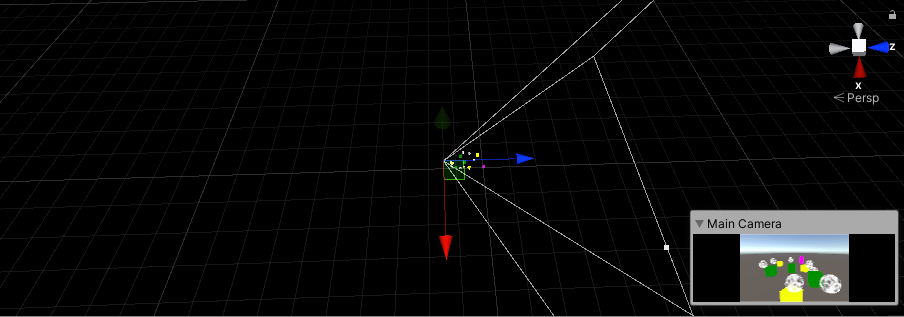
3.4:画Gimos

3.5 画Unity UI

游戏窗口有但是Editor窗口没有UI 显示

4:多摄像机

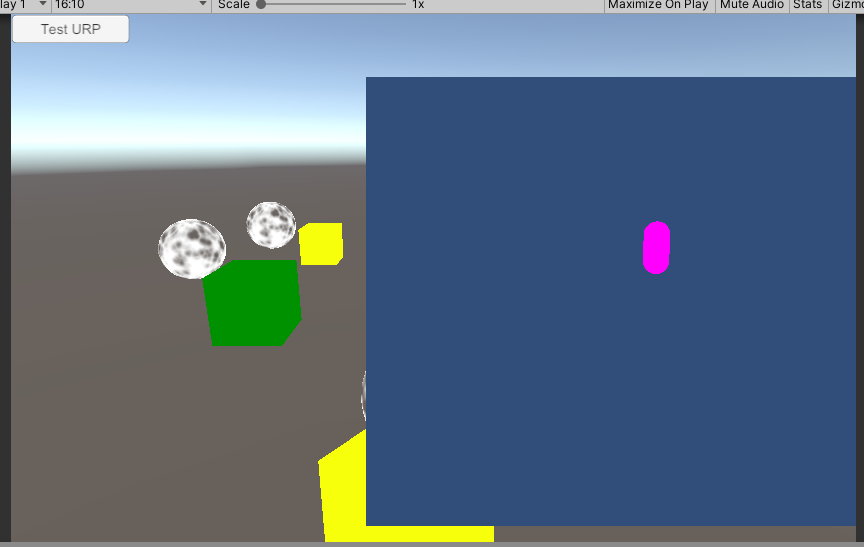
4.1 两个摄像机

两个摄像机同一个范围会合并到一起,要有不同的范围进行渲染。

4.2 处理改变的BufferName

4.3 图层

4.4 清除标志 Cull的层和摄像机的清除标志

课程2 Draw Call

目标：写一个HLSL 着色器

支持SRP Batch ,GPU Instancing 和动态合批

配置 材质属性对于每一个物体和随意的画很多

创建半透明材质和 镂空材质

1：着色器

* 1. 不发光的Shader
  2. HLSL程序
  3. Include 保护
  4. 着色器功能

我们是应该使用float还是half

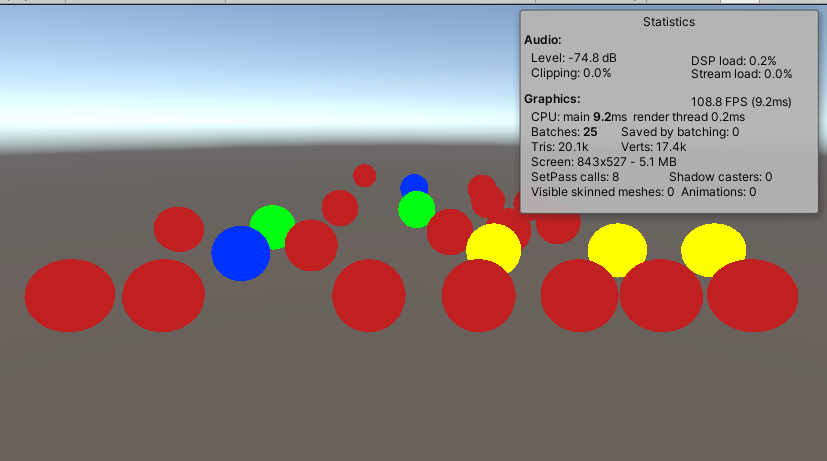
* 1. 空间转换
  2. 核心库 Common.HLSL SpaceTransform.HLSL
  3. 颜色

