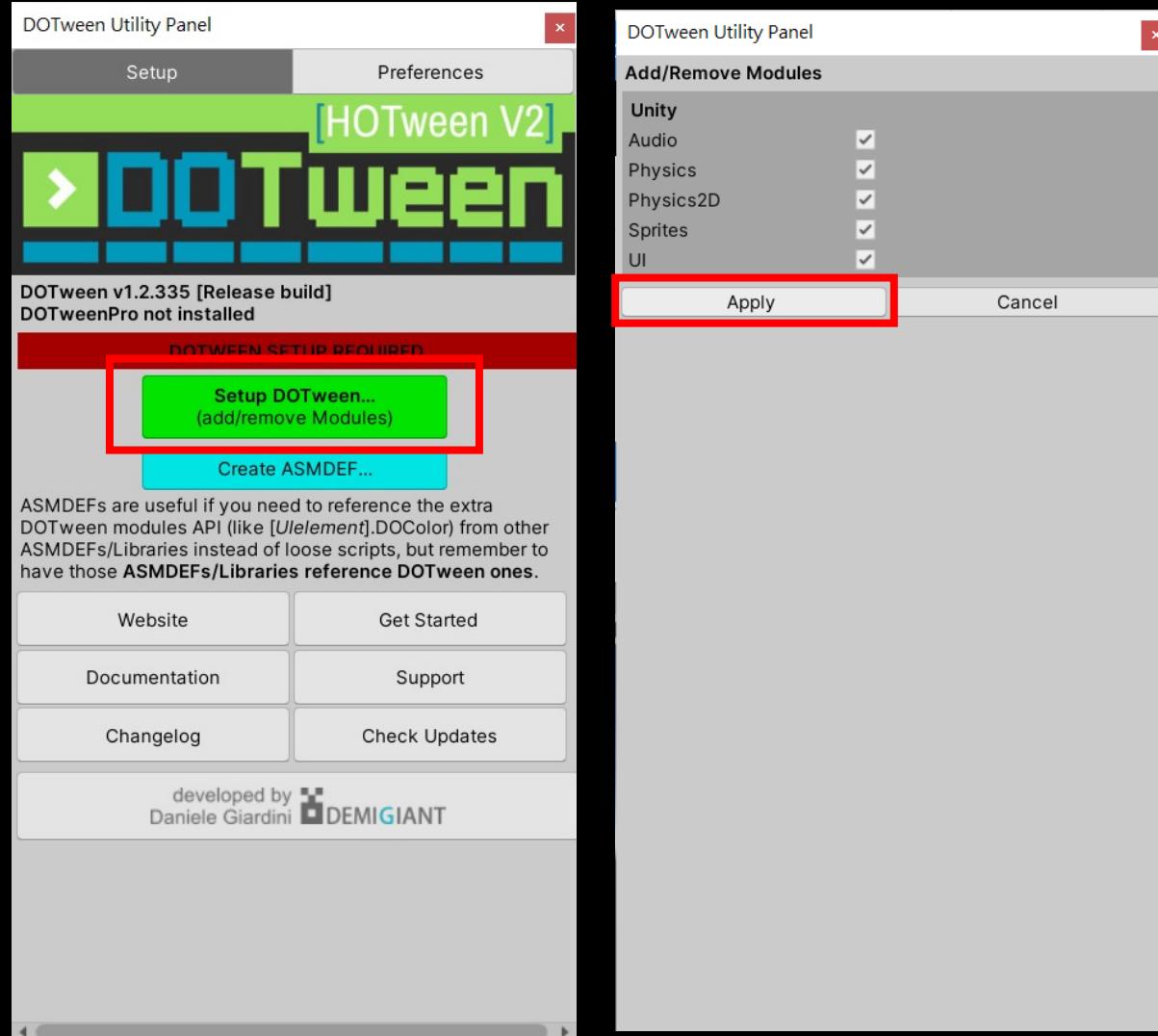
# DOTween全都要



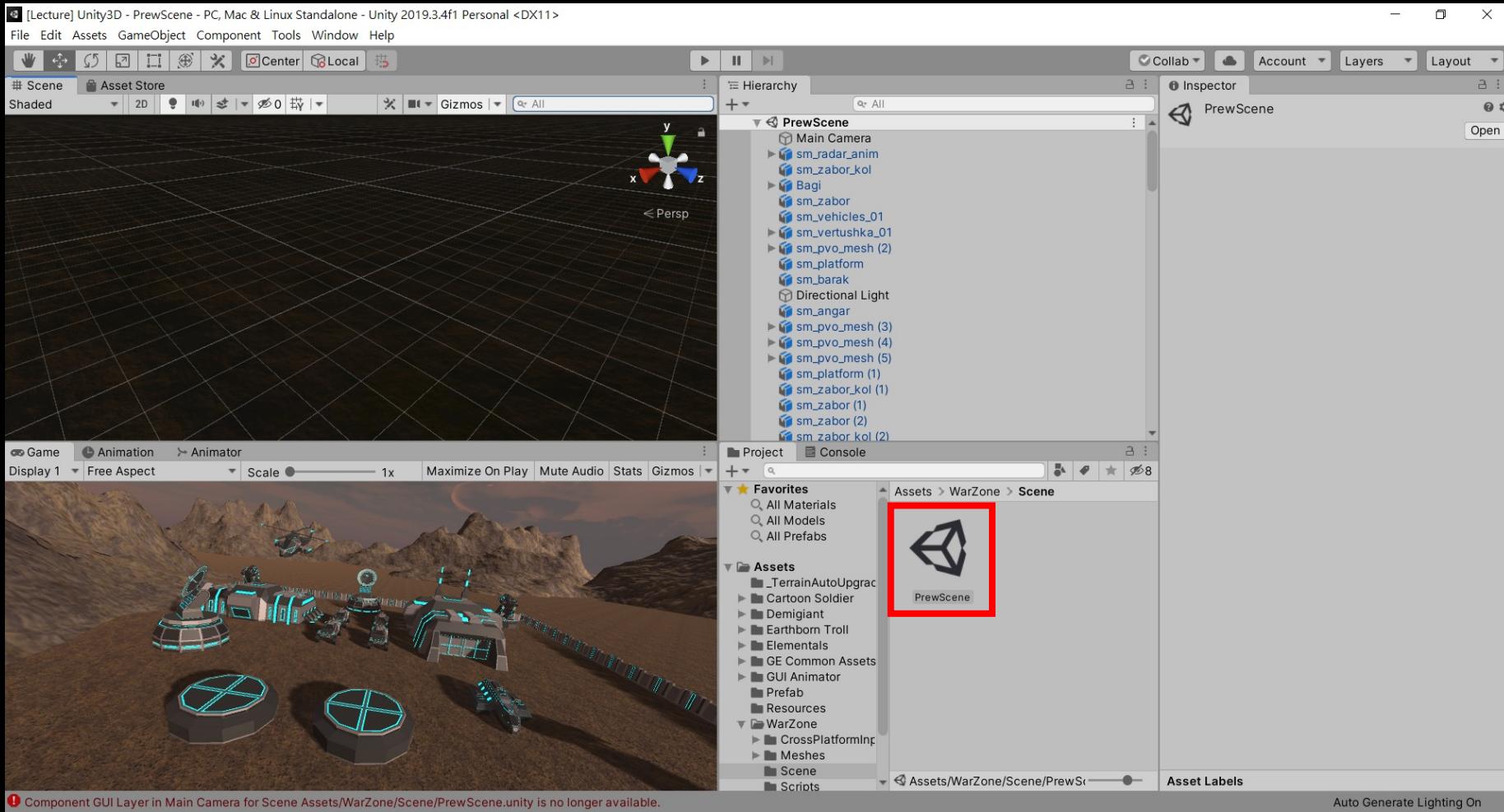
# 需要Setup



# 跟著點就對了



# 打開 Assets/WarZone/Scene/PrewScene.unity

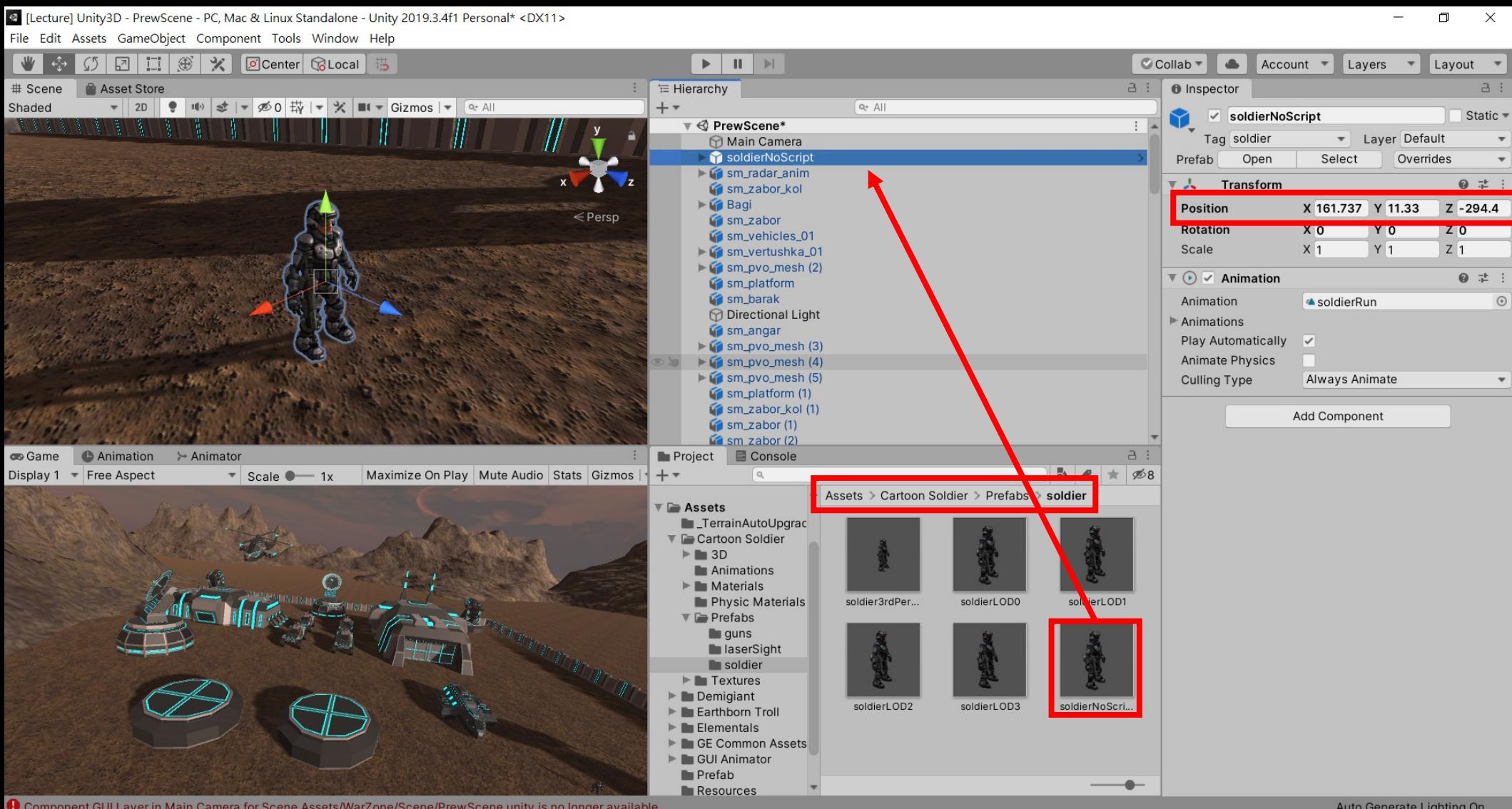


# 3D Animation

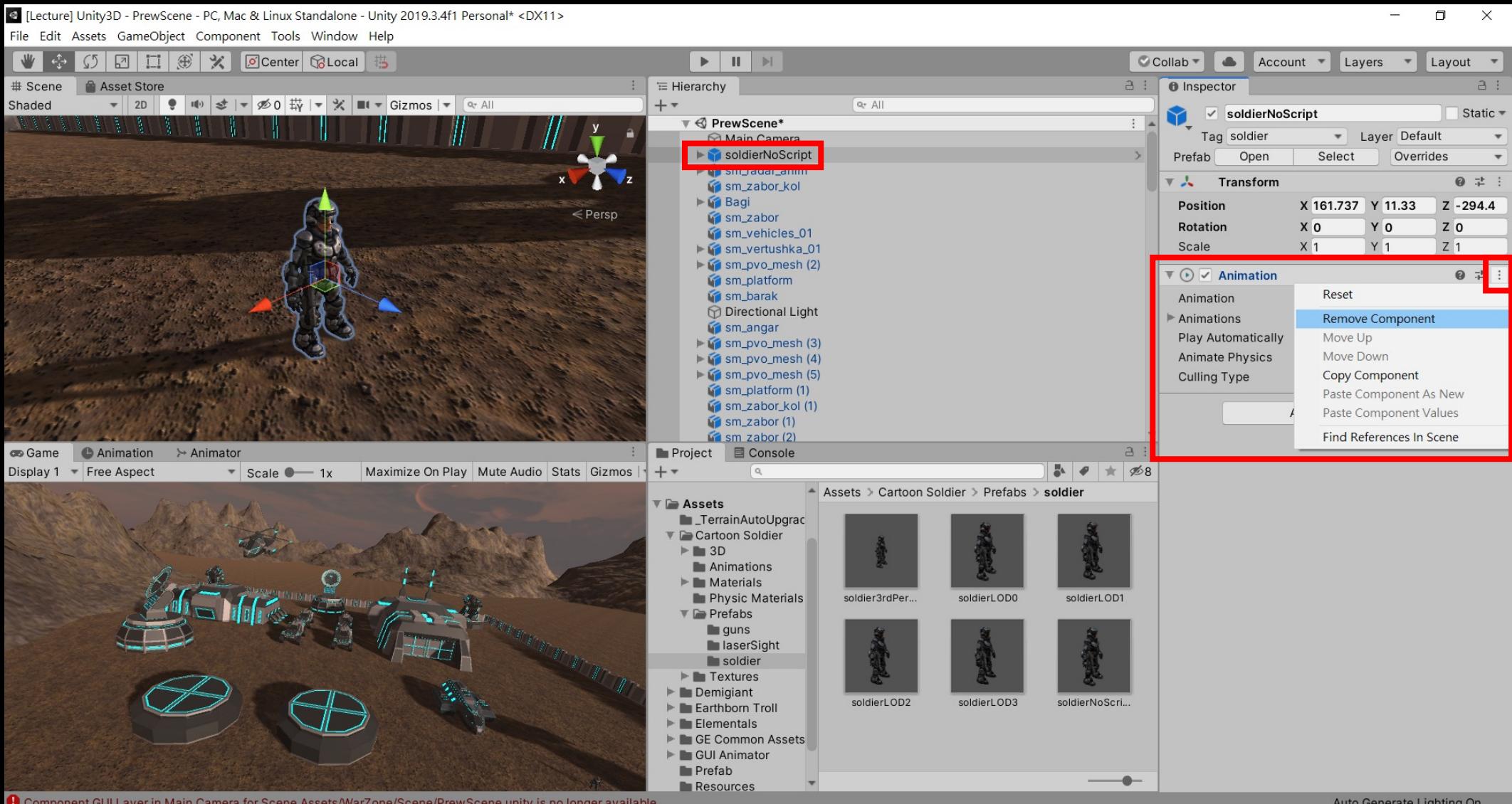
# Animation系統

- Unity內部的Animation系統有兩套
  - Animation (Legacy)
  - Animator Controller
- 兩套系統互不相容，無法同時使用
- 現在新的教學都是走Animator Controller，所以我們跟風
- 但是今天用的素材原本都是Animation Legacy
- 要改Import設定才能在新的Animator Controller中使用

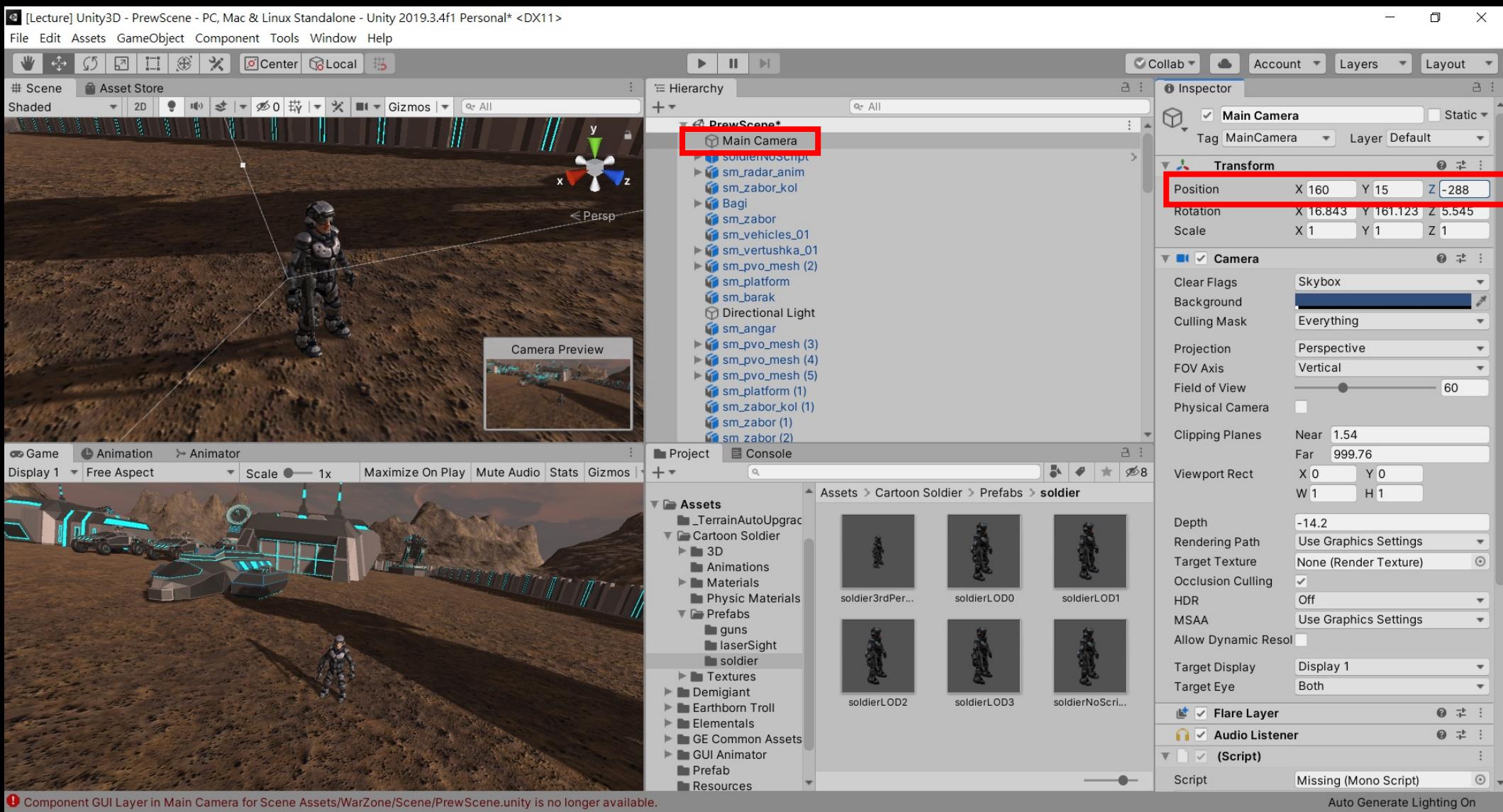
# 把Soldier模型放入場景內並設定位置



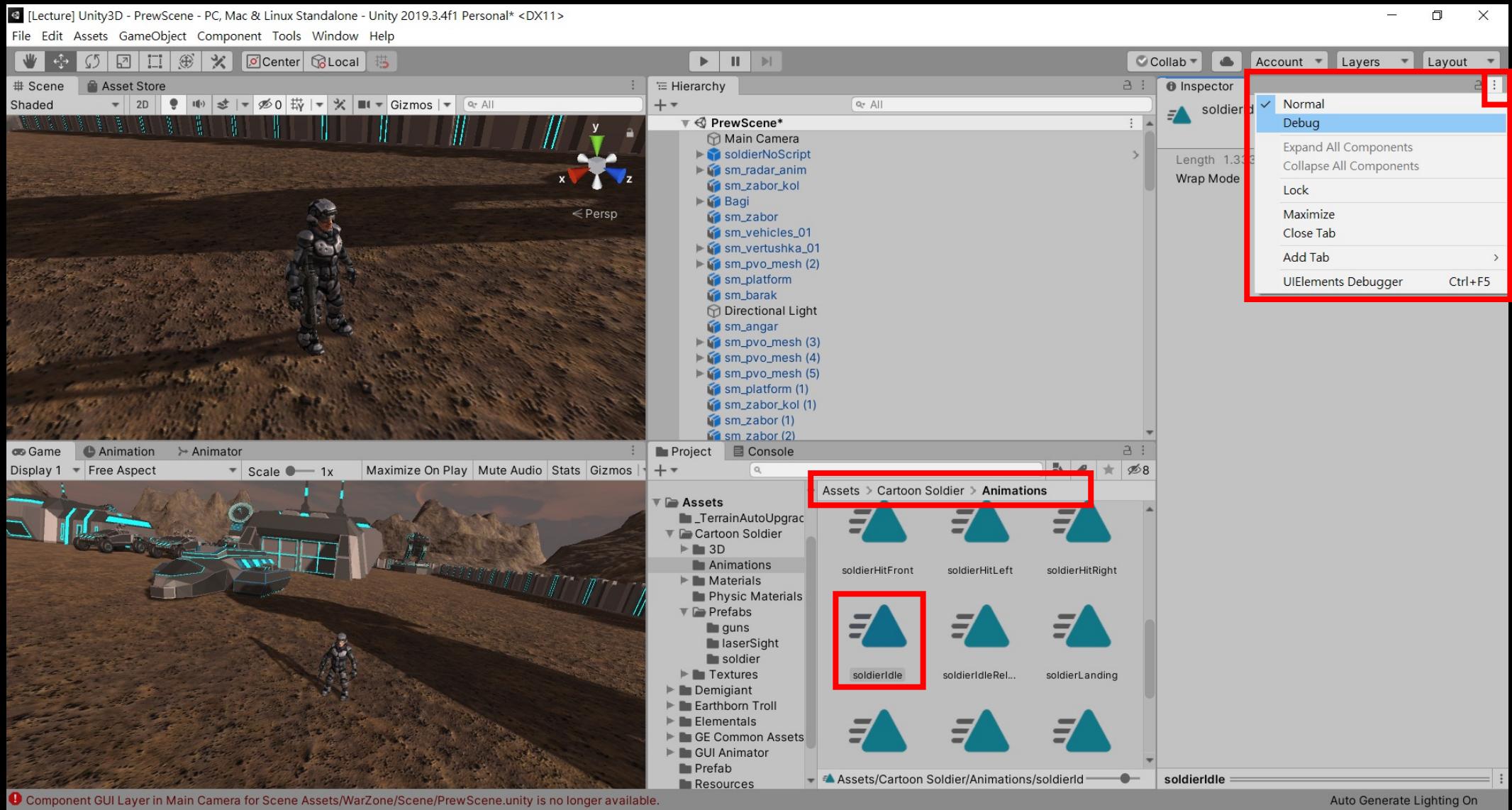
# 移除舊版的Animation (Legacy)



