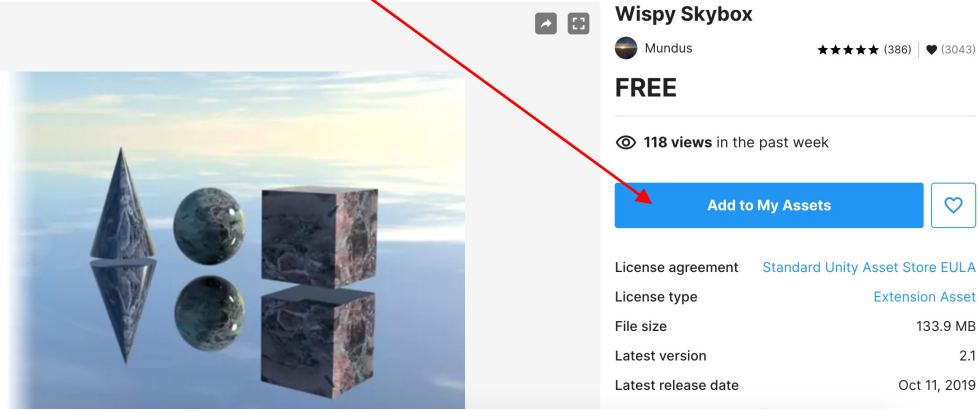
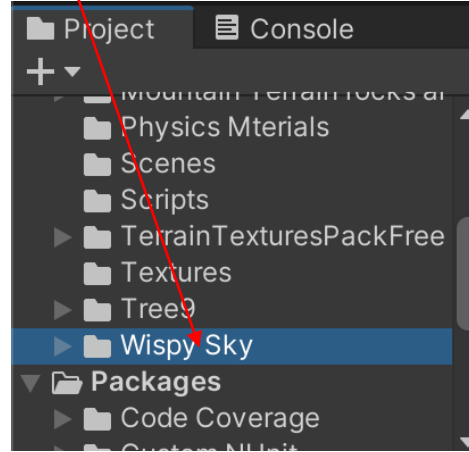


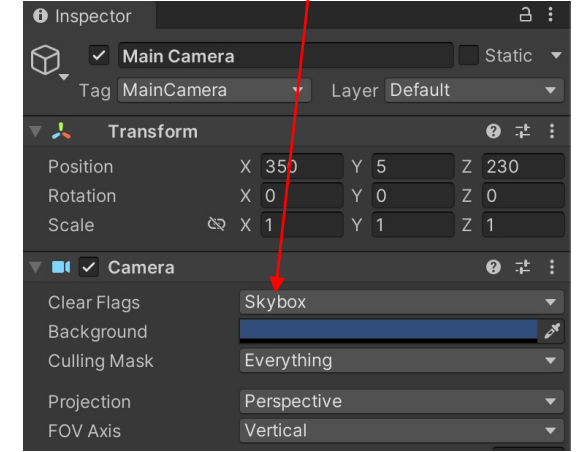
①Unity Assets StoreのサイトでWispy Skybox検索
→ Add to My Assetsをクリック



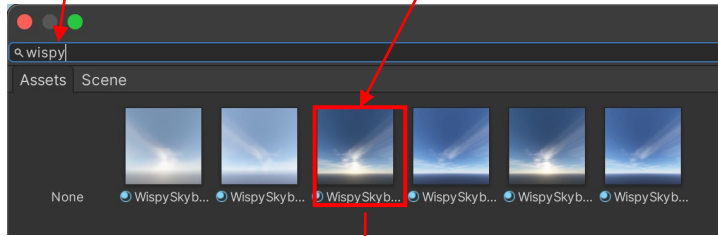
②Download、Import後Projectウィンドーに
Wispy Skyのフォルダを確認