每一个材质球配置颜色\_BaseColor

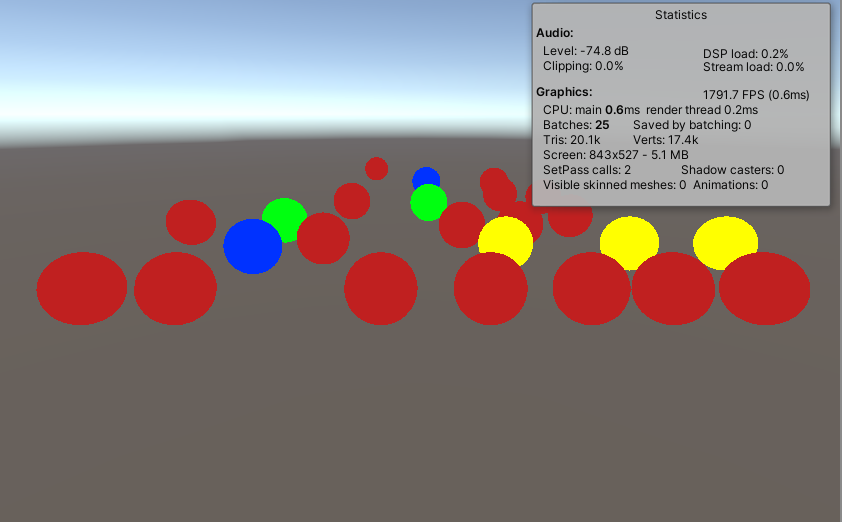
1. 合批
   1. SRP Batch

SRP Batch并不会减少Draw Call的数量而是Shader更精简,缓存材质属性在GPU 并不必发送给每一个Draw CallSRP批不是减少draw调用的数量，而是使它们更精简。它在GPU上缓存材质属性，所以它们不必在每次draw调用时都被发送。这既减少了必须通信的数据量，也减少了CPU在每次draw调用中必须做的工作。但这只有在着色器遵循统一数据的严格结构时才有效。

开启之前



开启之后



1. Save by batching Set Pass Call减少到2的数量
   1. 许多颜色

Unity简单得会把相同变体得Shader进行合批。OnValidate()当组件被加载和改变得时候进行触发,SRP Batch并不能处理逐物体得材质属性

* 1. GPU Instancing

#include "Packages/com.unity.renderpipelines.core/ShaderLibrary/UnityInstancing.hlsl"

Instancing Buffer

GPU Instancing 必须使用相同的材质 SRP Batch是使用相同的Shader变体 材质属性单独改变,批处理大小是会有限制的。

* 1. 画许多实例化网格

使用Instancing API的时候 必须材质开启Enable GPU Instancing 多少Draw Call是取决与平台的,每个Draw对应的最大Buffer是不同的。

* 1. 动态合批

会有顶点限制和物体尺寸限制 静态合批需要标记和会增大内存

* 1. 配置合批

1. 半透明

3.1混合方式

Src Alpha 是要画的图像

* 1. 不写入深度
  2. Alpha纹理
  3. Alpha 裁剪

0或者更小的值 片段会被裁剪掉

Alpha Test 2450 在不透明渲染之后进行渲染,丢弃片段让GPU 优化变得不可能,

3.5 Shader 特征

**shader\_feature 根据配置勾选材质生成变体 Multi都生成**

3.6Cutoff 每一个物体

3.7生成更多的Alpha Test物体

第三课时 方向光

用法向量计算光照

支持4盏方向光

应用BRDF

制作点光源的半透明材质

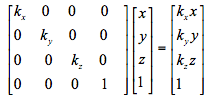
‘ 创建自定义GUI 预先设置

1 灯光

模拟灯光和物体表面相交

* 1. 灯光Shader
  2. 法线向量

世界空间法线向量,取决与灯光和表面的角度 表面法线



如果三个缩放系数k都相等，我们称为统一缩放，否则是非统一缩放，因为非统一缩放会拉伸或挤压模型，所以会改变与模型相关的角度和比例，这在法线变换时很重要，如果非统一变换，直接使用变换顶点的变换矩阵就会出错

