

Shader Graph

Unity Lab

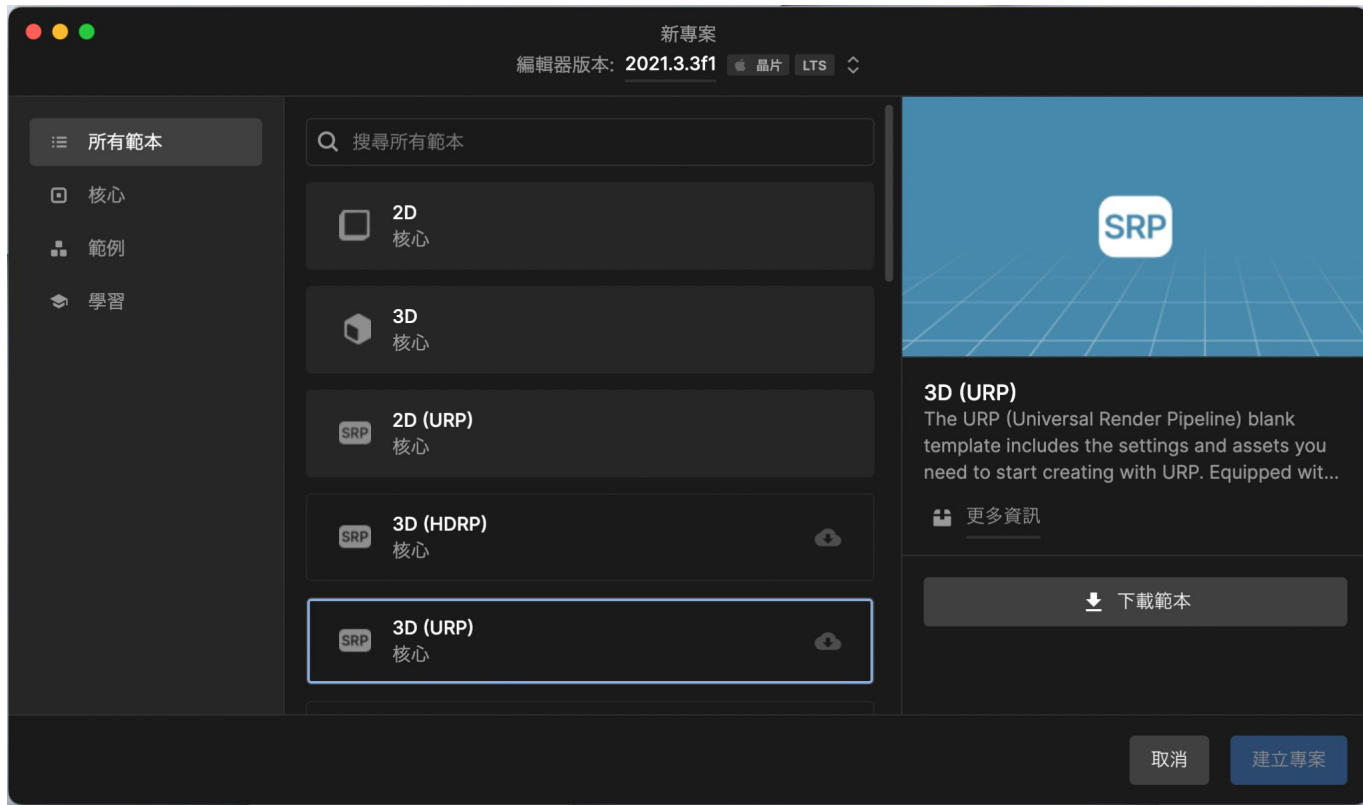
Shader Graph

Shader Graph is a tool that enables you to build shaders visually. Instead of writing code, you create and connect nodes in a graph framework.

Shader Graph gives instant feedback that reflects your changes, and it's simple enough for users who are new to shader creation.

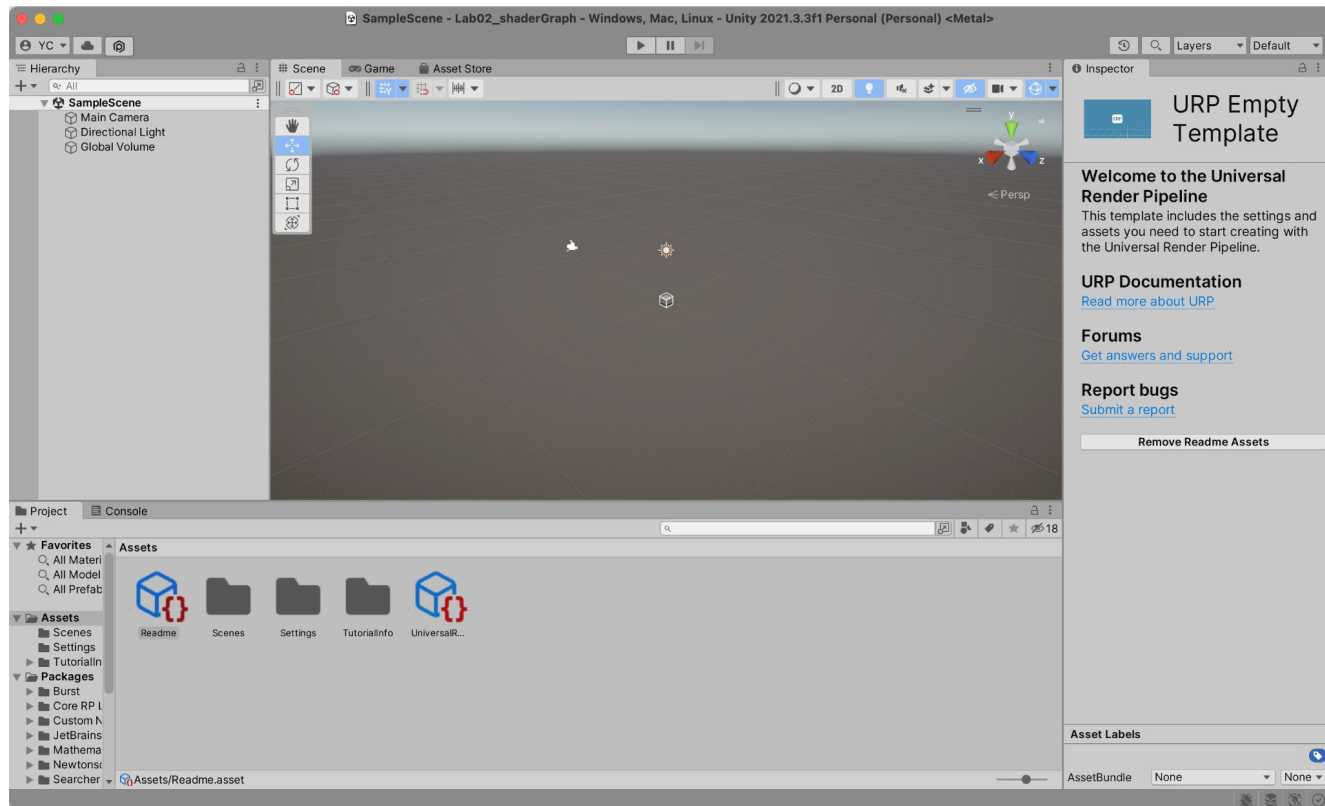
- 圖形化介面
 - 不用寫程式
- 自定義 Shader

開新專案



3D(URP) 或是 3D(HDRP)
才能使用 Shader Graph

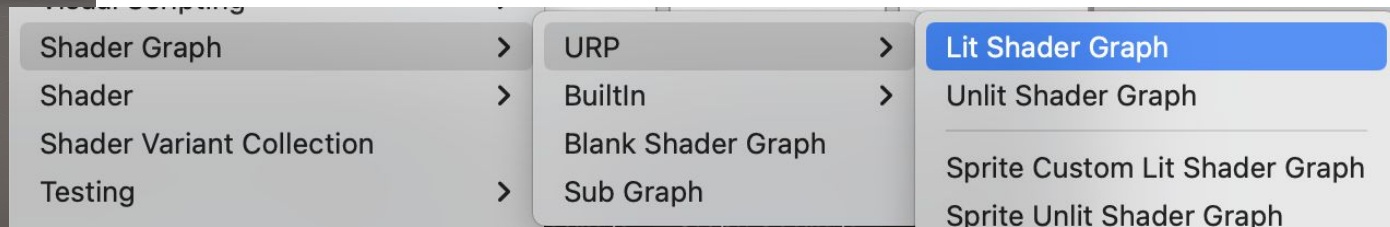
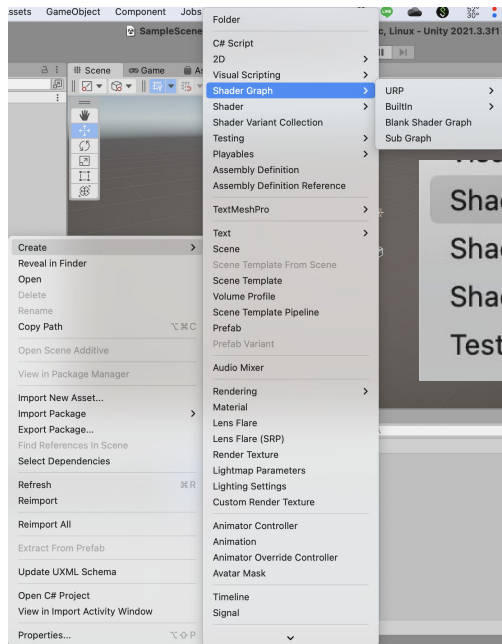
開新專案