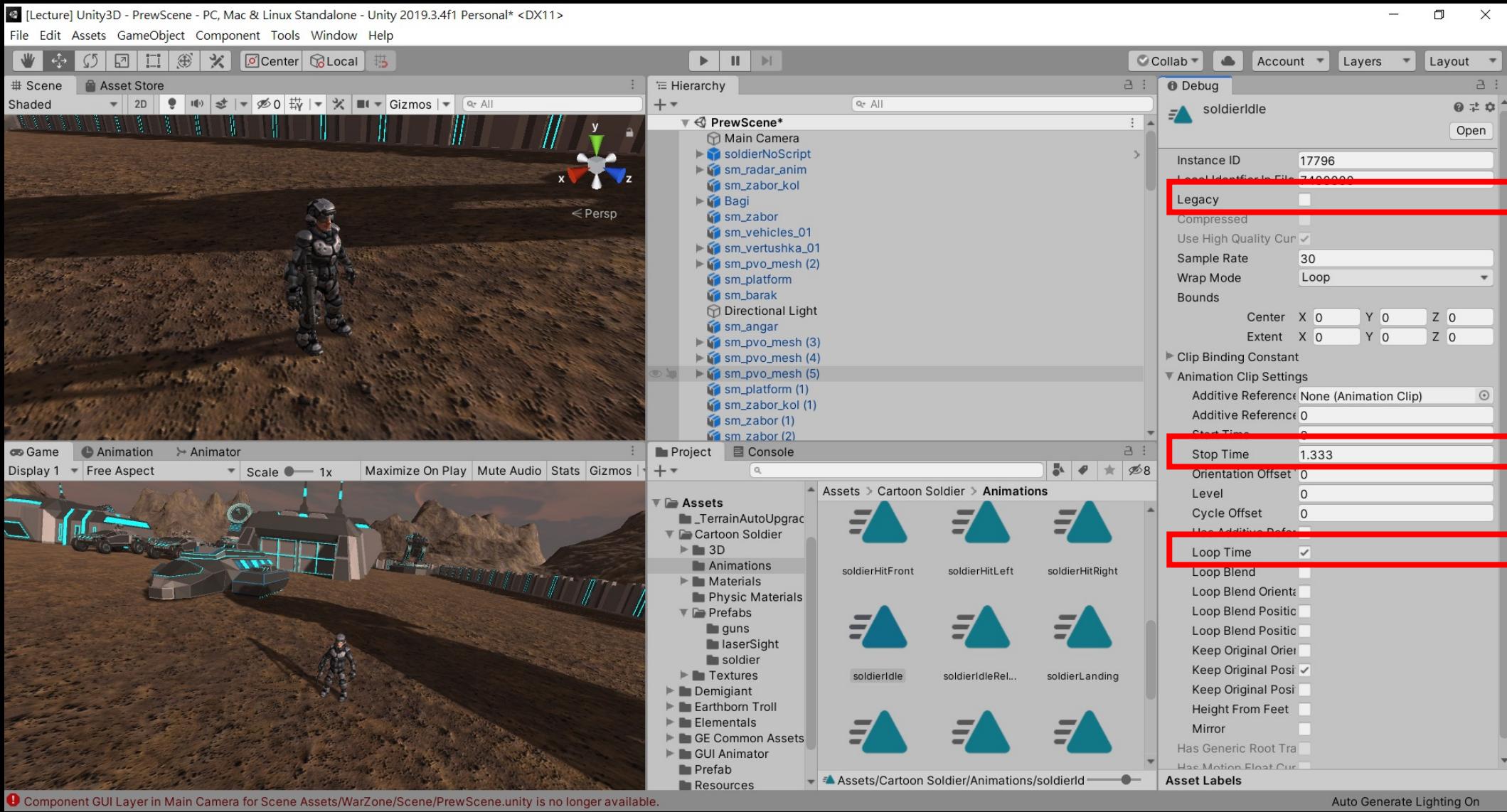
# 把Camera放到看得到主角的位置



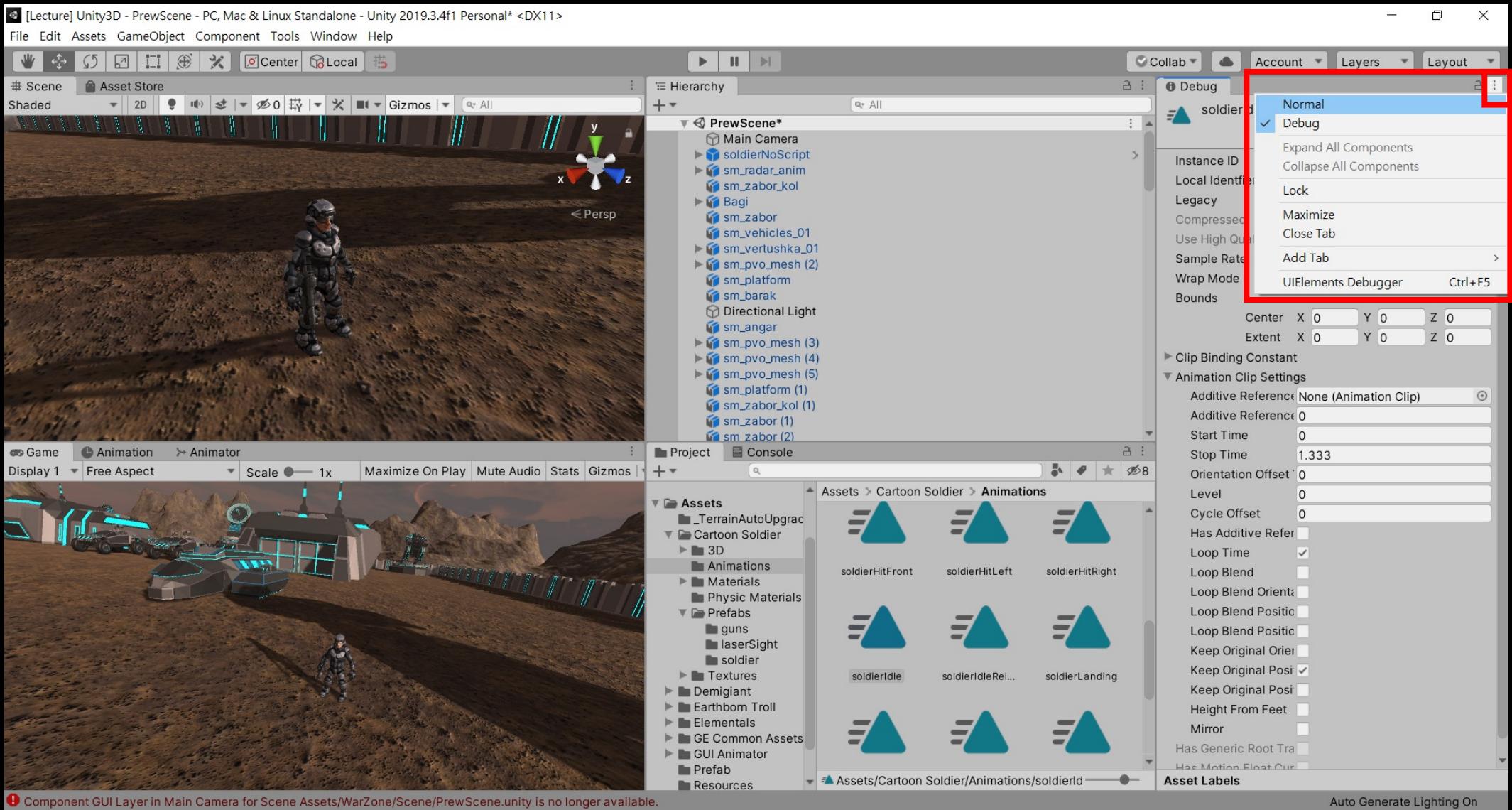
# 把soldieridle.anim調成Debug Mode



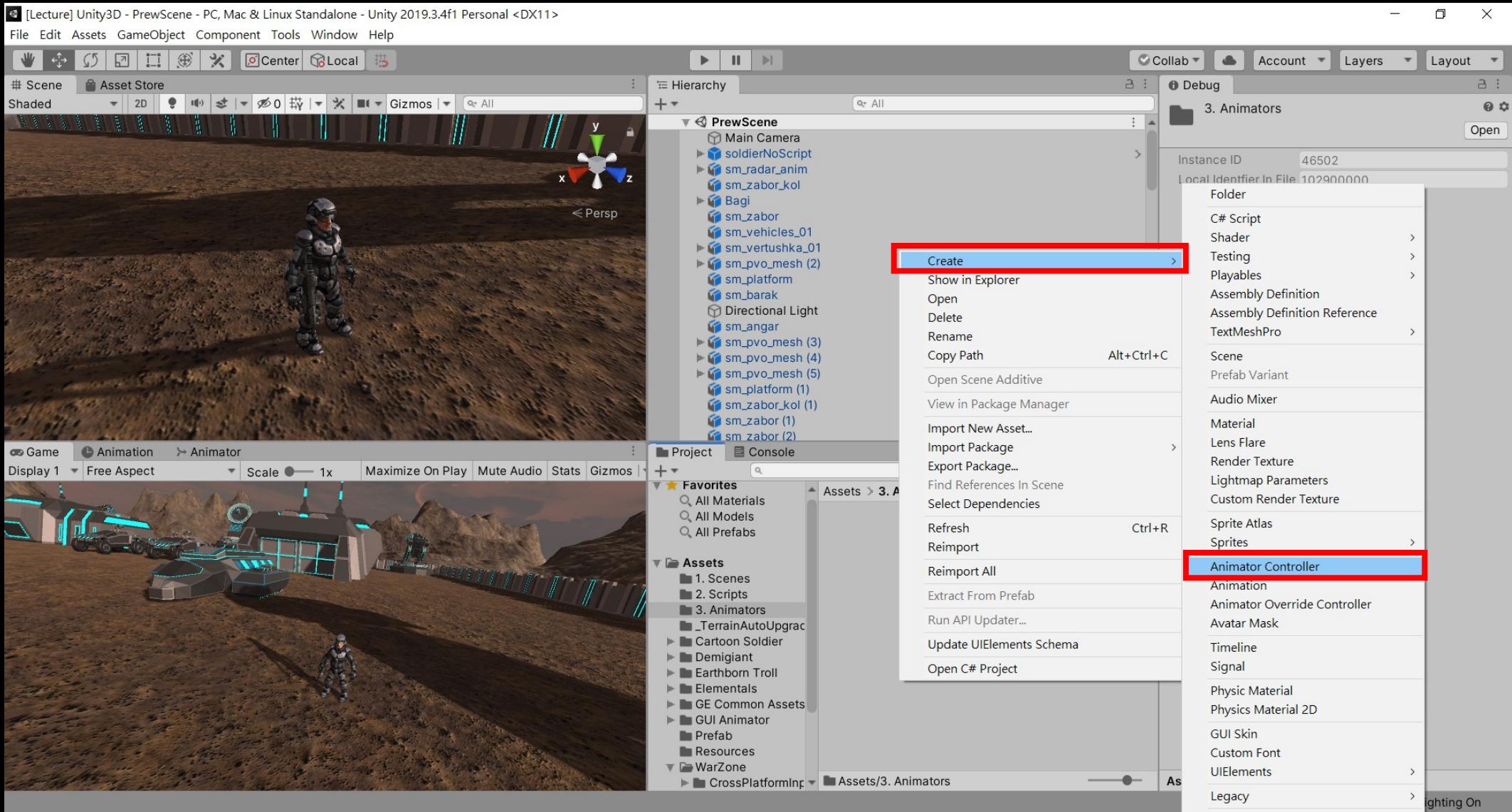
# 設定Legacy、Clip長度和Loop Time



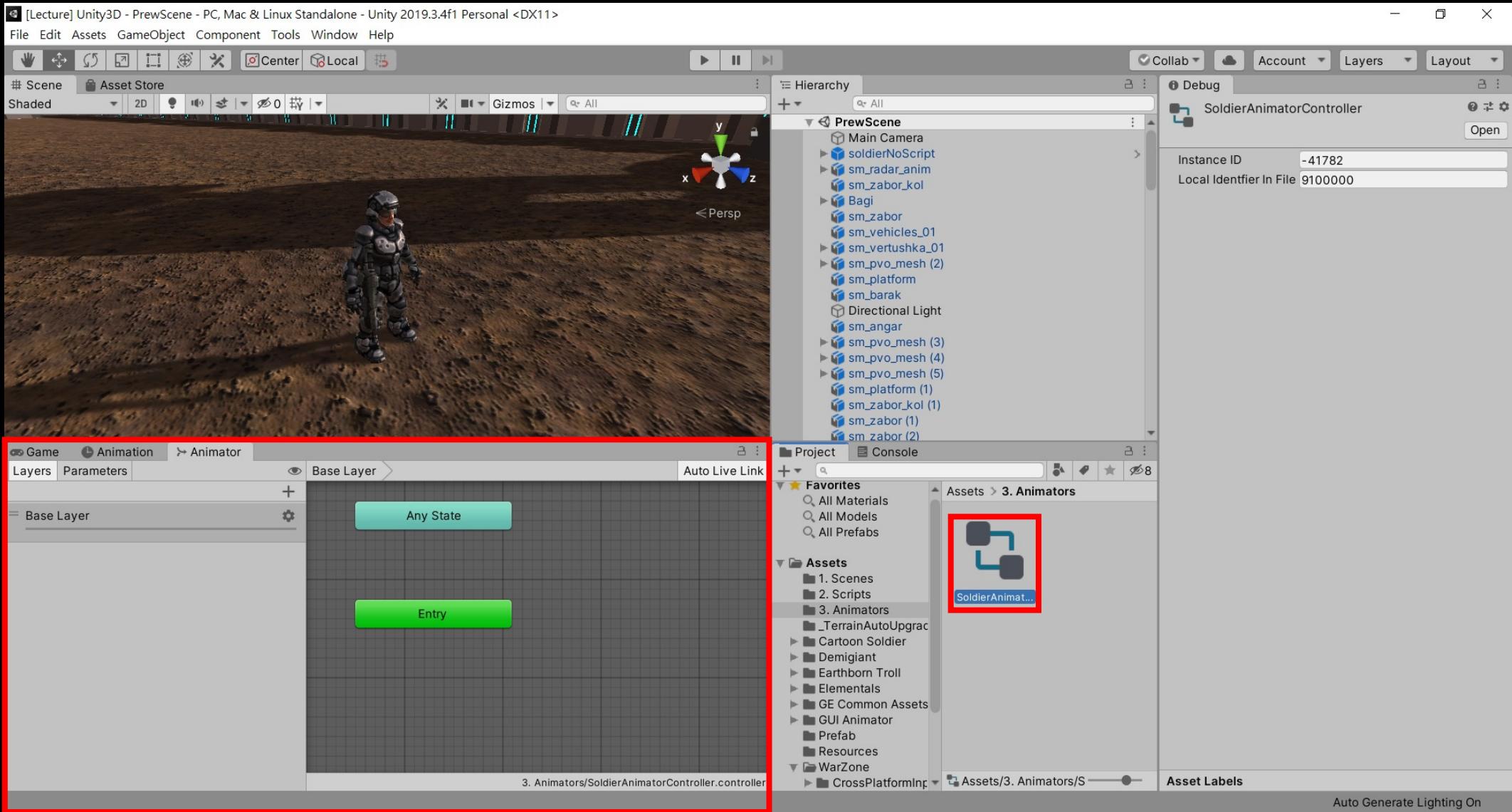
# 把inspector調回Normal



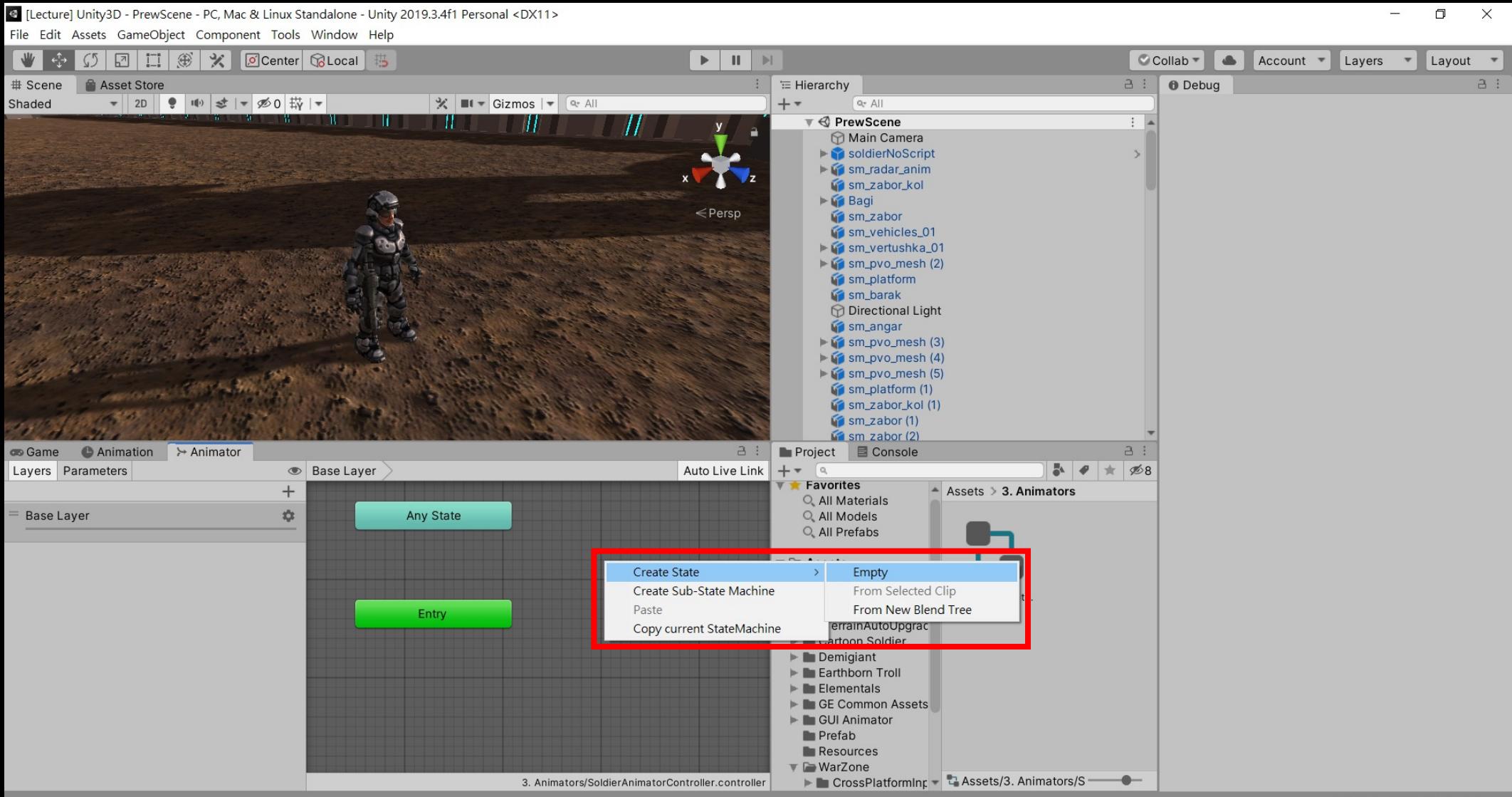
# 右鍵->Create->Animator Controller



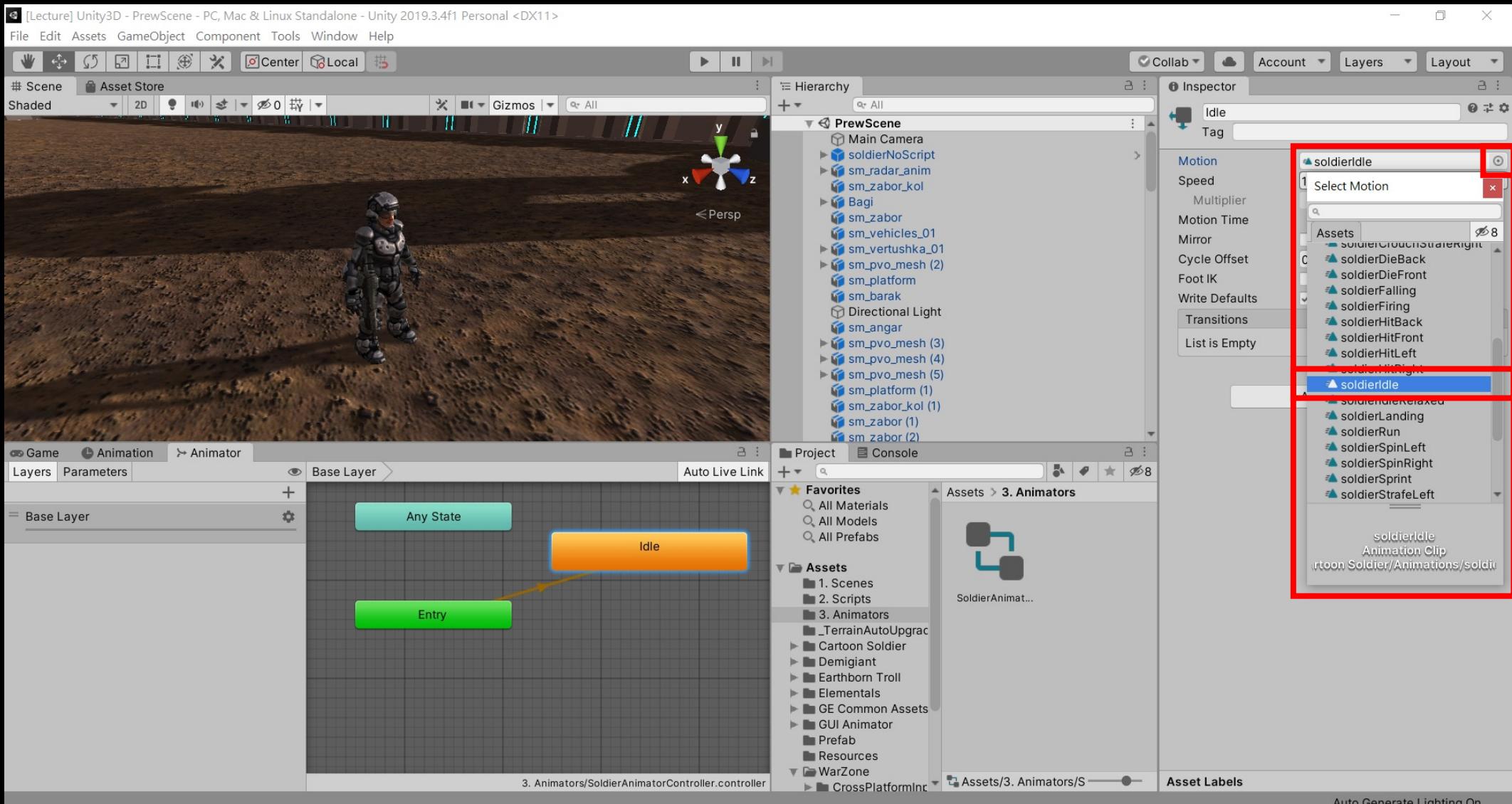
# 雙擊剛剛創建的Animator Controller



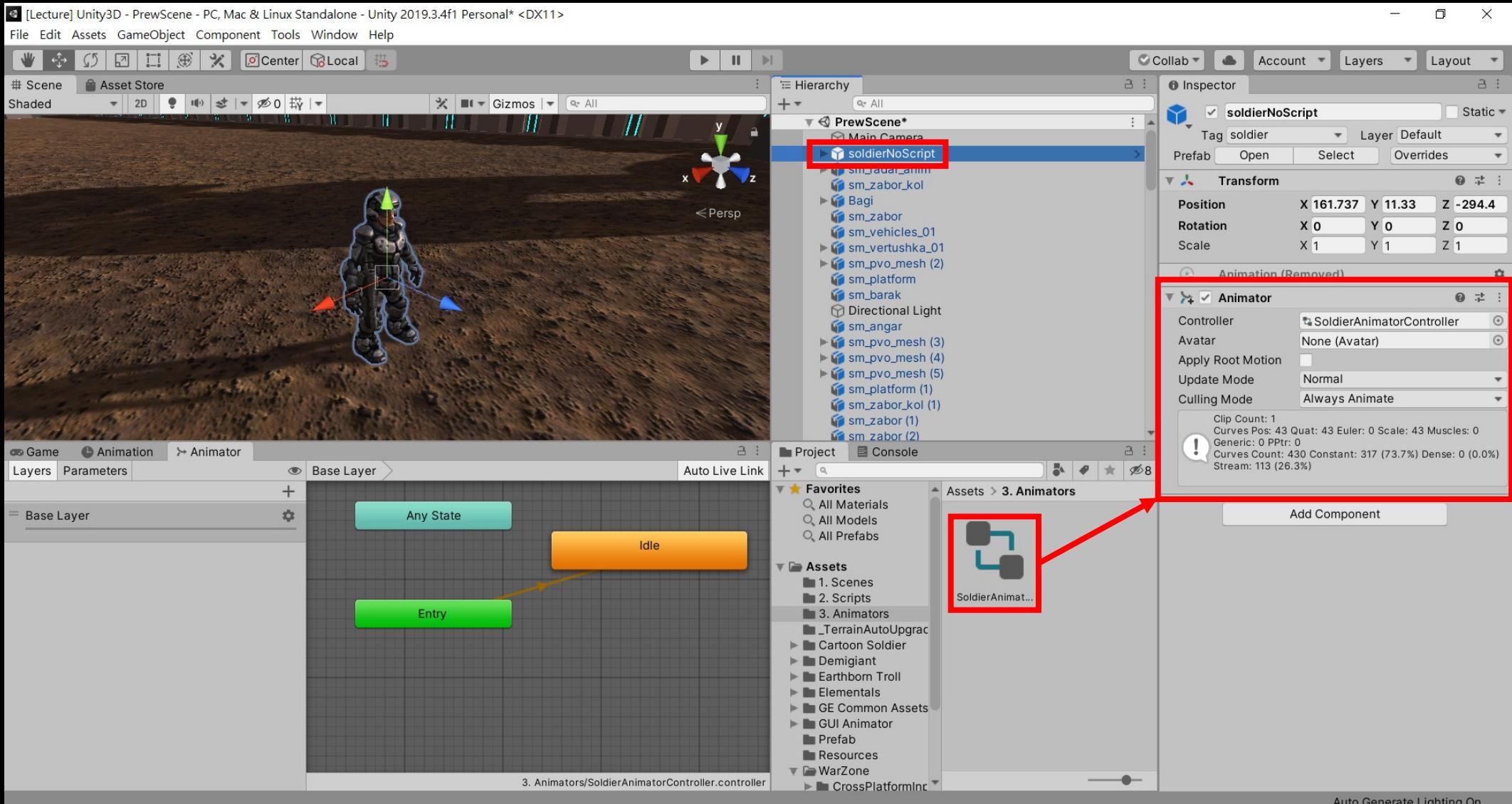