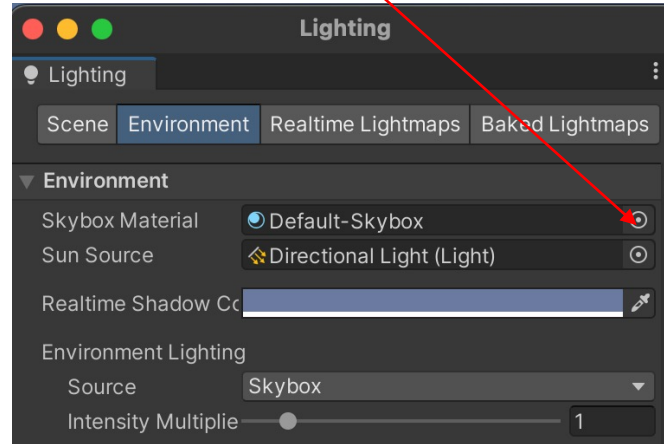
③Main Cameraのインスペクターの
Clear FlagsにSkybox確認



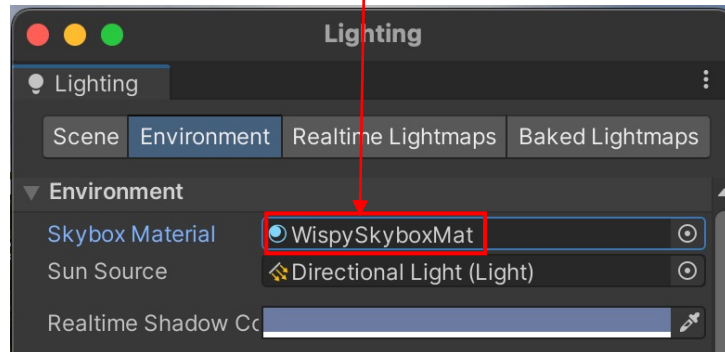
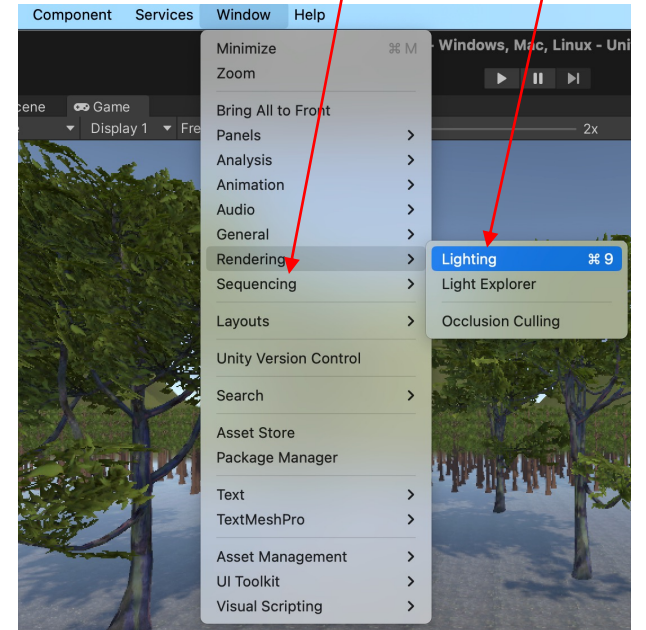
⑥「wispy」で検索→「WispySkyboxMat」を
クリック



