

完全に理解する

# Unity Shader 基礎Ⅱ

# 本日のおしながき

- プロパティについて
- シェーダを書いてみる
- C#も書いてみる



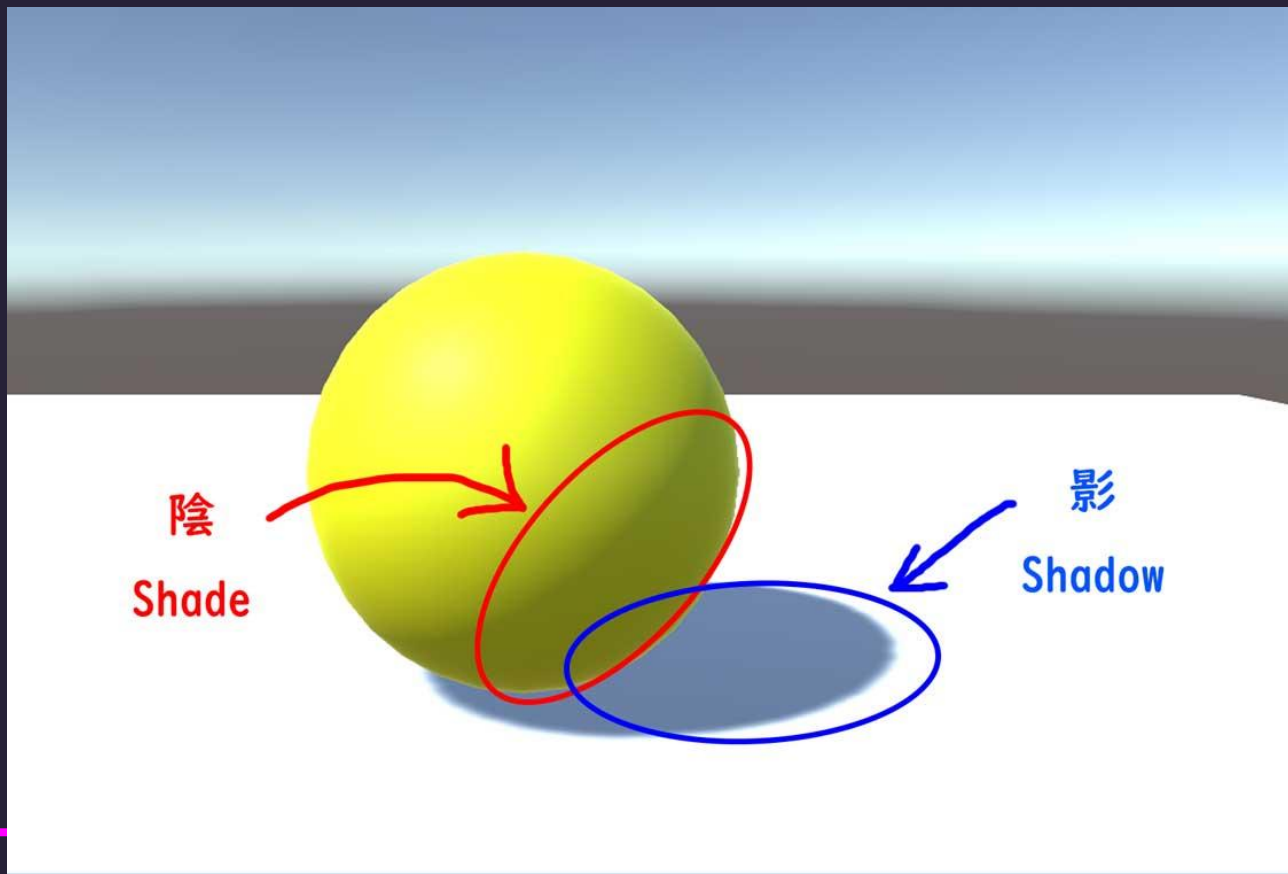
復習と補足！！



# 影と陰の違い

陰  
Shade

影  
Shadow

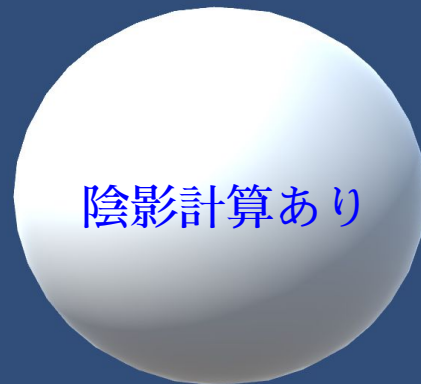


つまりシェーダープログラムとは

陰影の計算を行うためのもの



# なぜ陰影計算を行うか



# シェーダーの種類

- ・ バーテックスシェーダー (頂点シェーダー)
- ・ ジオメトリシェーダー
- ・ ピクセルシェーダー



# Unityのシェーダーの種類

- ・ Surface
- ・ Fragment



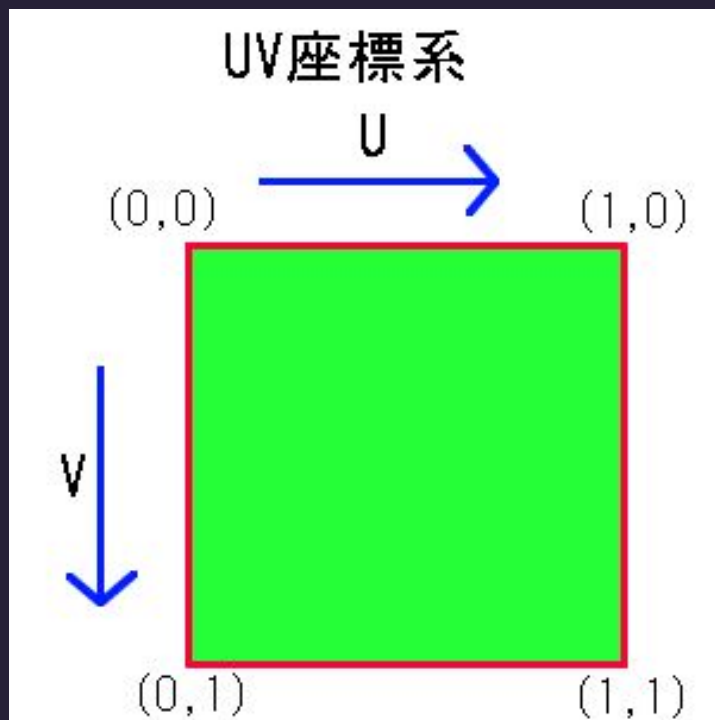


# Unityでシェーダーを書く理由

- ・最適化
- ・面白い表現・特殊な表現



# UV座標系



プロパティについて



# Properties

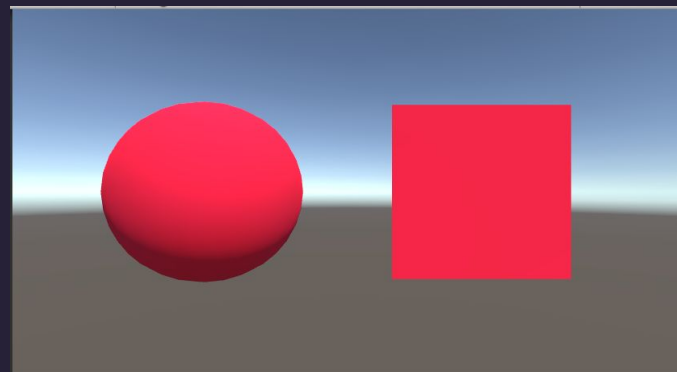