# 在狀態機中右鍵新增狀態並取名為Idle



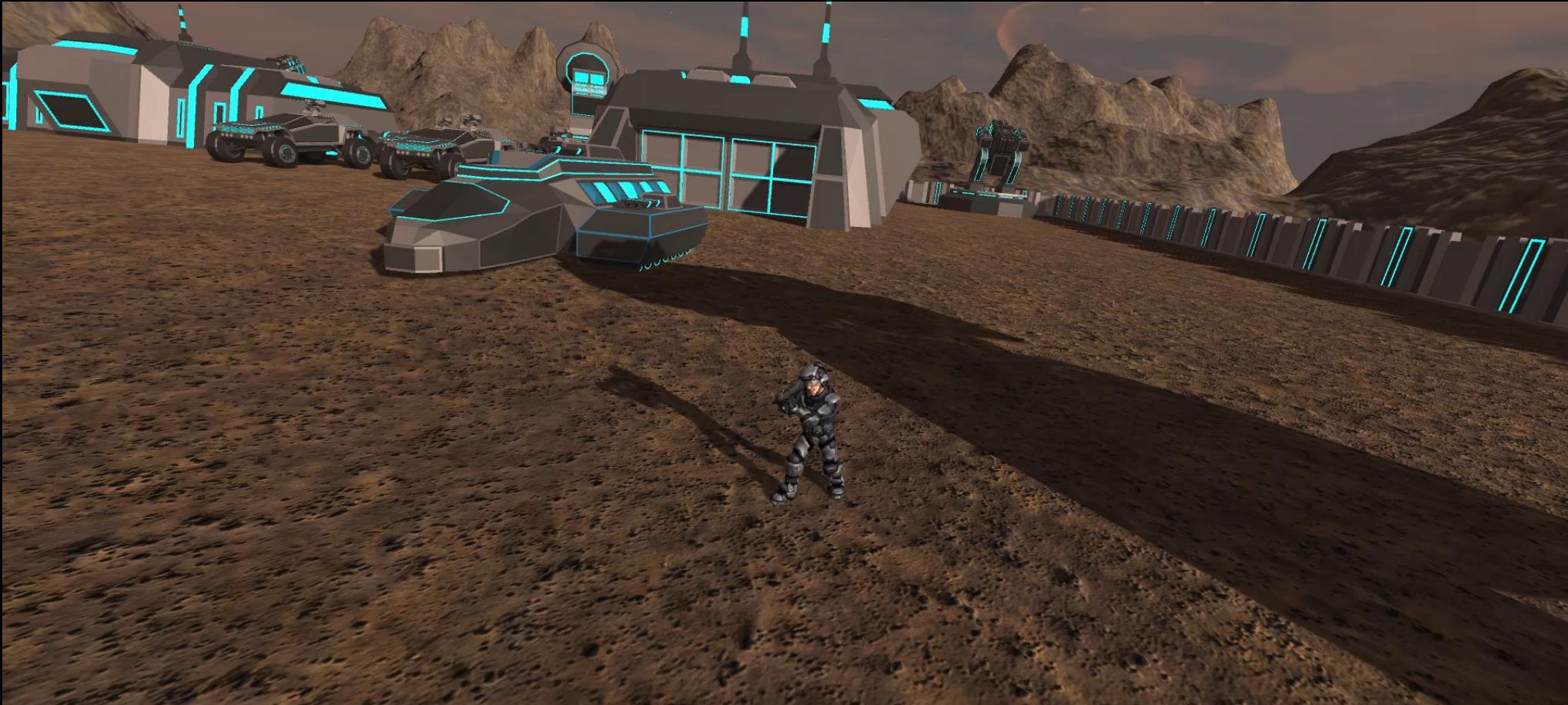
# 指定Animation Clip



# 把Animator Controller拖曳給主角

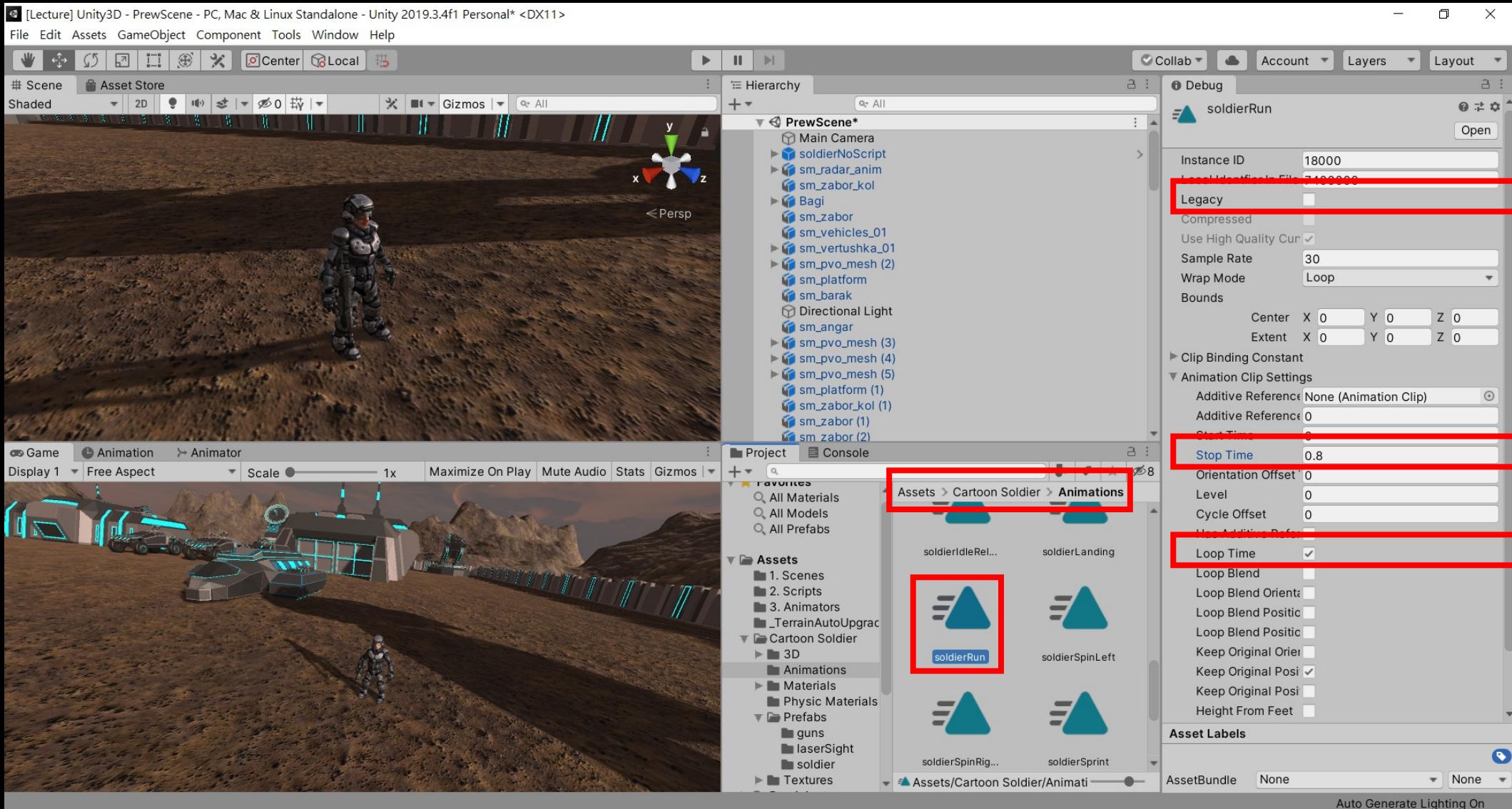


執行就可以看到主角動起來啦

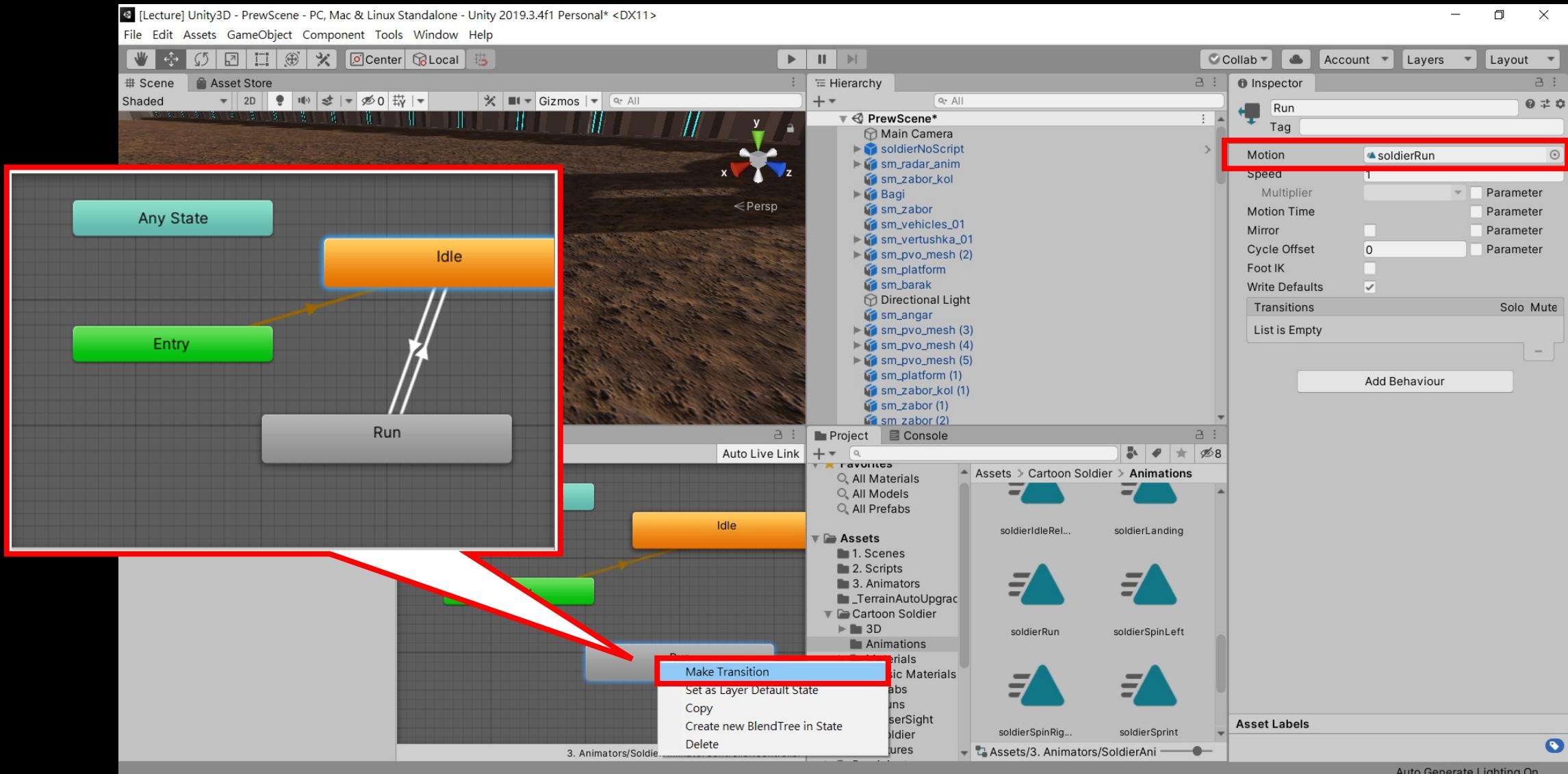


3D角色移動

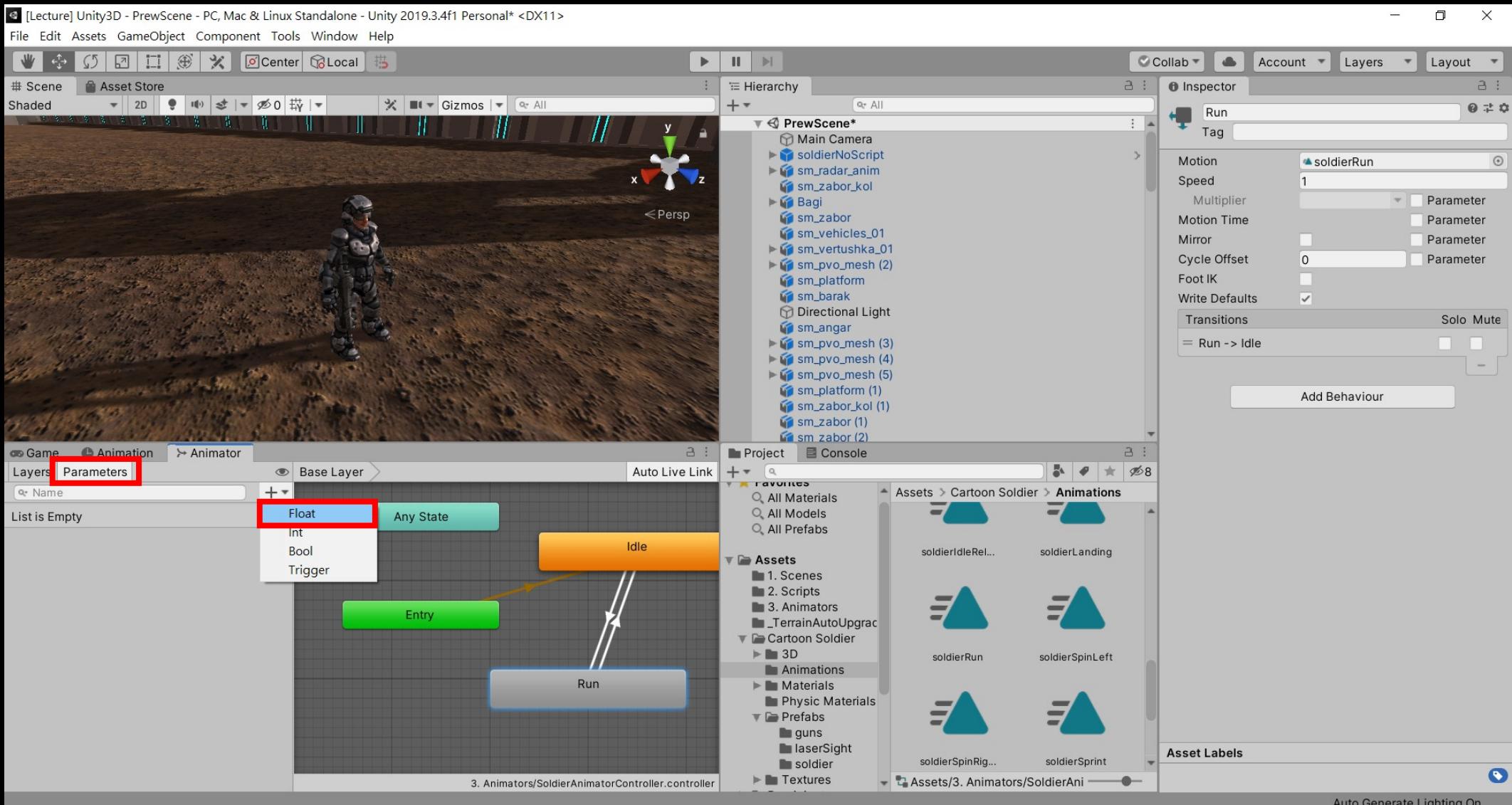
# 修改SoldierRun.anim



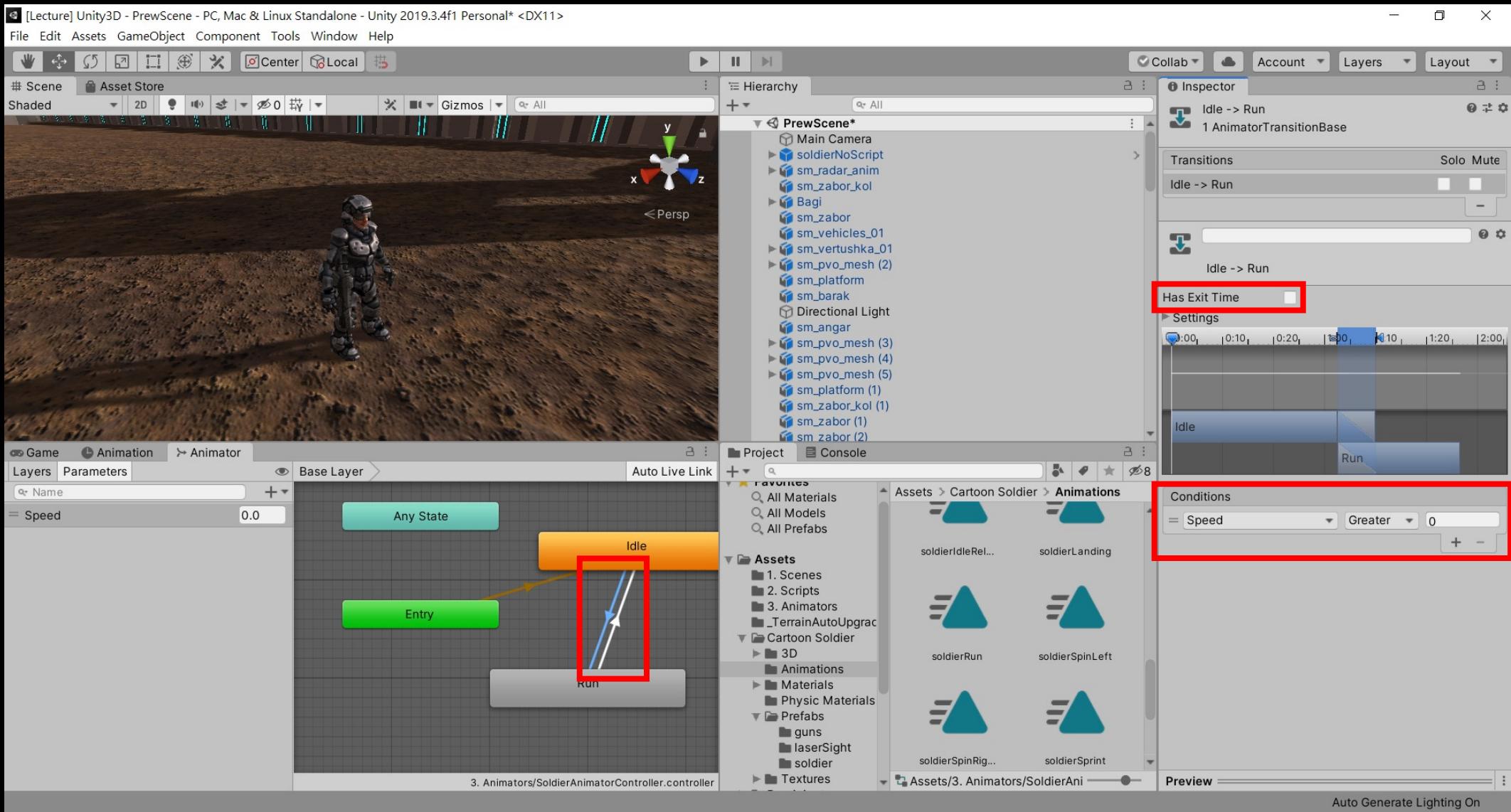
# 新增Run並且右鍵給他箭頭



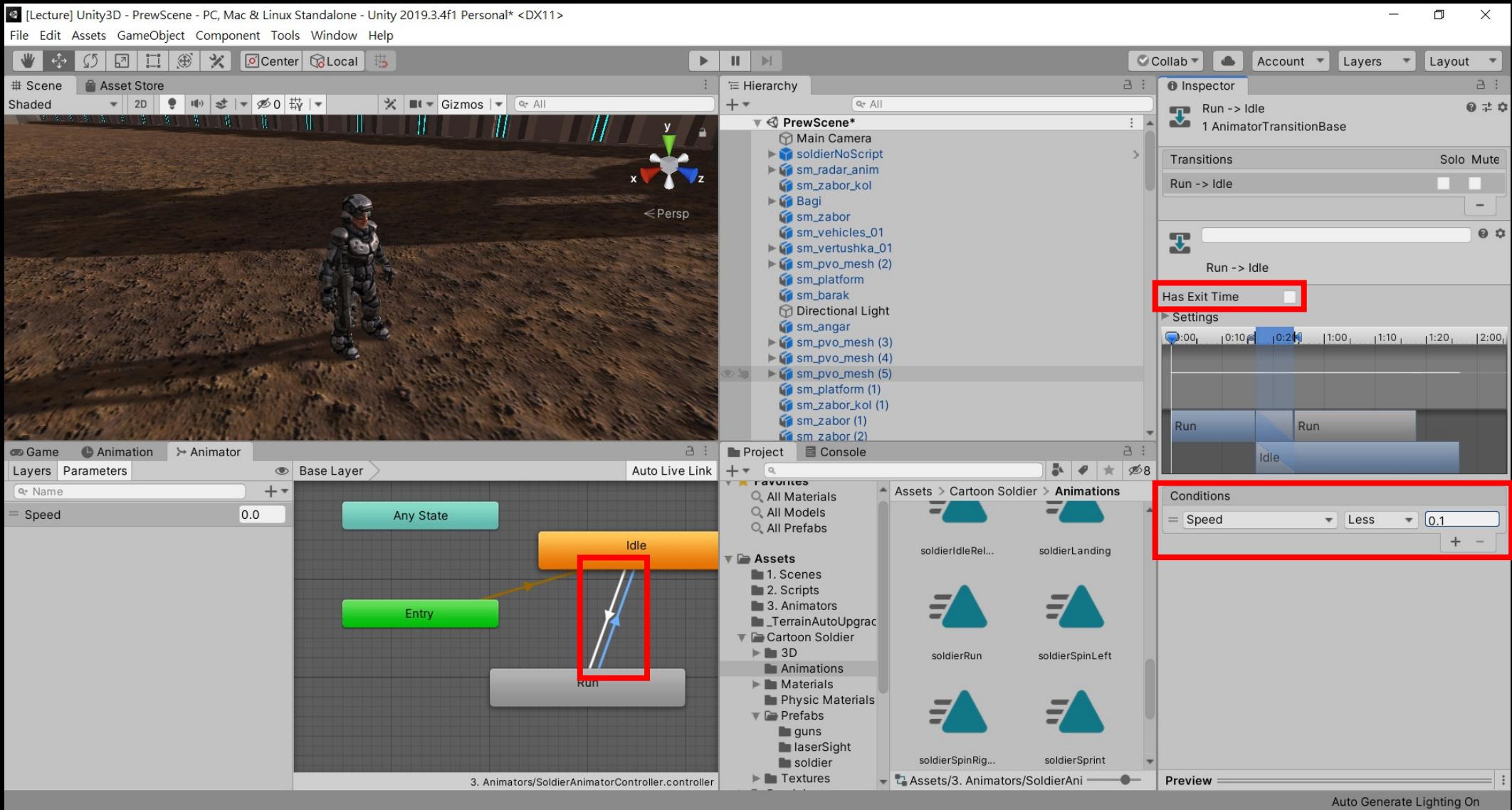
# 新增一個float參數並取名為Speed



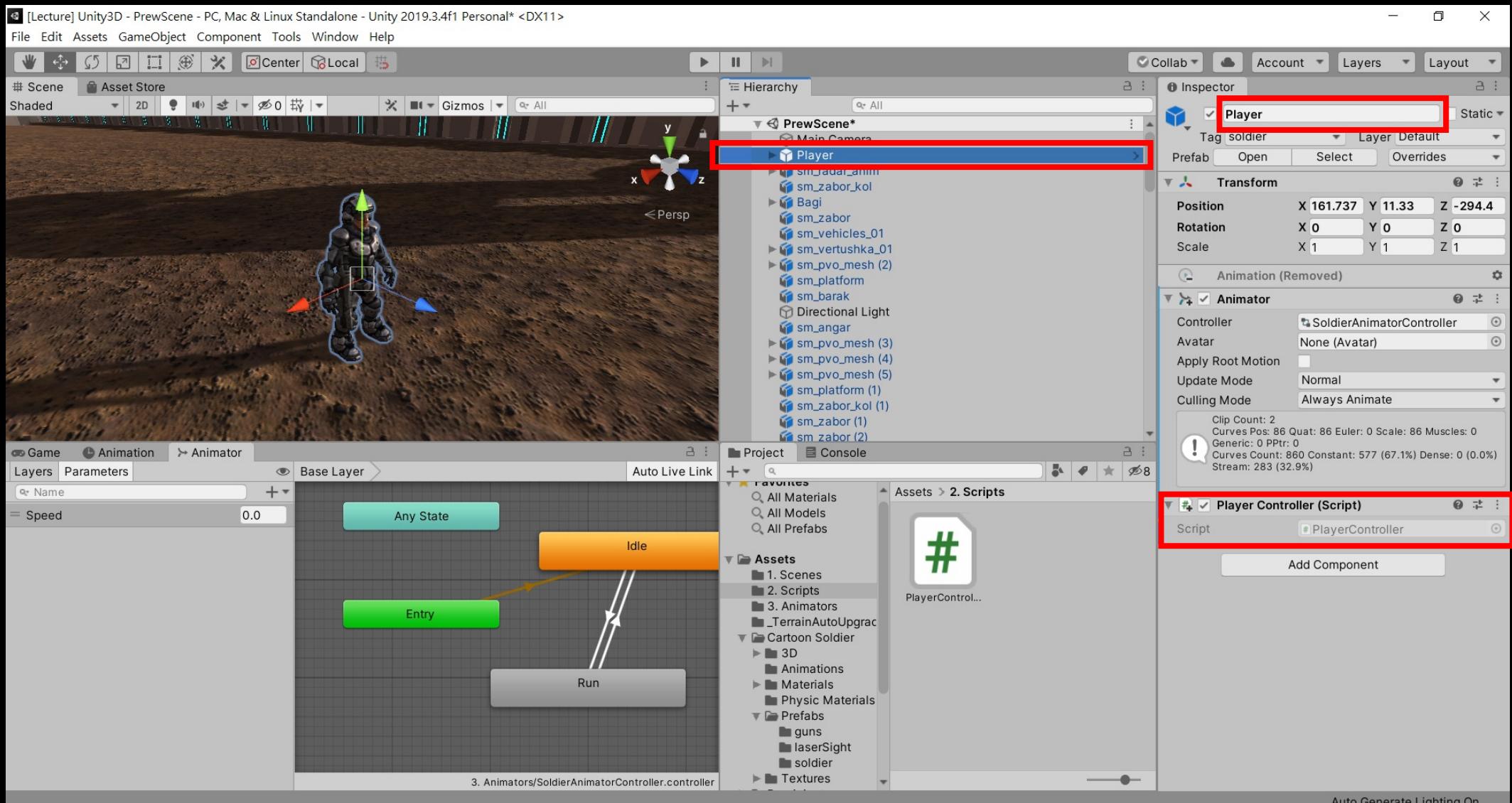
# Idle->Run 條件設定



# Run->Idle 條件設定



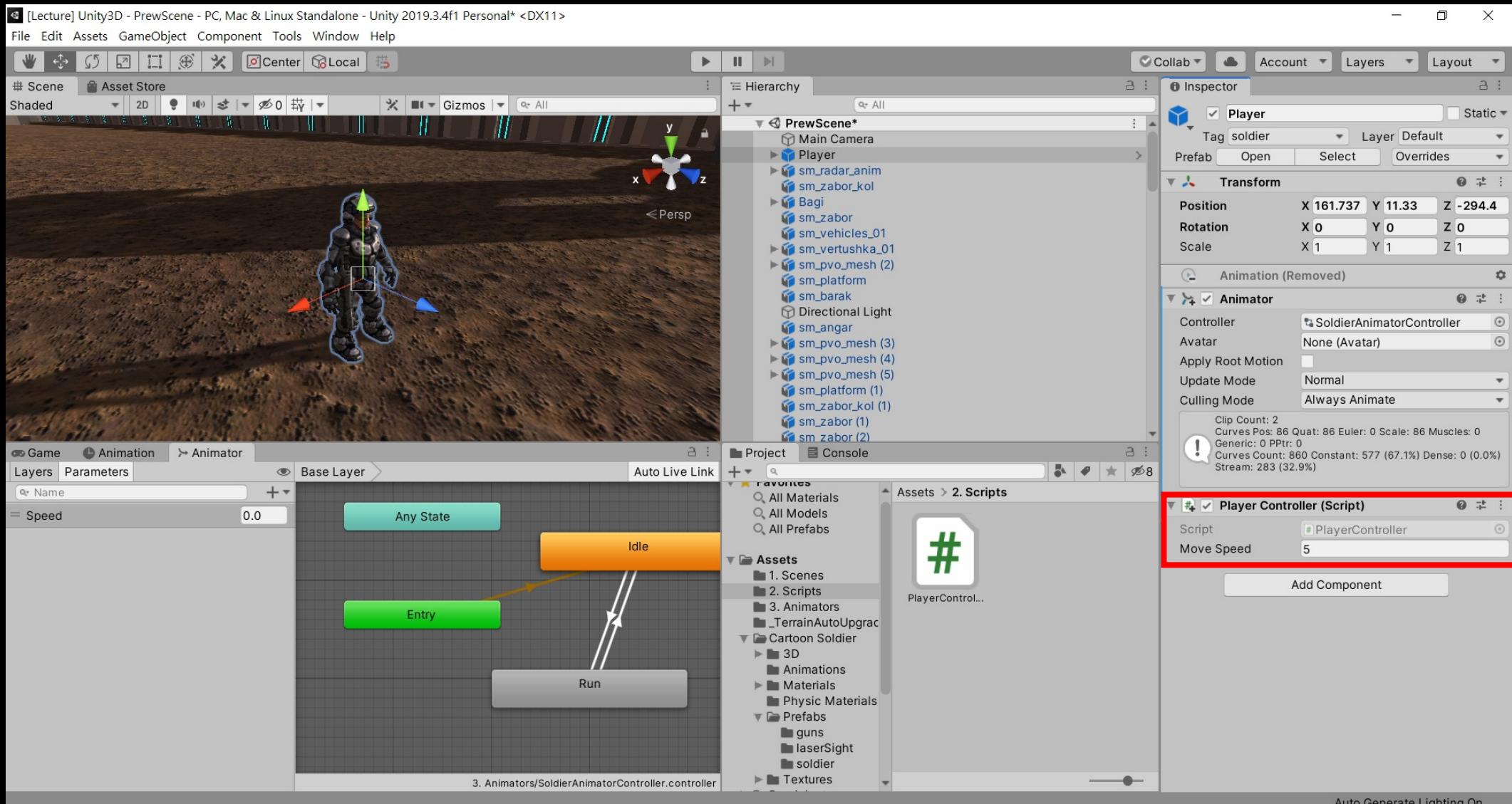