1.3内插法线

1.4 表面属性

1.5计算灯光

2 灯光

2.1灯光的结构

光的放行指明的是光是从哪个方向来

2.2 灯光功能

2.3 发送灯光数据到GPU

2.4 可见灯光

2.5 多个方向光

2.6 Shader 循环

2.7 Shader 目标

3 BRDF

* 1. 进来的灯光
  2. 输出的光线 光线反射和光线的散射

分散的高光反射、

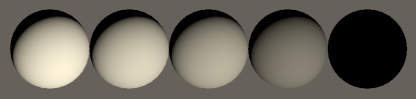
3.3表面属性

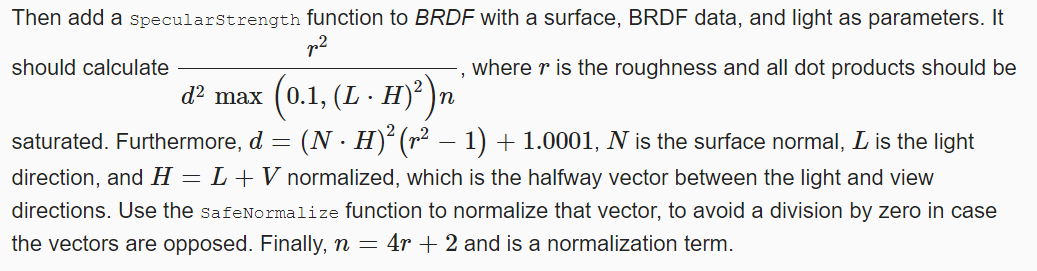
金属度和光滑度

* 1. BRDF属性

3.5反射率

金属度越强 漫反射越弱 能量守恒



* 1. 高光
  2. 粗糙度
  3. 视野方向
  4. 高光强度 计算高光系数
  5. 

增加金属度 漫反射颜色逐渐变黑 高光反射金属的颜色

Mesh Instance

4半透明

4.1预乘Alpha

预乘alpha和漫射有效地将物体变成玻璃

可以把漫反射影响也变成透明 的颜色完全模拟玻璃效果

5 Shader GUI

5.2设置属性和关键字

5.3 预设置按钮

第四章 方向光阴影

渲染和采样阴影RT

支持多个方向光阴影

用 Cascaded 阴影图 –>会增加Draw Call 软阴影技术

混合 淡入淡出 和 过滤阴影

1渲染阴影

投射阴影物体和接收阴影物体

1.1阴影设置

渲染阴影的距离和阴影图的大小,阴影生成的最大距离

1.2 通过设置

1.3 Shadow 类

虽然从逻辑上来说阴影是照明的一部分，但它们相当复杂。

1.4 灯光和阴影

GetShadowCasterBounds 当前灯光没有照射任何物体，但是依然进行shadowmap的绘制 看是否绘制ShadowMap

1.5 创建阴影图集

创建阴影贴图是通过在纹理上绘制阴影投射对象来完成的，格式为shadowMap格式

我们只能在首次声明一个纹理时才释放它，目前我们只在有方向阴影要渲染时才这么做。最明显的解决方案是只有当我们有阴影时才释放纹理。但是，不声明纹理会给WebGL 2.0带来问题，因为它将纹理和采样器绑定在一起。当一个材质和我们的着色器加载而纹理缺失时，它将失败，因为它将得到一个默认的纹理，这将与阴影采样器不兼容。我们可以通过引入一个着色器关键字来生成省略阴影采样的着色器变体来避免这一点,少引入shader变体