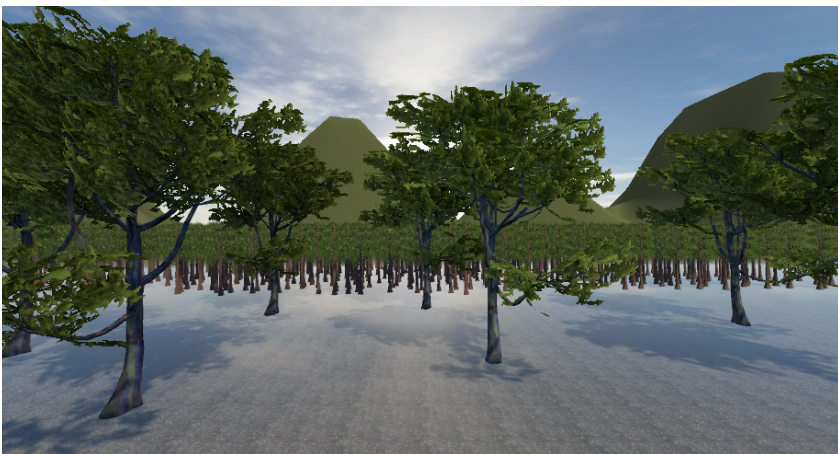
⑤Environmentのここをクリックして



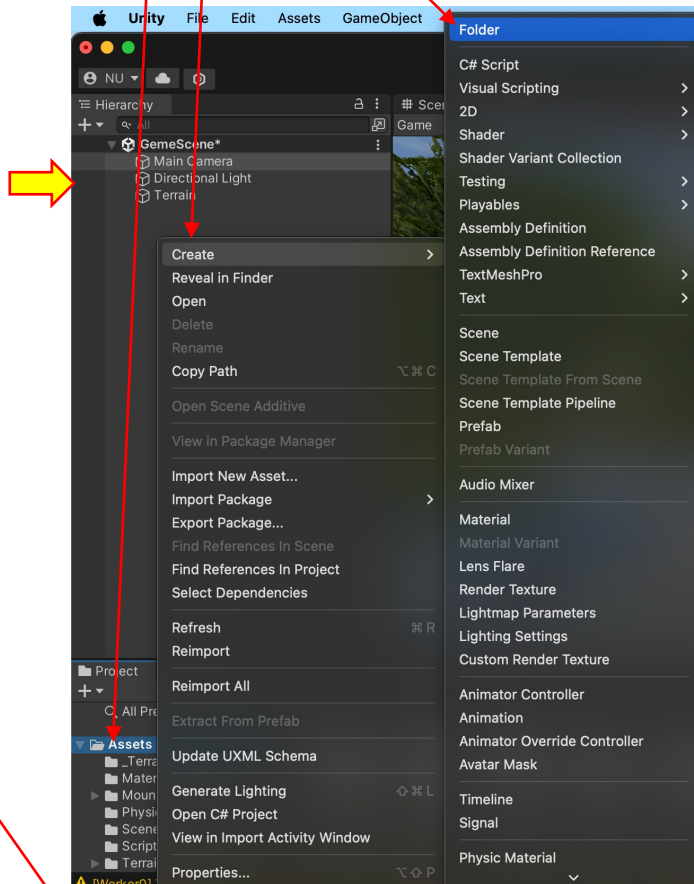
④Window→ Rendering→ Lighting
クリック



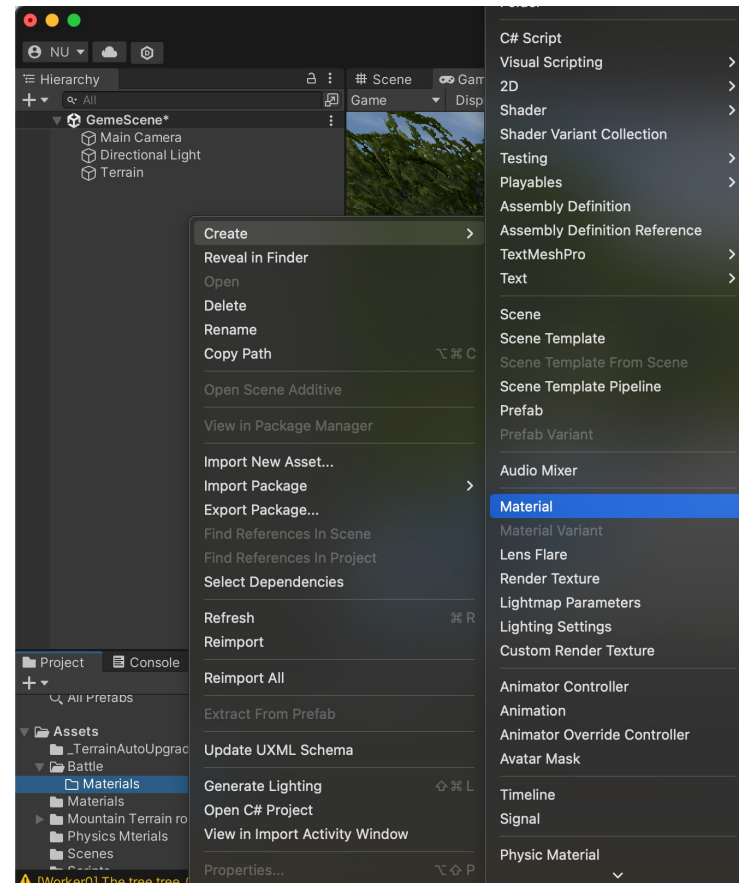
⑦空が画面に反映



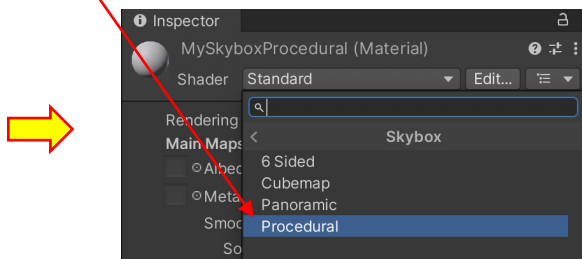
