**SRP Batcher**

**SRP Batcher的作用是什么？**

SRP Batcher不会减少Draw Call的数量，但可以减少Set Pass Call的数量，并减少Draw Call命令的开销。

**什么是Set Pass Call？**

Set Pass Call代表改变渲染状态，当切换材质或者切换同一材质中Shader的不同Pass进行渲染时都会触发一次Set Pass Call。比如我们渲染1000个相同的物体和渲染1000个不同的物体，虽然两者Draw Call都是1000，但是前者Set Pass Call为1，后者还是1000。切换渲染状态往往比Draw Call更耗时，这也是URP不再支持多Pass的原因。

**SRP Batcher的工作原理是什么？**

CPU不需要每帧都给GPU发送渲染数据，如果这些数据没有发生变化则会保存在GPU内存中，每个Draw Call仅需包含一个指向正确内存位置的偏移量。SRP Batcher会在主存中将模型的坐标信息、材质信息、主光源阴影参数和非主光源阴影参数分别保存到不同的CBUFFER（常量缓冲区）中，只有CBUFFER发生变化才会重新提交到GPU并保存。

**SRP Batcher是否被打断的依据是什么？**

SRP Batcher是否会被打断的判断依据是Shader变种，即使物体之间使用了不同的材质，但是使用的Shader变种相同就不会被打断，传统的批处理方式是要求使用同一材质为前提的。

**SRP Batcher的实现步骤**

1. 材质的属性都需要在常量内存缓冲区CBUFFER里定义。将\_BaseColor定义在名字为UnityPerMaterial的常量缓冲区中。

CBUFFER\_START(UnityPerMaterial)  
 float4 \_BaseColor;  
CBUFFER\_END

2. 在UnityInput.hlsl中把几个矩阵定义在UnityPerDraw的常量缓冲区中。

CBUFFER\_START(UnityPerDraw)  
 float4x4 unity\_ObjectToWorld;  
 float4x4 unity\_WorldToObject;  
 float4 unity\_LODFade;  
 real4 unity\_WorldTransformParams;  
CBUFFER\_END

3. 至此，我们的Shader已经兼容SRP Batcher了，我们Statistics，材质面板和FrameDebugger中查看SRP Batcher信息。

**什么情况会破坏SRP Batcher？**

若想给相同的物体设置不同的颜色，那么每个物体都需要使用一个相同的材质并通过MaterialPropertyBlock对象调整颜色。我们发现SRP Batcher失效了。