1.6 影子第一步

1.7 渲染

渲染多少盏平行光阴影是分Tile的, 阴影贴图的想法是我们从光的角度渲染场景，只存储深度信息，这个结果告诉我们光在击中物体之前传播了多远。

每增加一盏方向光,物体画阴影多一倍的DrawCall 在灯光位置画多次物体 全部叠加到一张ShadowMap 分Tile

1.8 阴影投射 Pass

1.9 多个灯光

2 采样阴影

2.1阴影矩阵

构建阴影矩阵,具体从哪一个Tile里面采样阴影图

近裁剪平面和远裁剪平面 决定摄像机看到的深度范围

2.2 储存阴影数据每一盏光

2.3 Shadow HLSL

双线性对常规的ShadowMap是没有意义的。

2.4 采样ShadowMap ->转换到阴影空间的矩阵直接是Unity进行提供的。

计算阴影的衰减, 然后添加一个GetDirectionalShadowAttenuation函数，返回shadow attenuation，给定的方向阴影数据和一个surface，这个surface应该是在世界空间定义的。它使用平铺偏移量来检索正确的矩阵，将表面位置转换为阴影平铺空间，然后对图集进行采样。块的片段是并行进行着色的

2.5 灯光衰减 动态分支和静态分支

问题出现摩尔纹和黑方块, 摩尔纹是自阴影导致的,就是阴影ShadowMap的分辨率有限patterns 提高分辨率并不能消除 ShadowMap对齐的方向是灯光的方向而不是摄像机的方向,一个光使用另外一个光的阴影。现在这些无效的阴影保持可见.黑方块跟阴影距离有关系。

3 Cascaded 级联的阴影图

比率增大ShadowMap分辨率 更近的地方用更高的分辨率,需要一个可变的ShadowMap分辨率,基于阴影接收器的视野距离。级联阴影映射是这个问题的解决方案。这个想法是阴影施法器被多次渲染，所以每个光在图集中得到多个瓷砖，称为瀑布。第一个级联只覆盖靠近相机的一个小区域，连续的级联缩小，以相同数量的texel覆盖一个越来越大的区域。着色器然后为每个片段采样最好的级联。

3.1设置

3.2 渲染Cascades

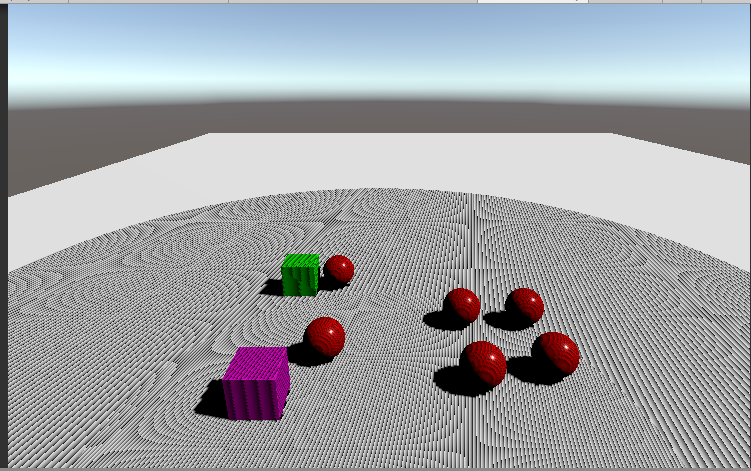
每多一个Cascade 就会多一倍的DrawCall 会在画一遍,多一盏光也会多一倍的Draw Call。开四倍 地板Plane画了四次。