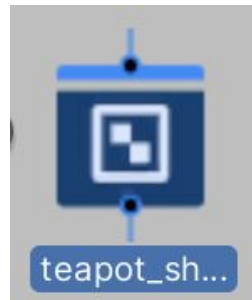
Shader Graph

按右鍵選擇 Create -> Shader Graph 就
可以建立 Shader Graph

這次的範例使用 URP -> Lit Shader Graph

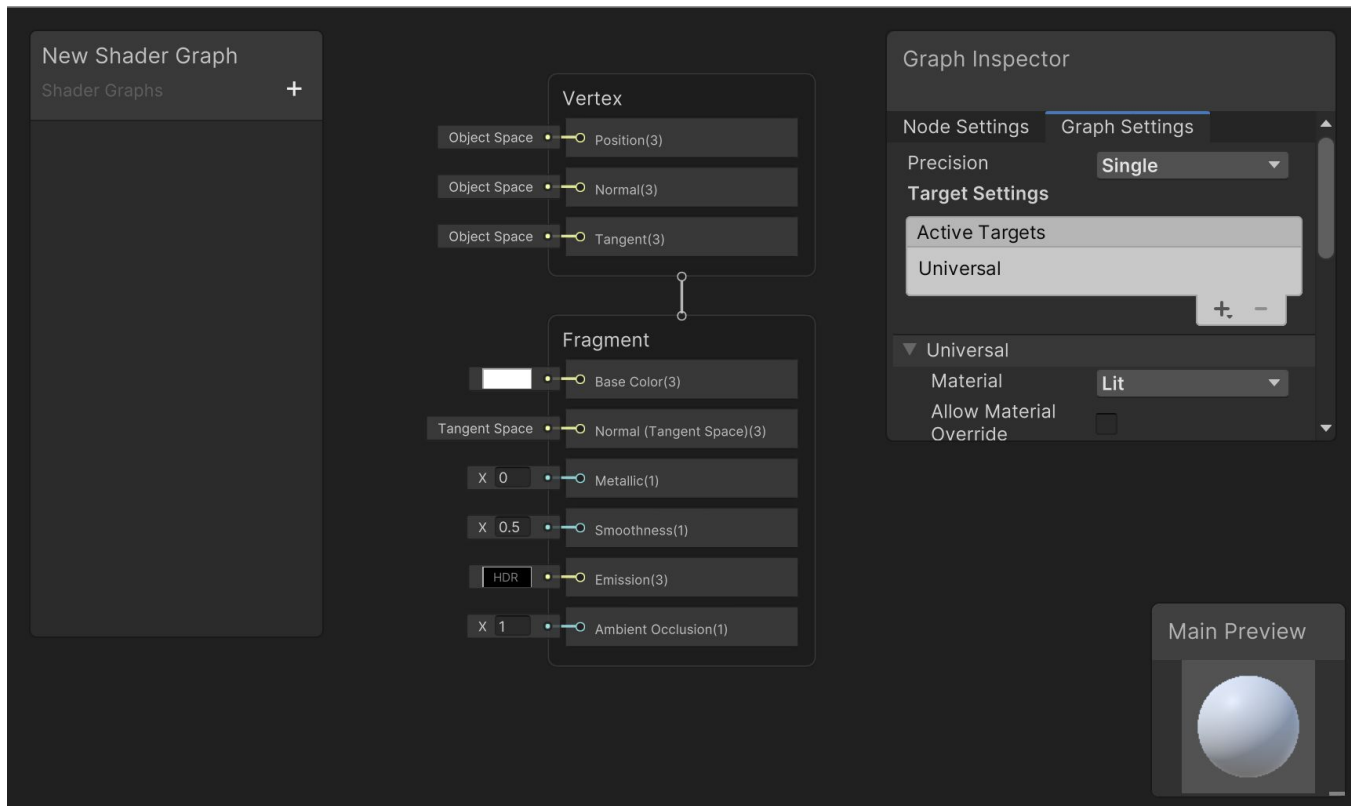


成功建立 shader graph



Shader Graph

進入編輯 Shader
Graph 的畫面



Shader Graph

可以隨意調整 Fragment 的參數

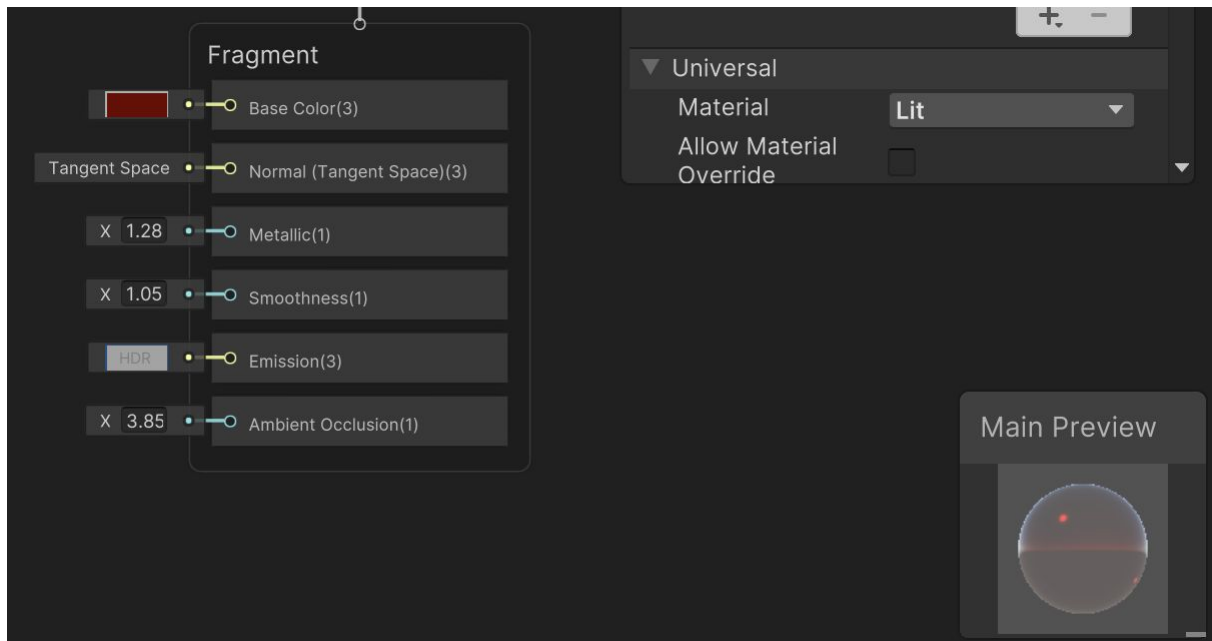
Base Color: 底色

Metallic: 金屬度

Smoothness: 光滑度

Emission: 環境光色

Ambient Occlusion: 環境光遮蔽

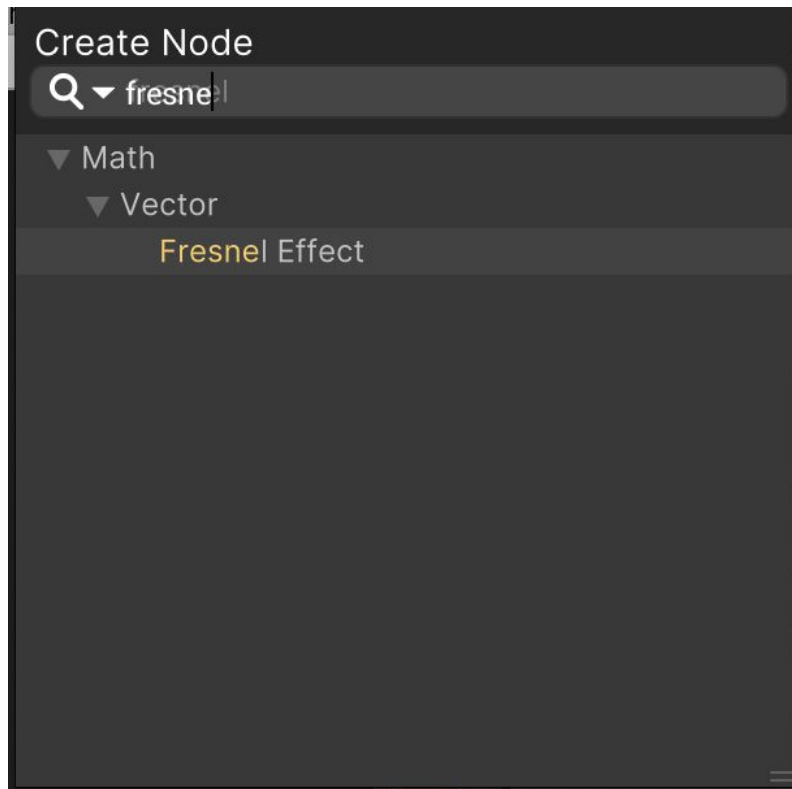


新增節點

- 預設的環境光是均勻的
- 透過新增節點 (Node) 來自定義環境光色

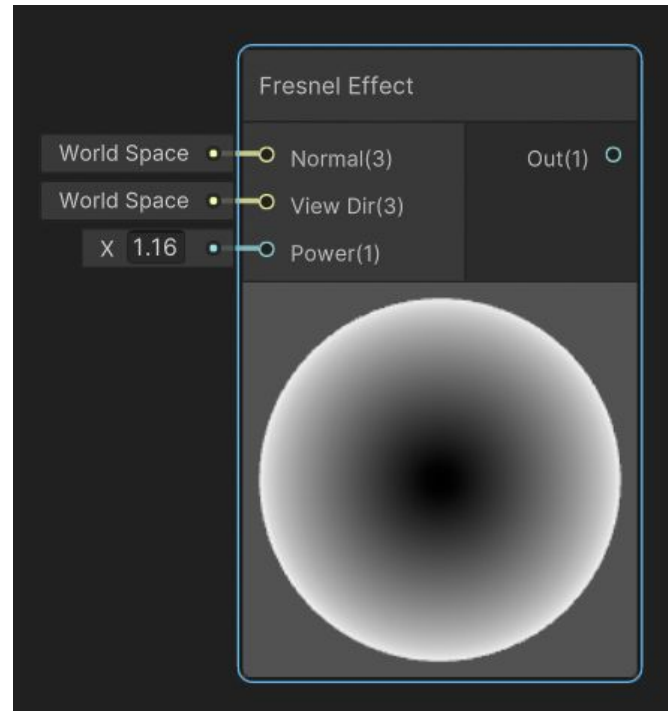
新增節點

- 按空白鍵新增節點
- 搜尋 Fresnel Effect 並新增
 - 邊緣光特效



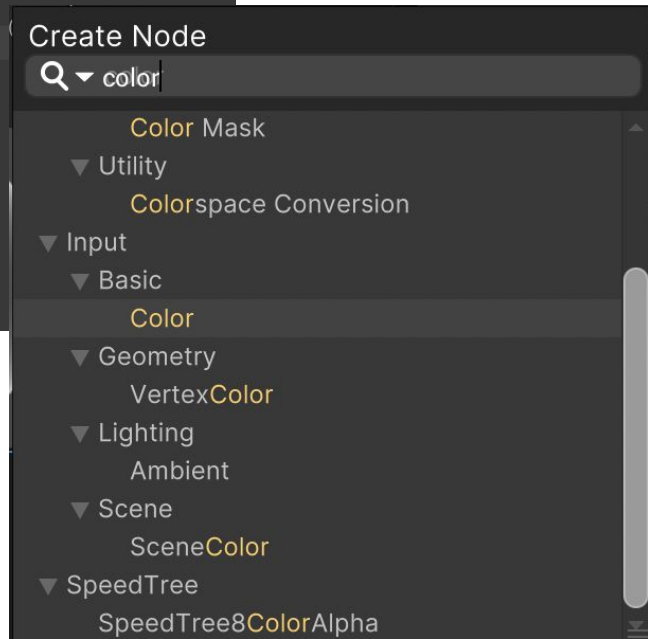
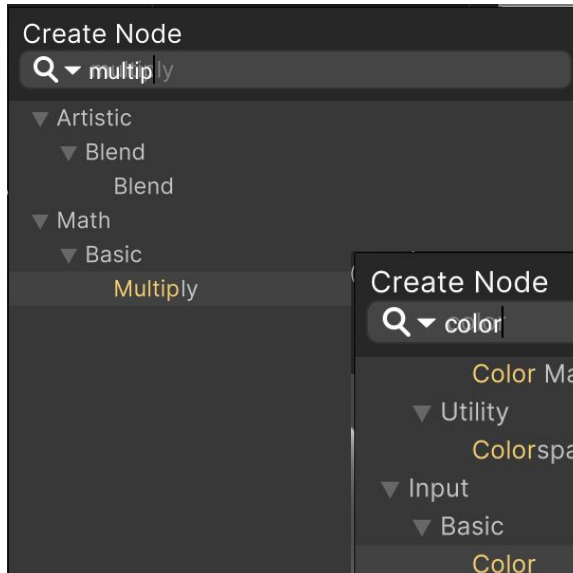
邊緣光特效

