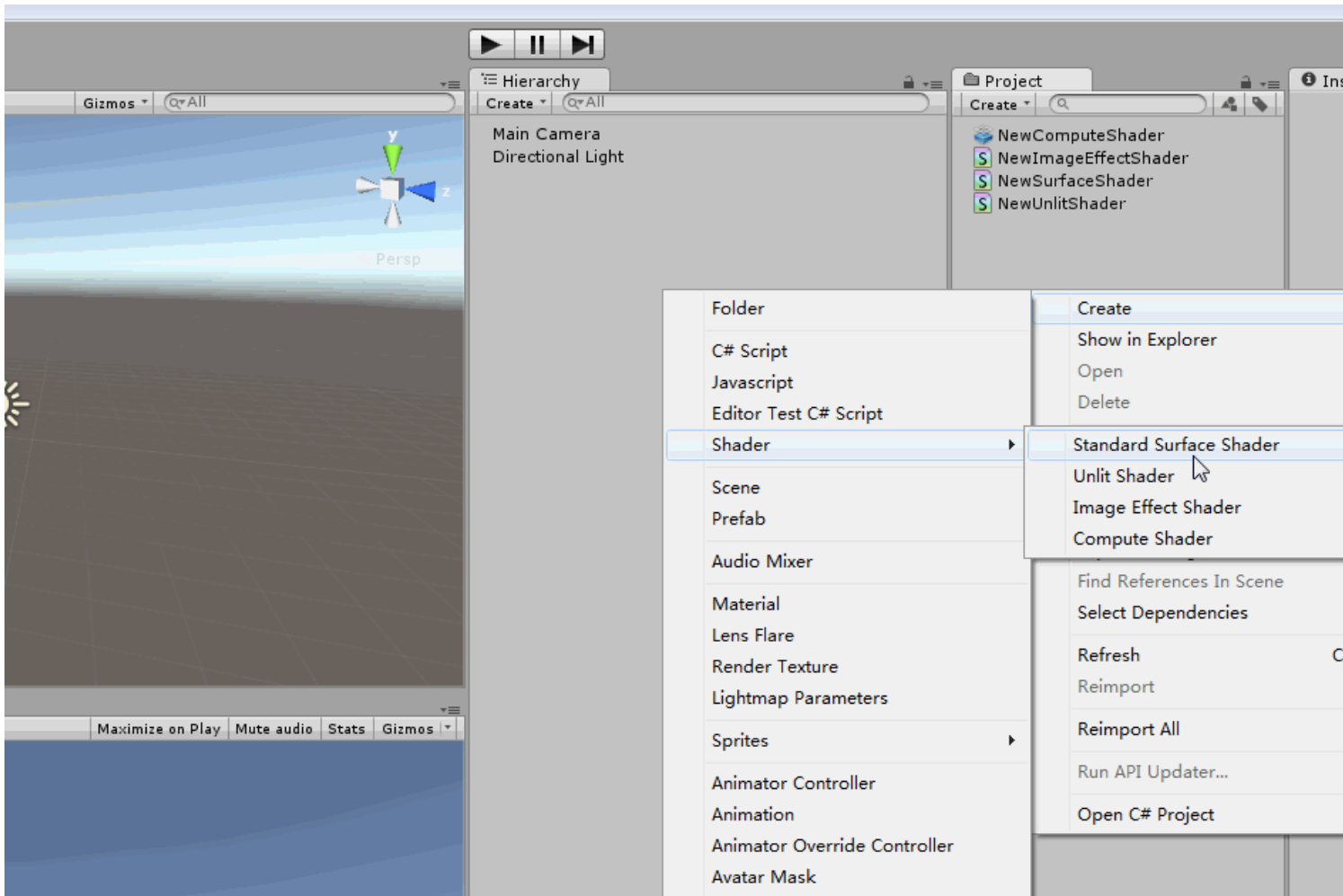


Unity Shader_1-----固定函数着色器 & 一些属性

一、Unity Shader的创建



如上图所示,

其中: Standard Surface Shader 为一个包含了标准光照模型(??)的表面着色器模板;

Unlit Shader 则会产生一个不包含光照(但包含雾效)的基本的顶点(Vertex)/片元(Fragment)着色器;

Image Effect Shader 提供了实现各种屏幕后期处理效果的一个基本模板;

Compute Shader 为一种旨在利用GPU的并行性来进行一些与常规渲染流水线无关的计算的一种特殊Shader。

二、Unity Shader 之 ShaderLab

首先, ShaderLab是Unity提供的编写Unity Shader的一种说明性语言, Unity Shader都是基于ShaderLab编写的。

ShaderLab基础结构如下:

```
Shader "ShaderName" //常用格式为 "Custom/MyShader", 也是该Shader所在位置
{
    Properties{
        //属性
```

_Color ("Main Color", Color) = (1,1,1,0)

Internal name Inspector title Property type Default value

内部名称

编辑器中显示的名称

属性类型

默认值

渲染队列	渲染队列描述
Background	这个队列被最先渲染。它被用于skyboxes等。
Geometry	这是默认的渲染队列。它被用于绝大多数对象。不透明几何体使用该队列。
AlphaTest	通道检查的几何体使用该队列。它和Geometry队列不同，对于在所有立体物体绘制后渲染的通道检查的对象，它更有效。
Transparent	该渲染队列在Geometry和AlphaTest队列后被渲染。任何通道混合的（也就是说，那些不写入深度缓存的 Shaders ）对象使用该队列，例如玻璃和粒子效果。
Overlay	该渲染队列是为覆盖物效果服务的。任何最后被渲染的对象使用该队列，例如镜头光晕。


```

6     _Specular("Specular",color)=(1,1,1,1)
7     _Shininess("Shininess",range(0,8))-4
8     _Emission("Emission",color)=(1,1,1,1)
9     _MainTex("_MainTex",2d)=" "
10    _SecondTex("_SecondTex",2d)=" "
11  }
12
13  SubShader {
14
15    pass{
16      //      color(1,0,0,1)
17      //color[_Color]
18      material
19      {
20        diffuse[_Color]
21        ambient[_Ambient]
22        specular[_Specular]
23        shininess[_Shininess]
24        emission[_Emission]
25      }
26      lighting on
27      separatespecular on
28
29      settexture[_MainTex]
30      {
31        combine texture * primary double
32      }
33      settexture[_SecondTex]
34      {
35        combine texture * previous double
36      }
37
38    }
39  }
40
41 }
42

```