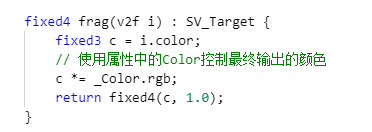
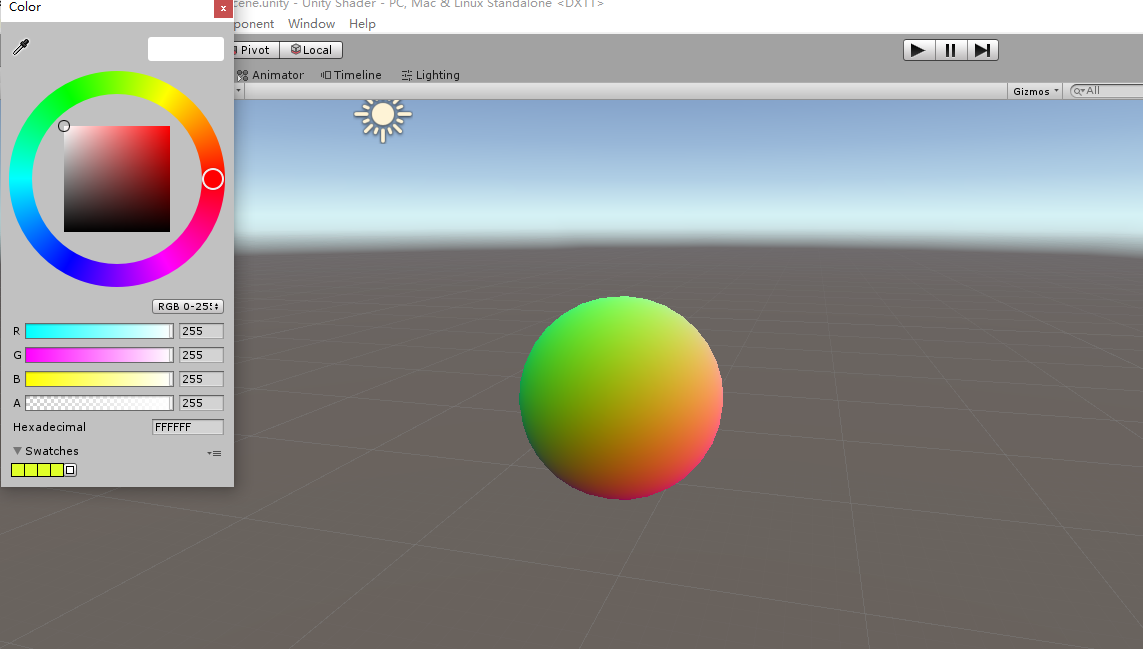
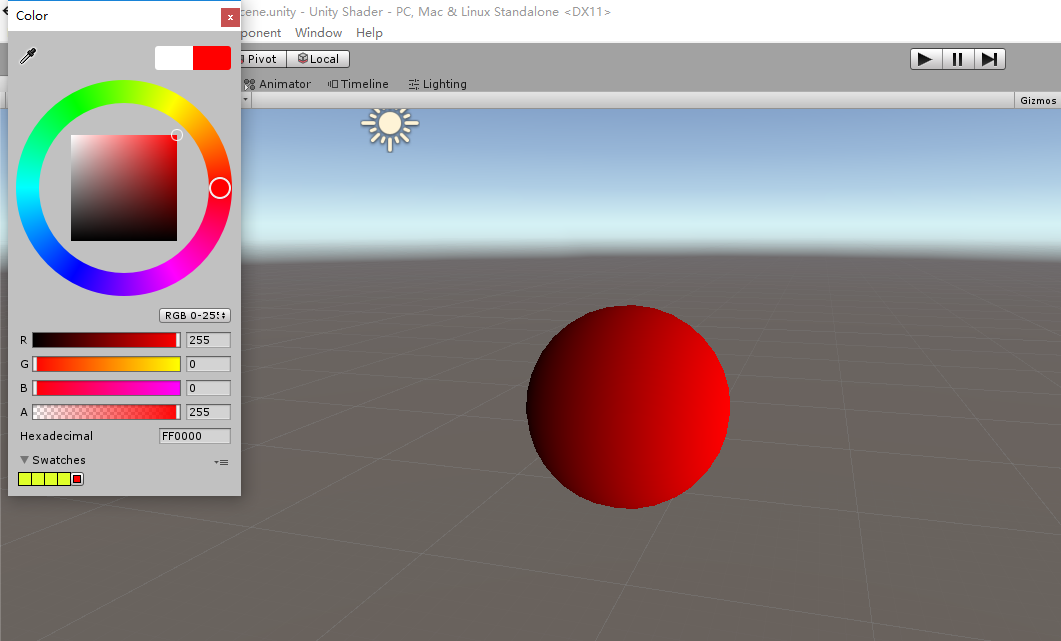
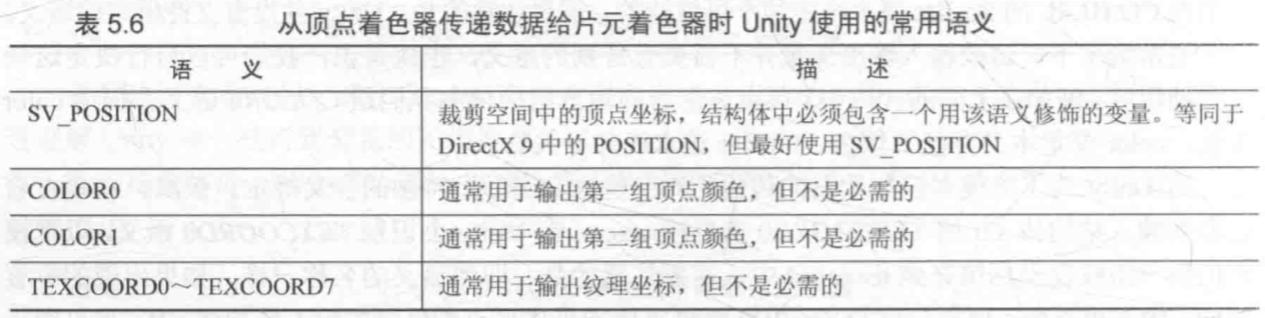
# 开始学习Unity Shader

* 简单的顶点/片元着色器
  + UnityShader基本结构
  + Shader "Custom/SimpleVertex\_fragment\_Shader"
  + {
  + Properties {
  + // 属性
  + }
  + SubShader {
  + // 针对显卡A的SubShader
  + Pass {
  + // 设置渲染状态和标签
  + // CG代码片段
  + CGPROGRAM
  + // 代码片段编译指令
  + #pragma vertex vert
  + #pragma fragment frag
  + // 代码段
  + ENDCG
  + //其他设置
  + }
  + // 其他Pass
  + }
  + SubShader {
  + // 针对显卡B的SubShader
  + }
  + // 回调
  + FallBack "Diffuse"
  + }
* 创建一个简单的顶点/片元着色器
  + Shader "UnityShader/Chapter\_5/SimpleVertex\_fragment\_Shader"
  + {
  + SubShader
  + {
  + Pass
  + {
  + CGPROGRAM
  + #pragma vertex vert
  + #pragma fragment frag
  + float4 vert (float4 v: POSITION) : SV\_POSITION
  + {
  + return UnityObjectToClipPos (v);
  + }
  + fixed4 frag() : SV\_Target
  + {
  + return fixed4(1.0, 1.0, 1.0, 1.0);
  + }
  + ENDCG
  + }
  + }
  + FallBack "Diffuse"
  + }
  + 分析：
  + #pragma vertex vert和 #pragma fragment frag
    - 告诉Unity哪个函数包含了顶点着色器和片元着色器的代码。其中vert和frag就是之前定义的函数名
  + vert函数具体定义
    - v：这个顶点的位置，由POSITION指定
    - SV\_POSITION：顶点着色器输出是裁剪空间中的顶点坐标
  + Frag函数
    - SV\_Target：告诉渲染器把输出颜色存储到一个渲染目标（这里输出到默认的帧缓存）
* 添加更多的参数
  + Shader "Unity Shaders Book/Chapter 5/Simple Shader" {
  + SubShader {