- 改變 Power 的值來調整特效強度



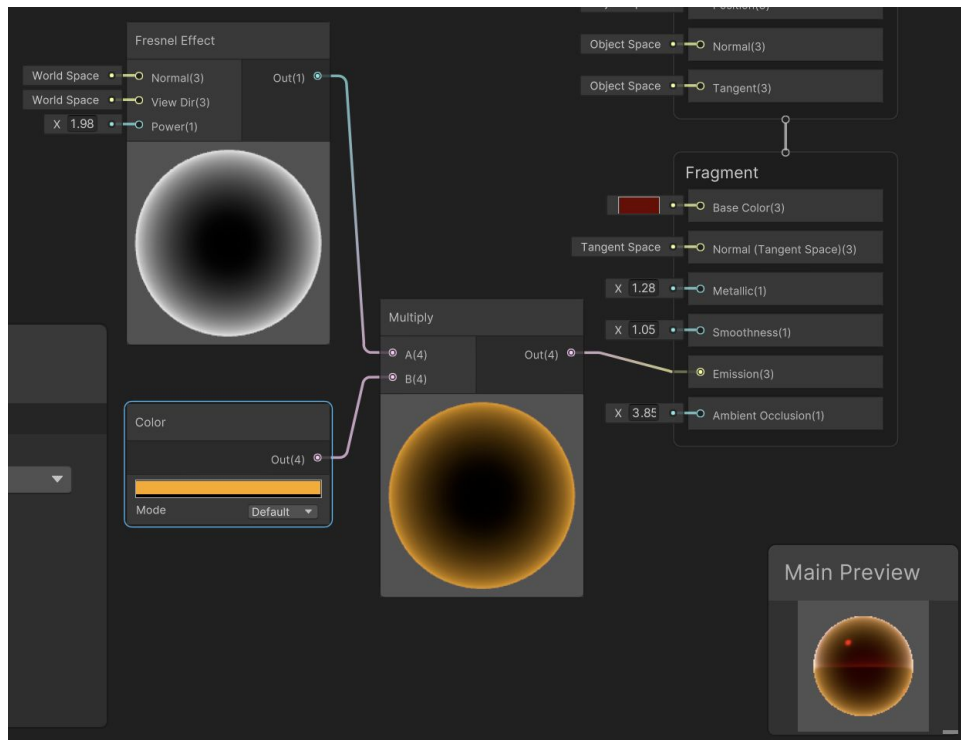
邊緣光特效

- 新增 Color 和 Multiply 節點
 - Multiply 能將向量或是變數相乘



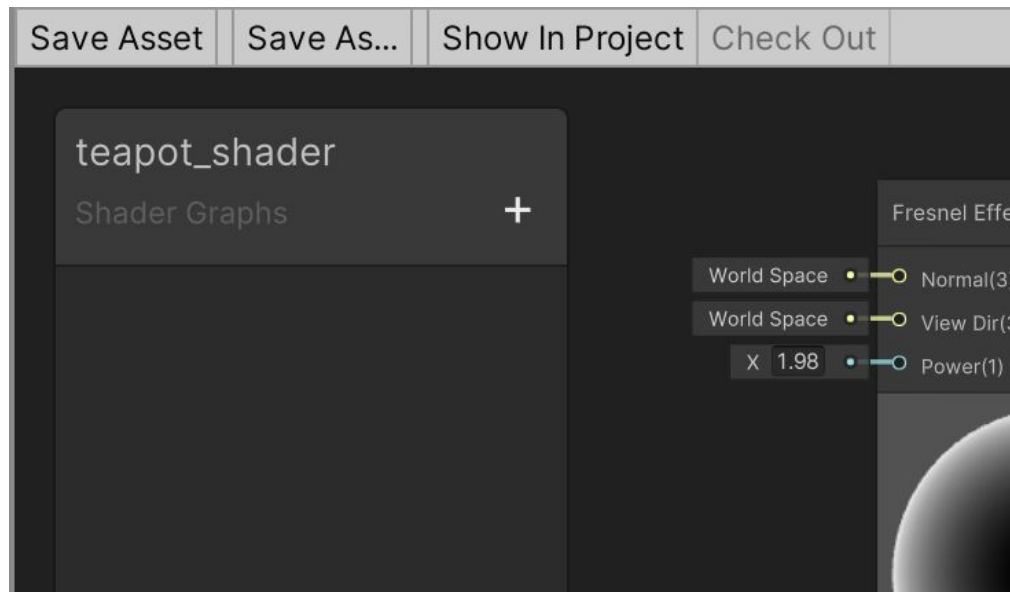
連接節點

- 將 Color 和 Fresnel Effect 結合
 - 將特效加上顏色



儲存檔案

- 點選左上角 Save Asset 儲存



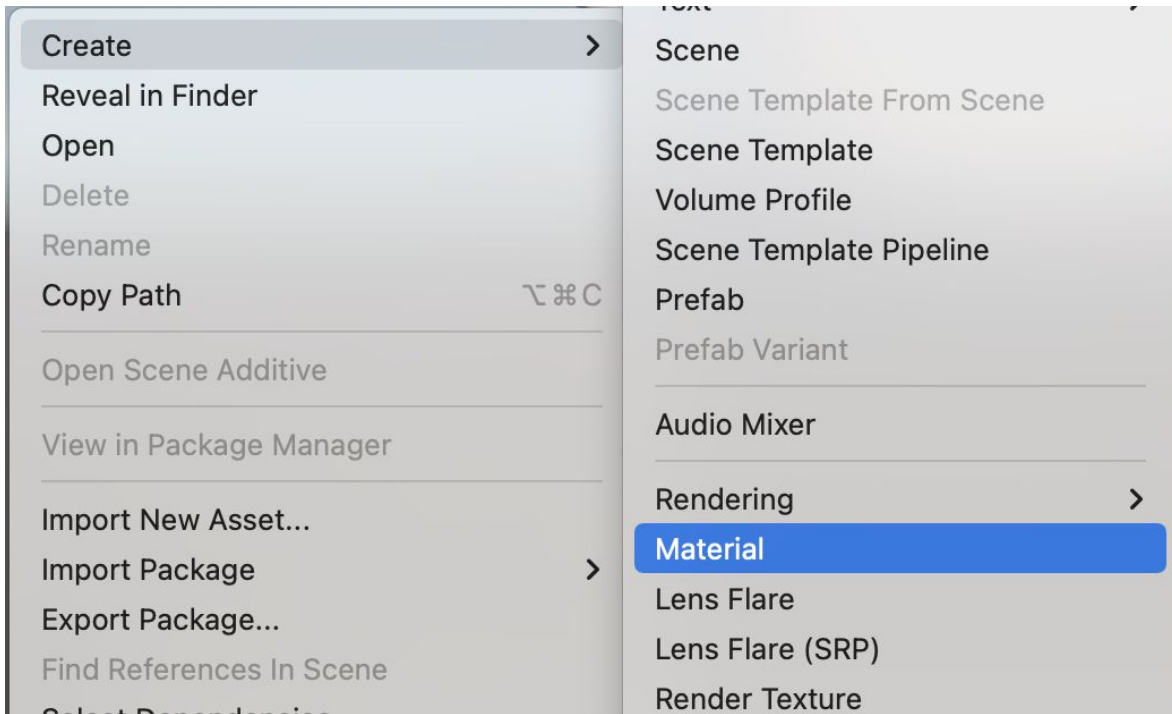
套用 Shader - 下載模型檔案



下載模型檔案
並加入 Unity

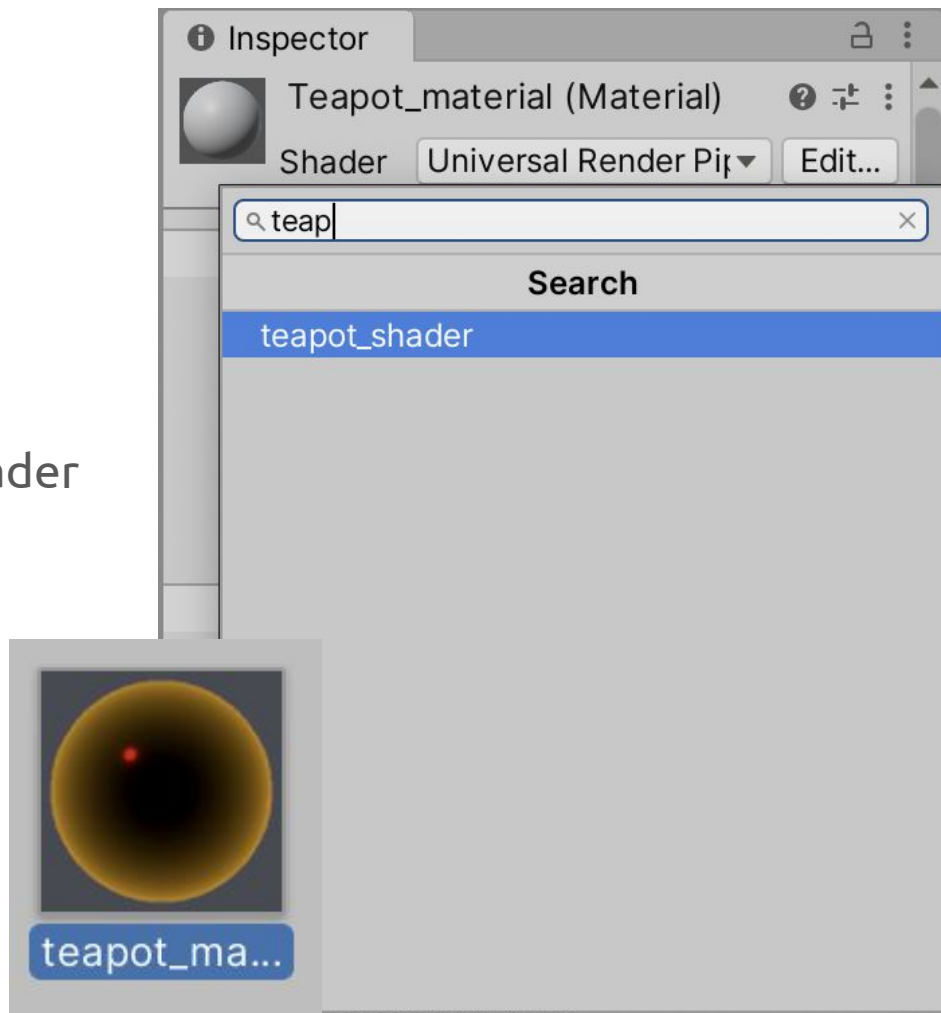
套用 Shader

- 新增 Material



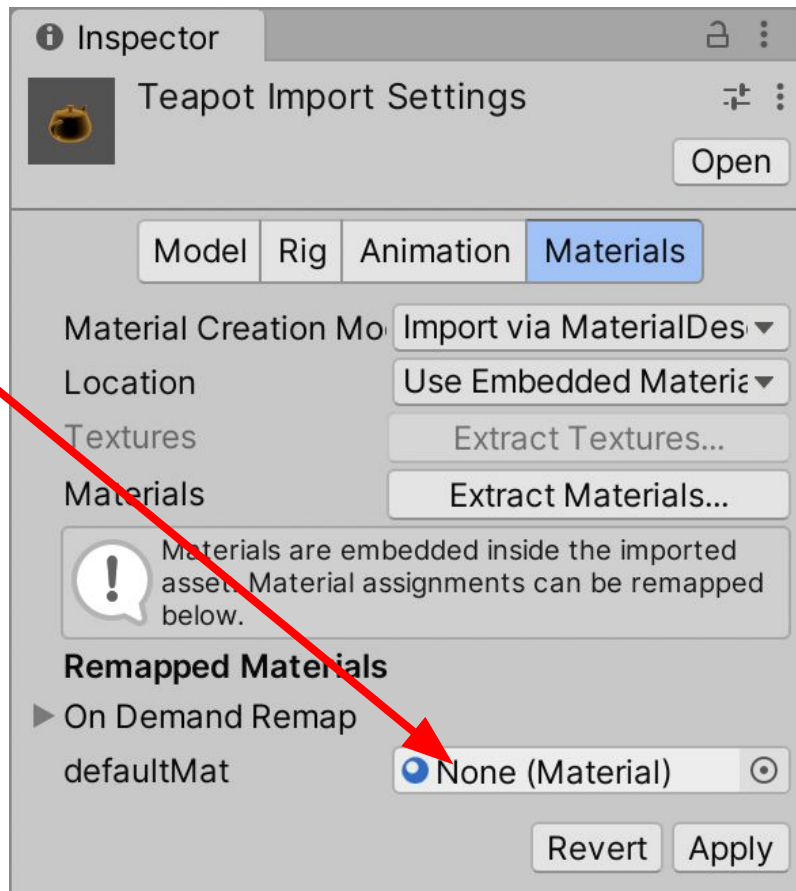
套用 Shader

- 在 Material 的 Inspector 中更換 shader
 - 選擇剛才建立的 shader graph
- 確認 Material 的圖示變更

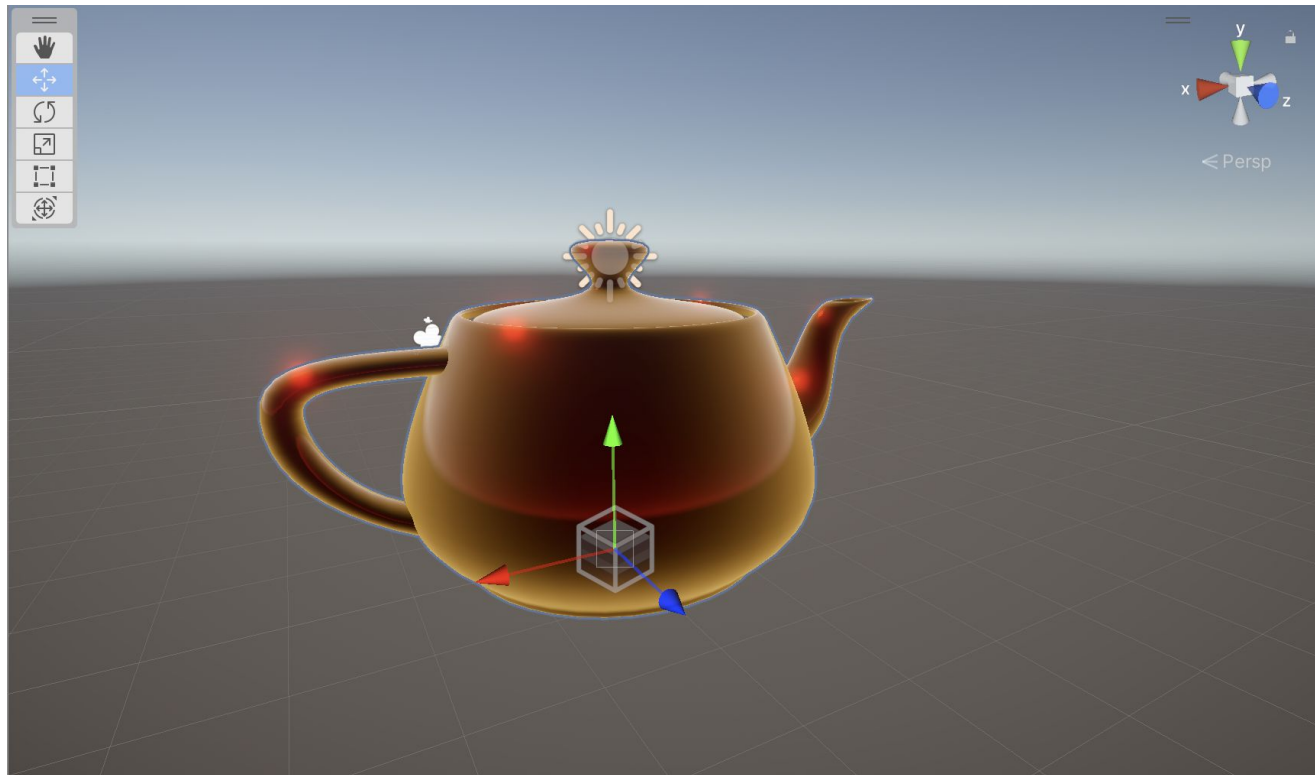


更換 Shader

- 選擇剛才加入的模型
- 進入 Materials tag
- 將剛才建立的 Material 拉到 defaultMat
- 點選 Apply