# 主角改名Player並新增PlayerController.cs



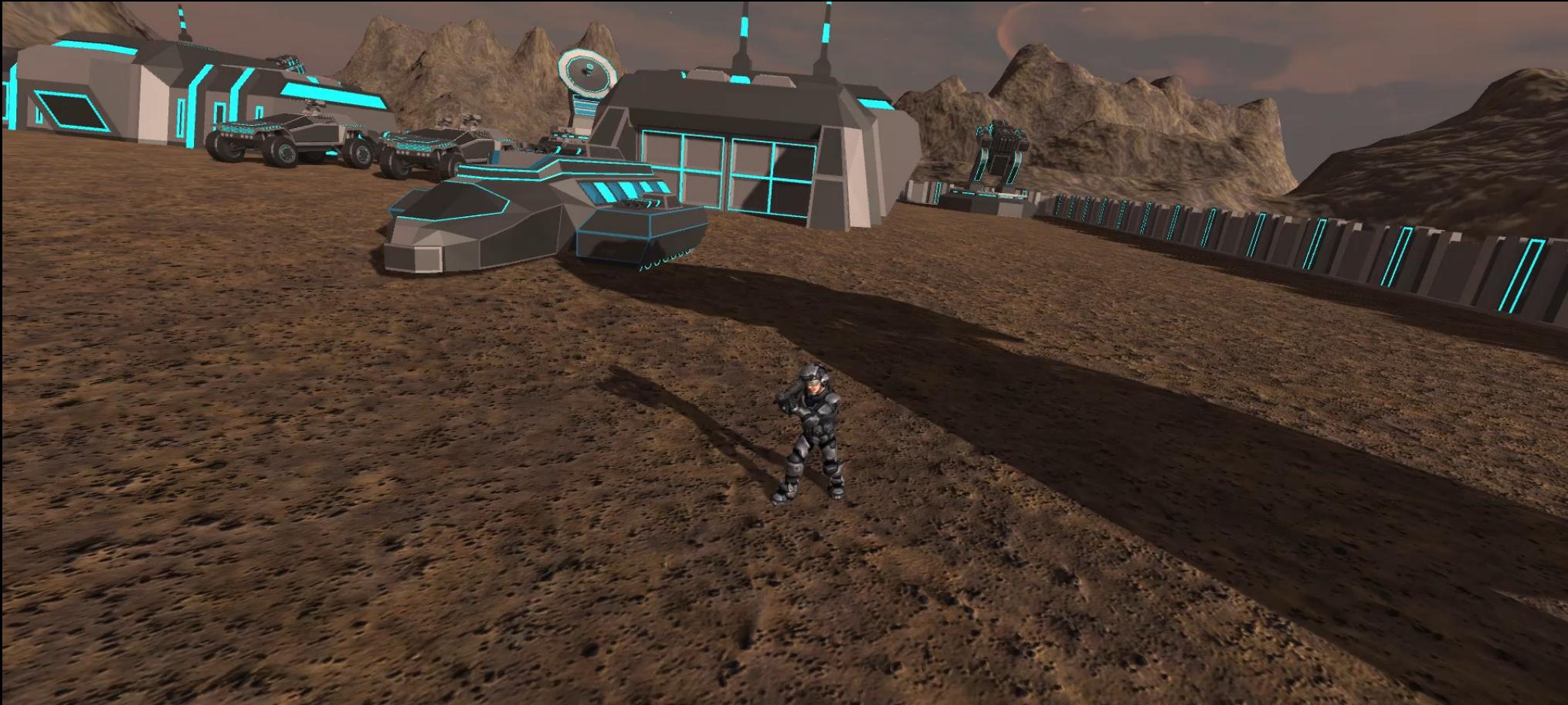
# PlayerController.cs

- <https://bit.ly/37dl0x9>

# MoveSpeed設定成5

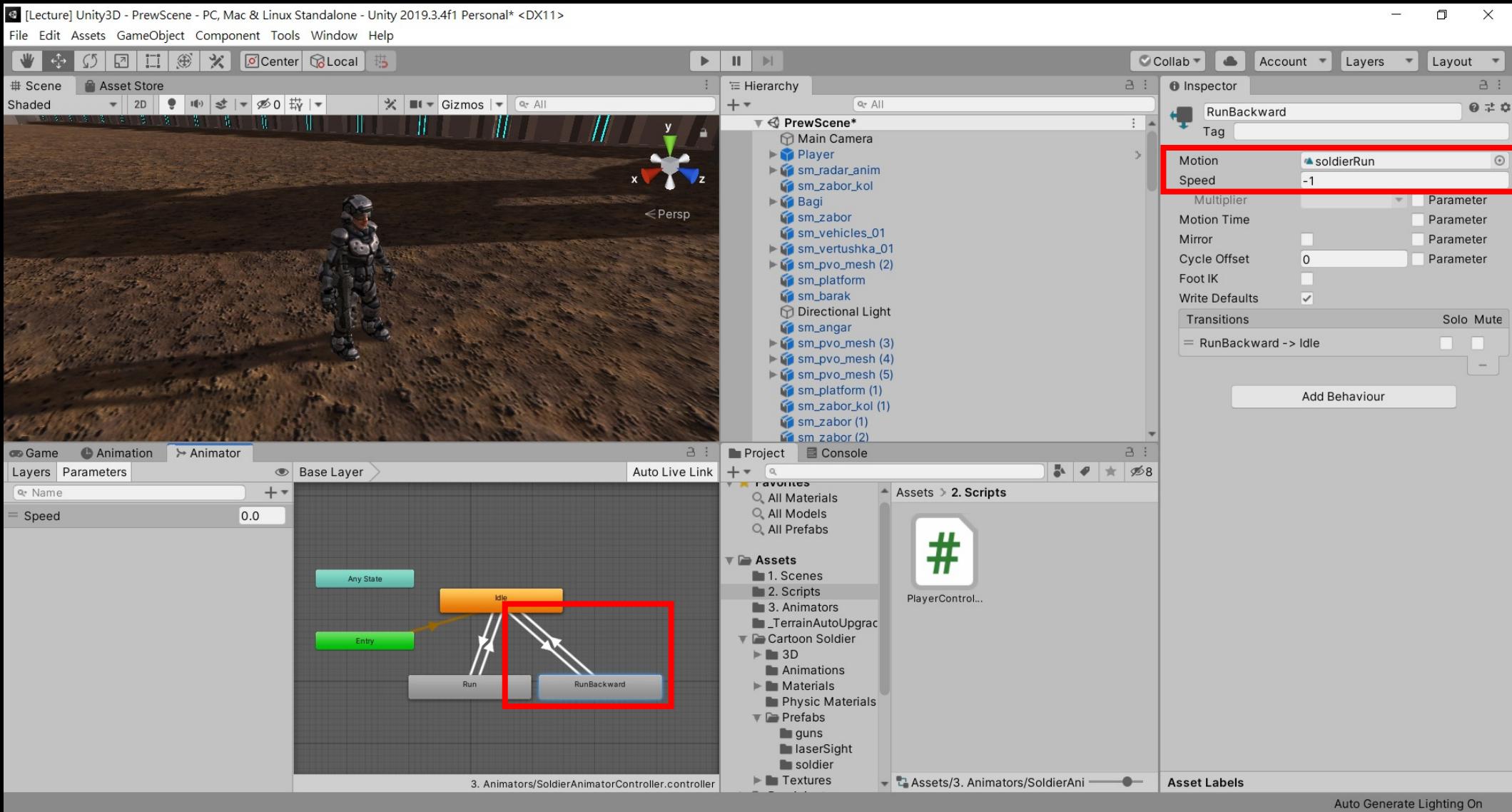


按 “W” 角色會往前跑啦

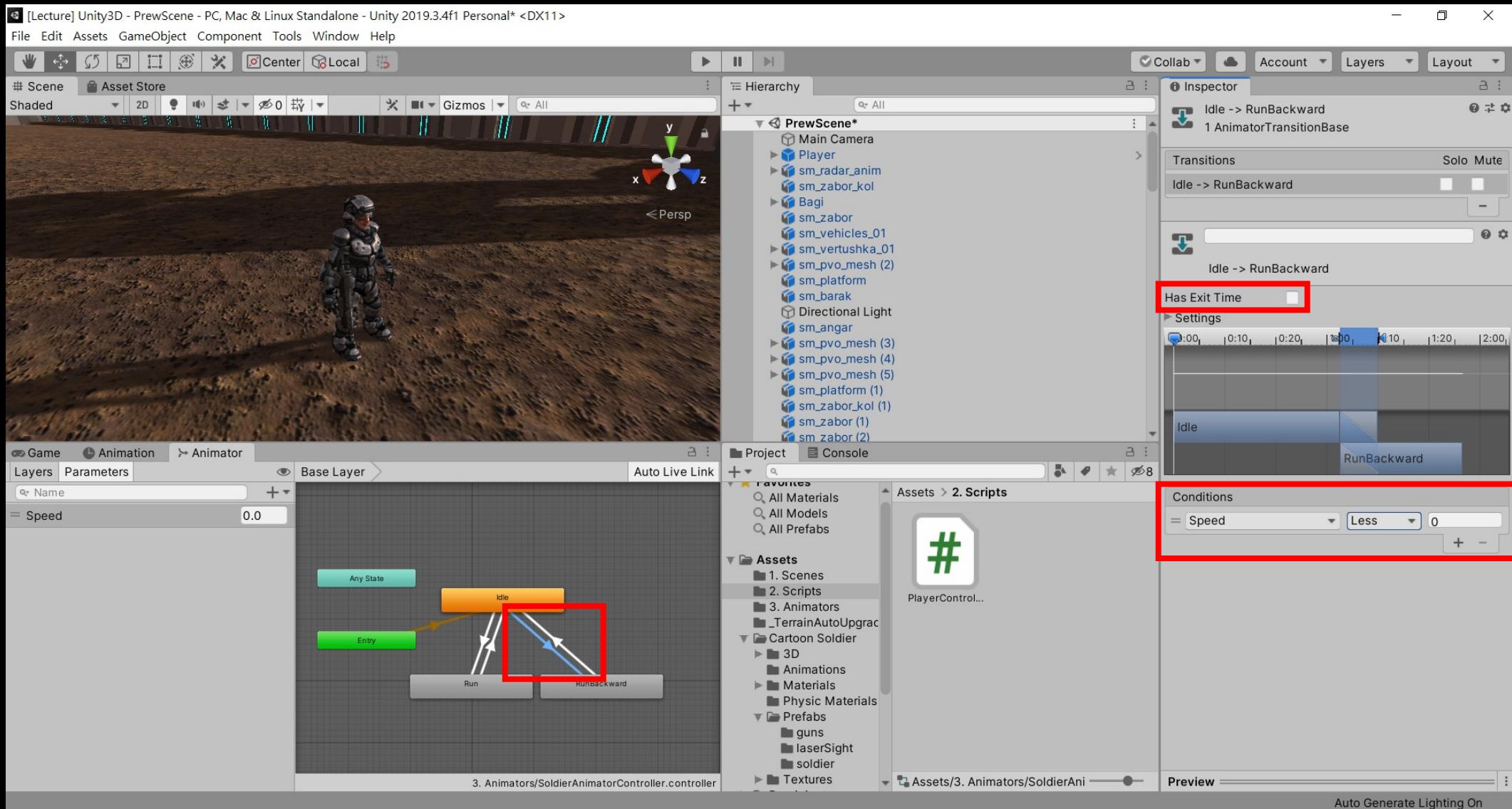


往後走 = 動畫反過來播

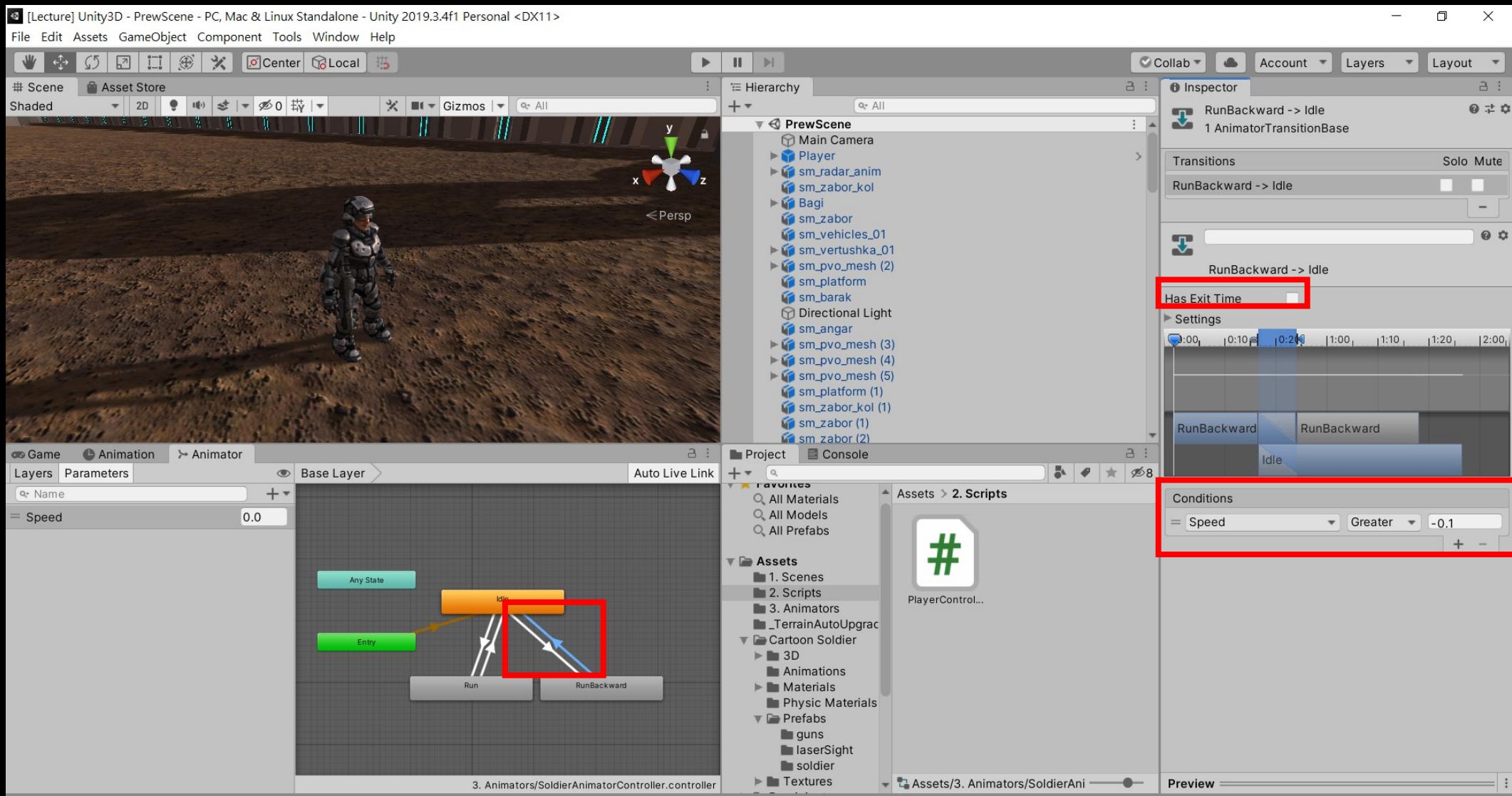
# 新增狀態RunBackward



# Idle->RunBackward 條件設定



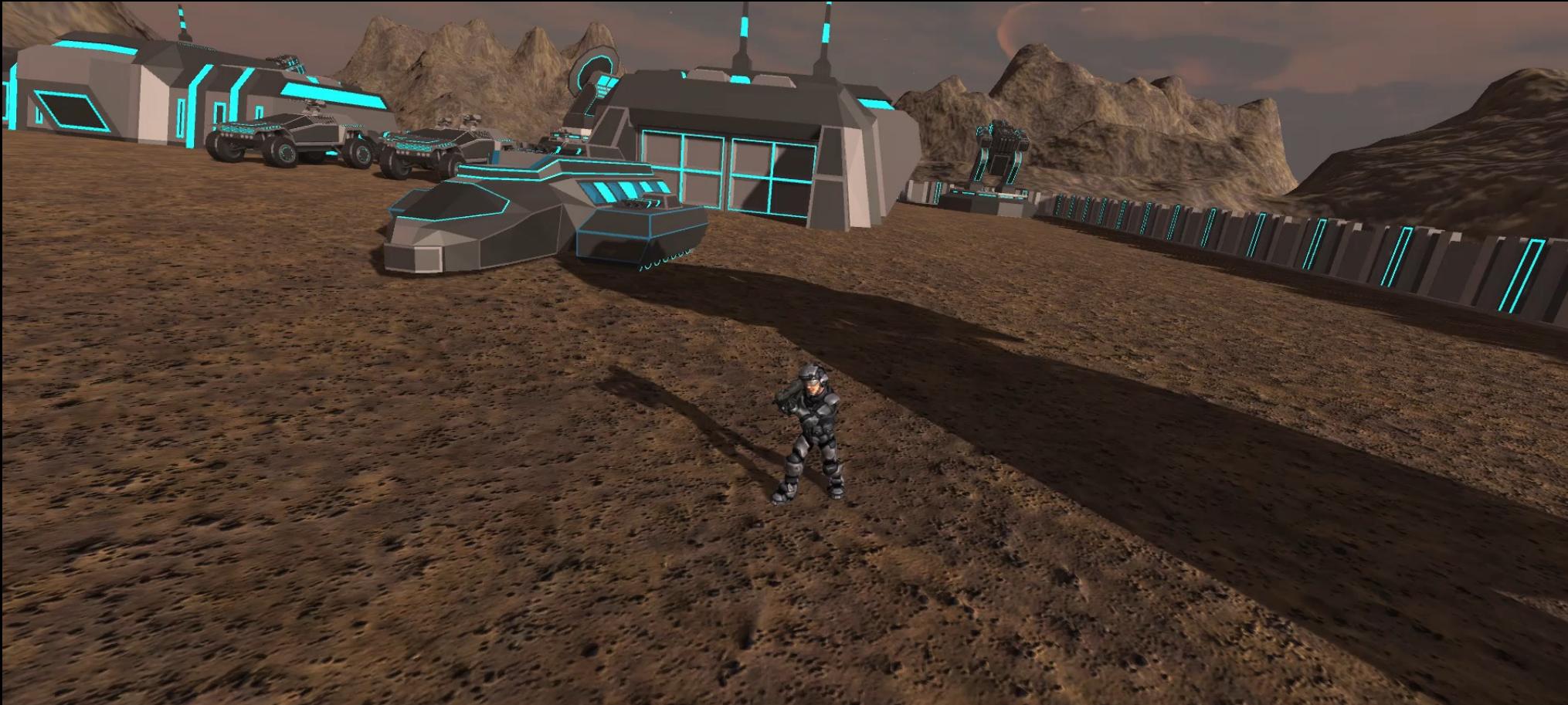
# RunBackward->Idle 條件設定



# 修改PlayerController.cs

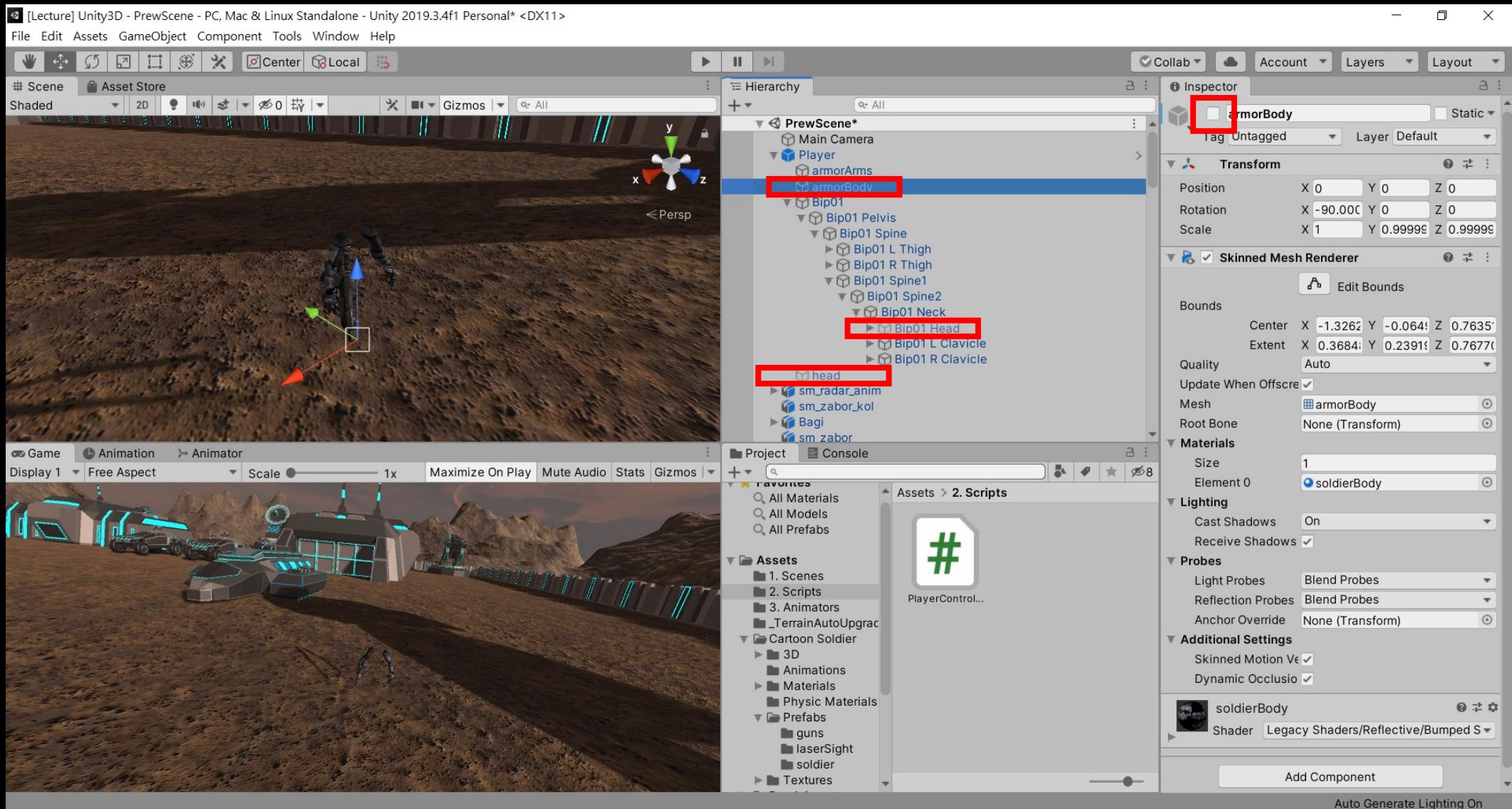