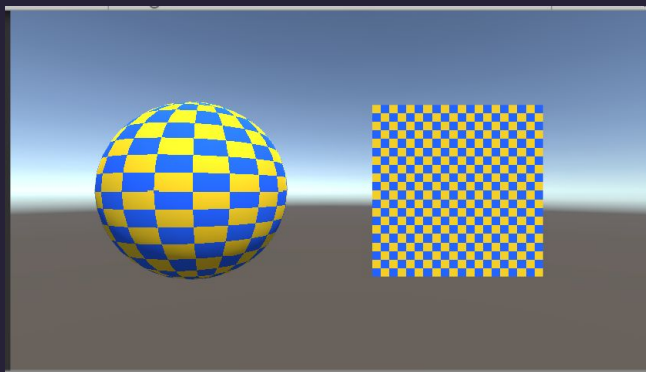
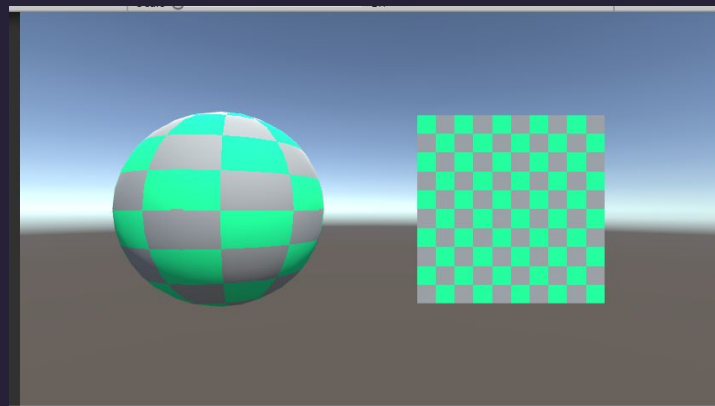
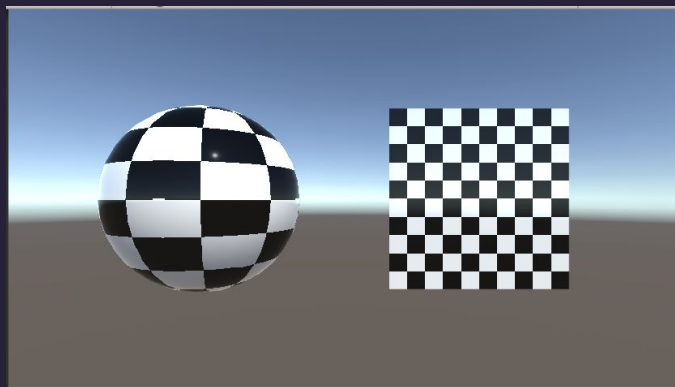
ShaderLab: Properties



シェーダーを書いてみる



# その1 市松模様シェーダーを改造する



C#も書いてみる



# 動的にマテリアルを触りたい場合はコレ

```
using UnityEngine;

public class ColorChanger : MonoBehaviour
{
    Material material;

    [SerializeField]
    Texture2D texture;

    void Start()
    {
        material = GetComponent<MeshRenderer>().material;
    }

    void Update()
    {
        material.SetFloat("_Hoge", 0.0f);
        material.SetColor("_Color", Color.white);