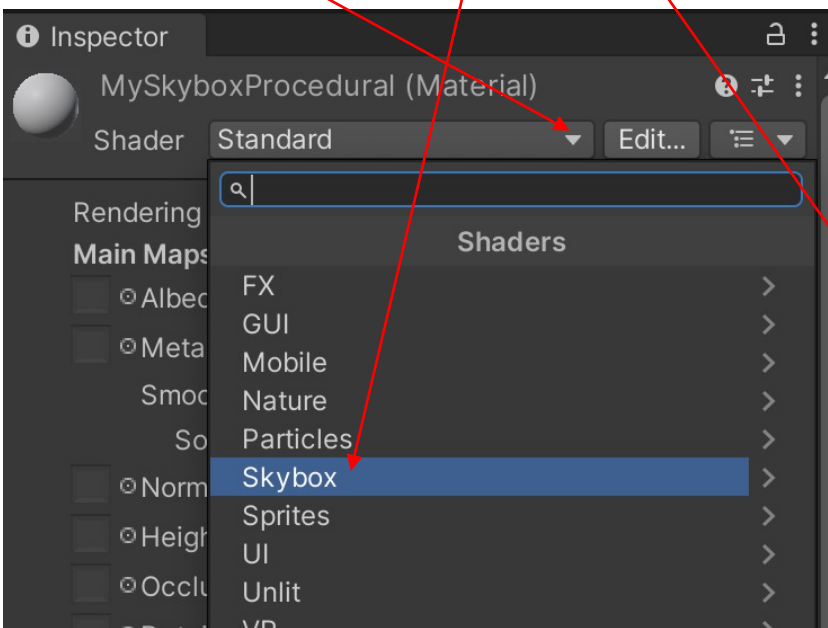
⑧Assetsフォルダ上で右クリック
→ Create → Folder → フォルダ名
をMaterial



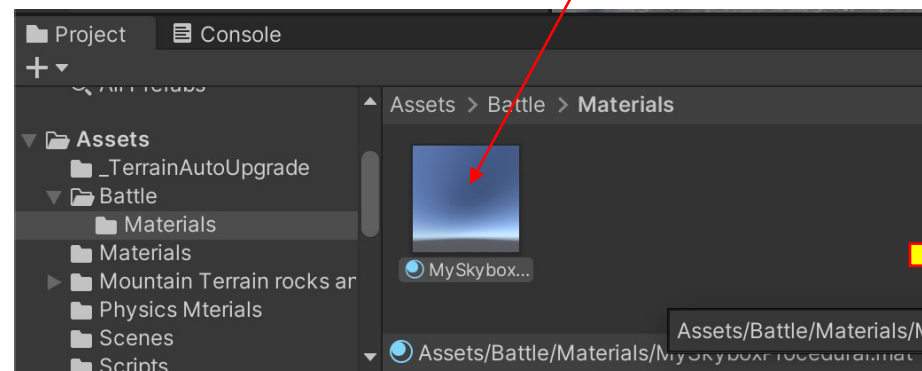
⑨ Material上で右クリック → Create → Material
フォルダ名を「MySkyboxProcedural」とする