- <https://bit.ly/2Wff8wW>

角色也可以前後移動啦

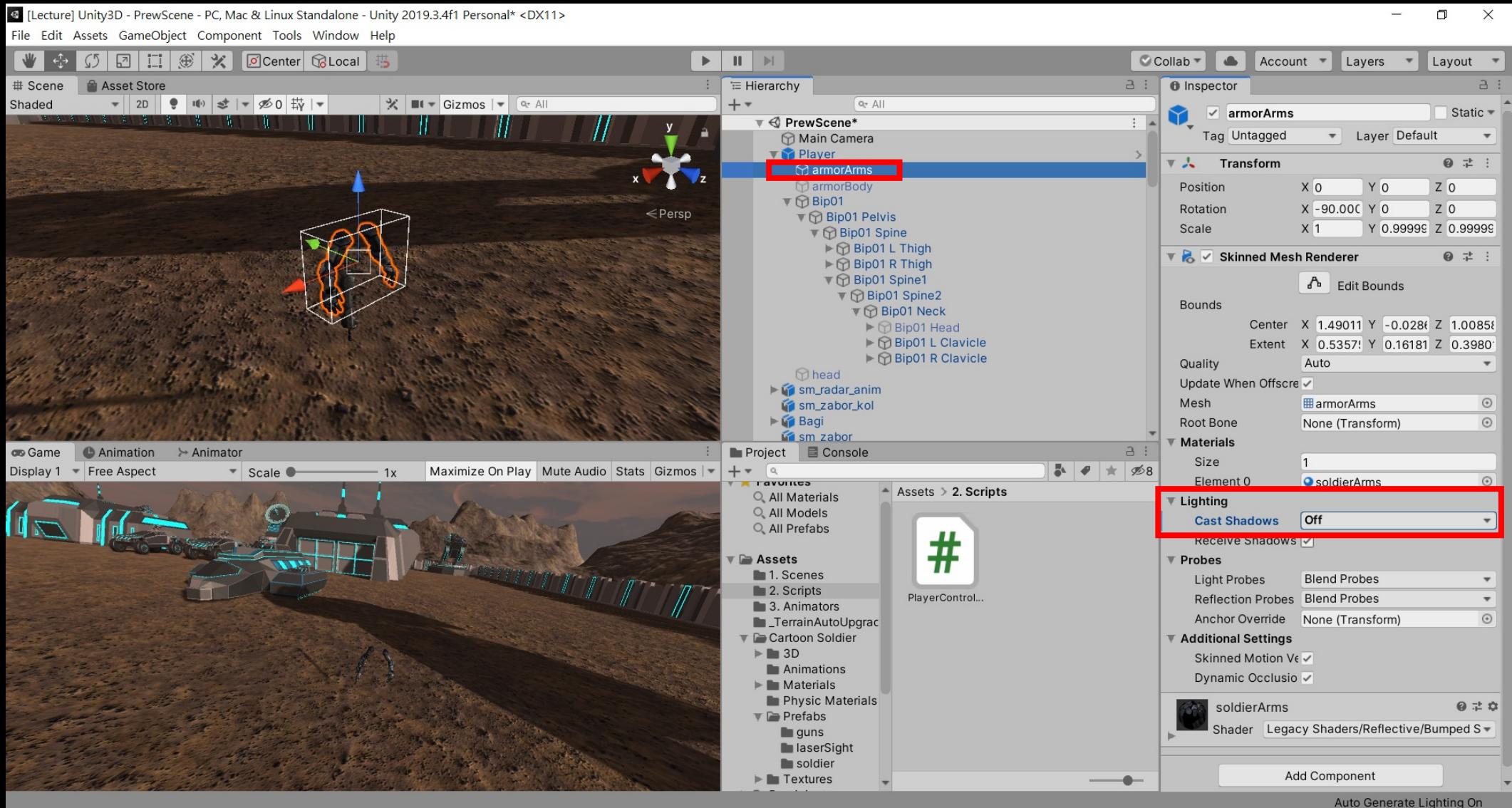


第一人稱視角

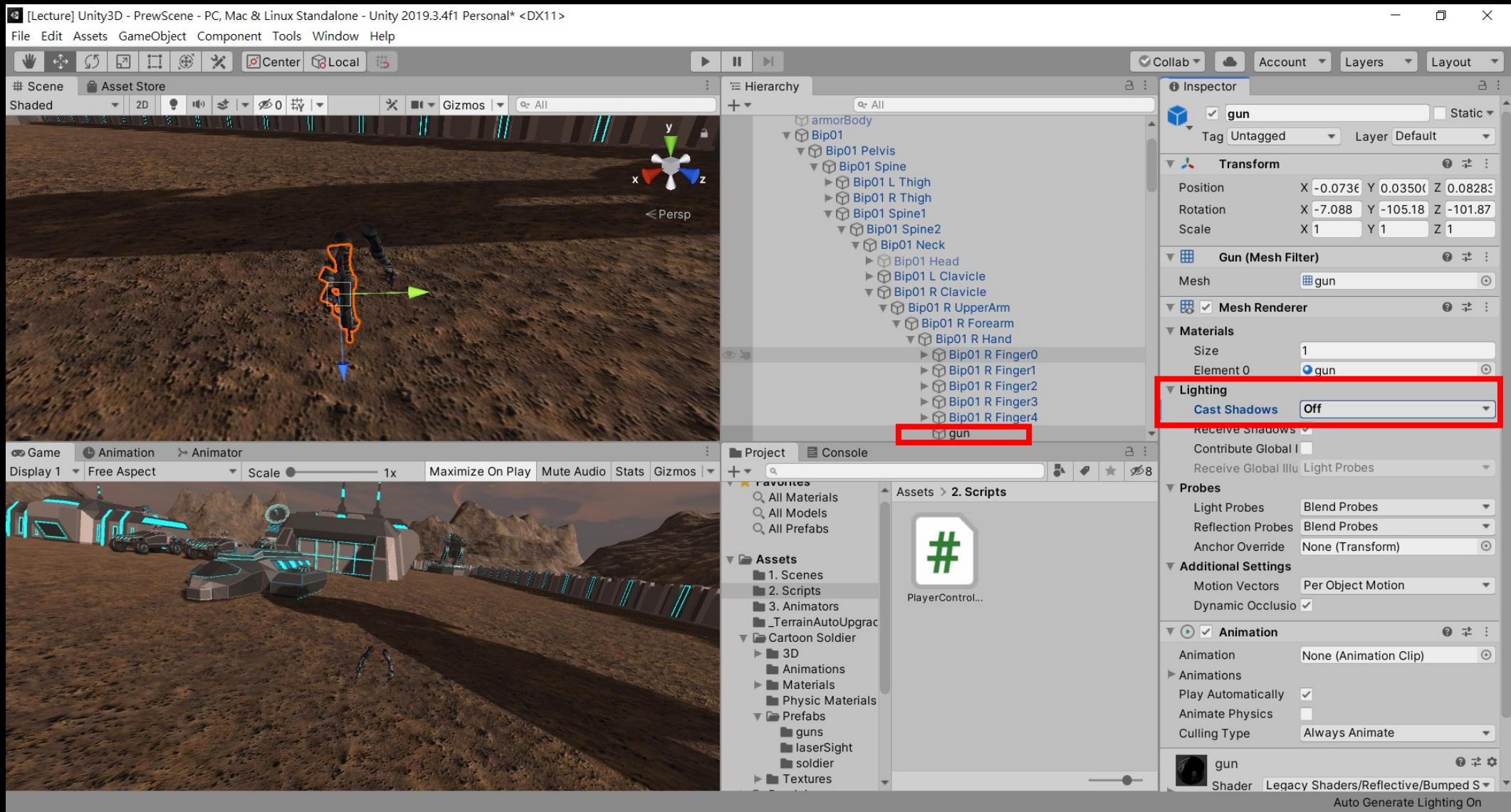
# 關閉armorBody, Bip01 Head, head



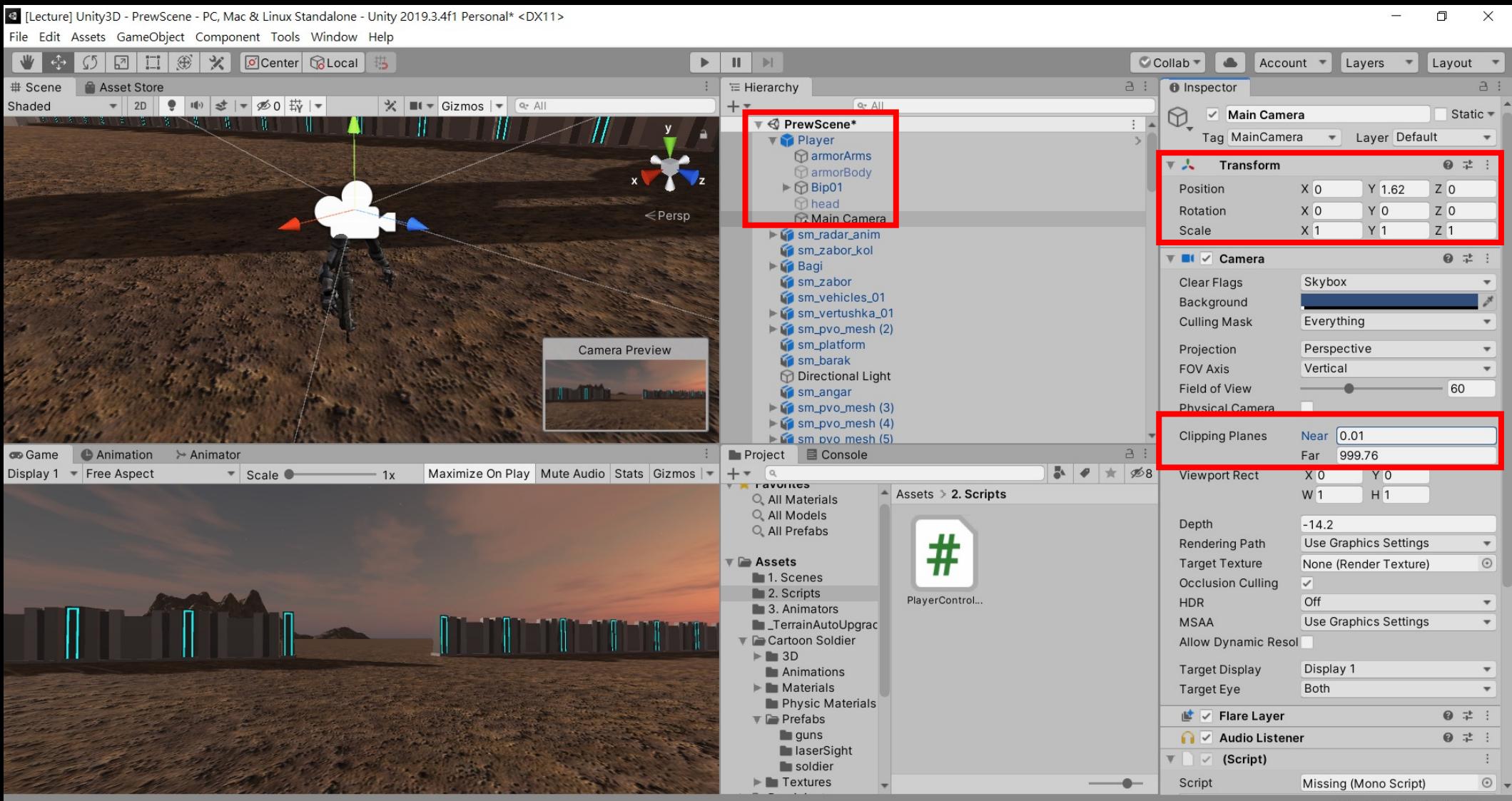
# 關閉ArmorArms的Cast Shadows



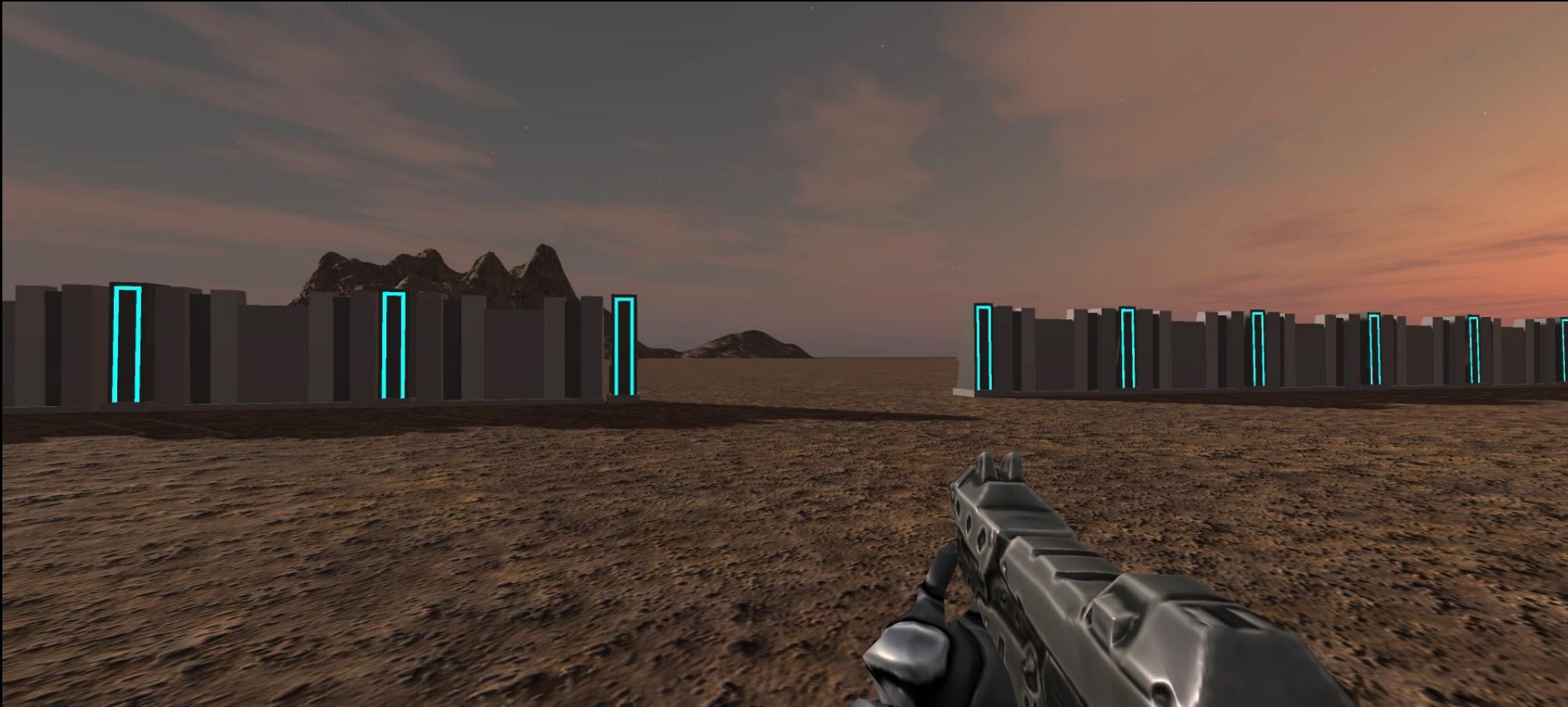
# 關閉gun的Cast Shadows



# 把Main Camera放在Player底下



# 第一人稱視角的移動

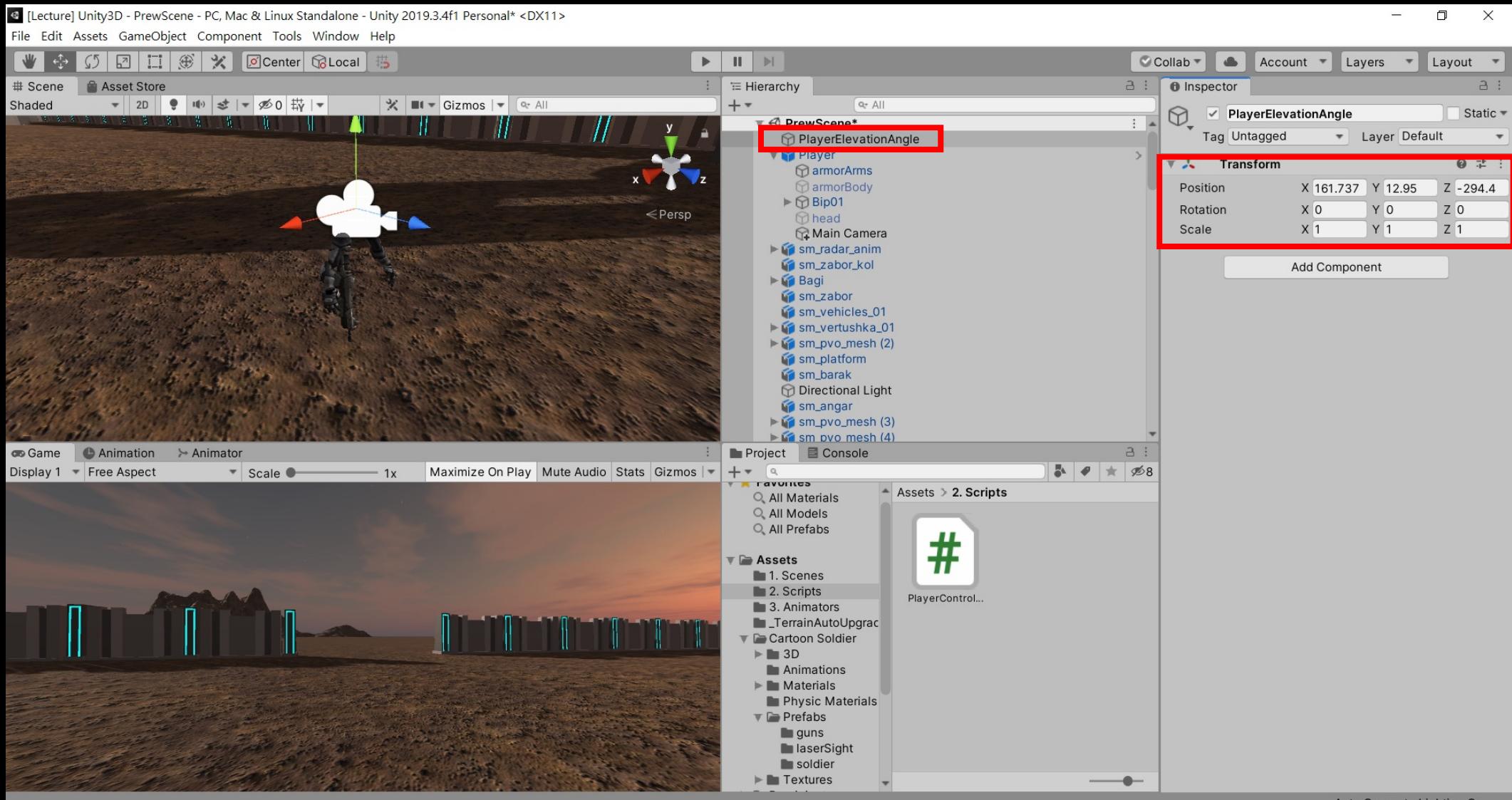


# 第一人稱視角控制與移動

# 旋轉主角視野

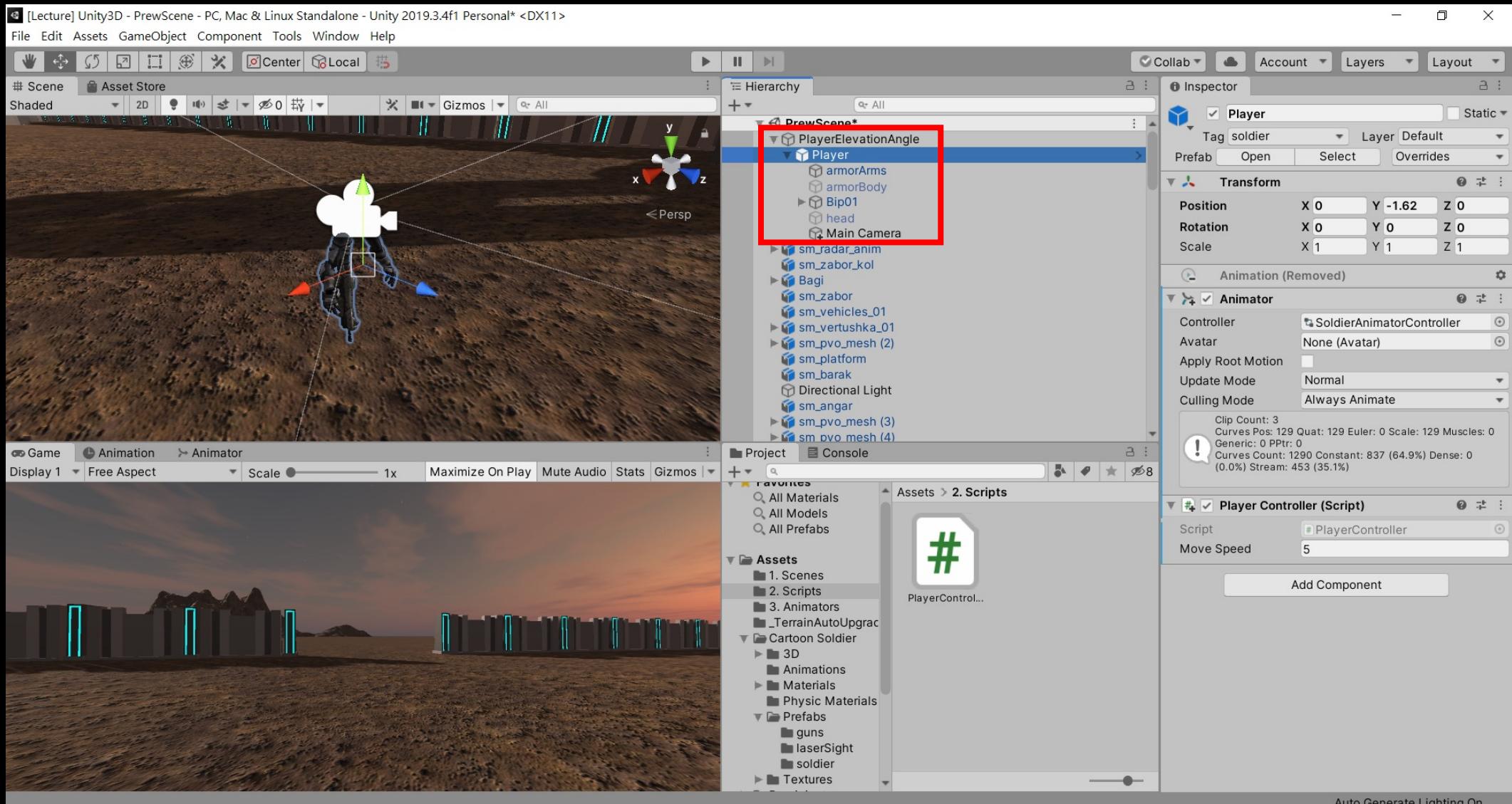
- 當滑鼠左右移動時,應該以主角為圓心沿著Y軸旋轉
- 當滑鼠上下移動時,應該以攝影機為圓心沿著X軸旋轉

