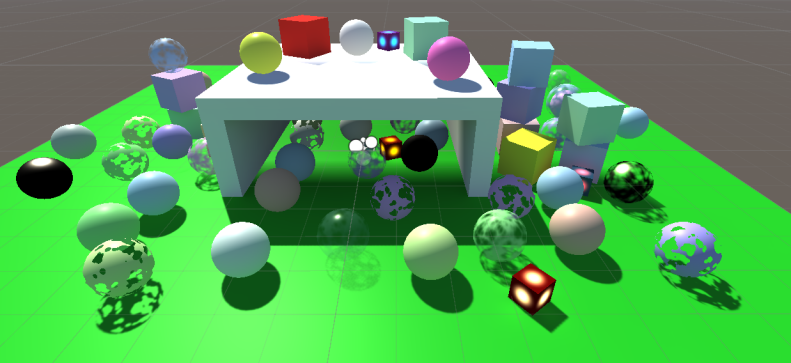
本部分，尝试融合烘焙的阴影和实时阴影。

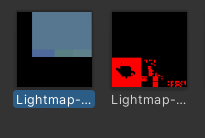
使用光照贴图的阴影时，不会受到最大阴影距离的限制，烘焙的阴影不会被剔除，但是也无法更改。

减少最大阴影距离的情况：



将混合照明模式切换到 Shadowmask 。阴影遮罩混合光照有两种使用方式，可通过项目设置的 Quality 配置。我们先使用该 Distance Shadowmask 模式。另一种模式就叫Shadowmask ，我们将在后面介绍。

