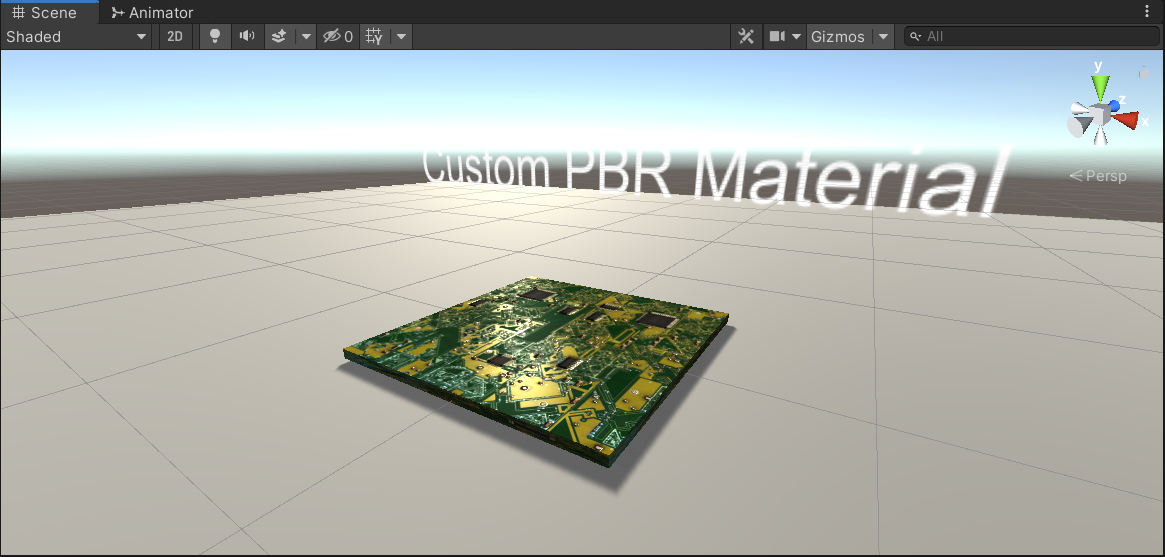
**自定义Shader编写**

**一、 简述**

功能：能实现基本的仿PBR光照（实际原理为Blinn Phone光照模型）和电路板发光流动效果。



**二、 过程**





**1、顶点着色器：**构建切线空间矩阵，便于在片元着色器将贴图中切线空间的法线信息转到世界空间下，方便参与光照的计算；

**2、片元着色器：**计算世界坐标法线、光照方向、顶点位置在世界空间下返回视角的方向，并运用至漫反射和高光反射公式中，最后将其相加即可。其中注意ARM贴图的三个通道，分别为AO，粗糙度和金属度。采样一次分别取RGB三个通道参与运算即可。

**3、流光效果：**采样电路板贴图，并以uv其中一个方向为主，加上\_Time.y让其流动起来即可。注意贴图的采样器需要是Repeat。

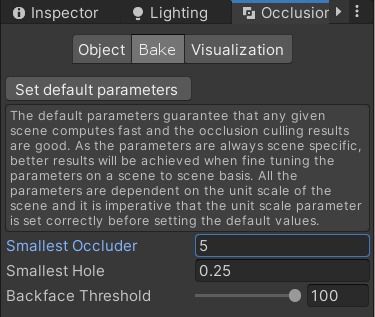
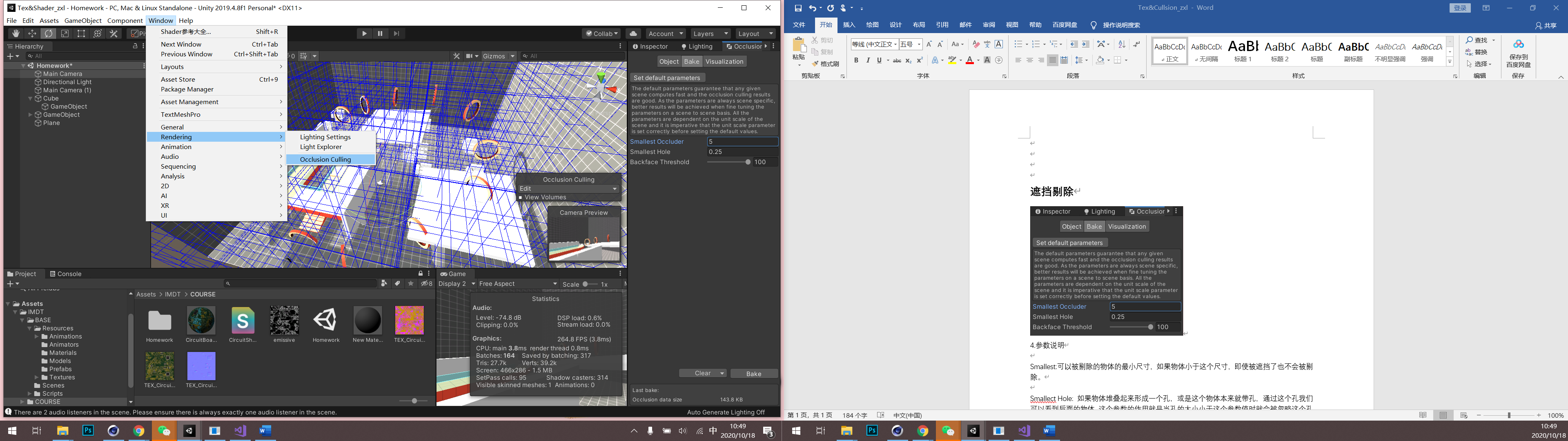
**遮挡剔除**