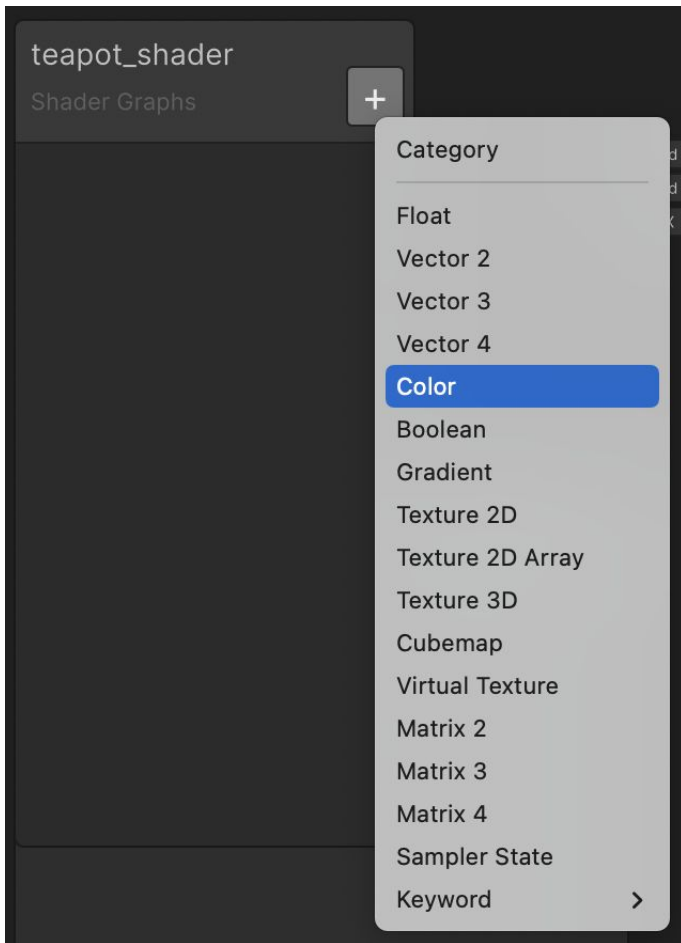


套用成功



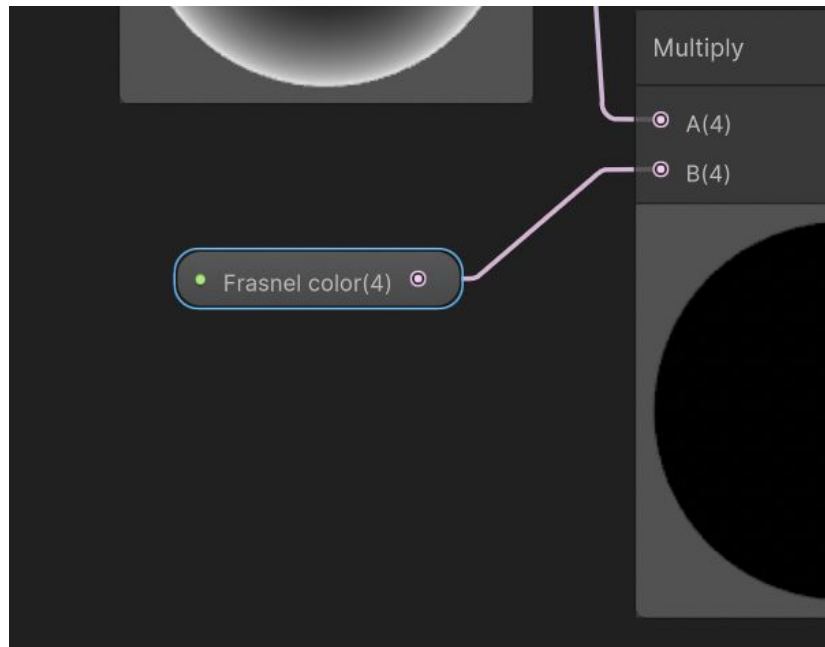
建立外部參數

- 可以從外部修改值的參數



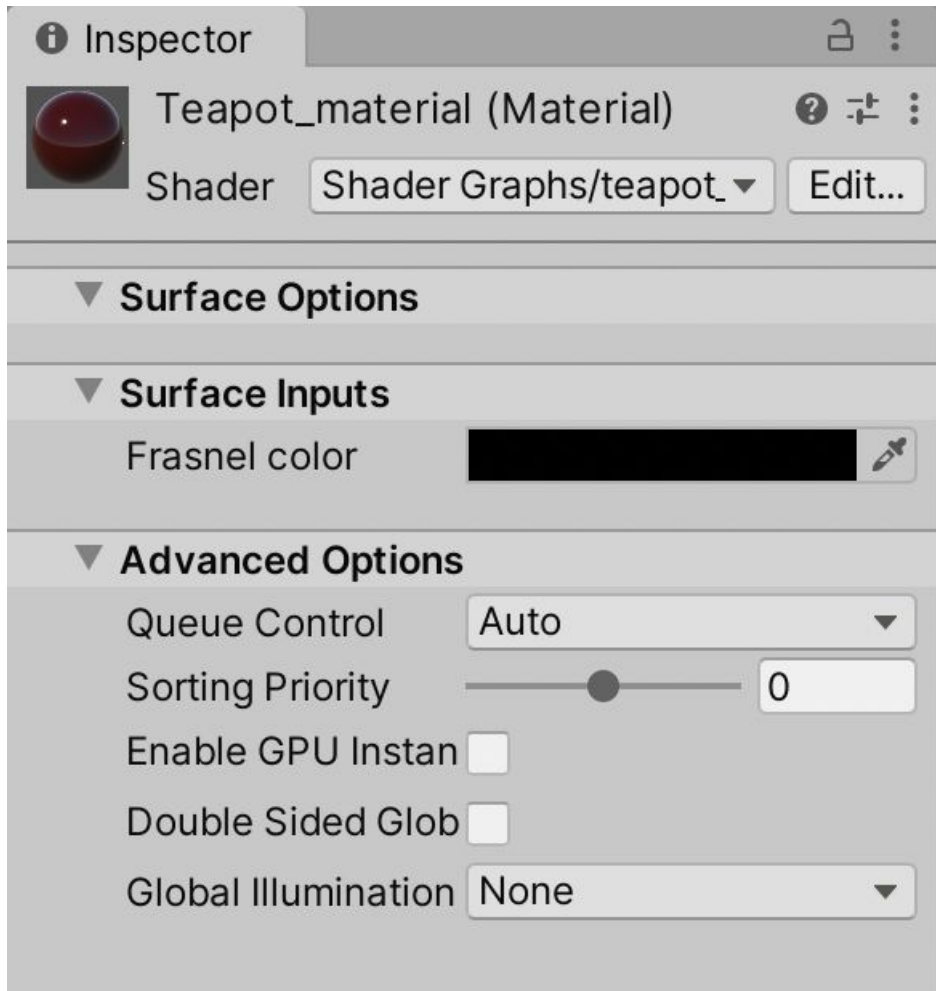
建立外部參數

- 可以拉到 Shader Graph 中當成 Node



建立外部參數

- 可以在 Inspector 看到外部參數



進階篇 — Diffuse and Specular

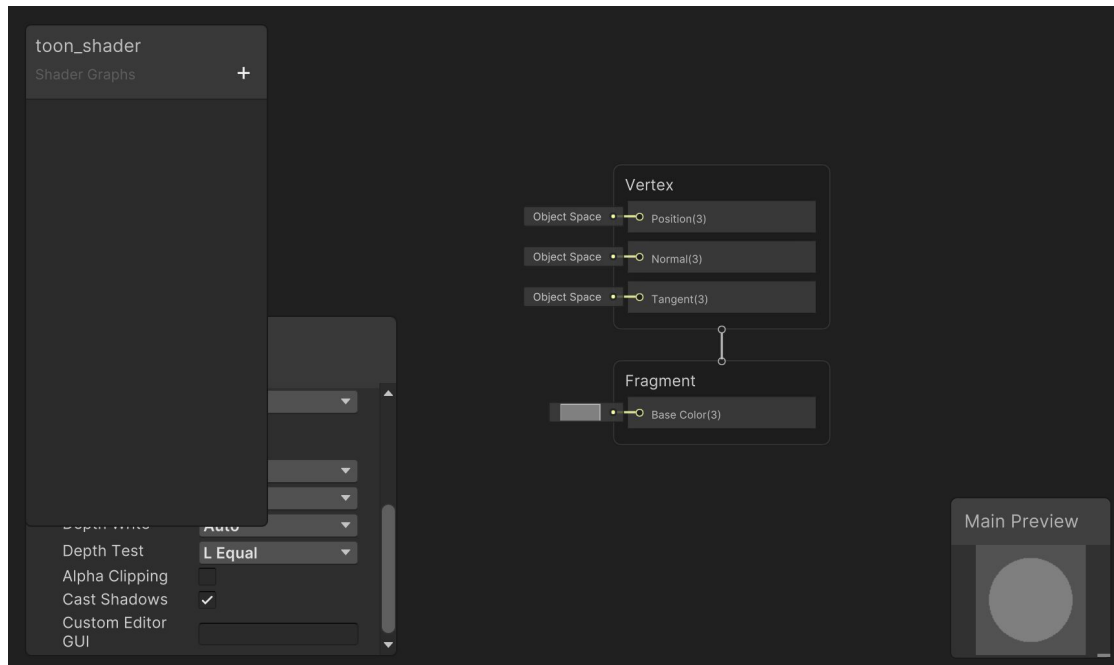
建立 Shader Graph

- 這次建立 Shader Graph -> URP -> Unlit Shader Graph



建立 Shader Graph

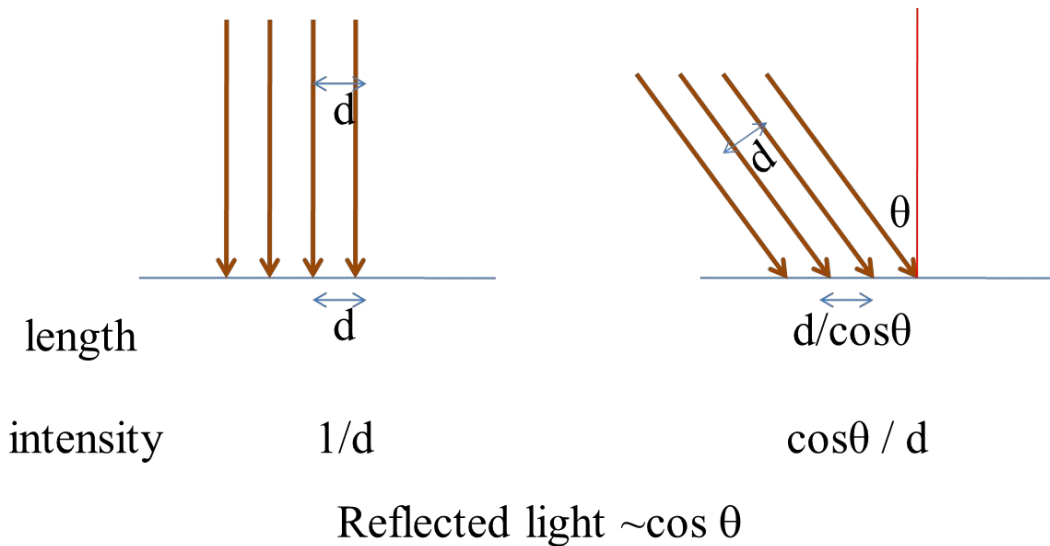
- Fragment 只有 Base Color



Shading 基本概念 - diffuse

- Diffuse

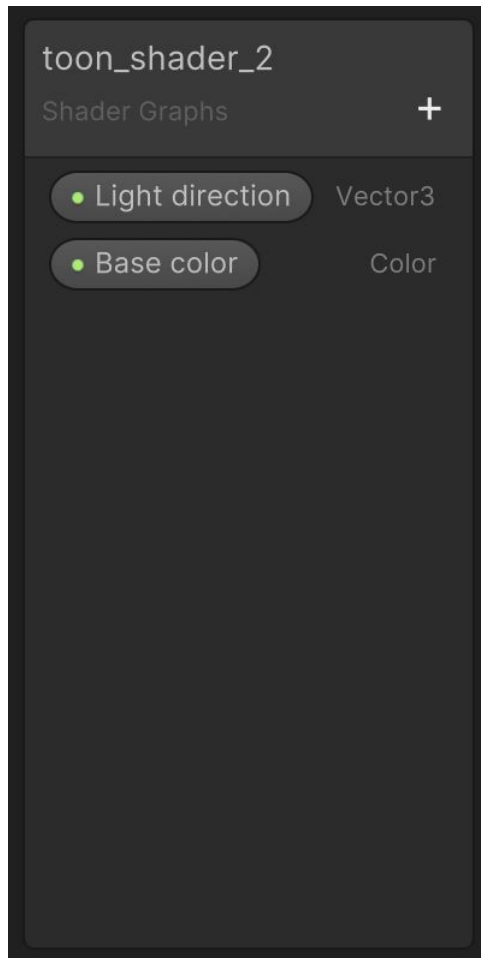
- N: 法向量
- L: 光源向量
- L 需要 normalize



$$\mathbf{N} \cdot \mathbf{L}$$

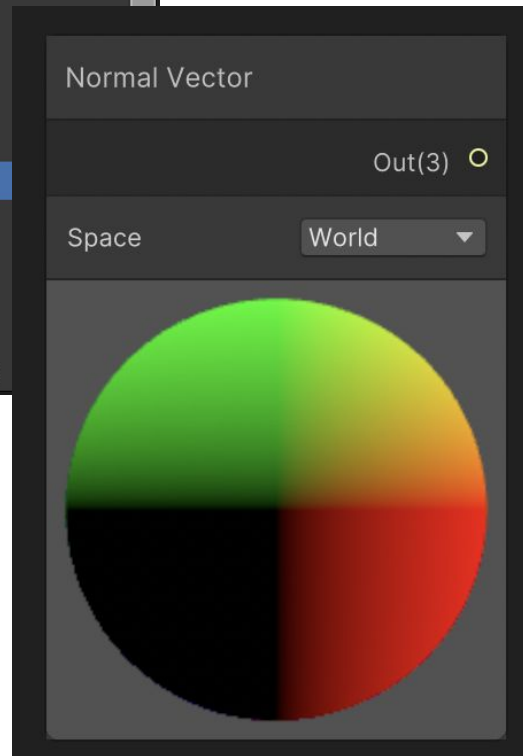
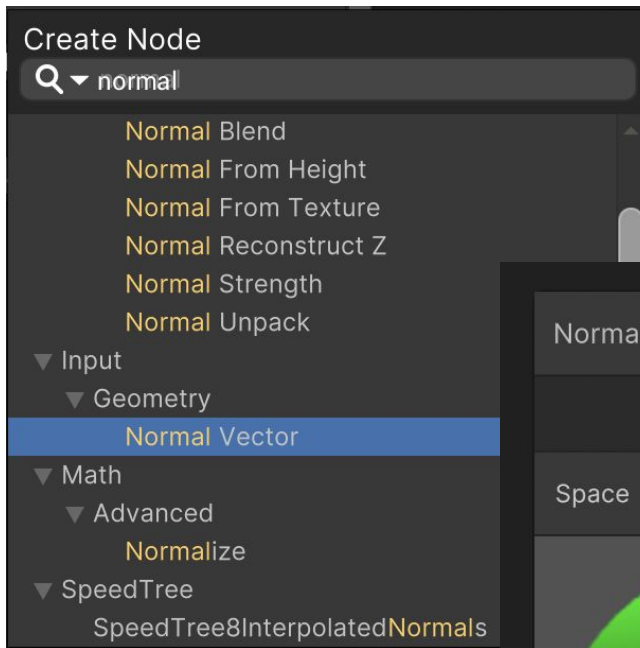
Diffuse