与之前的烘焙不同的是，该模式还会额外烘焙阴影遮罩贴图：

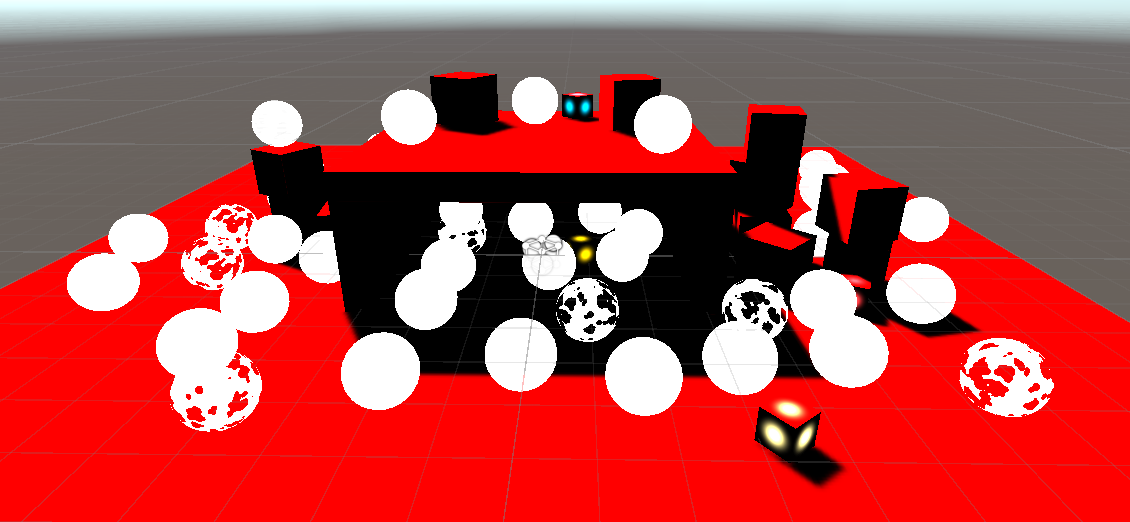


阴影遮罩贴图包含了我们这个混合模式的定向光的阴影衰减，表示对全局光照有贡献的所有静态物体所投射的阴影。数据存储在红色通道中，因此该贴图是黑色和红色的。

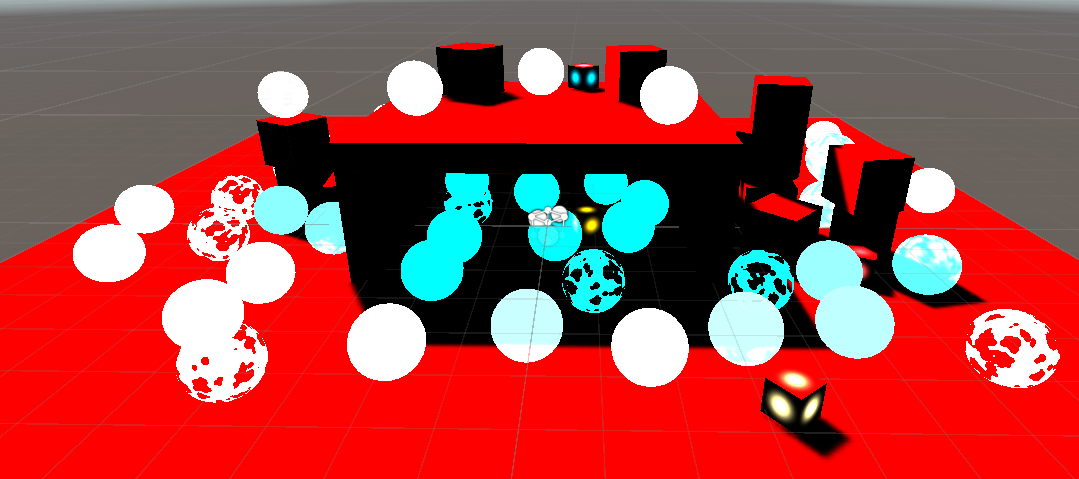
我们需要检测方向光，在包含shadowmask时设置关键字。

然后在gi里采样ShadowMask，在计算光照时把采样结果传给ShadowData。

得到的采样结果如下：

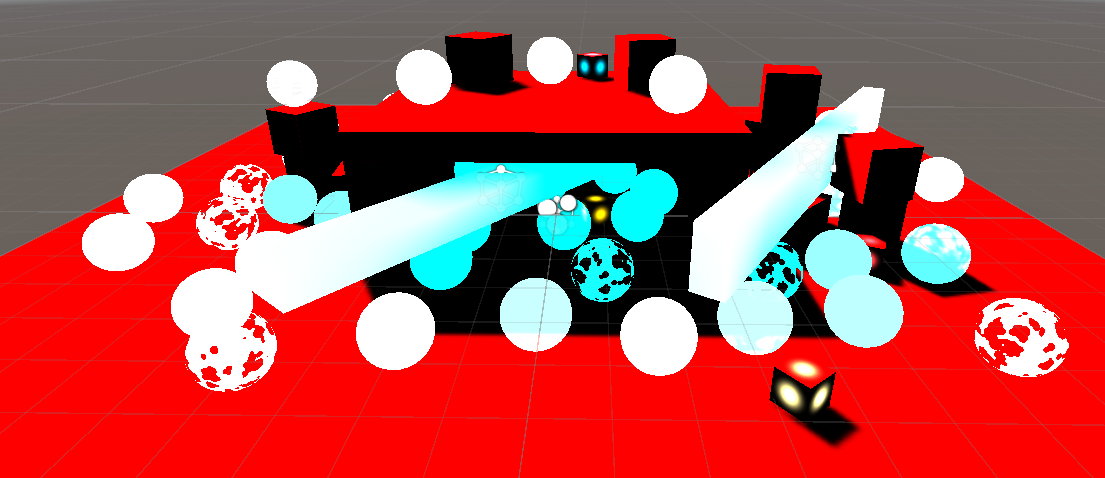