详情请见文件夹内视频。

**一、 简述**

功能：遮挡剔除功能可以降低被渲染对象个数，从而降低每帧渲染时间。

原理：避免对因为完全被其他物体挡住而不可见的物体的渲染。



**二、参数说明**

参数调整都应该从一个较大的值开始，不断调小，直到得到满意的结果。当测试发现烘培时间过长或者烘培数据过大时，就应该尝试调大参数。

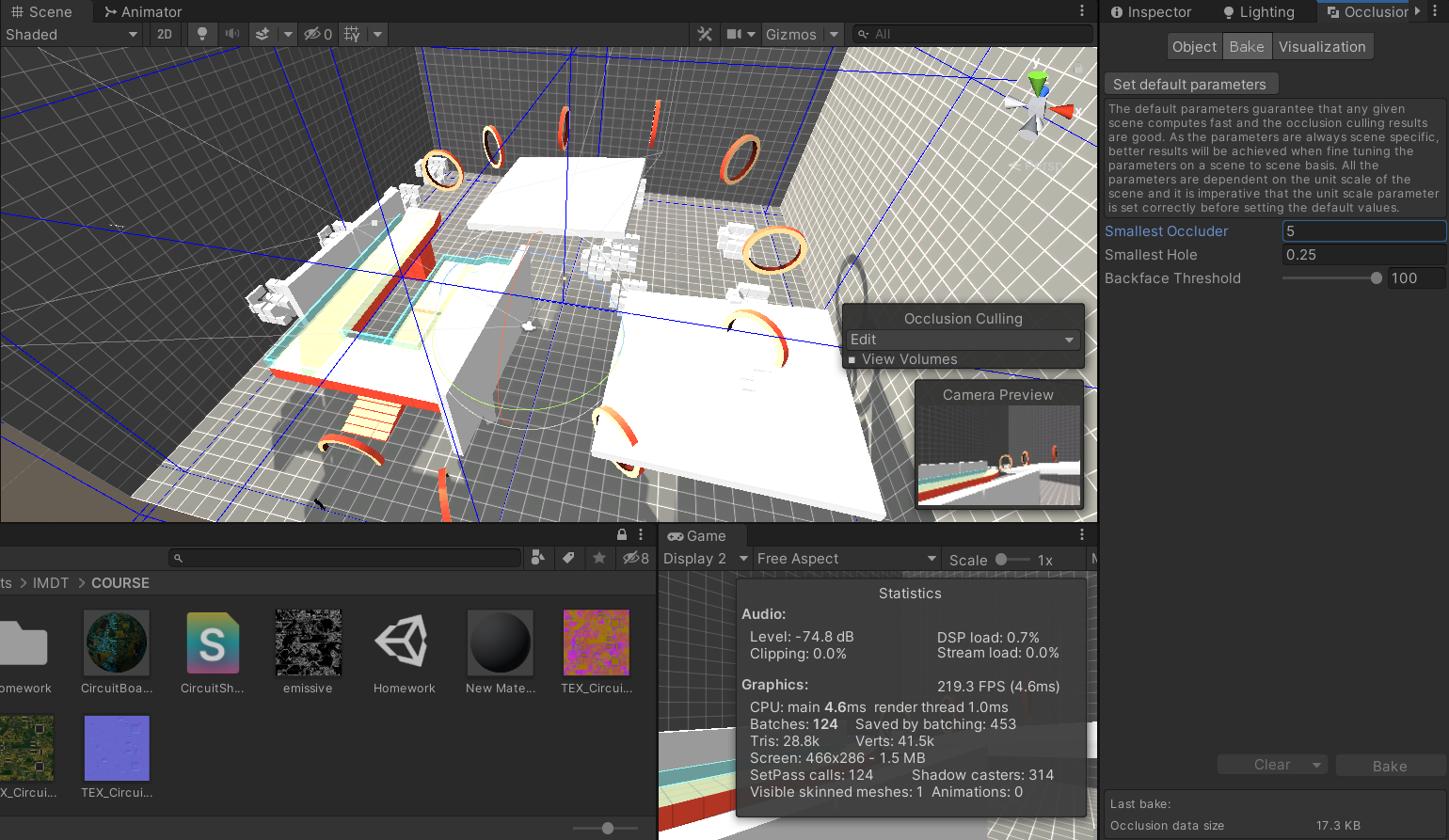
1、Smallest Occluder:可以被剔除的物体的最小尺寸，设得过大会导致剔除成功率下降（但是应该不会产成剔除错误），过小会导致性能问题。