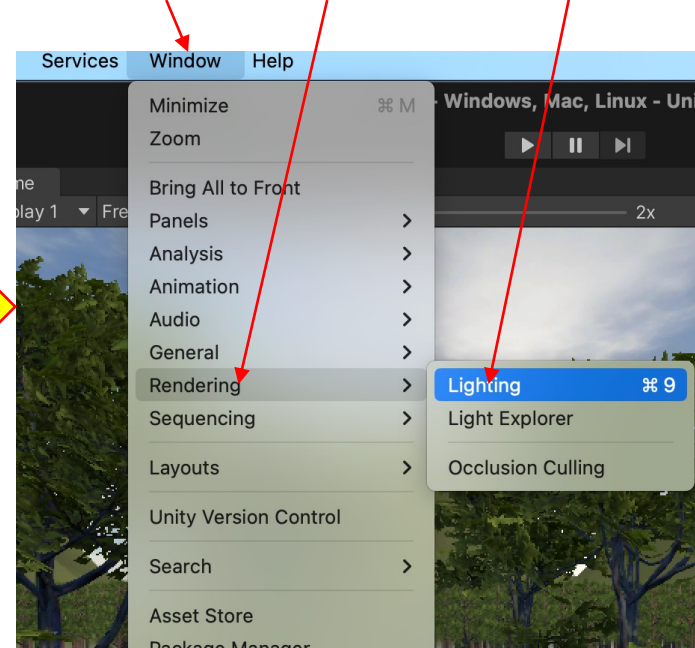
⑩ Materialを選択してインスペクターの
ShaderのプルダウンSkybox → Procedural選択



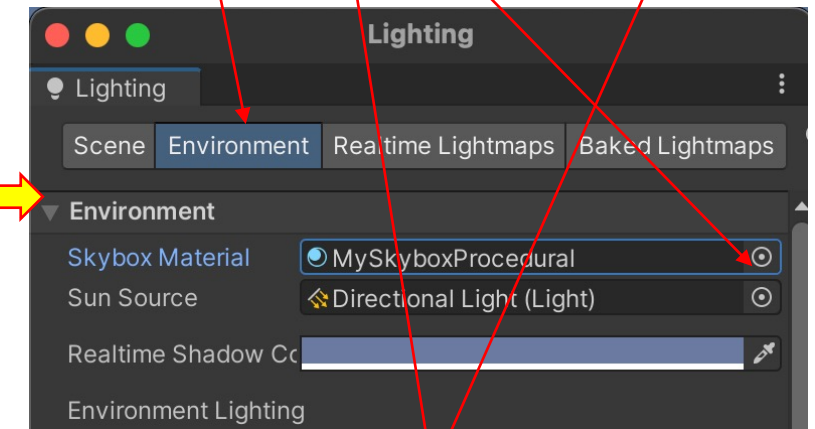
⑪ Materialフォルダ内にMySkyboxProcedural
ができています



