1. Camera 组件Depth属性决定不同相机渲染顺序，从小到大渲染。
2. Camera.AddCommandBuffer添加CommandBuffer后，CommandBuffer会一直在摄像机里面，调用RemoveCommandBuffer移除。
3. 模糊的时候是 C#迭代for循环，还是shader循环
4. Formal场景里主相机Clipping Planes的Near为0.02时，深度贴图无法生成，改为0.3就好了
5. 同一个相机用CommandBuffer做后处理，最低限度需要两个CommandBuffer，进行三次Blit：

\_opaqueBuffer.*Blit*(*BuiltinRenderTextureType*.*CurrentActive*, \_renderTex);

\_opaqueBuffer.*Blit*(\_renderTex, buffer0);

\_transparentBuffer.*Blit*(buffer0, *BuiltinRenderTextureType*.*CurrentActive*, \_mat, 1);

1. 优化5，使用一个CommandBuffer，

\_opaqueBuffer.*Blit*(tempTex, *BuiltinRenderTextureType*.*CameraTarget*, \_mat, 3);

1. Bloom原理：

一共四个pass：

第一个pass：对source预过滤亮度不合格的，得到buffer0

第二个pass：对buffer0多次降采样最终得到buffer1

第三个pass：对buffer1多次升采样最终得到buffer0

第四个pass：将最原始source与buffer0相加

1. 为什么要渐进降采样和升采样？

每次降采样原来分辨率1/2，目标图片上一个像素点，对应原来图片周围四个像素点。这样原来图片中所有像素点都会被采样到。如果直接降采样1/4，因为是对图片像素点周围四个像素点取平均值，所以有12个点不会被采样到，丢失了。

参照https://catlikecoding.com/unity/tutorials/advanced-rendering/bloom/