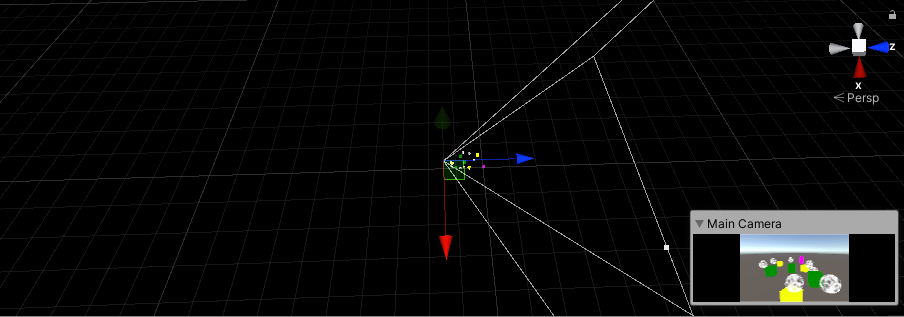
3.4:画Gimos

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

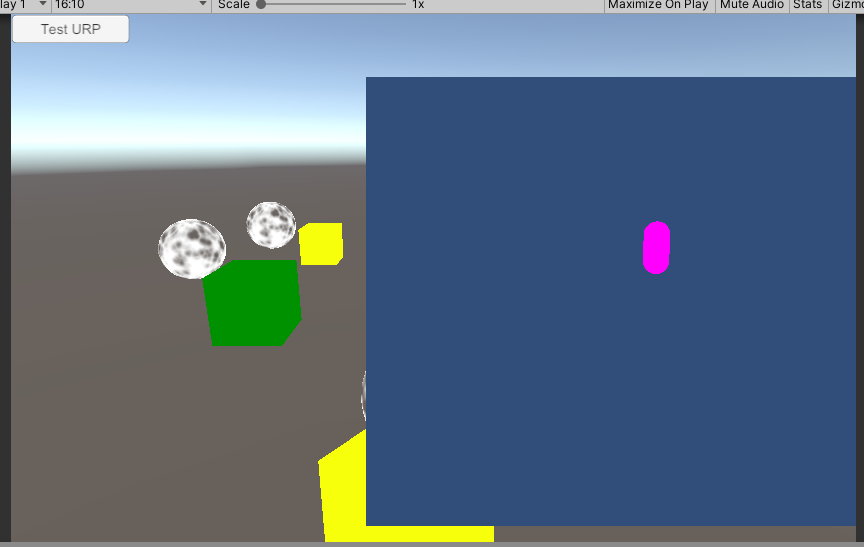
4.1 两个摄像机

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

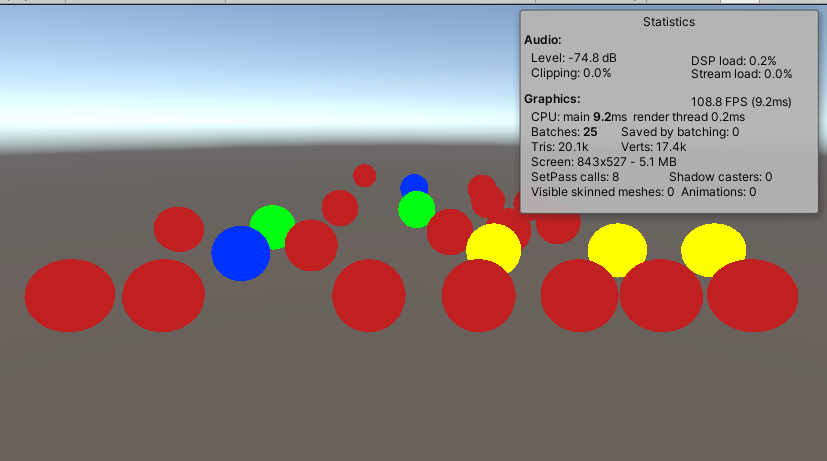
* 1. 空间转换
  2. 核心库 Common.HLSL SpaceTransform.HLSL
  3. 颜色

每一个材质球配置颜色\_BaseColor

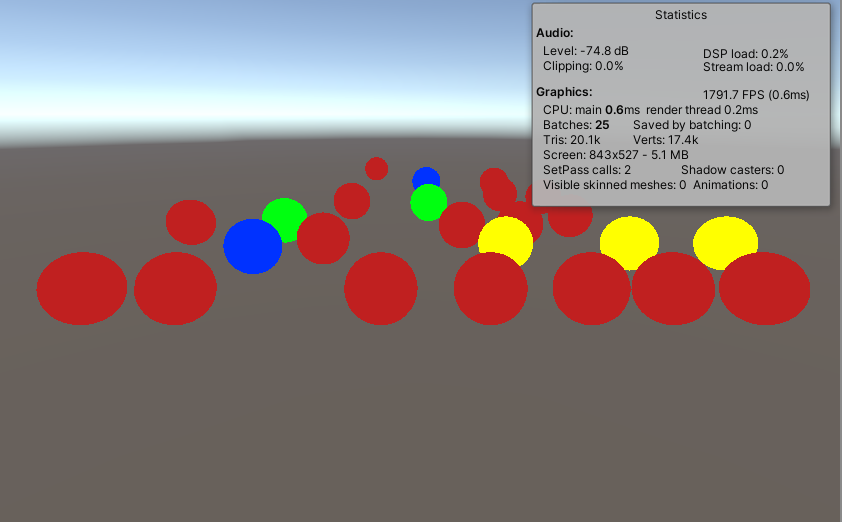
1. 合批
   1. SRP Batch

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

2.2 许多颜色