        material.SetTexture("_Texture", texture);
    }
}
```




# 動的にマテリアルを触りたい場合はコレ

using UnityEngine;

```
material.SetFloat("_Hoge", 0.0f);  
material.SetColor("_Color", Color.white);  
material.SetTexture("_Texture", texture);
```

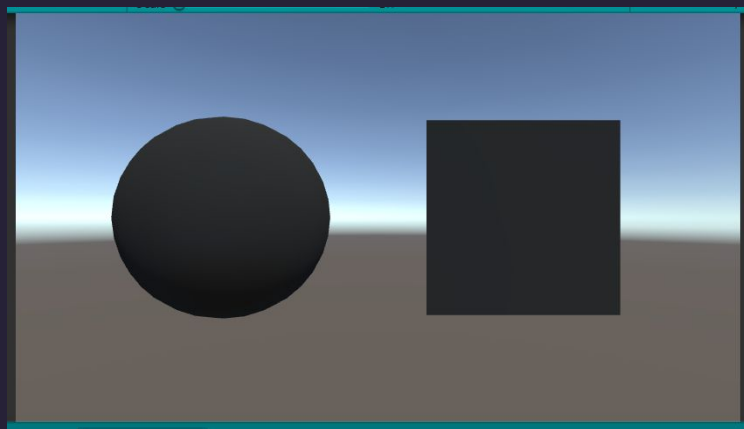
```
void Update()  
{
```

```
    material.SetFloat("_Hoge", 0.0f);  
    material.SetColor("_Color", Color.white);  
    material.SetTexture("_Texture", texture);  
}
```



シェーダーで宣言されている  
変数名を書く  
プロパティで用意する必要は  
ない

## その2 色が変わるShaderとコンポーネントを作る



キー入力