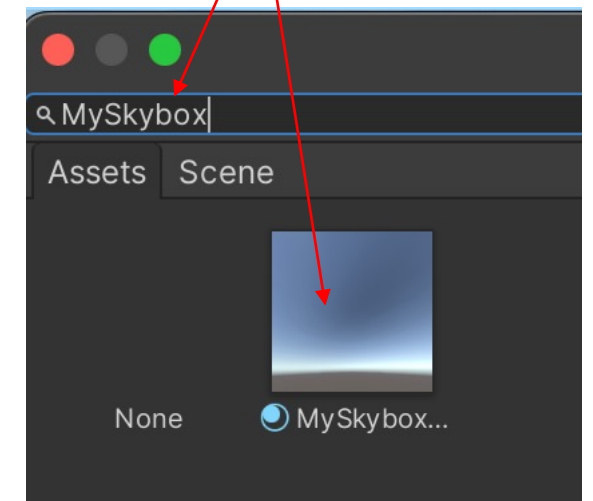
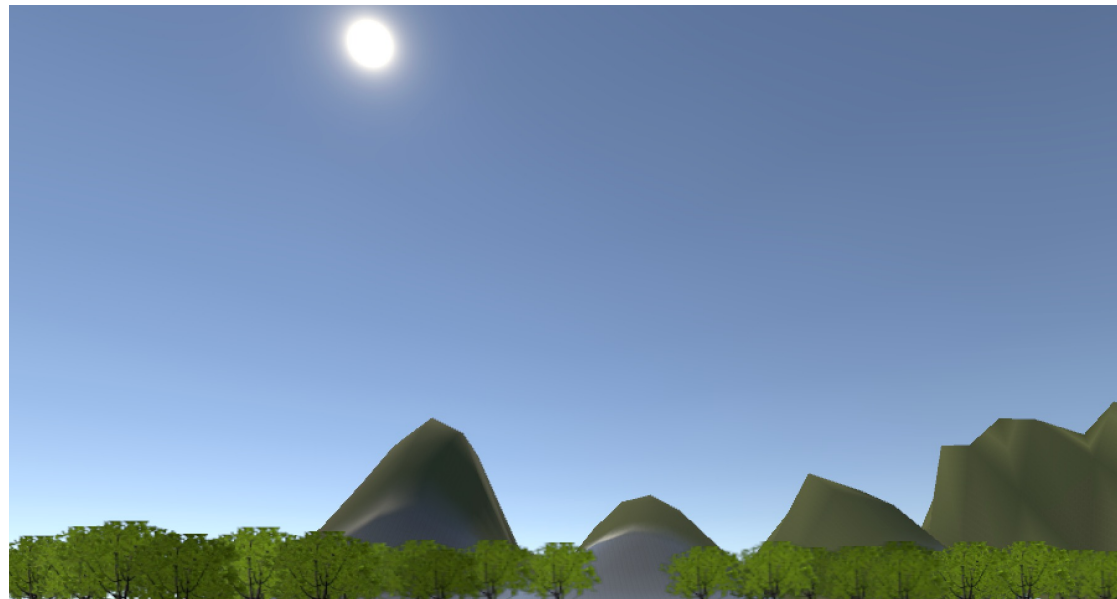
⑫ Window → Rendering → Lighting



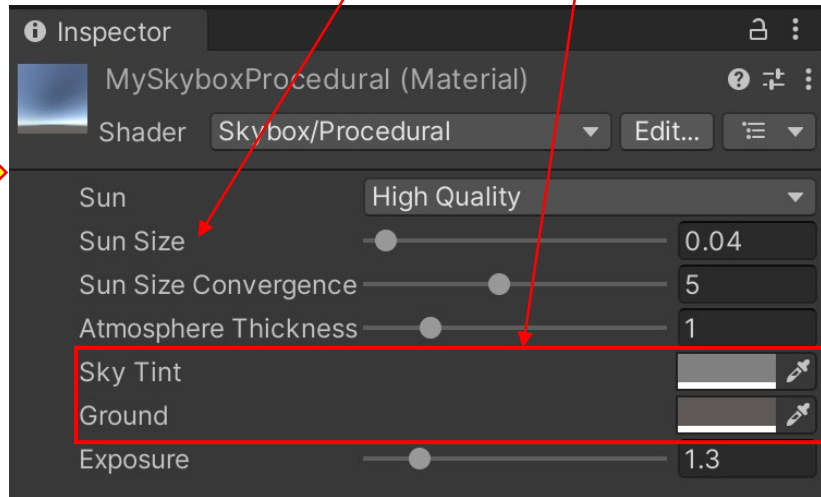
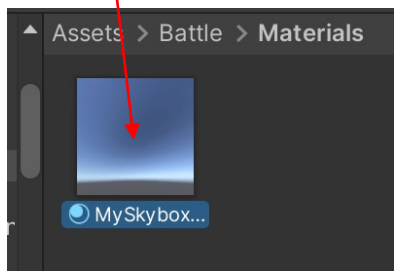
⑬ Environment → ここをクリック → 「MySkybox」
で検索 → 「MySkyboxProcedural」を
選択