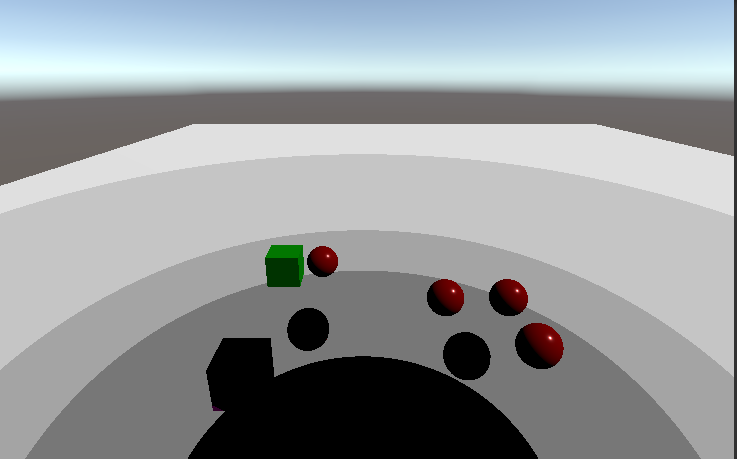
3.3 剔除球 优化

检查片段是否在这个球里面,球是确定片段是否在哪一级的Cascade里面

3.4 采样Cascade

世界空间表面位置 剔除球决定片段在哪一级的Cascade里面。根据片段世界位置到球体中心的距离和球体半径进行比较。可视化阴影Caca





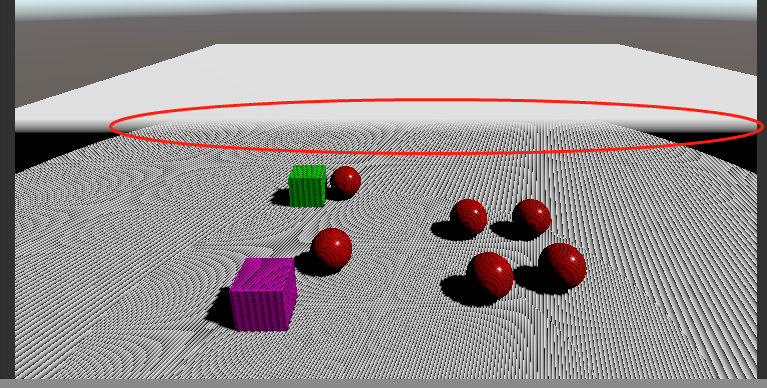
* 1. 剔除阴影采样

3.6最大距离

最大距离是基于视野空间深度,而不是相机的位置距离,

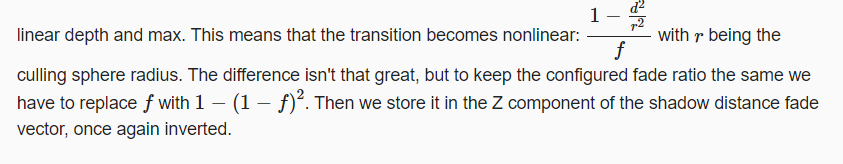
3.7阴影衰减

增加一个衰减项产生一个过渡,边界处在阴影消失的边界的地方~~



3.8 衰减 Cascades 级联贴图

两个Cascades之间做一个过渡。利用一个非线性的衰减



4:阴影质量

问题:acne这是由表面的不正确的自阴影造成的，没有与光的方向完全对齐 简单的增加分辨率是不能解决acne

4.1：深度Bias

增加恒定的偏移,问题Peter平移。

4.2 Cascade 数据