- Shader Graph 無法取得光源方向
 - 使用外部參數定義
- 順便定義 Base Color



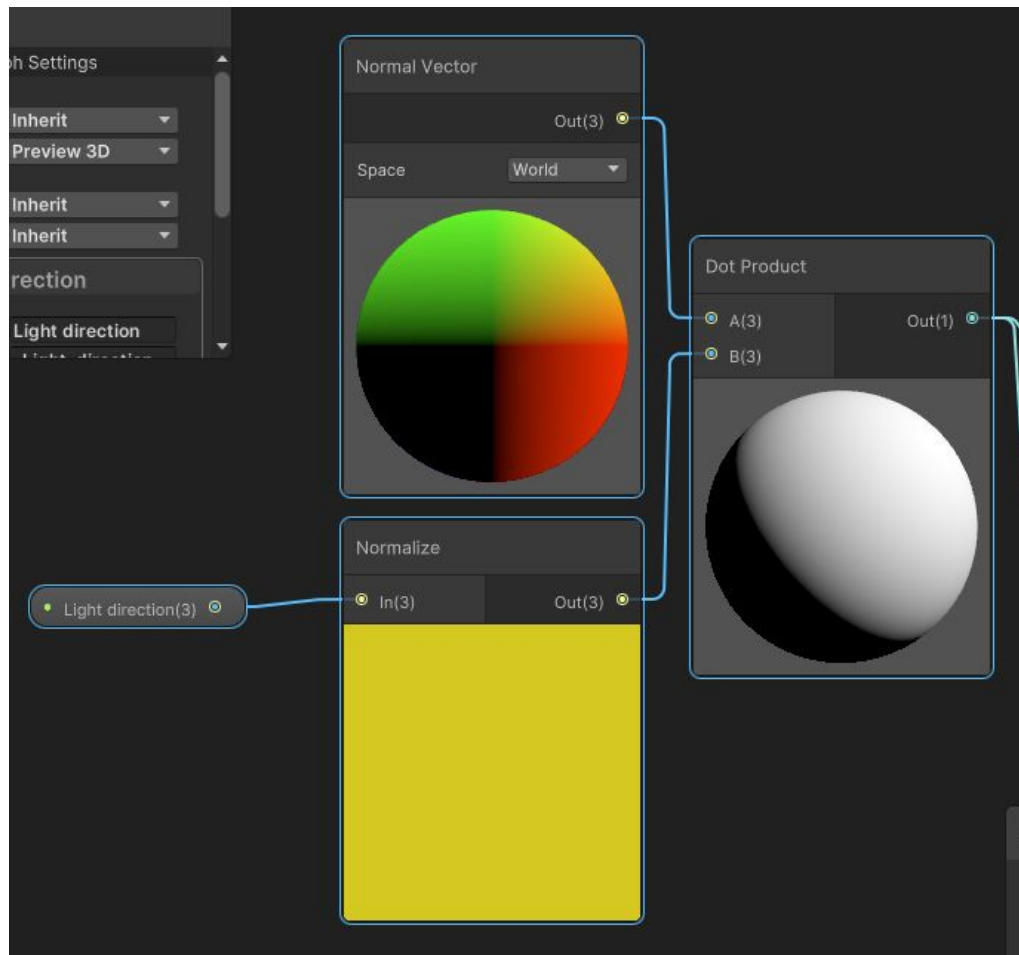
Diffuse

- 建立 Normal node
 - 取得法向量 N



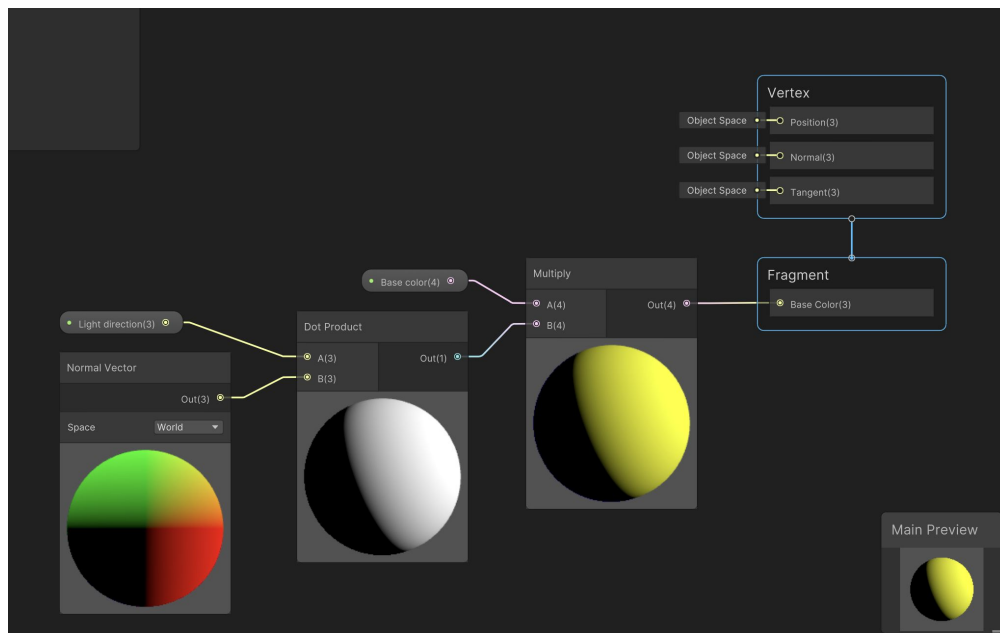
Diffuse

- 結合 Dot Product
 - 調整光源方向



Diffuse

- 和底色相乘並連接到 Base Color



Shading 基本概念 - Specular

- Specular (Phong Shading)