実行結果 太陽が現れる

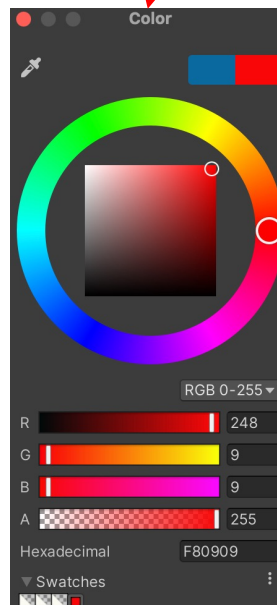
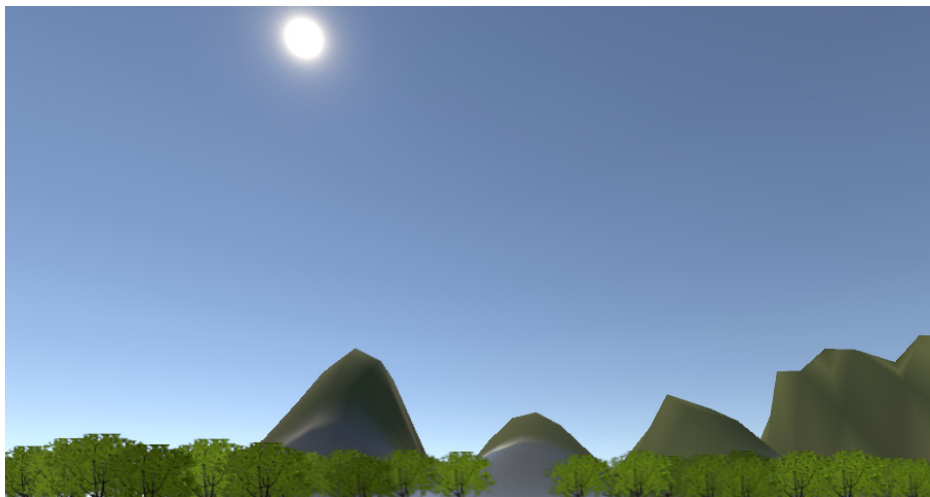


⑭ここをクリックしてインスペクタ画面→ 太陽の大きさや空の色を
変えることができる

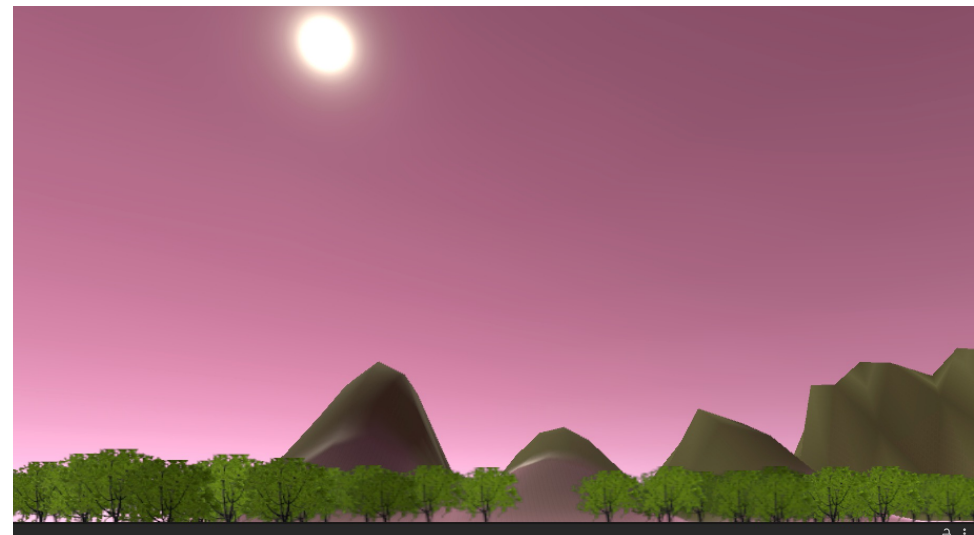


←ここをクリックすると

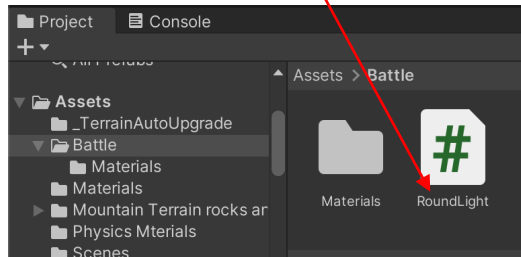
Sky Tint: 空に色
Ground: 地平線より下の色



←色を選択する



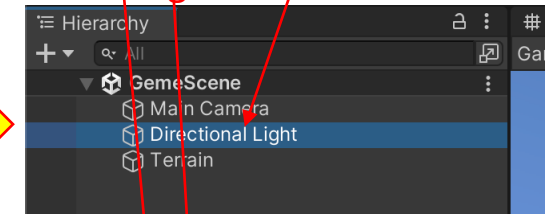
⑮プロジェクトのフォルダ上(ここではbattle)で右クリック→ Create → C# Script → ファイル名を「RoundLight」とする