# 建立新的空物件PlayerElevationAngle

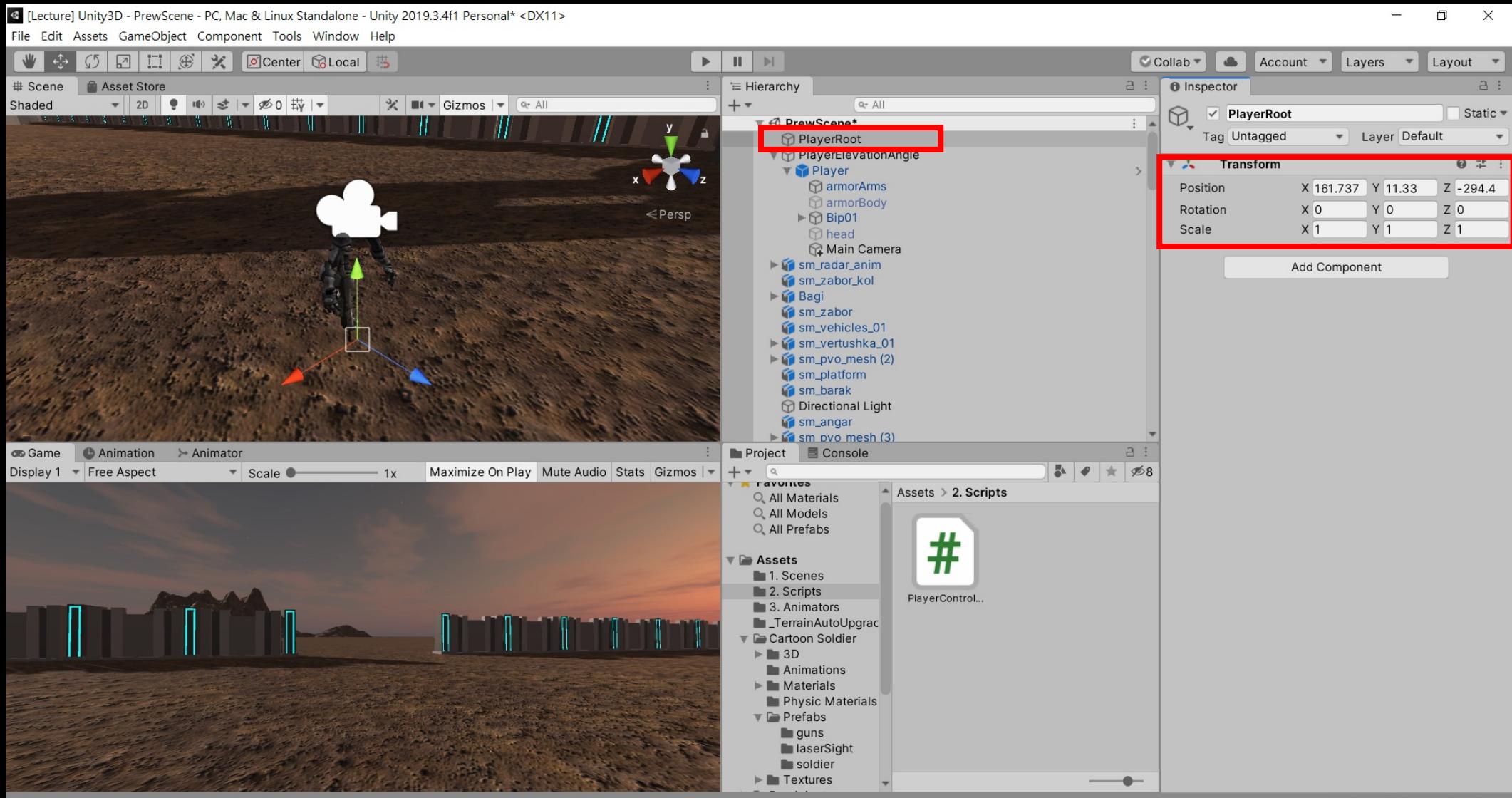


Auto Generate Lighting On

# 把Player放到PlayerElevationAngle底下

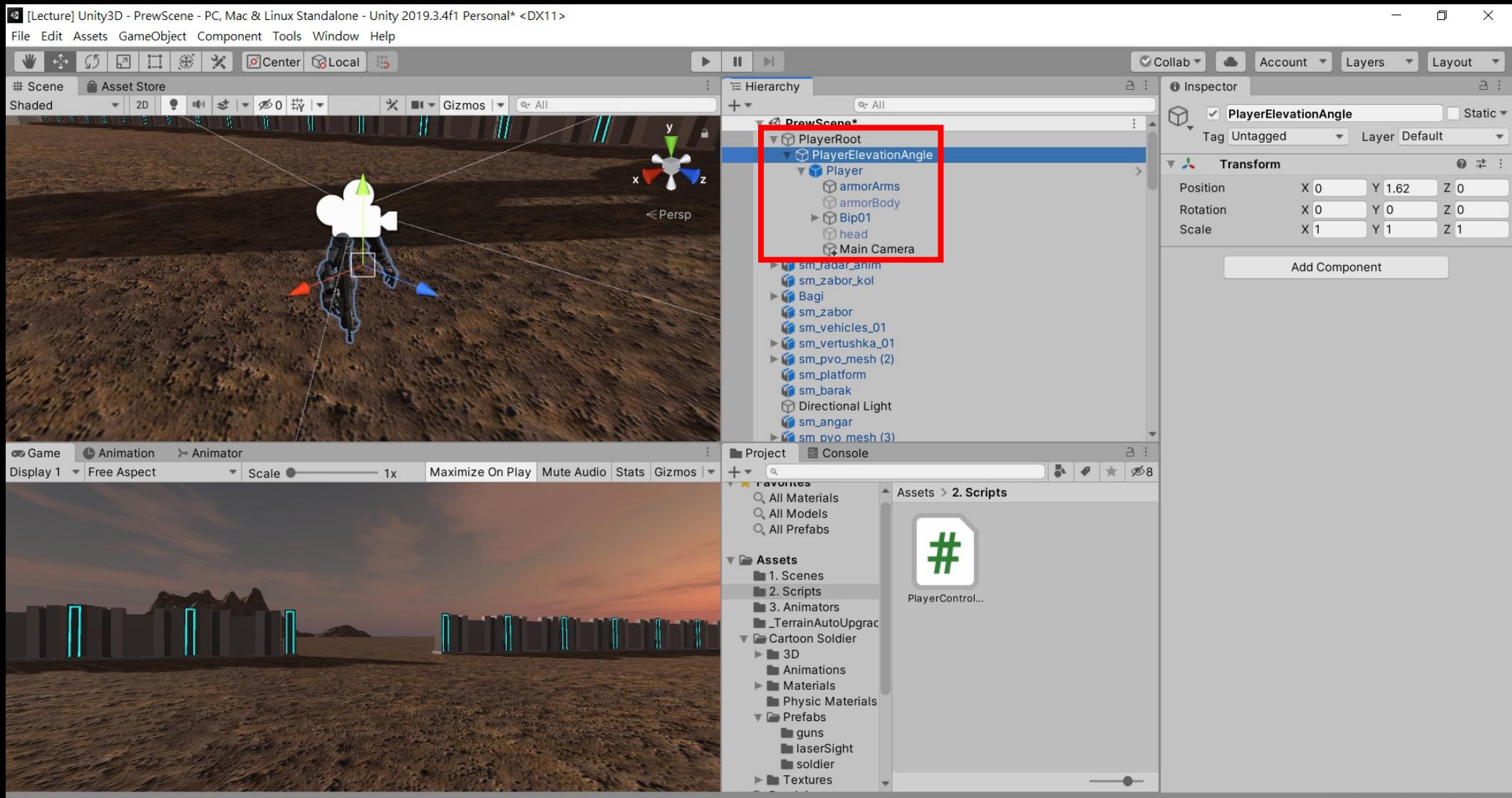


# 建立新的空物件PlayerRoot



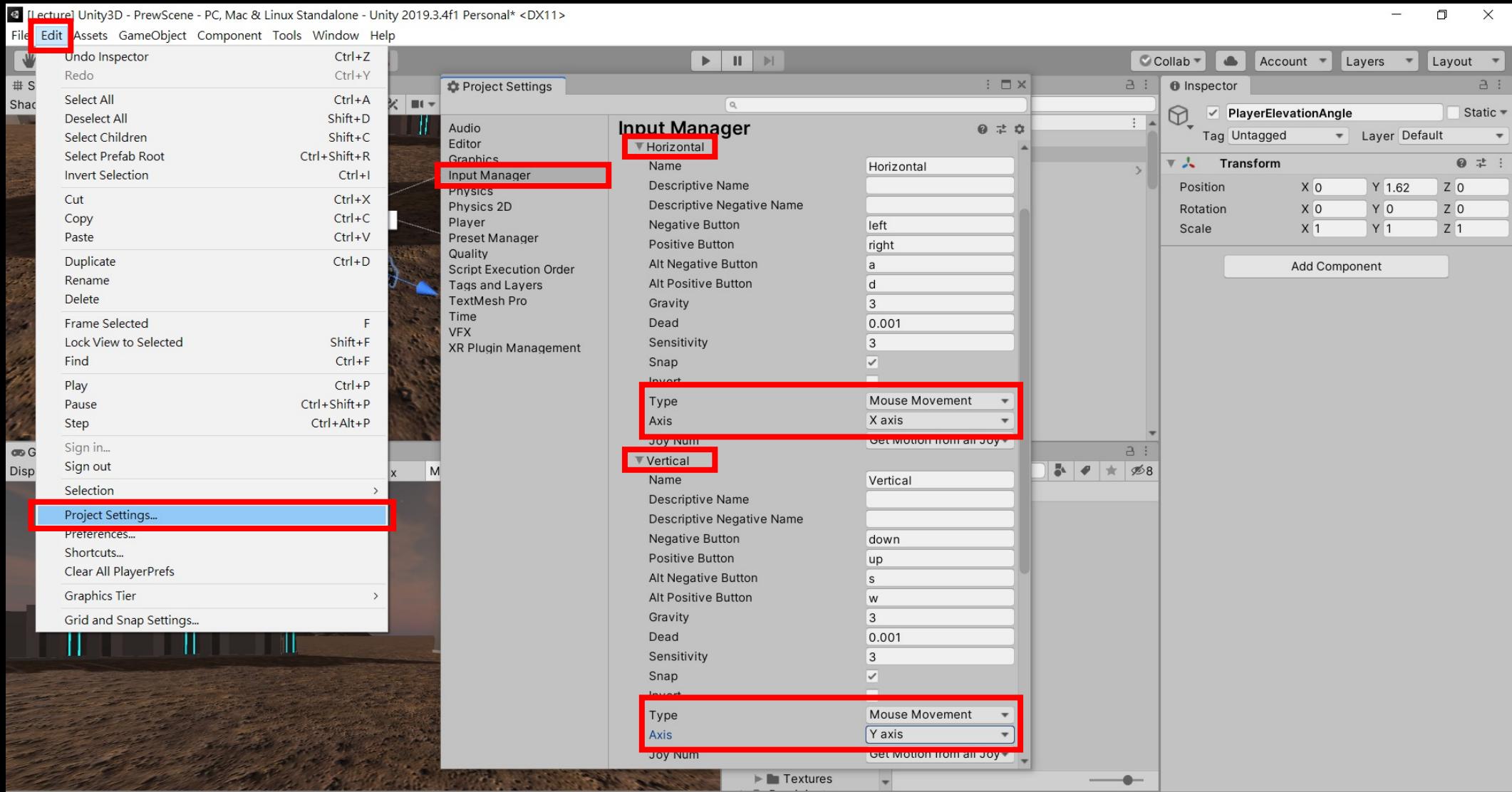
Auto Generate Lighting On

# 把PlayerElevationAngle放到PlayerRoot底下



Auto Generate Lighting On

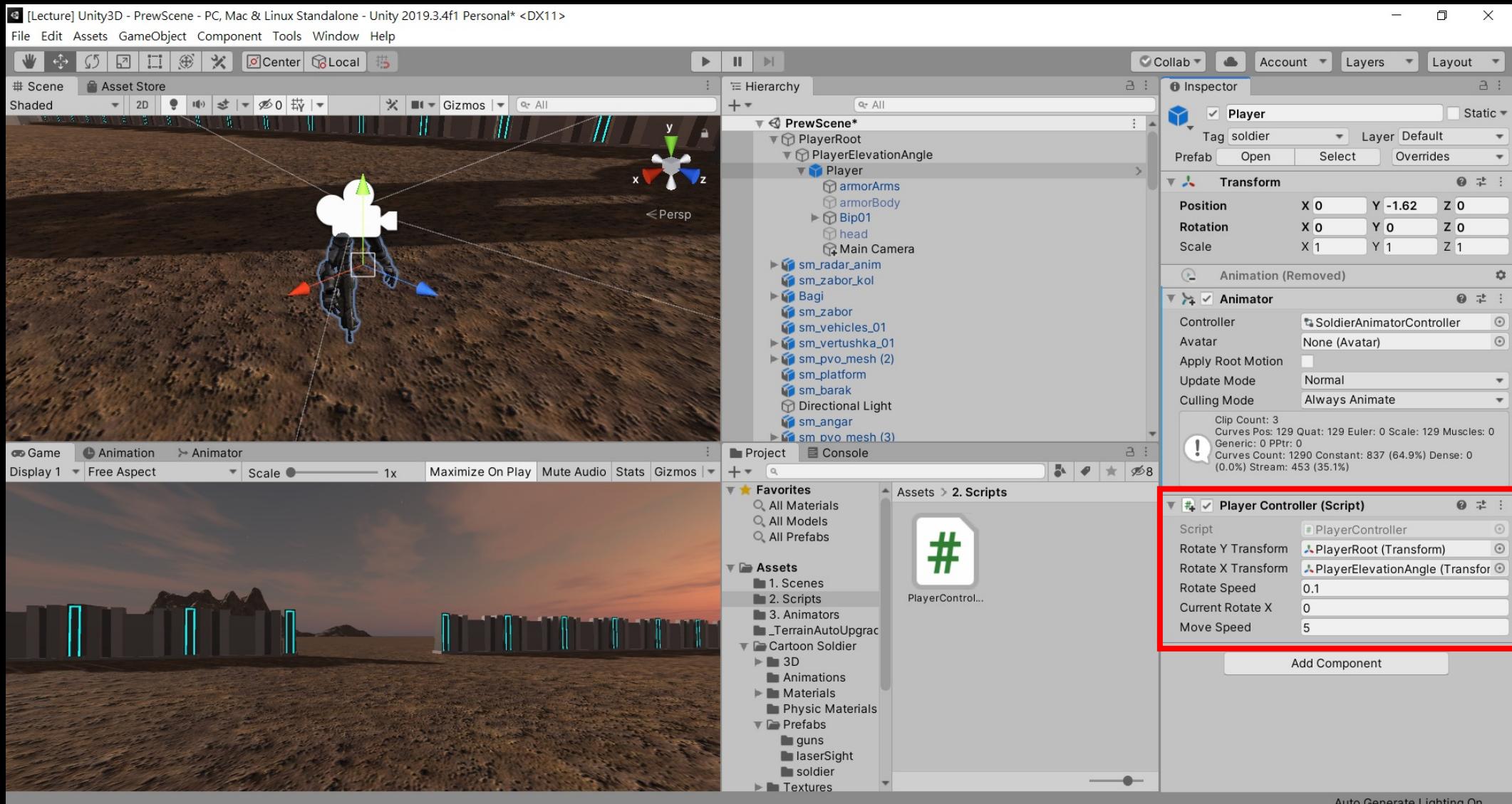
# 定義Horizontal和Vertical



# 修改PlayerController.cs

- <https://bit.ly/349eCoP>