- v : View Vector

- 物體相對於觀察者的向量

- r : 反射向量

- α : 光滑度

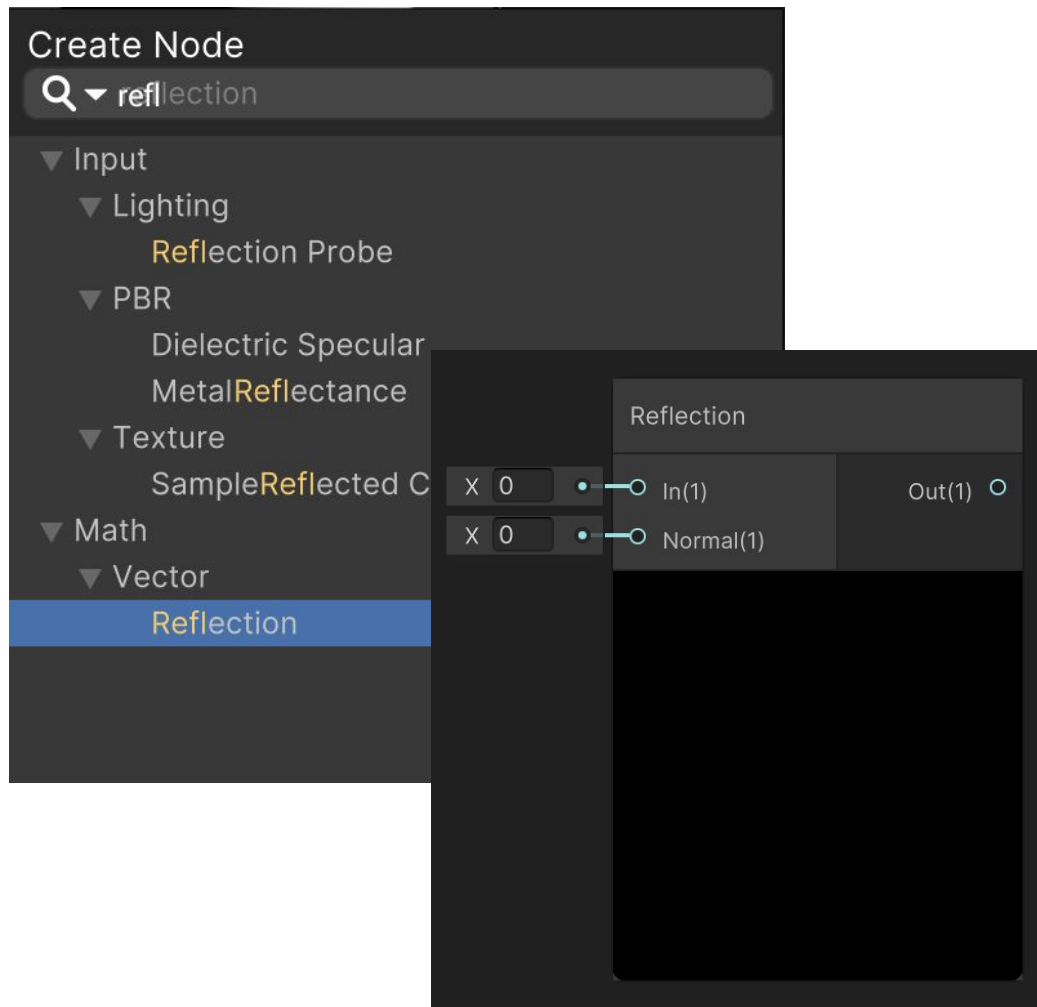
- v, r 需 normalize

$$(v \cdot r)^\alpha$$

$$r = \text{reflect}(-L, N)$$

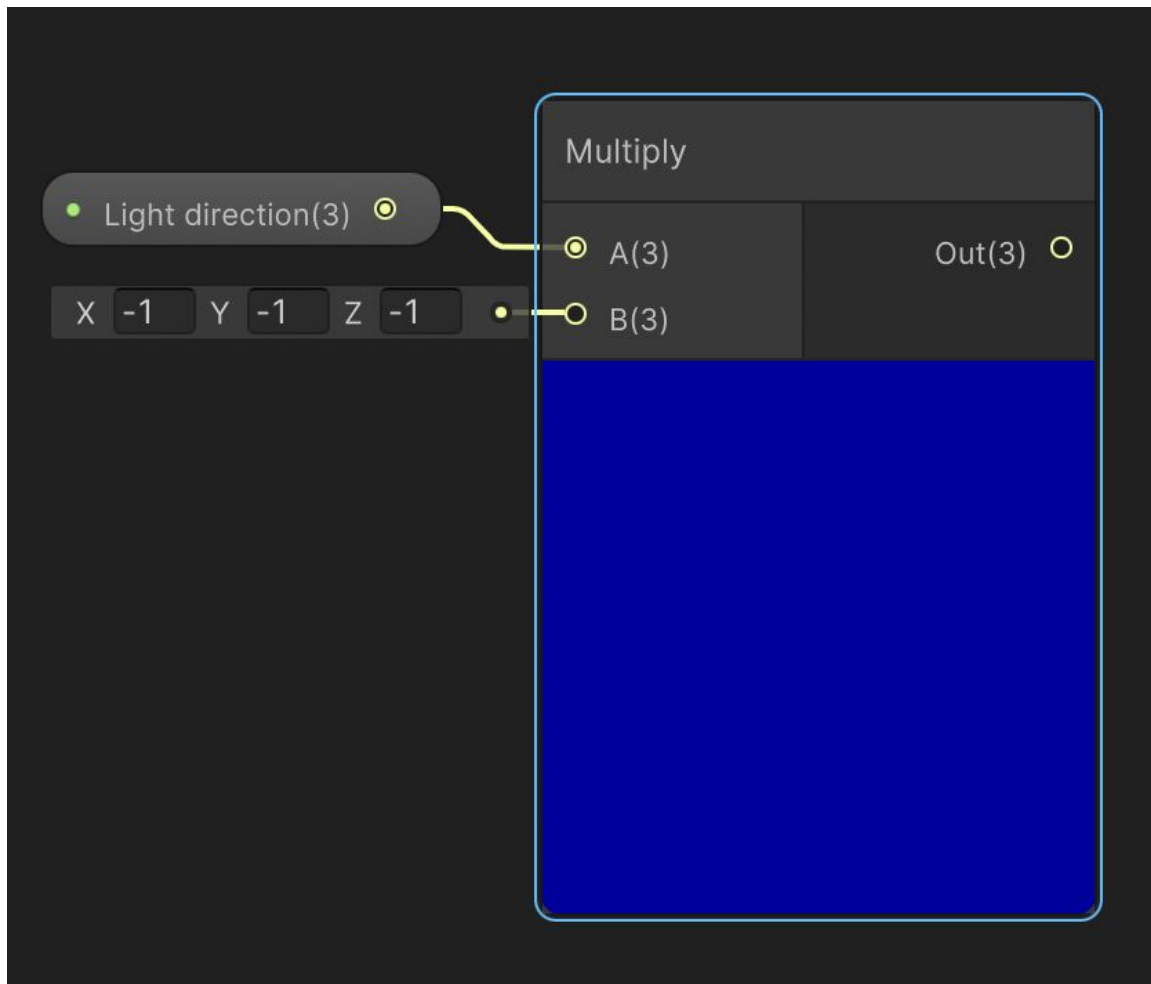
反射向量

- 新增 Reflection 節點



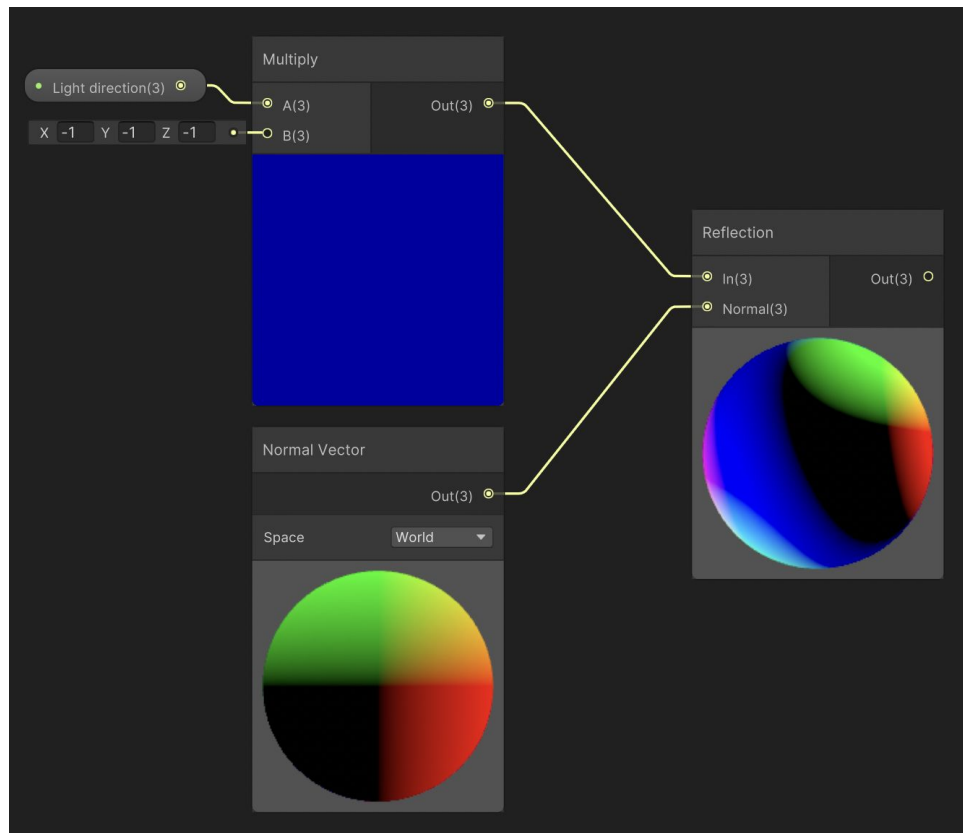
反射向量

- 計算 $-L$



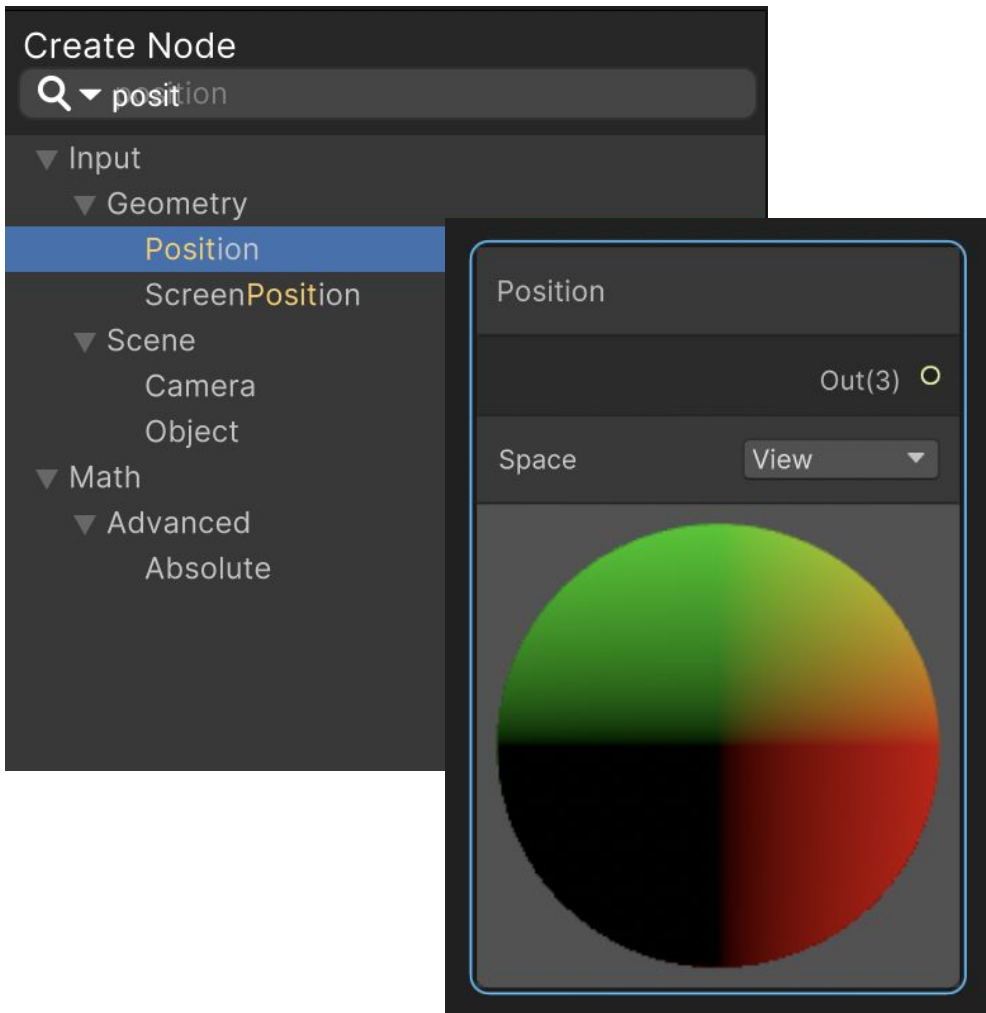
反射向量

- 計算出 v



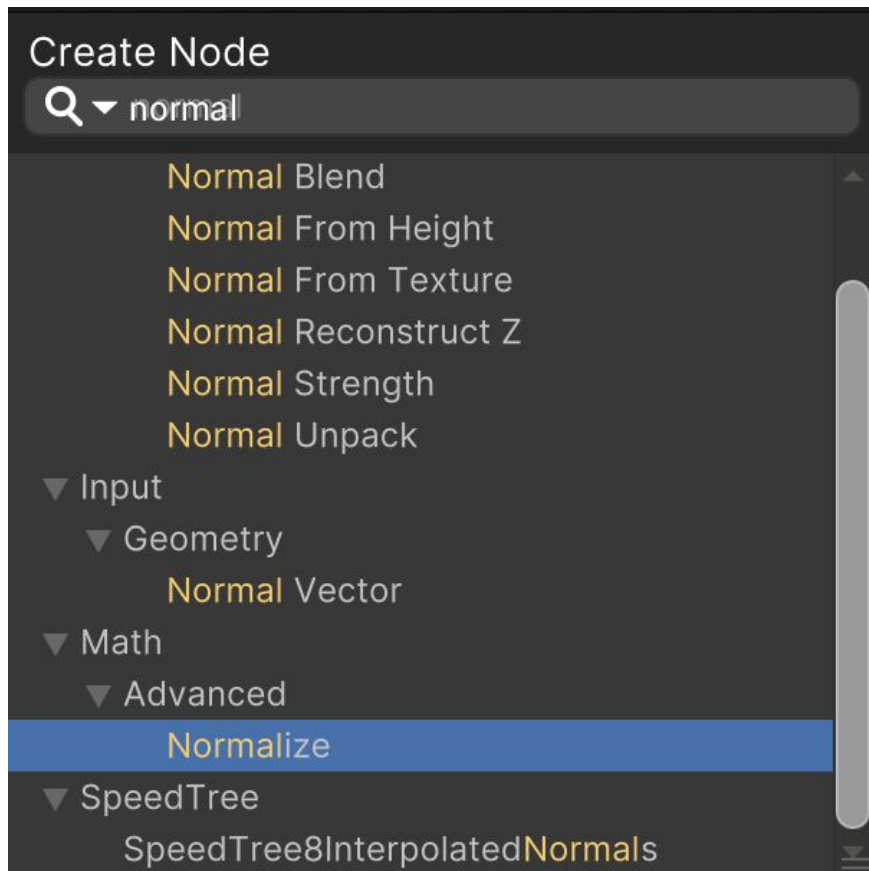
View Vector

- 建立 Position 節點
 - Space 選擇 View



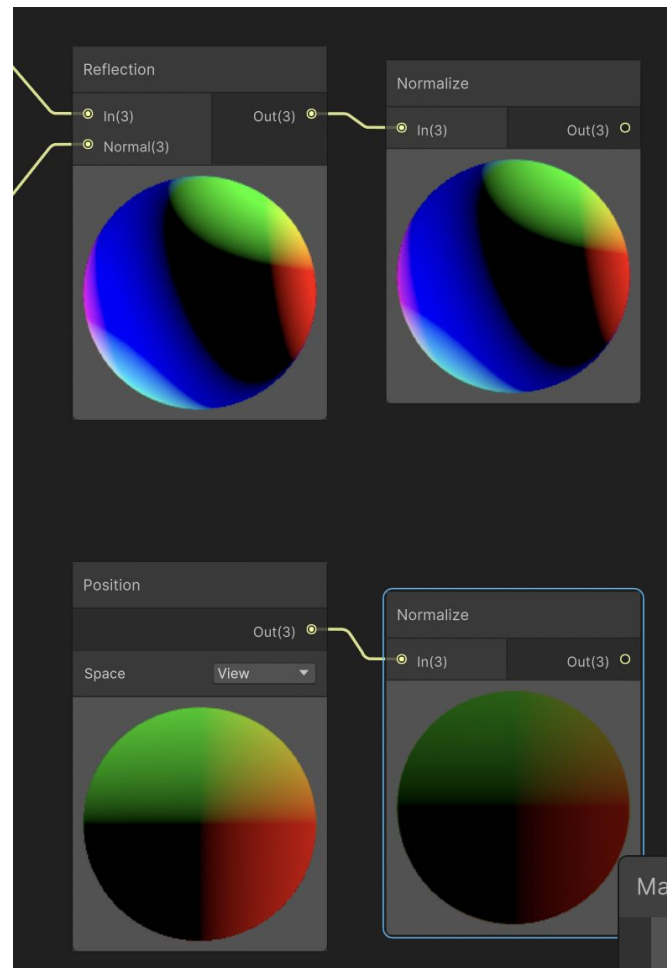
Normalize

- 新增 normalize 節點



Normalize

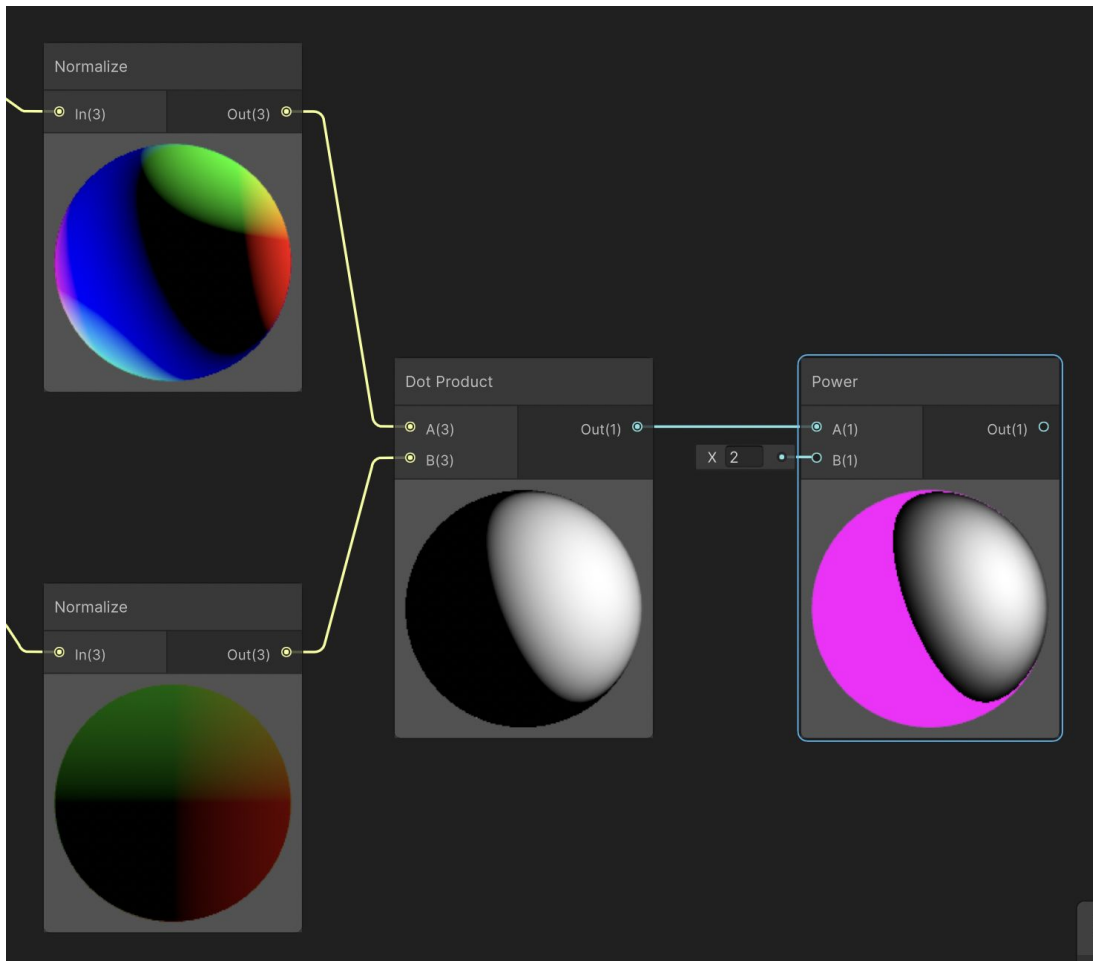
- 分別將 v , r normalize



Specular

- 連接 dot 和 power 節點