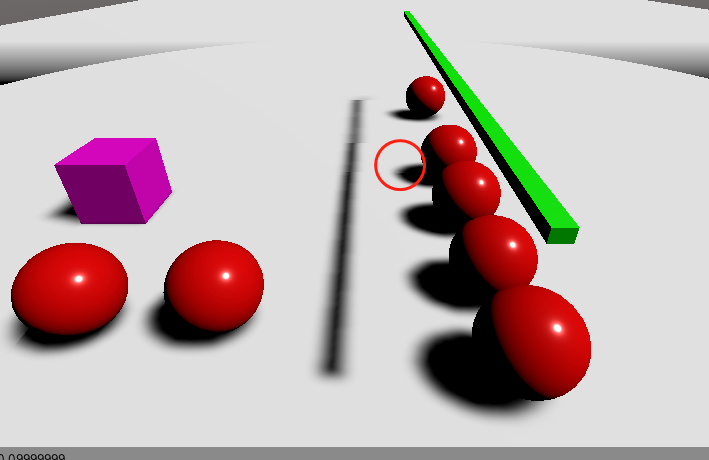
acne 取决于世界空间纹素的大小

4.3 Normal 偏移

不正确的自阴影发生是因为投射阴影者的深度texel覆盖了多个碎片,

4.4配置Bias

4.5 阴影平坠 ？

* 4.6 软阴影PCF过滤 百分比渐进过滤
* 4.7 混合Cascade
* 

4.8 抖动过渡

抖动 更省性能模式

4.9 剔除偏差 网易P8相当腾讯3-3 腾讯3-2

5半透明

5.1 阴影模式

5.2裁剪阴影

半透明 需要ShadowCaster Pass里面用clip 裁剪掉投射的图形。

5.3 抖动阴影

5.4 没有阴影

5.5 Unlit 阴影投射

第五课时 烘培灯光

灯光图和反射球

烘培静态全局光,采样Light Maps和反射球 和LPPVs

支持自发光表面

1 烘培静态光

提前存在LightMap里面和Probe，后者是全局光照的一部分,静态的不能在运行时进行改变,增加构建的时间的内存。

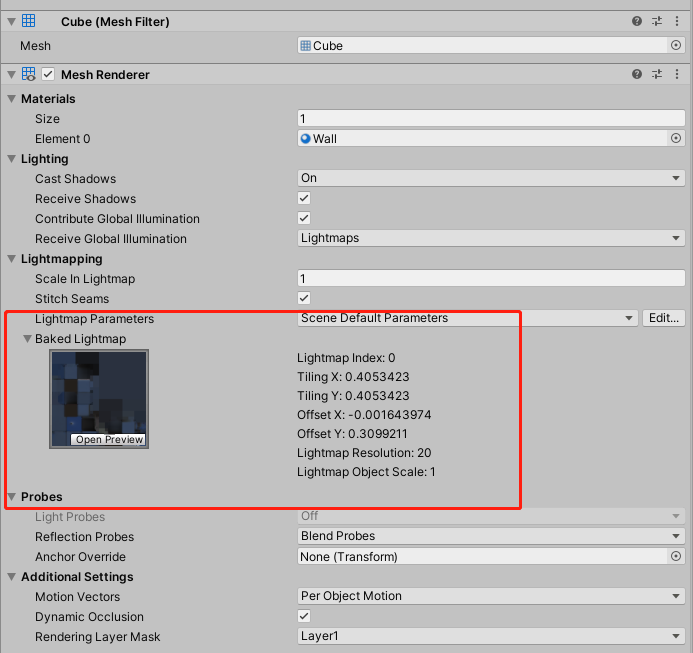
1.1场景灯光的设置

方向光模式跟NormalMap有关系？ 现在选择No-Directional