同样的，unity还有遮挡探针这一操作，用于计算动态物体的遮挡情况，类似的添加上去：

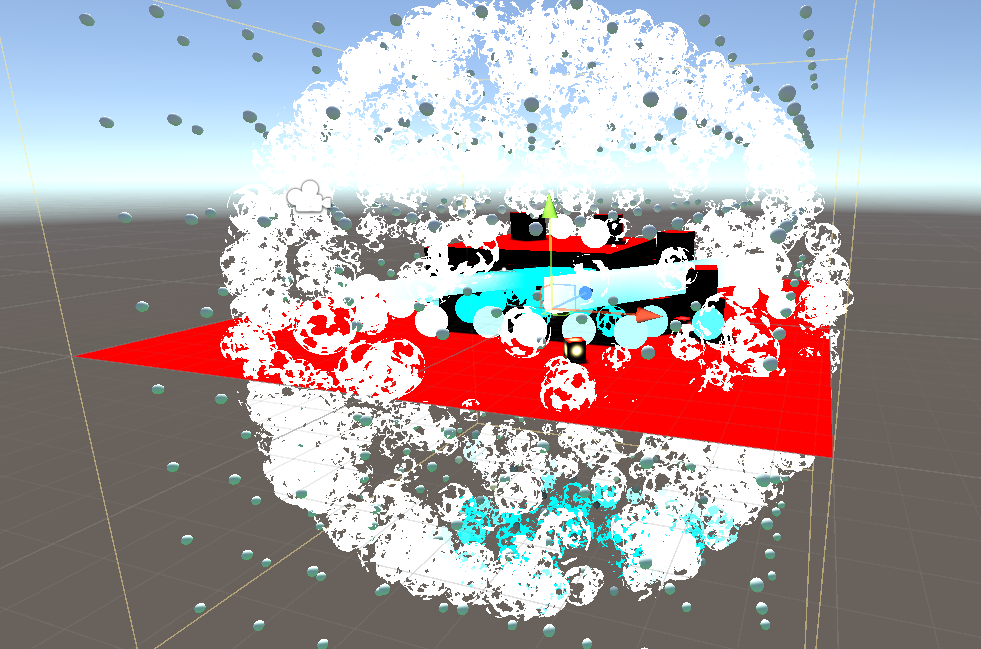


注意到，动态物体的gb通道是全1的，r通道和静态物体处理一样。

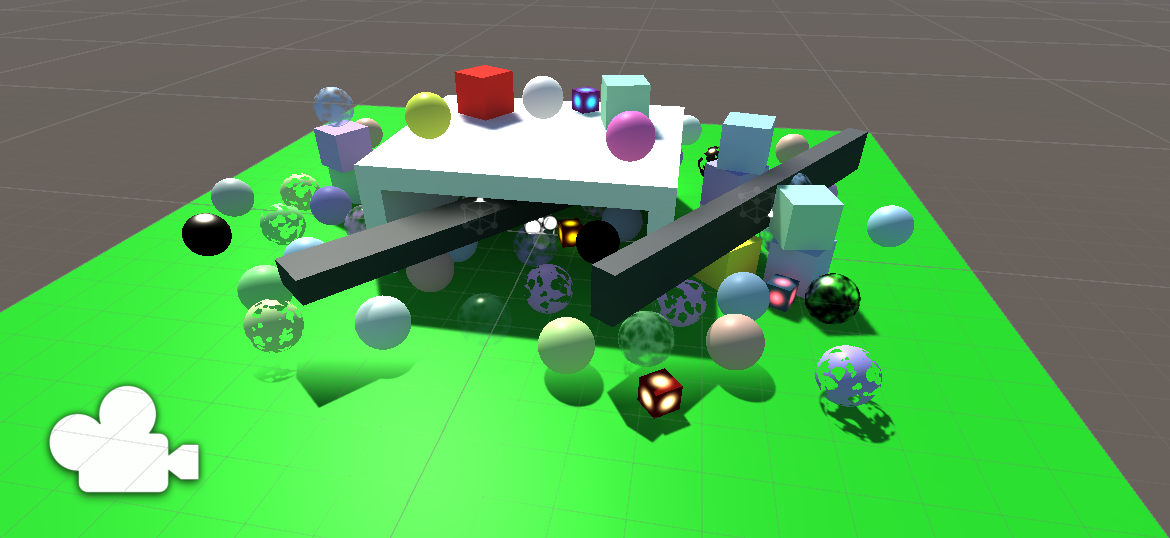
相似地加入LPPV的采样：



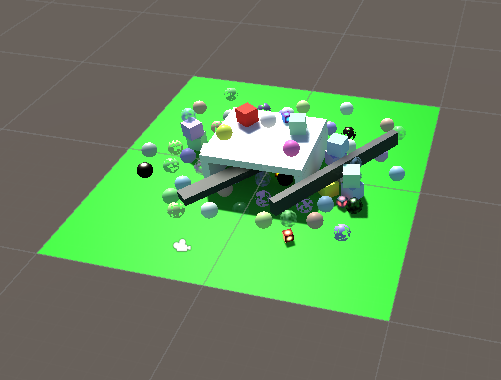
对于MeshBall，需要动态创建occlusionProbes并上传：



混合Bake的结果和实时阴影的结果，可以让静态物体的阴影不被淡化：

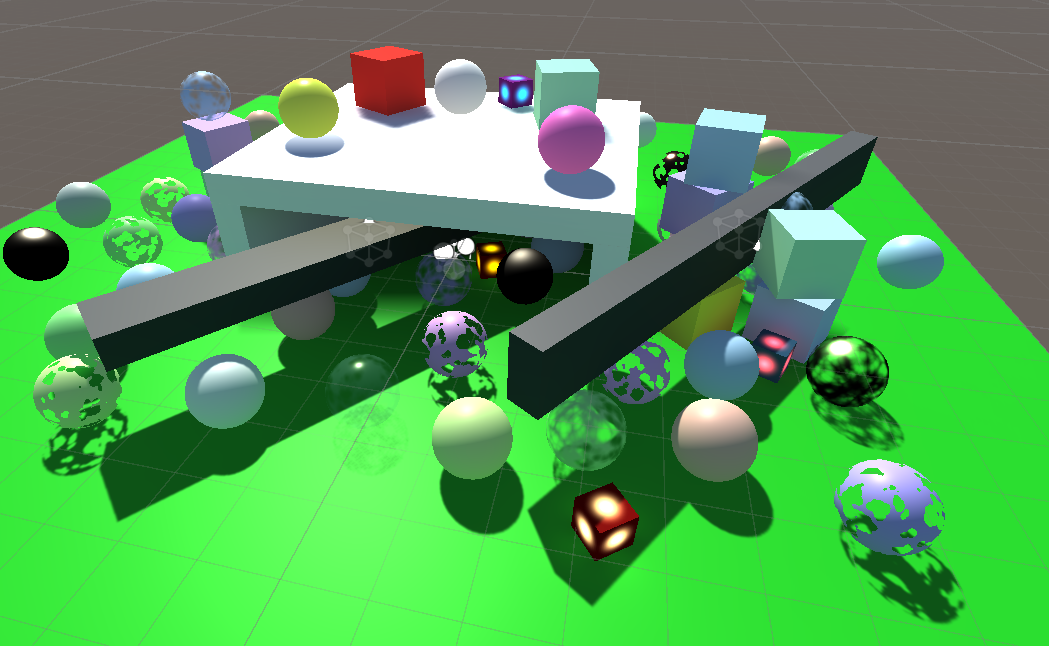