 - power 計算次方

$$(v \cdot r)^\alpha$$



Specular

- 新增光滑度外部參數
 - 調整光滑度

toon_shader_2

Shader Graphs



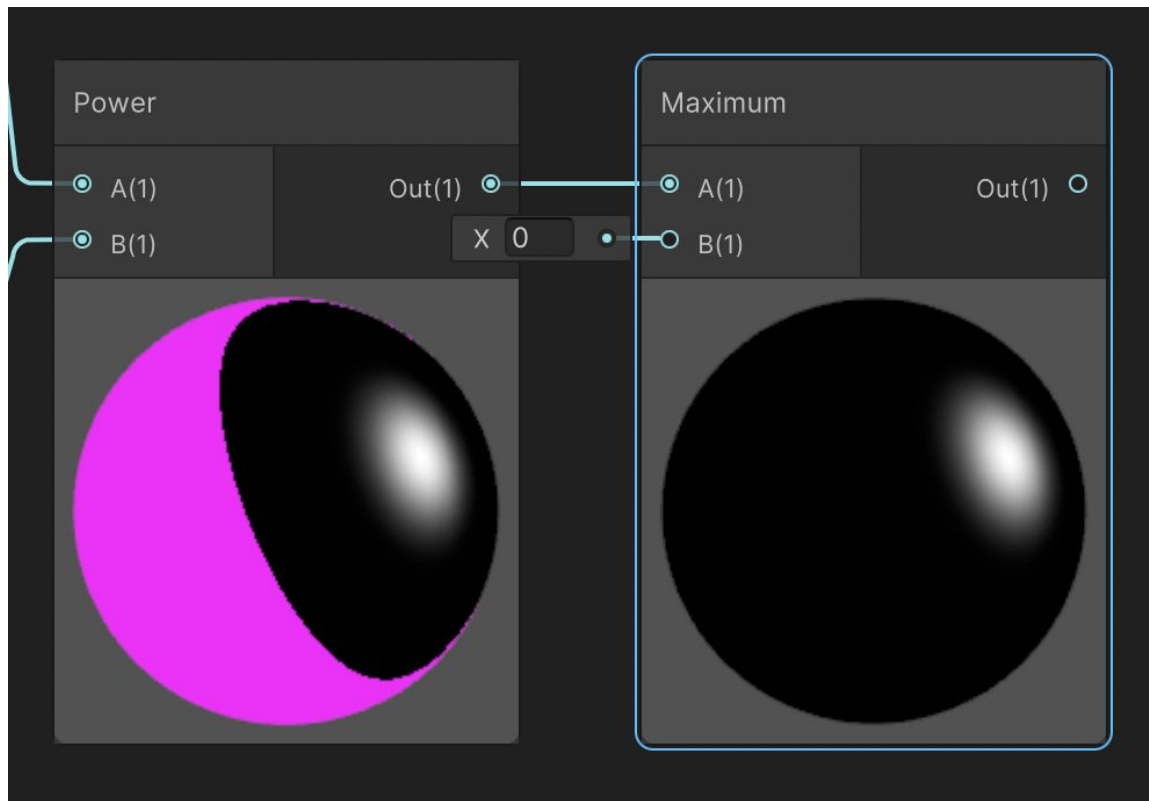
● Light direction Vector3

● Base color Color

● Smoothness Float

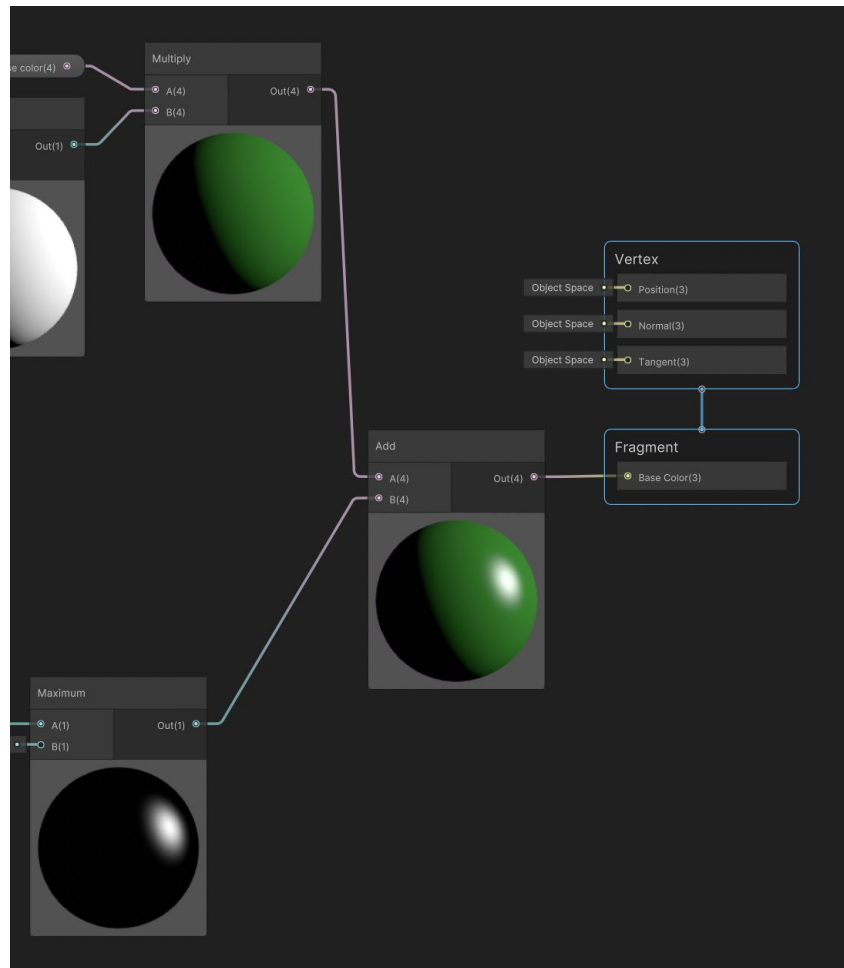
Specular

- 新增 Maximum 節點
 - 將小於 0 的部分設成 0



結合 diffuse 與 specular

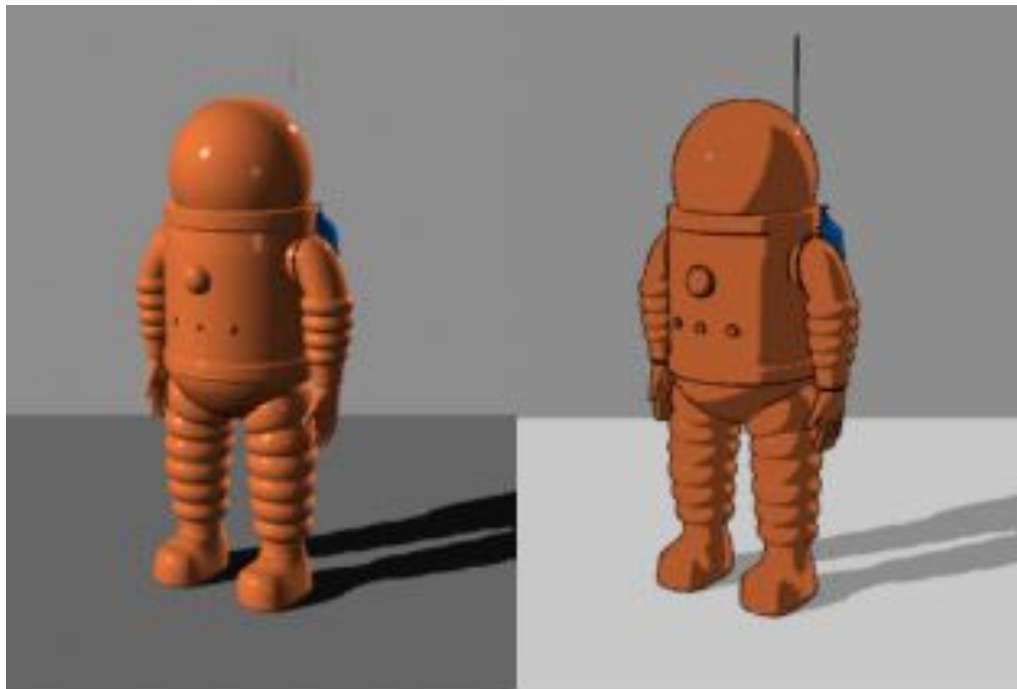
- 使用 Add 節點
 - 將兩個值相加



進階篇 — Toon Shading

Toon Shading

- 設定閾值
 - 在兩個閾值之間亮度相等



plastic shader

toon shader

設定閾值

- 使用 Comparison 及 Branch 節點
 - 比較並設定值

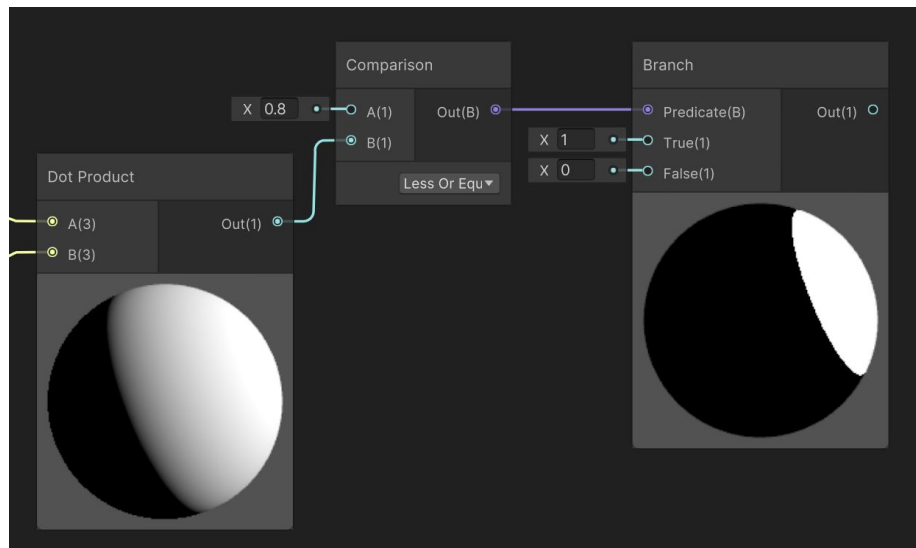
設定閾值

- Comparison

- 檢查 A(0.8) 是否小於 B(dot 的結果)