1.2 静态物体

灯光模式Mix，告诉Unity要对这盏光烘培间接光,除此之外这盏光还是一盏实时光。灯光设置勾选自动生成LightMap

勾选LightMap 使用物体会变成静态



* 1. 完全的Bake灯光

Bake 场景完全黑掉

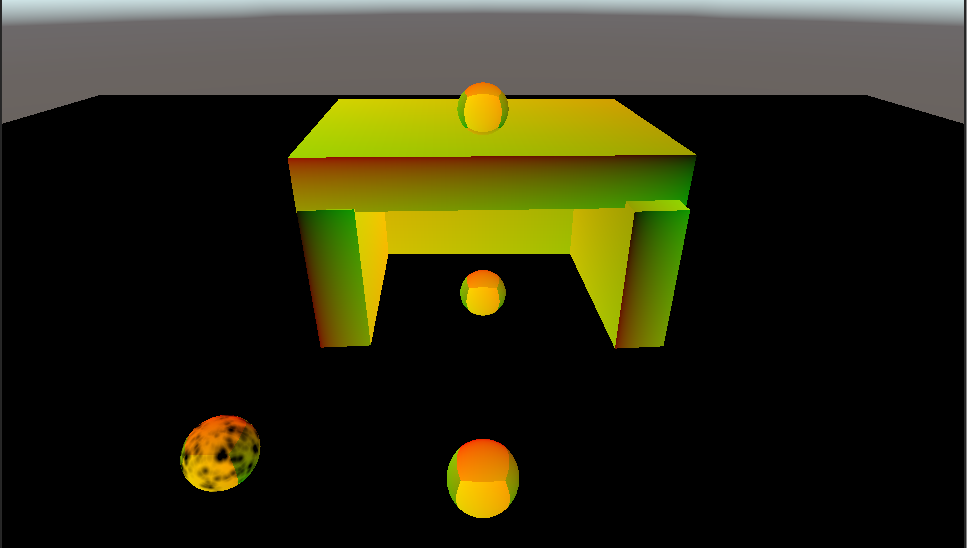
1. 采样 烘培灯光

2.1全局照明

全局灯光的高光是反射球是SSR Screen-space reflections 屏幕空间反射

2.2 LightMap 坐标

设置每个物体逐物体数据LightMaps灯光贴图DrawingSettings 里面进行设置



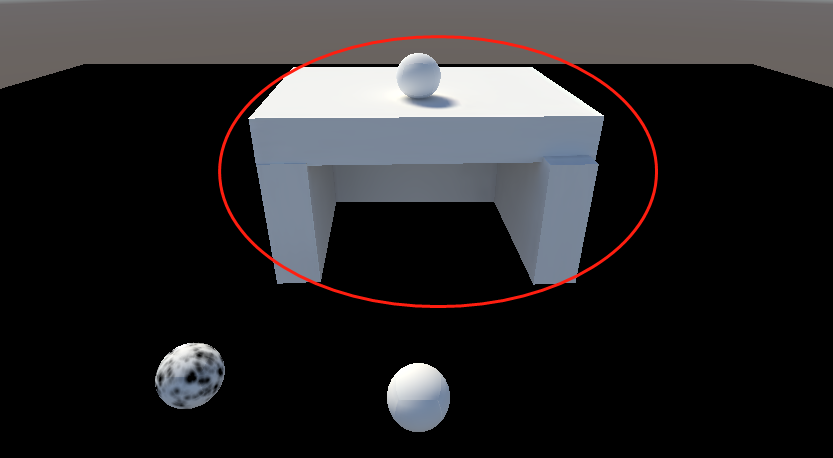
静态的物体会显示出UV 动态的物体 是黑色的没有UV信息

2.3 转换Light Map 坐标

Light Map UV坐标是每一个Mesh自动生成的或者导入网格数据的一部分。增加unityDynamicLightmapST 否则SRP Batch兼容性是会打断的。UnityPerDraw是在被需要的时候进行实例化。