  + Pass {
  + CGPROGRAM
  + #pragma vertex vert
  + #pragma fragment frag
  + struct a2v {
  + float4 vertex : POSITION;
  + float3 normal : NORMAL;
  + float4 texcoord : TEXCOORD0;
  + };
  + float4 vert (a2v v) : SV\_POSITION
  + {
  + return UnityObjectToClipPos (v);
  + }
  + fixed4 frag() : SV\_Target
  + {
  + return fixed4(1.0, 1.0, 1.0, 1.0);
  + }
  + ENDCG
  + }
  + }
  + }
  + 声明的新结构体a2v
    - 除了定点位置以外，新添加了模型空间的法线方向NORMAL、纹理坐标TEXCOORD0
    - 上述结构体就是最基本的结构体格式
    - 之后用结构体代替了vert函数之前的参数
* 顶点着色器和片元着色器之间的通信
  + 需要定义一个新的结构体v2f
  + Shader "Unity Shaders Book/Chapter 5/Simple Shader" {
  + SubShader {
  + Pass {
  + CGPROGRAM
  + #pragma vertex vert
  + #pragma fragment frag
  + // a2v => application to vertex shader
  + struct a2v {
  + float4 vertex : POSITION;
  + float3 normal : NORMAL;
  + float4 texcoord : TEXCOORD0;
  + };
  + // v2f => vertex shader to fragment shader
  + struct v2f {
  + float4 pos : SV\_POSITION;
  + fixed3 color : COLOR0;
  + };
  + v2f vert (a2v v)
  + {
  + v2f o;
  + o.pos = UnityObjectToClipPos (v.vertex);
  + // 将每一个顶点的法线[-1, 1]映射到[0, 1]作为颜色传递给片元着色器
  + o.color = v.normal \* 0.5 + fixed3(0.5, 0.5, 0.5);
  + return o;
  + }
  + fixed4 frag(v2f i) : SV\_Target {
  + fixed3 c = i.color;
  + return fixed4(c, 1.0);
  + }
  + ENDCG
  + }
  + }
  + }
* 使用属性
  + 
  + 
  + 
  + 
* Unity支持的常见CG/HLSL语义
  + 
  + 
  + 