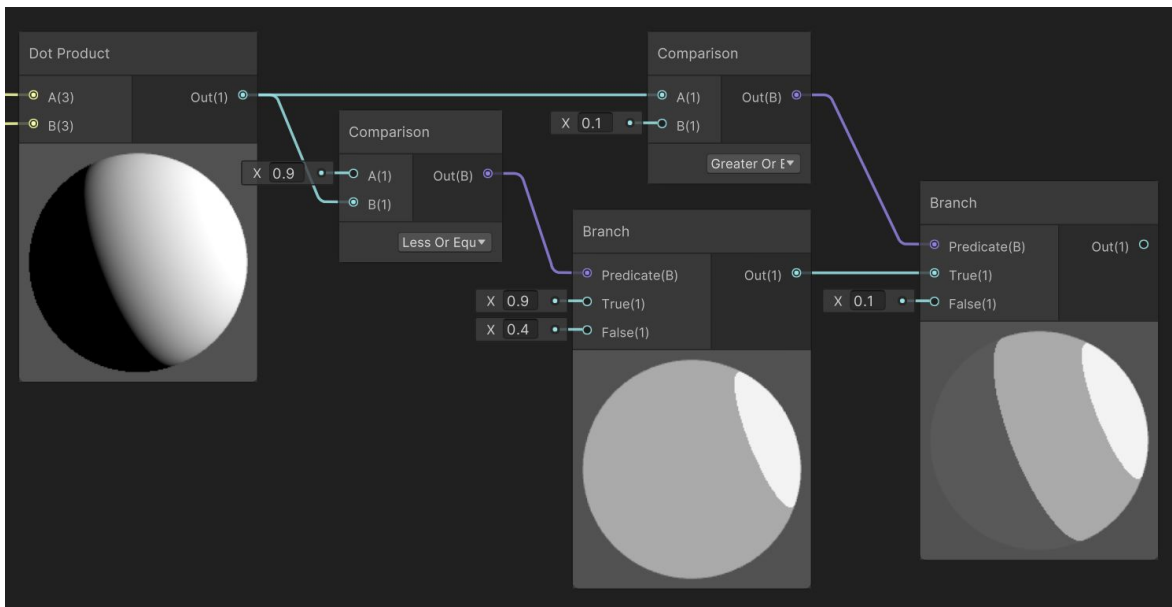
- Branch

- 接收比較結果
- True 輸出 1
- False 輸出 0



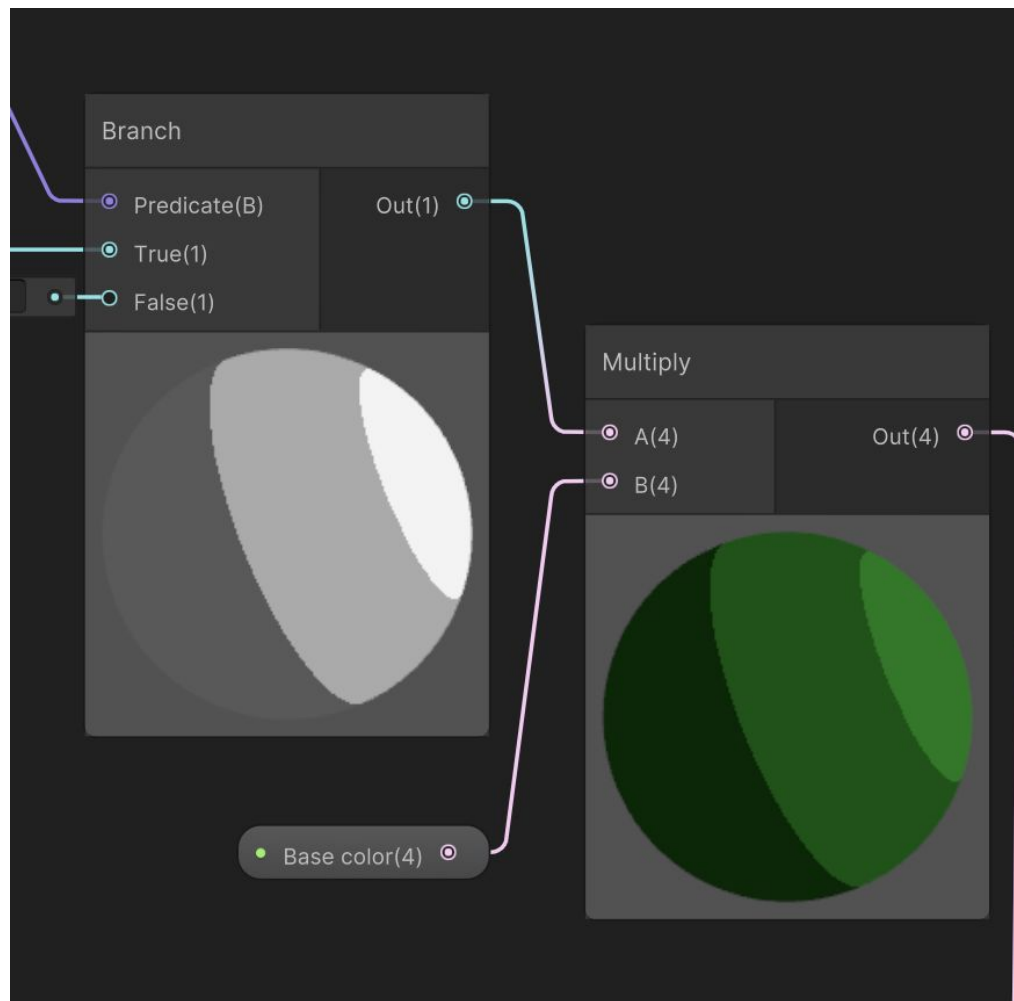
設定閾值

- 結合多個條件判斷
 - 調整閾值和輸出值



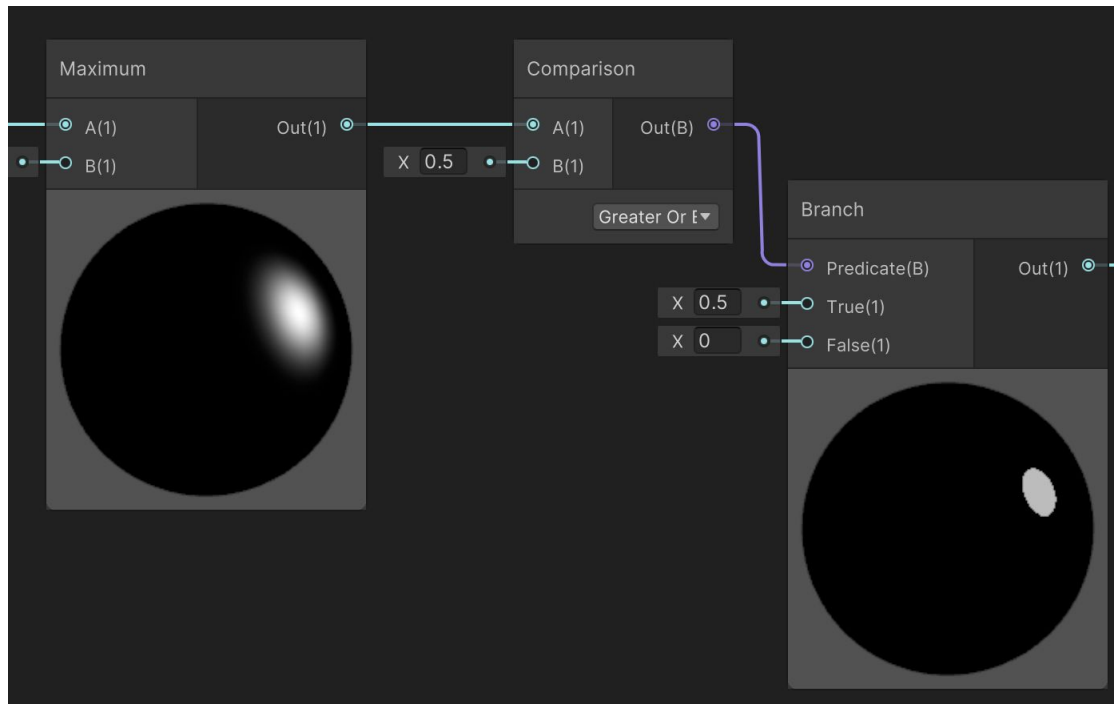
設定閾值

- 乘上底色



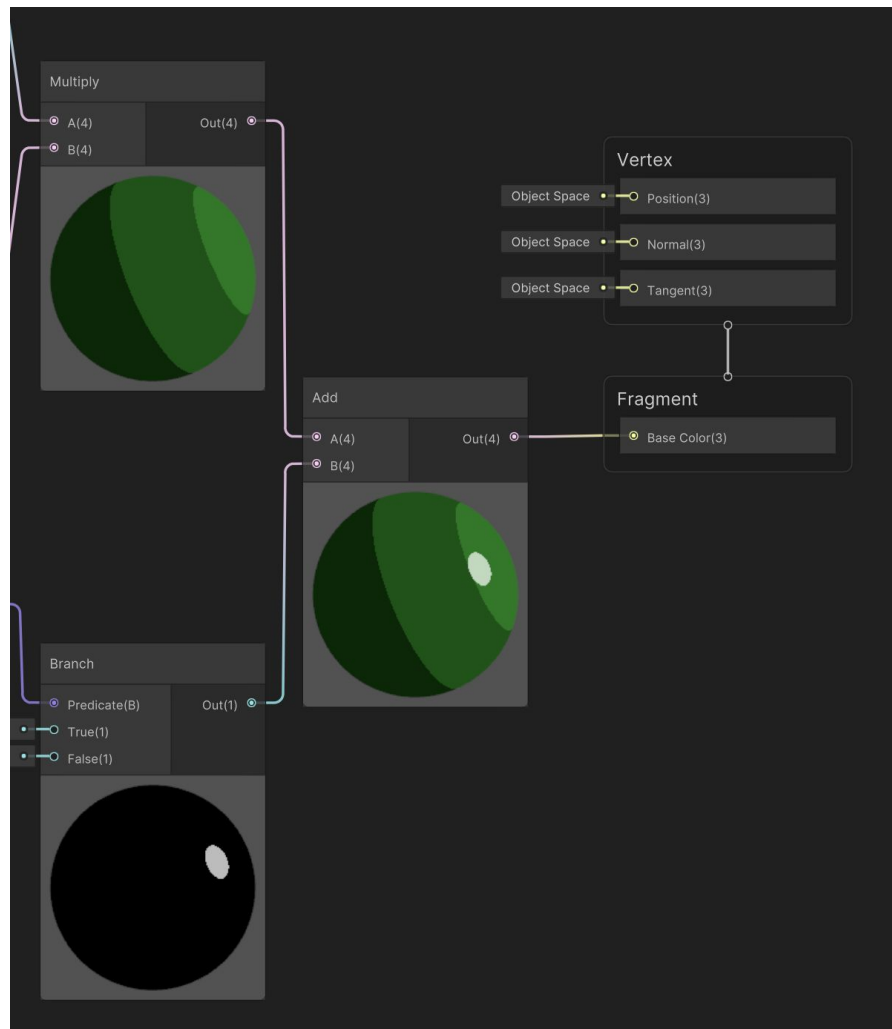
設定閾值

- Specular 也做相同處理



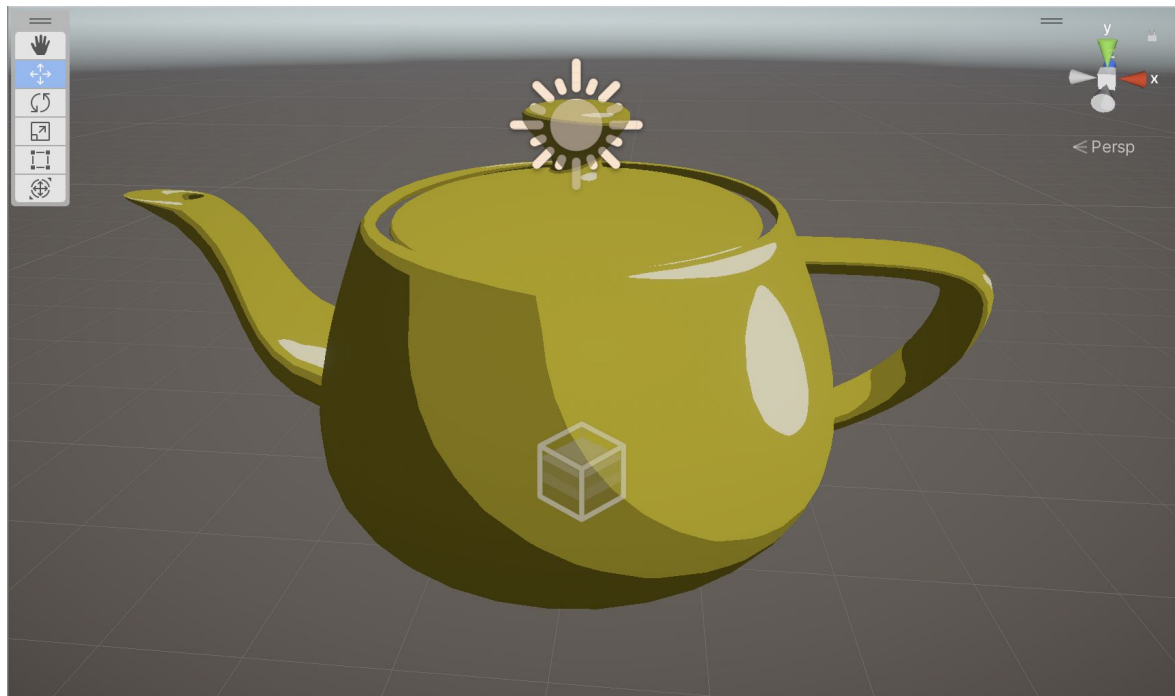
Toon Shader

- 完成 Toon Shader



套用 Shader

- 方法和前面相同



練習

- 繪製簡單的輪廓線
 - 提示: 利用 Normal 和 View Matrix



參考資料

- [Basics of Shader Graph - Unity Tutorial](#)