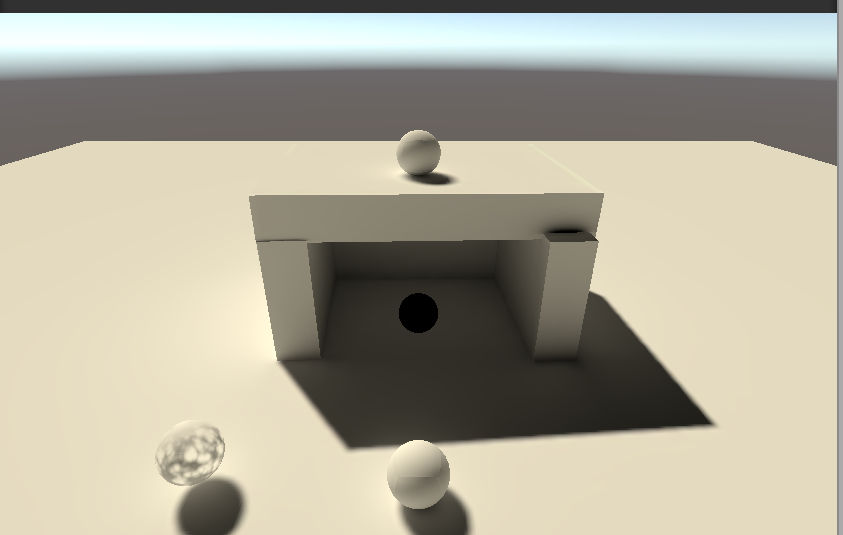
2.4 采样LightMap

采样成功LightMap

HDR不压缩 每次改变静态非静态重新Bake

2.5 禁用掉环境光

环境光强度设置为零 Bake出来就不包含天空球的间接光部分



1. 光探针

球谐光,L2球面谐波

* 1. Light 探针组

Bake之后才能看到动态物体受哪些光探针影响选中就可以,自动合并两个LightProbe区域。

* 1. 采样Probes

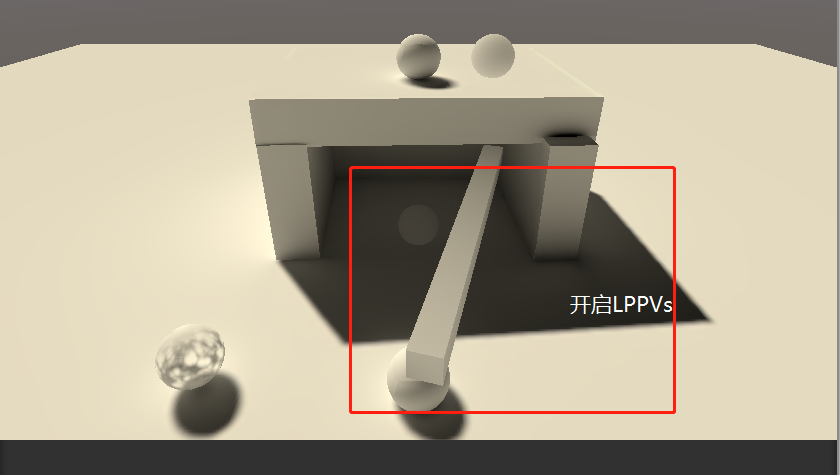
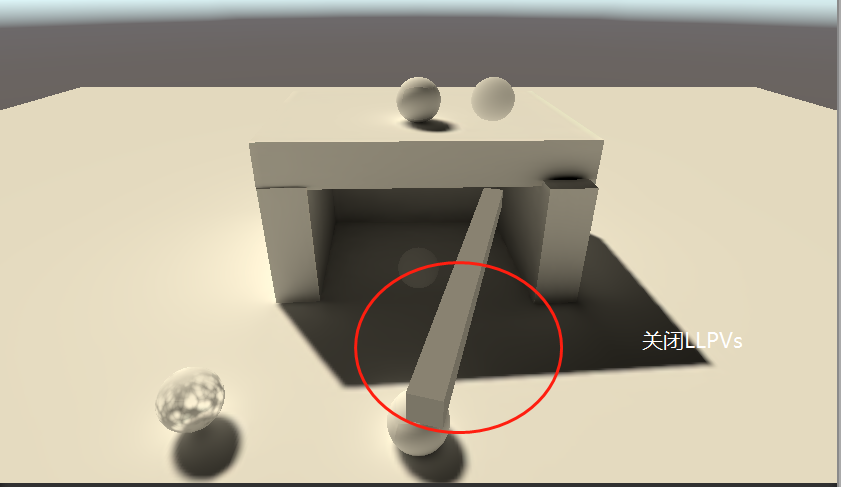
需要一个方向

* 1. LightProbe 体积

对大物体的支持，均匀黑暗是不正确的？在阴影的地方黑在外面应该间接光更亮一点才对。

* 1. 采样LPPVs

unity\_ProbeVolumeParams.x 进行判断的 世界坐标和法线信息进行采样



1. Meta Pass 干什么用的？ 是根据物体反射率反射到周围的颜色是怎么样的

Unity 烘培光的间接反射部分

4.1统一的输入

HLSLINCLUDE

4.2 Unlit

4.3 Meta 灯光模式 ？？？

4.4 lightMap 坐标

4.5 漫反射反射率

Mix灯光 实时光+间接光部分 细节更好

成功

5 自发光表面

一些表面他们自己会发光,然而这个影响会对Bake 灯光有贡献。

5.1 发光的光源

HDR配置

5.2烘焙自发光

自发光影响周围环境而且是静态的。

6 Bake 半透明

6.1 硬编码属性

6.2 复制属性

7 Mesh Ball

7.1 光探针

7.2 LPPV

动态物体的间接光部分 静态是LightMap 自发光也会成为一部分间接光部分

<https://blog.csdn.net/weixin_30412167/article/details/95799502> 解释MetaPass为全局光照提供反射率和自发光项

1. 静态阴影

Bake直接遮挡

Bake 静态阴影

结合实时光 Bake 阴影

Mix 实时光和Bake 阴影

支持四盏光的ShadowMask 光 支持更远的阴影