并且，我们还可以调整之前处理的逻辑，让超出范围时，完全使用烘焙的结果：

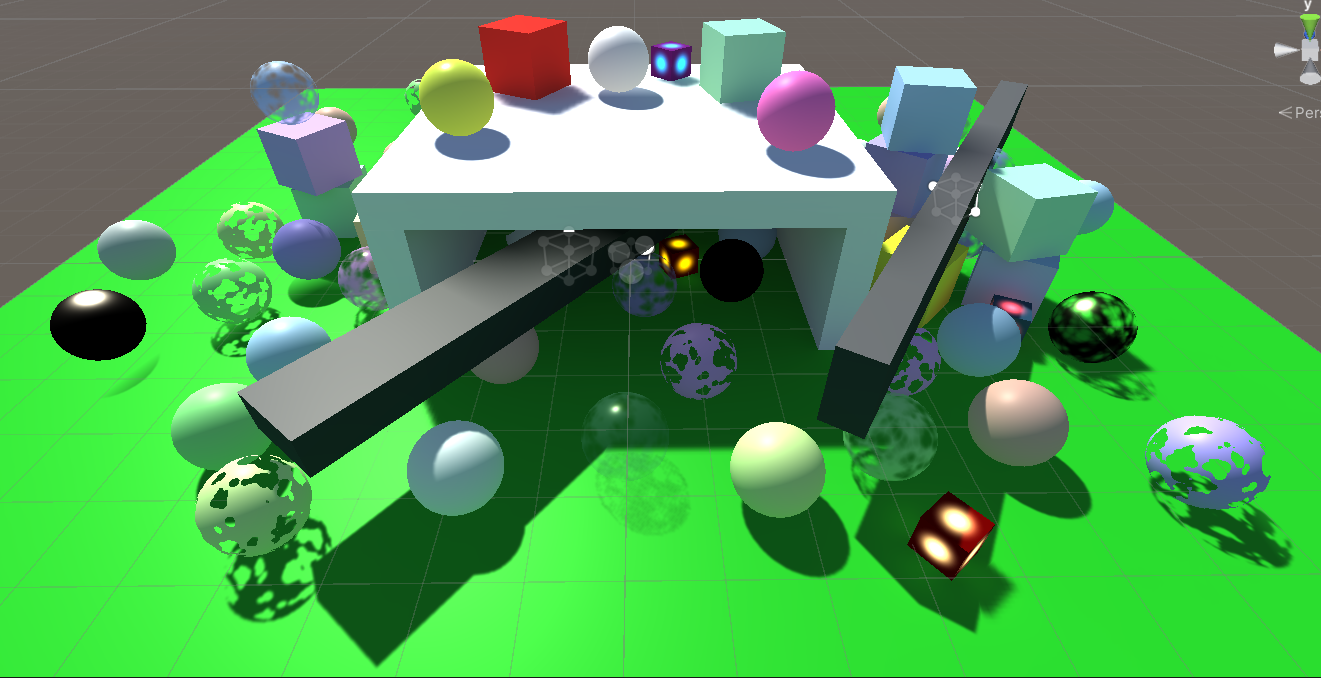


下面处理另一个模式：Shadowmask

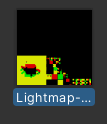
该模式下，静态物体不会生成实时阴影：



对于该模式，混合时，使用静态物体的烘焙，与动态物体的阴影取最小：



当使用多个光源时，不同光源的shadowMask会放在不同通道里，当超出4个时，多余的光会变成完全烘焙的。



把光源选择的通道作为shadowData上传，然后在采样时选择正确的通道即可。

结果如下：