# 設定PlayerController的參數

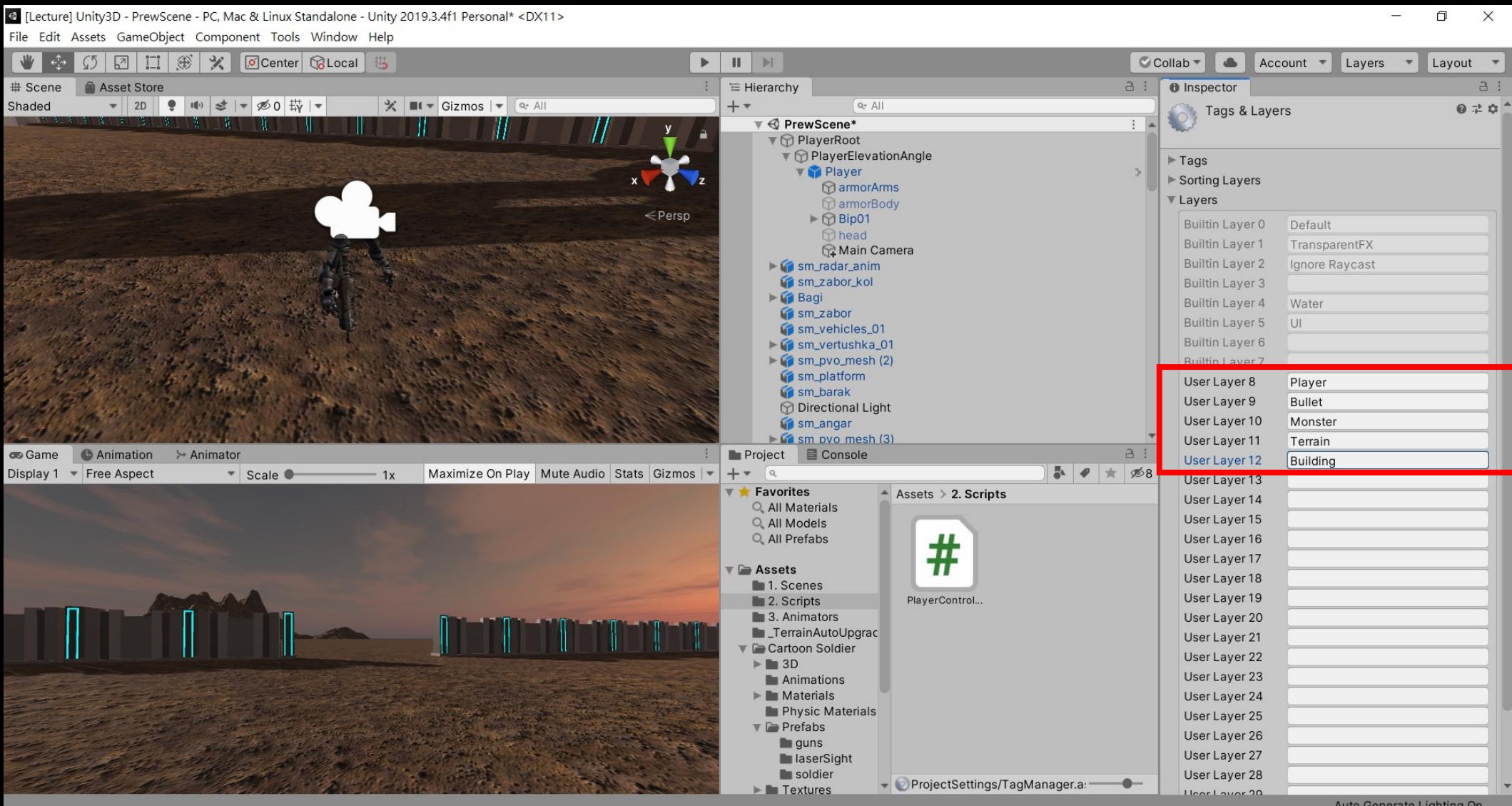


移動滑鼠可以自由轉動

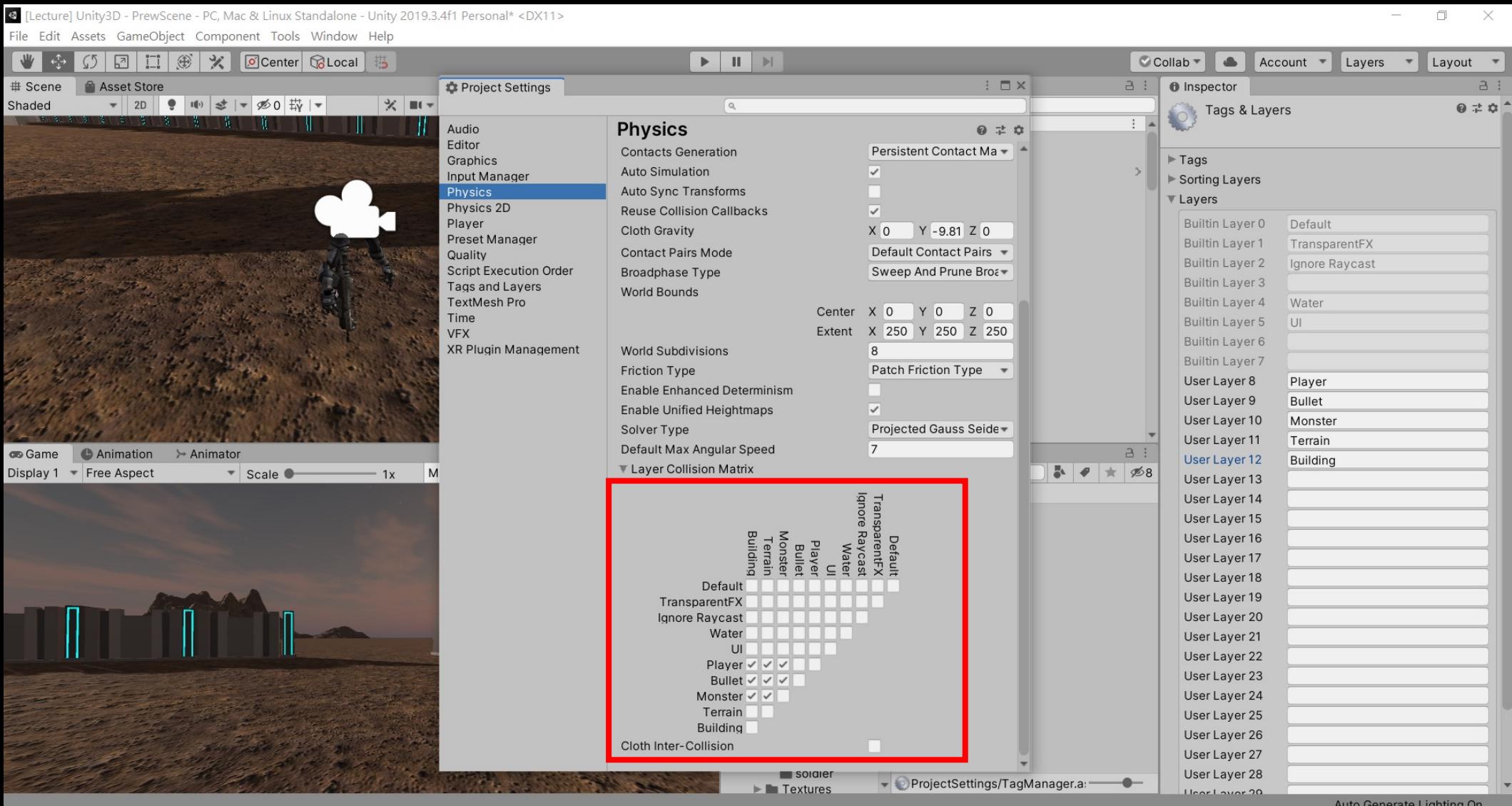


# 物理碰撞

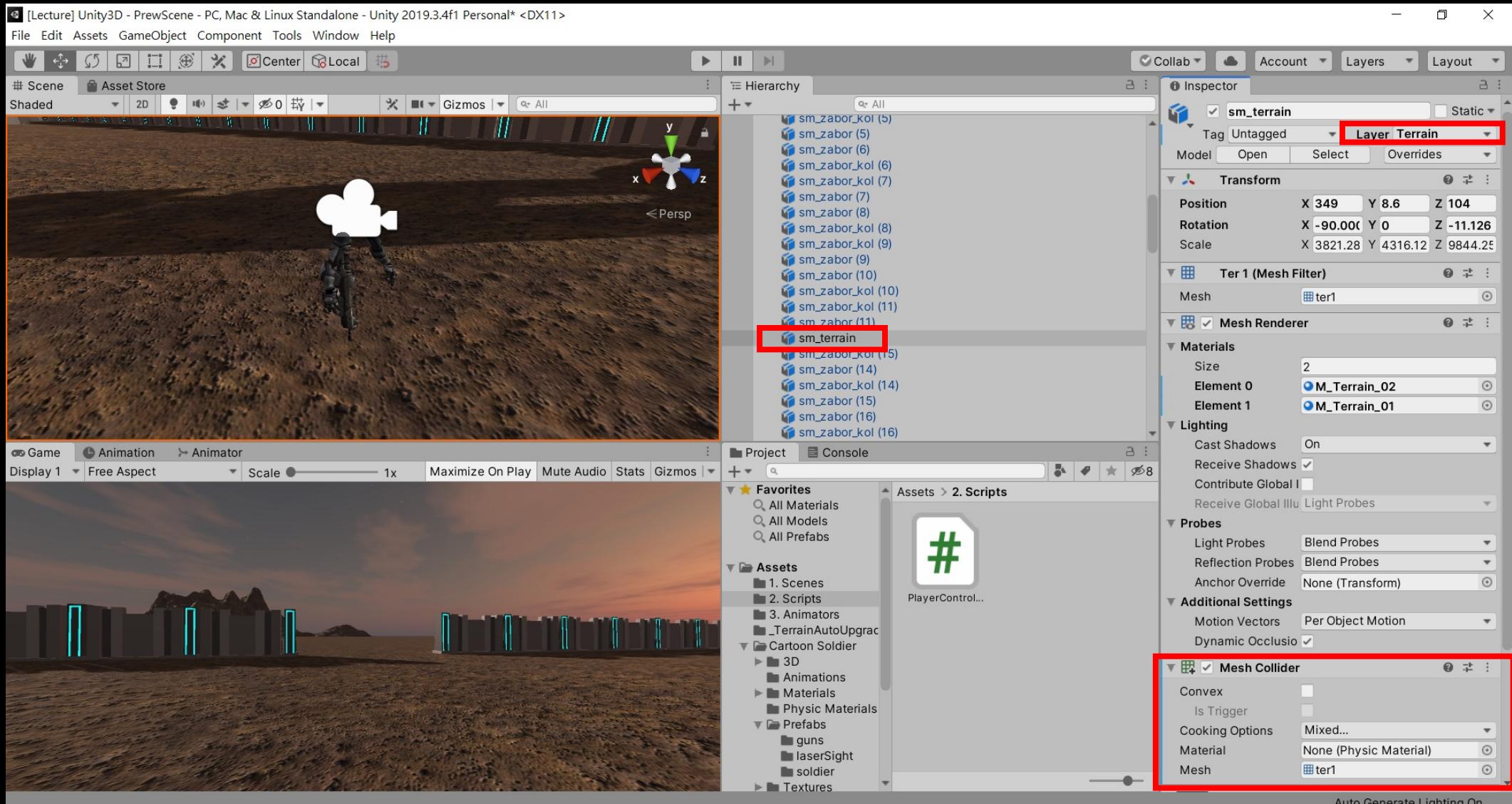
# 設定Layer



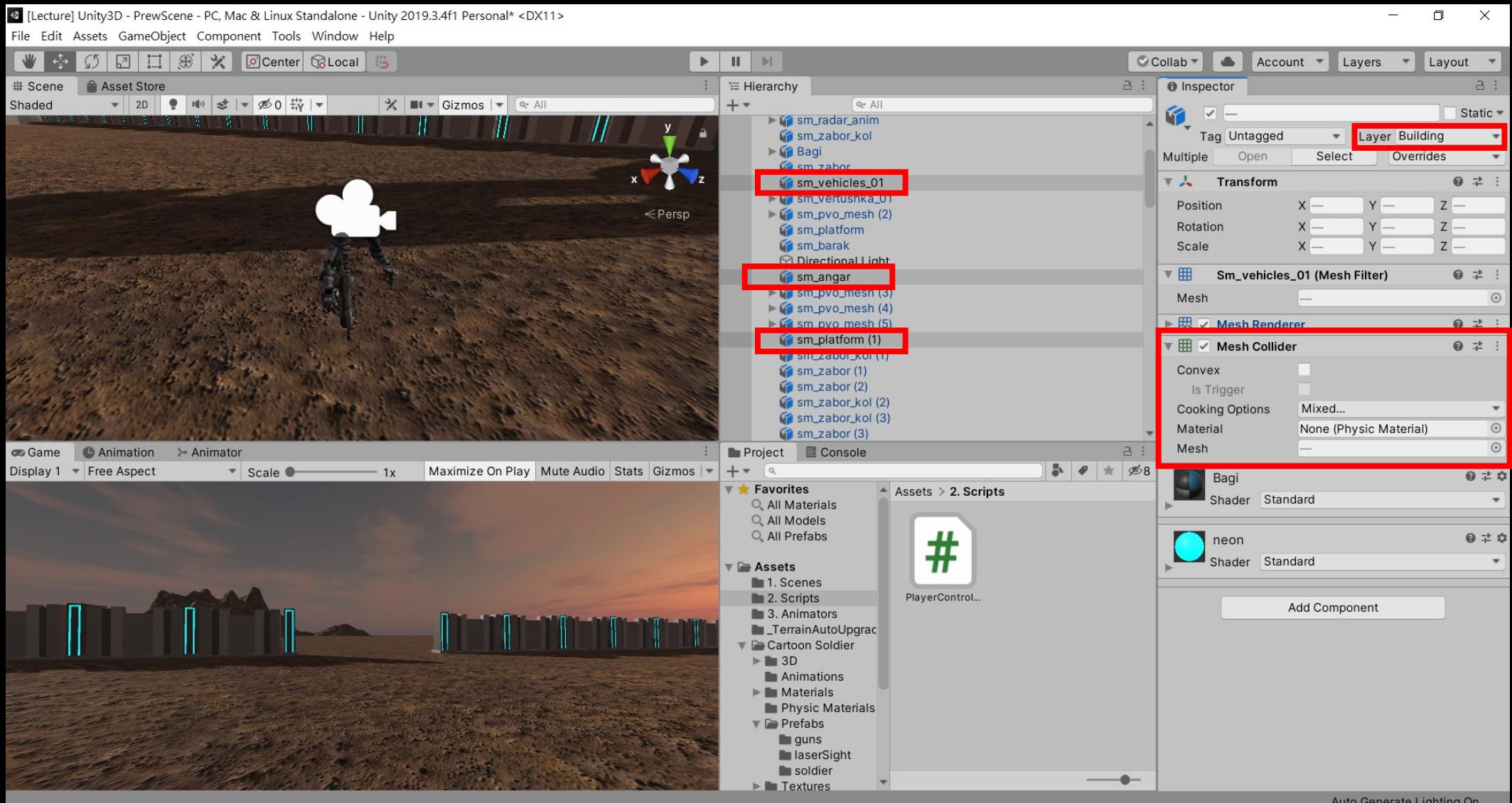
# 設定物理碰撞Matrix



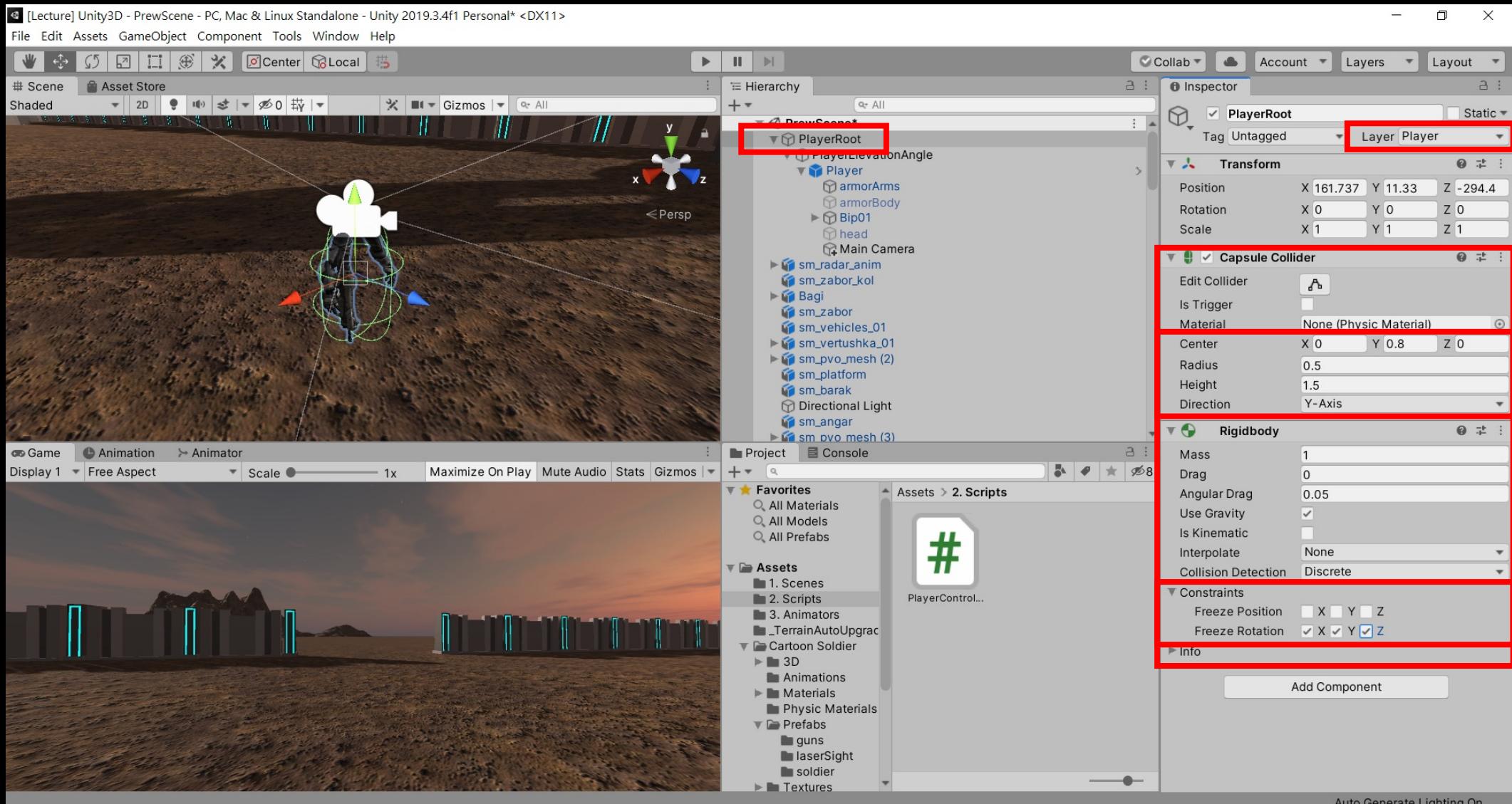
# 設定地板



# 設定要碰撞的建築物



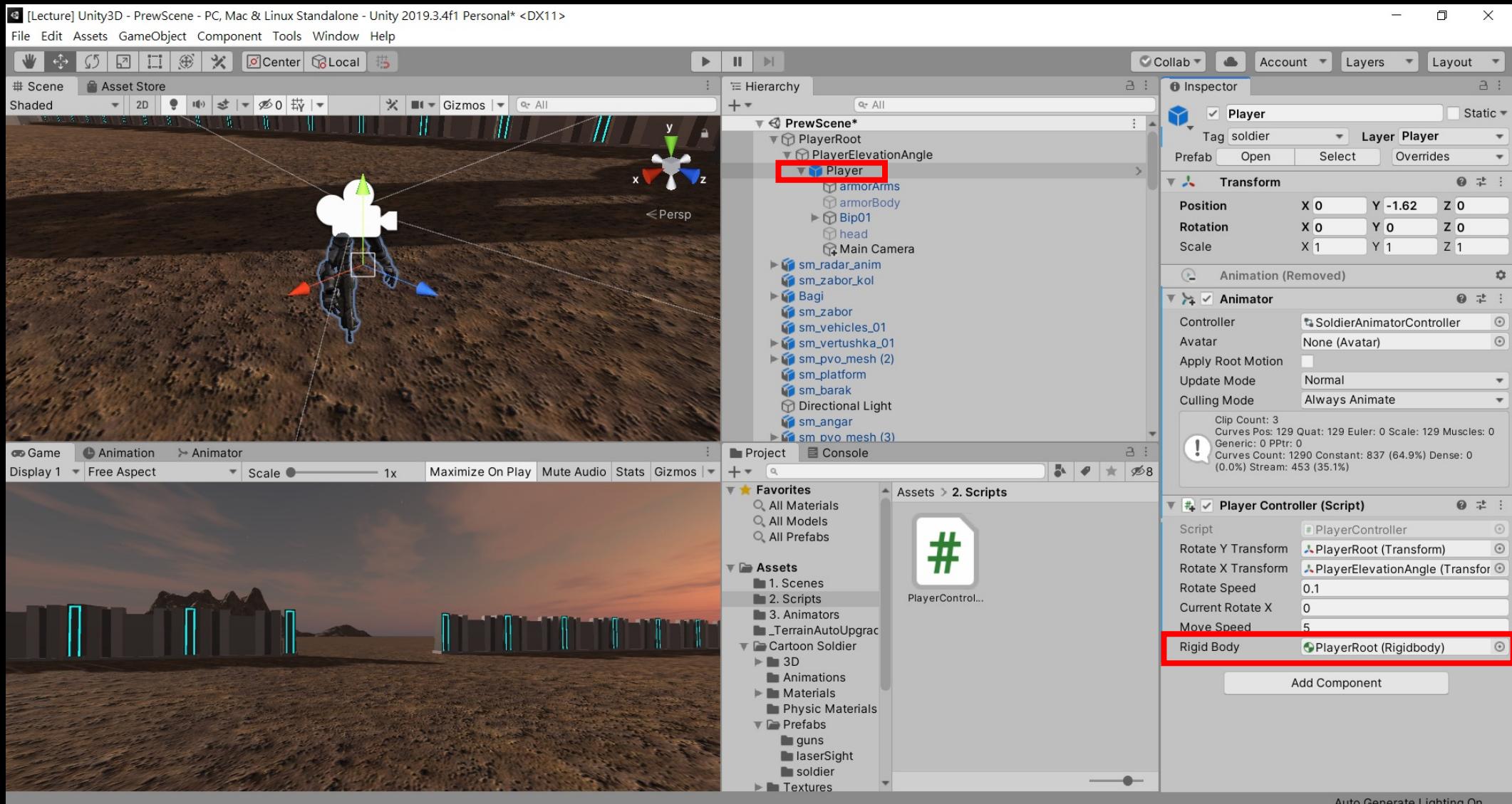
# 設定PlayerRoot



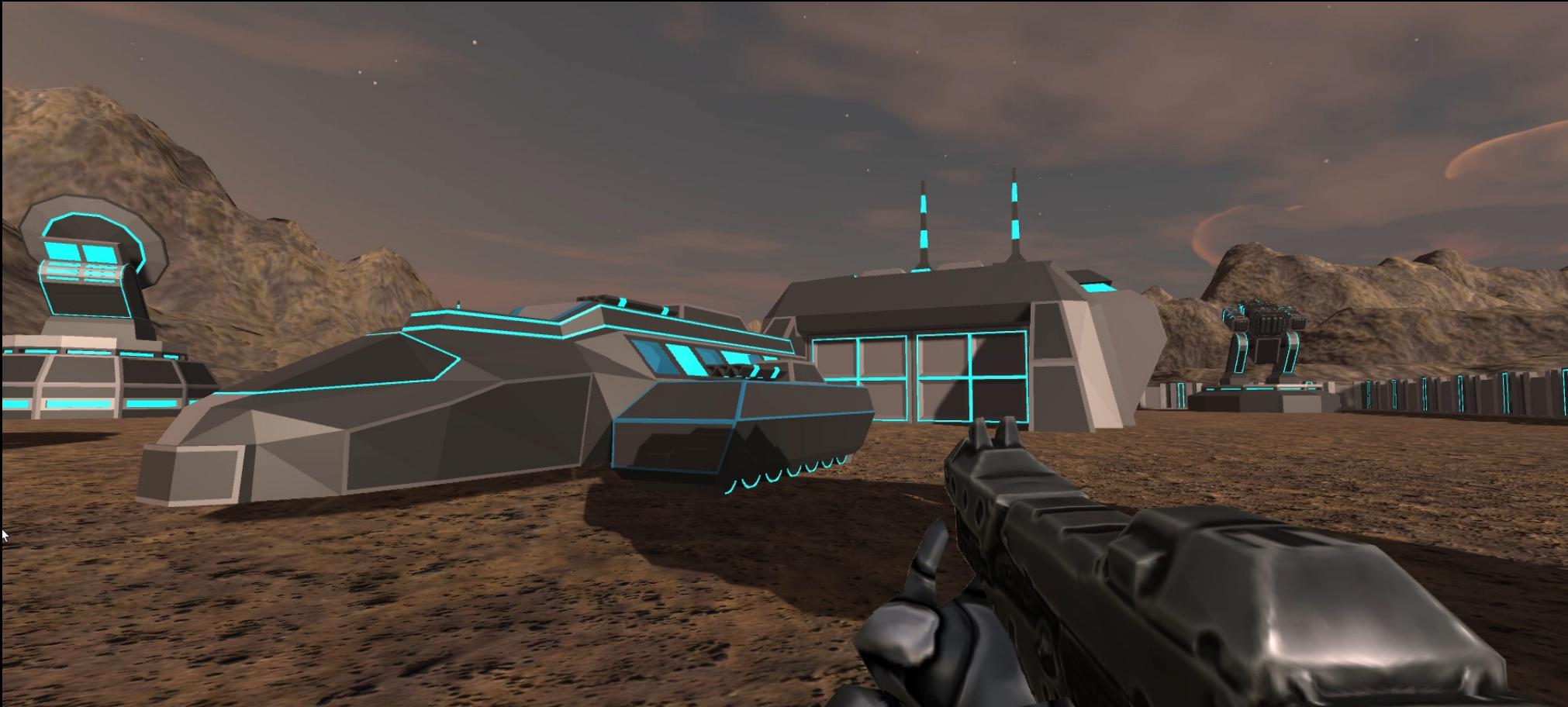
# 修改PlayerController.cs

- <https://bit.ly/2K5mfpf>

# 設定Player Controller上的RigidBody

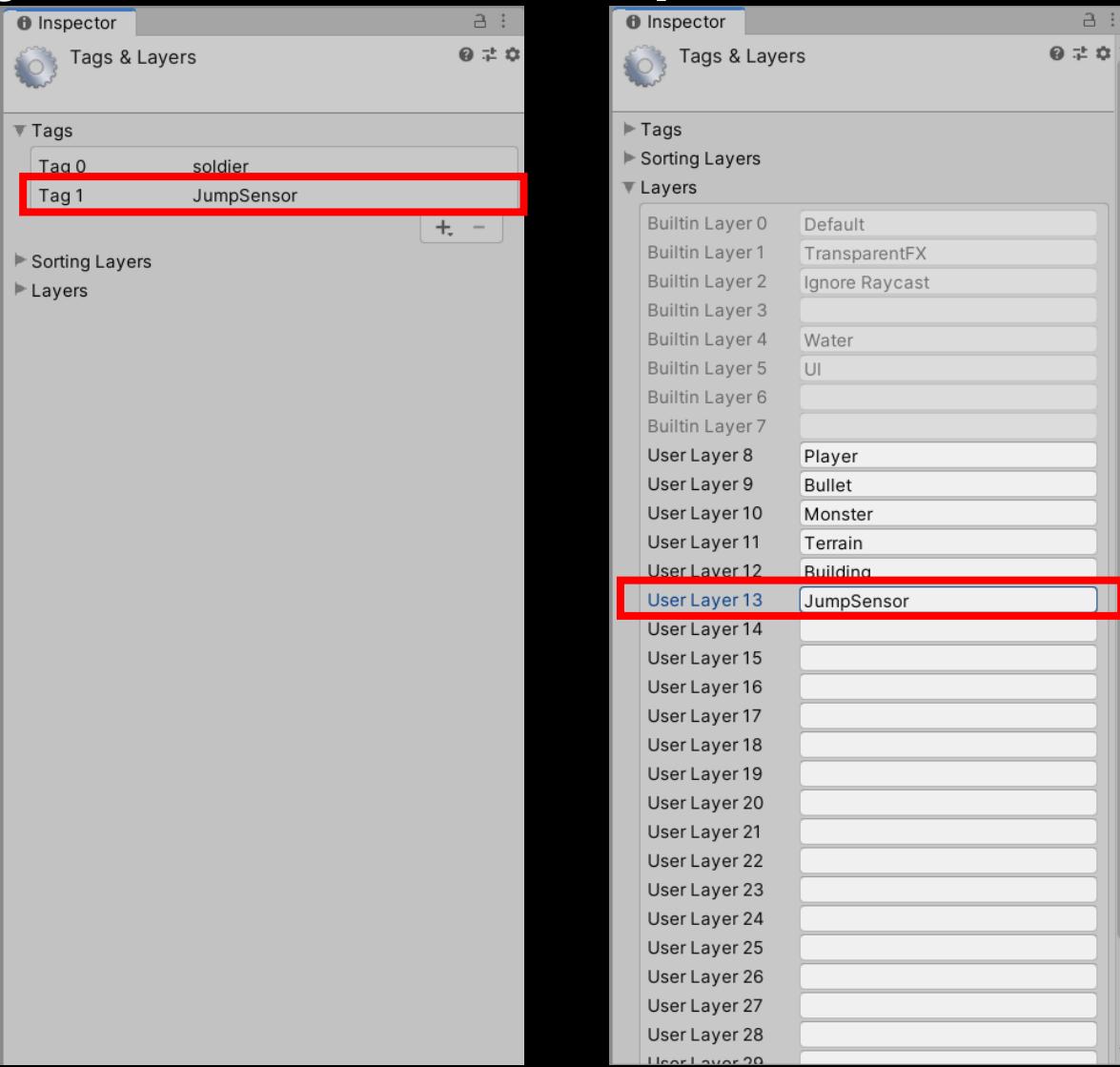