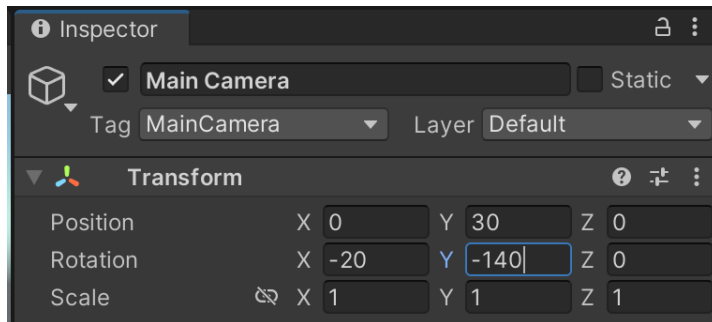
⑯「RoundLight」をダブルクリック後、以下のコードを書いて保存

```
RoundLight.cs ×
Users > ushisawanorihiko > Unity > Test > Assets > Battle > RoundLight.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class RoundLight : MonoBehaviour
6 {
7     private void Update()
8     {
9         transform.Rotate(new Vector3(0, -12)*Time.deltaTime);
10    }
11 }
12
```

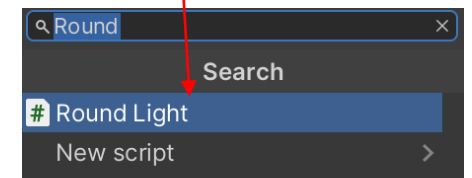
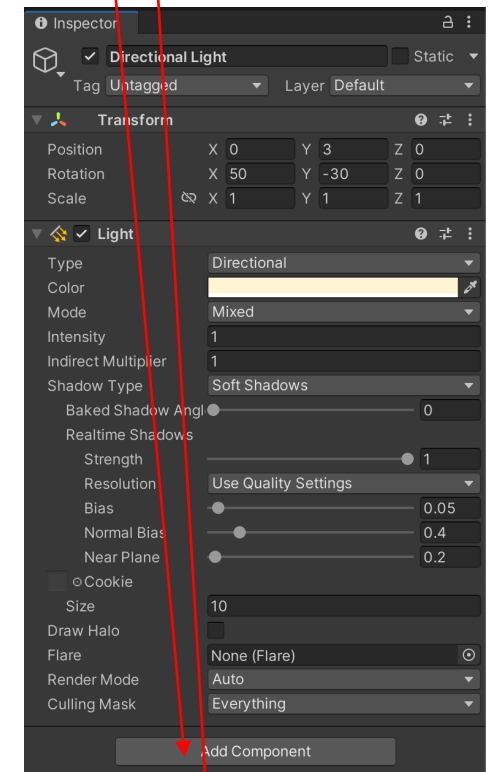
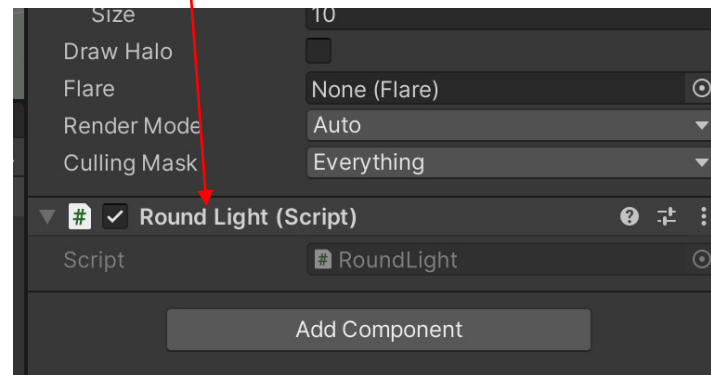
⑰HierarchyでDirectionalLightをクリック→インスペクター画面→Add Componentをクリック→Round Lightをクリック



⑱カメラの位置を変更



⑲ Scriptがリンクしたことを確認



実行結果