2、Smallect Hole: 如果场景中有带孔的物体需要能被视线穿透(例如墙上的洞, 房间的门)，那么需要将Smallest Hole设置为小于孔的直径

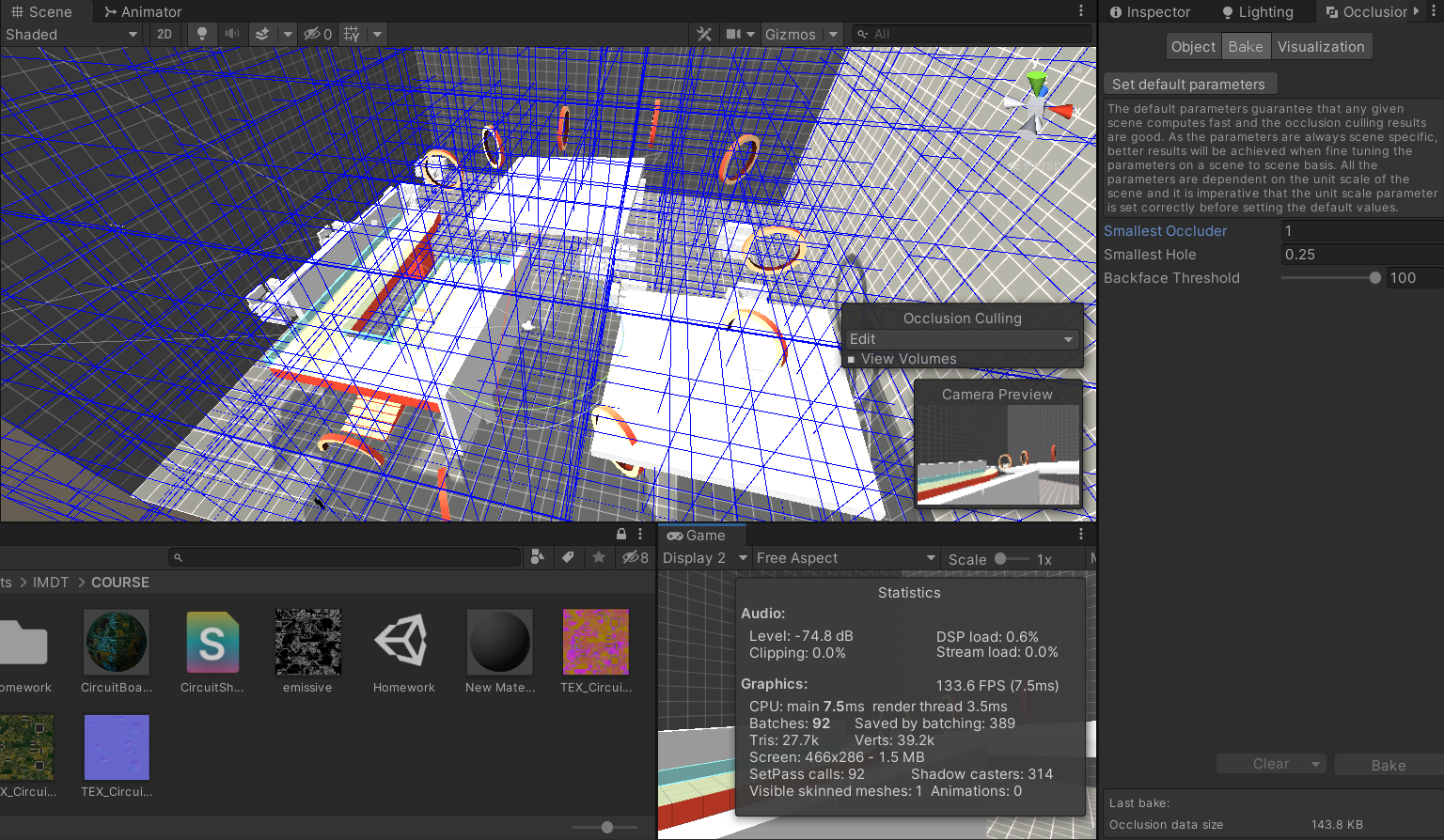
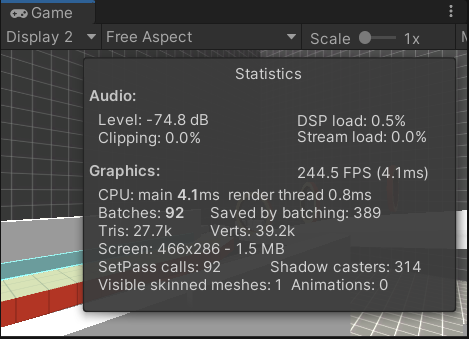
3、Backface Threshold : 设置背景剔除的阈值。本参数的引入是为了减少剔除数据大小，另一方面，设置不当会导致剔除错误（可见的物体被剔除了），比如会不小心把背景剔除掉。

**三、测试结果**

当剔除参数Smallest Occluder设置为：5的时候，我们可发现右下角的Occlusion data size为17.3kb，游戏运行的Statistics的Batches为124，帧数为219.3 FPS.



而当剔除参数Smallest Occluder设置为：1的时候，烘焙时间加长，右下角的Occlusion data size增加为143.8kb，游戏运行的Statistics的Batches减少为92，帧数增加为244.5 FPS.



通过以上测试，我们可以验证，适当的减少Smallest Occluder的大小，能在一定程度上减少DC和增加运行帧数，可是却带来了烘焙时间的延长和数据的增加。所以实际在项目中，Smallest Occluder的具体取值需要反复权衡，找到运行速度和数据载荷之间的平衡点。

当剔除参数Smallest Hole设置为：0.1的时候，我们可发现右下角的Occlusion data size为126.9kb（之前为143.8kb），游戏运行的Statistics的Batches为92（不变），帧数为195 FPS（ 之前为244.5 FPS）。故Smallest Hole也不能设置的过小，同样需要适中匹配。