有物理碰撞了

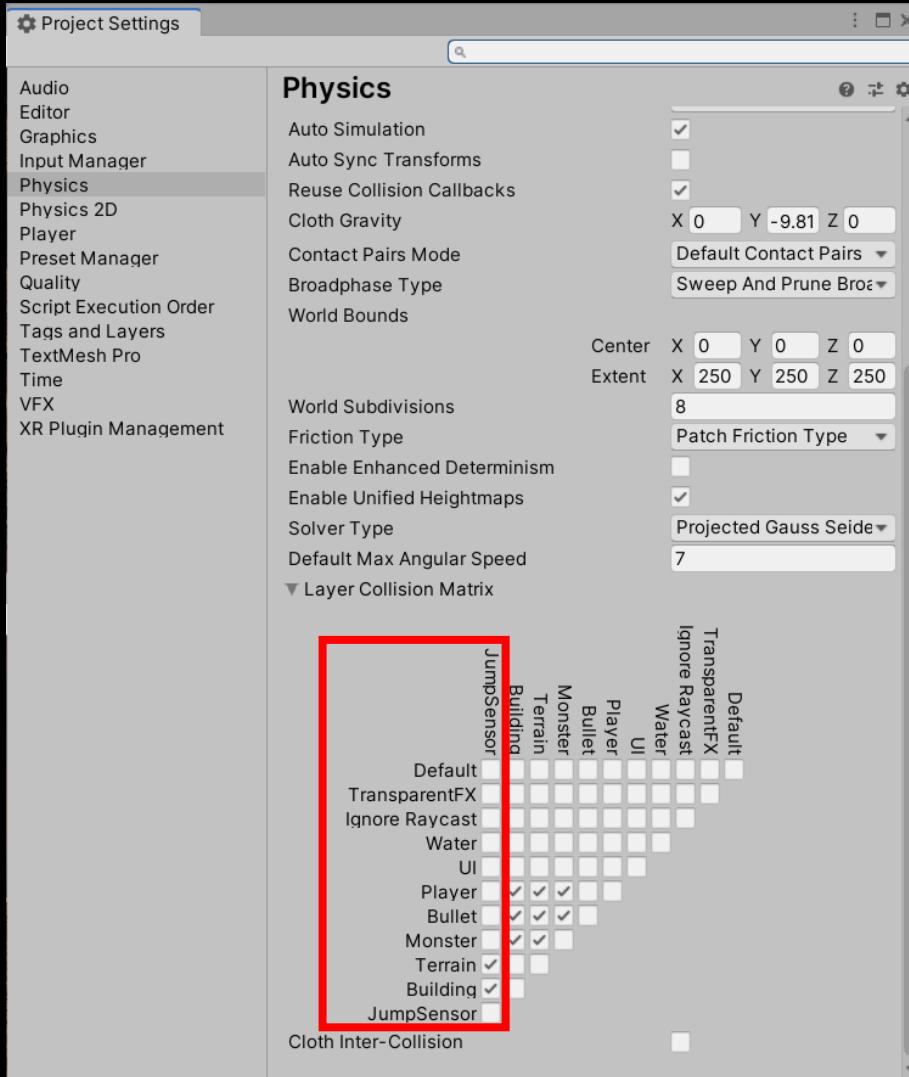


角色跳躍

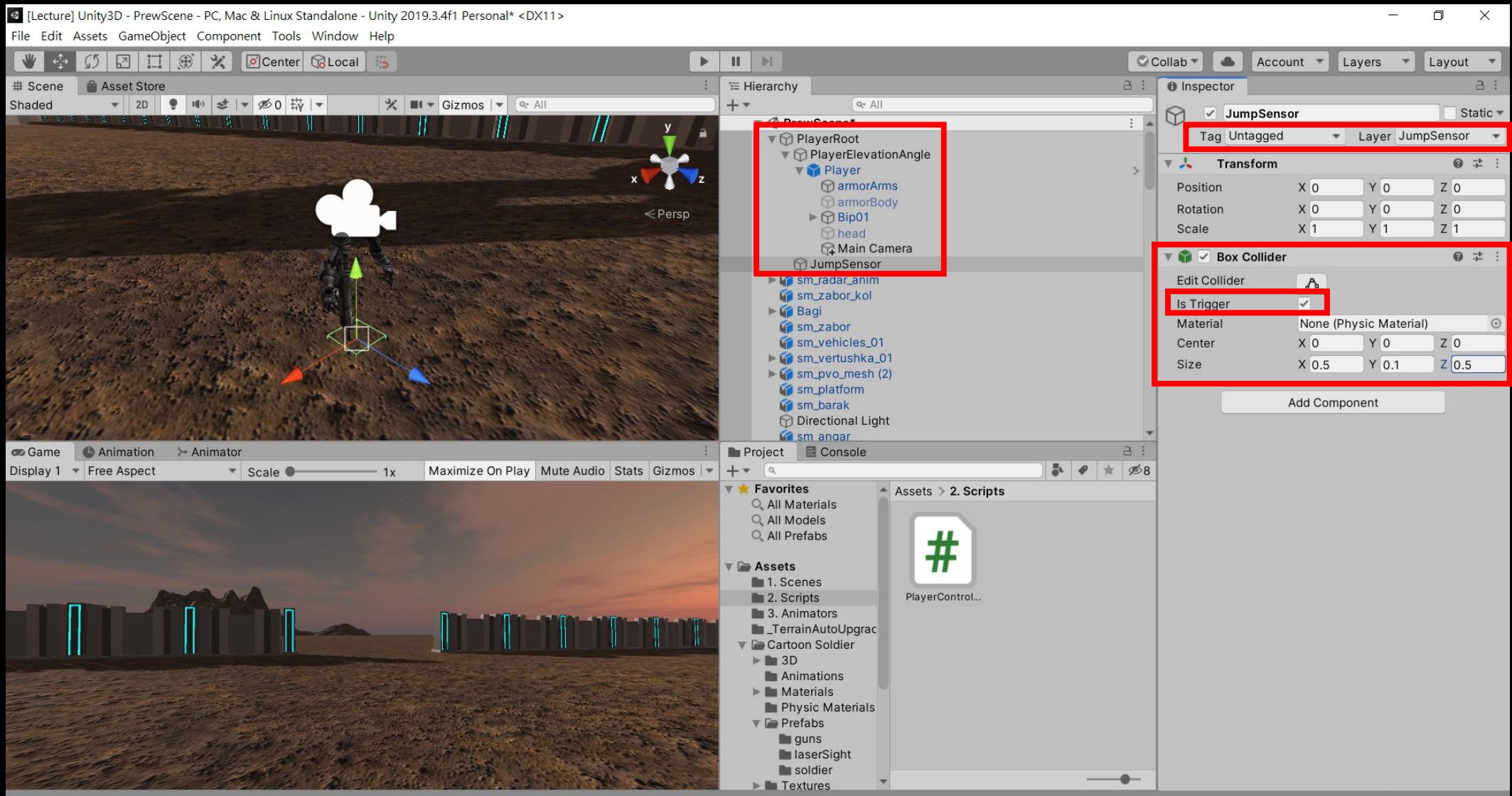
# Tags和Layers新增JumpSensor



# 修改Collision Matrix

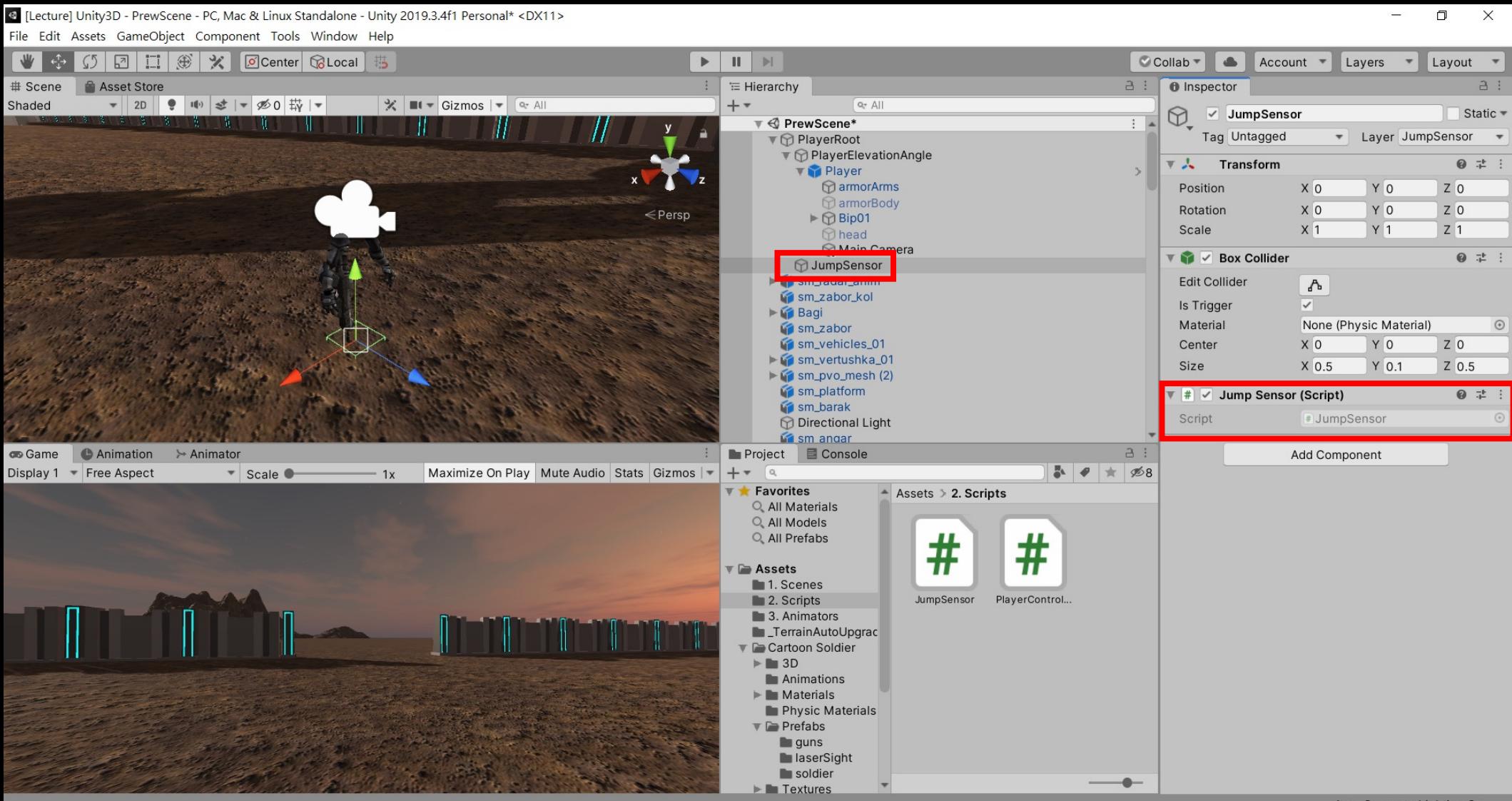


# PlayerRoot底下建立空物件JumpSensor



Auto Generate Lighting On

# 新增JumpSensor.cs



Auto Generate Lighting On

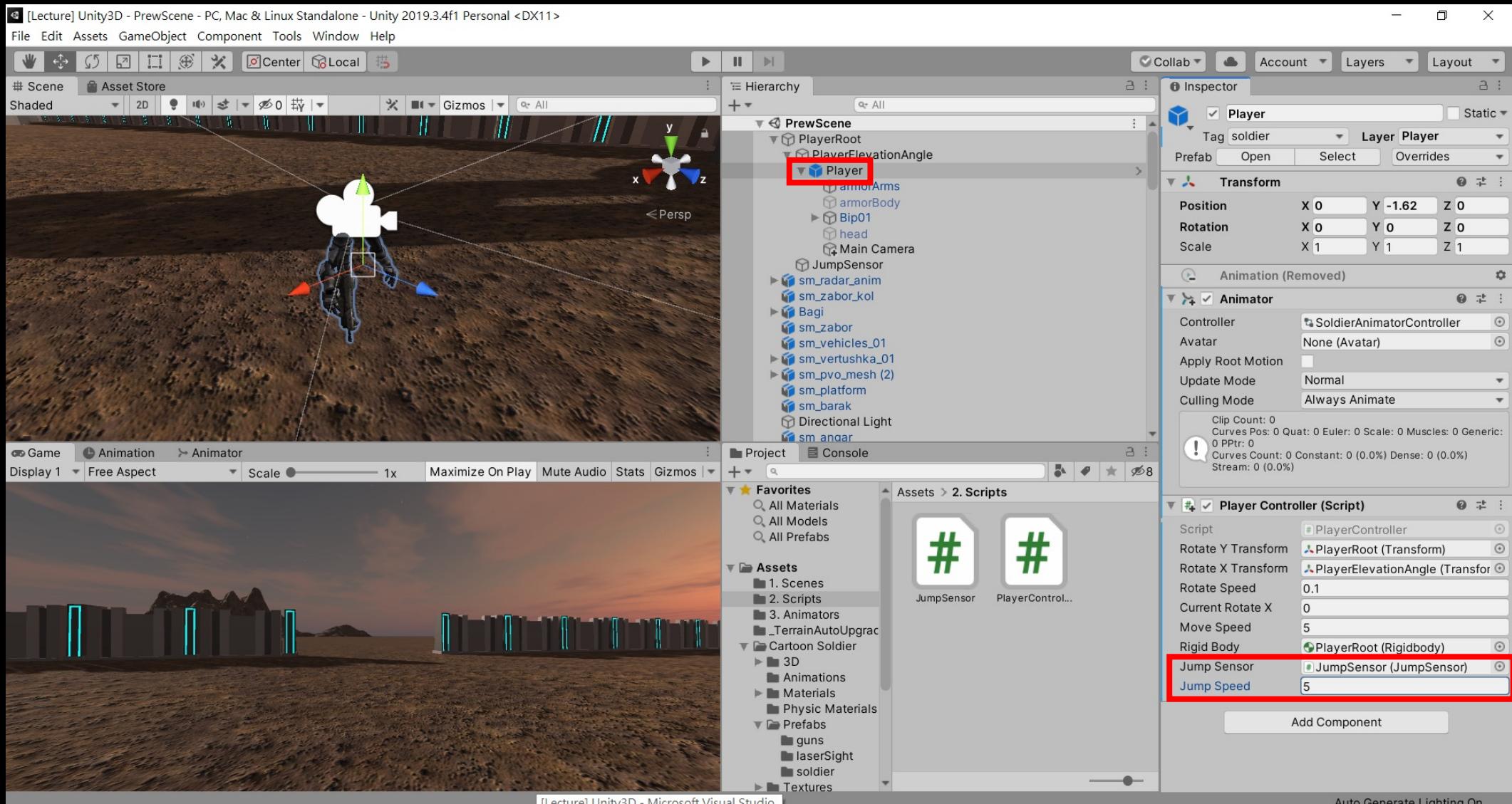
# JumpSensor.cs

- <https://bit.ly/3gWjFOF>

# 修改PlayerController.cs

- <https://bit.ly/3mfga6L>

# 指定JumpSensor和JumpSpeed



按空白鍵就可以跳起來了

