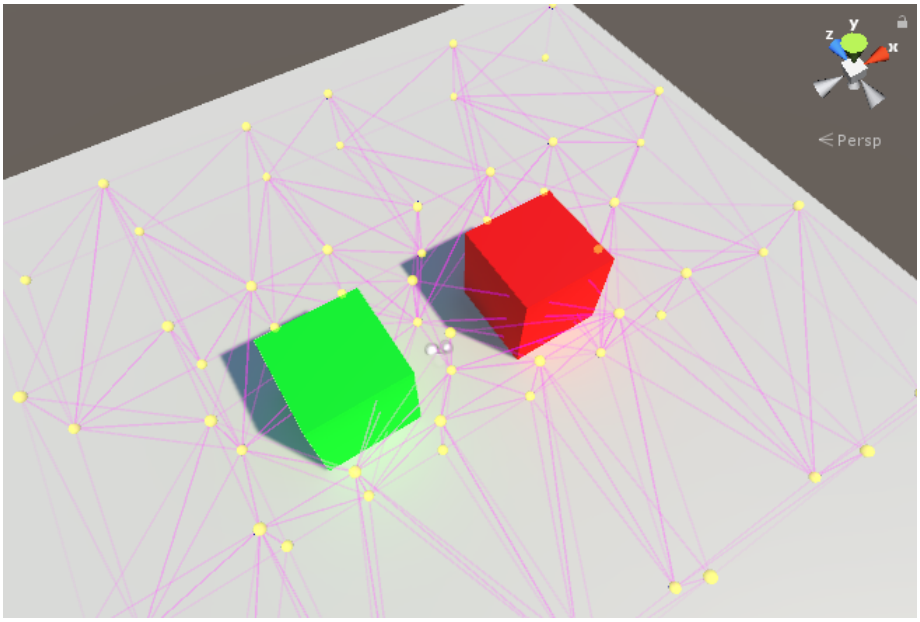
Light Probes

LightProbes (光照探针，光探测器？) 提供了一种方法用于捕获和使用 *穿过场景中空白空间的* 光**(light)**的信息。

和光照贴图相似(lightmaps)，Light Probes也存储关于场景中光照(lighting)的"baked"信息。两者的区别在于：lightmaps存储的是光线照射场景表面(surfaces)的光照信息，而light probes存储的是光线穿过场景空白空间的信息。

> *注：图形学中提到"bake"或"baked"或"烘焙"一般是指把场景中的光照等信息提前存储到纹理中，将来渲染时，使用这些纹理来达到"光照"效果。*



上图显示的是一个只有两个Cubes的场景，在两个Cubes周围旋转了放一些light probes(图中黄色的小球)。

Light Probes有两个主要的用途：

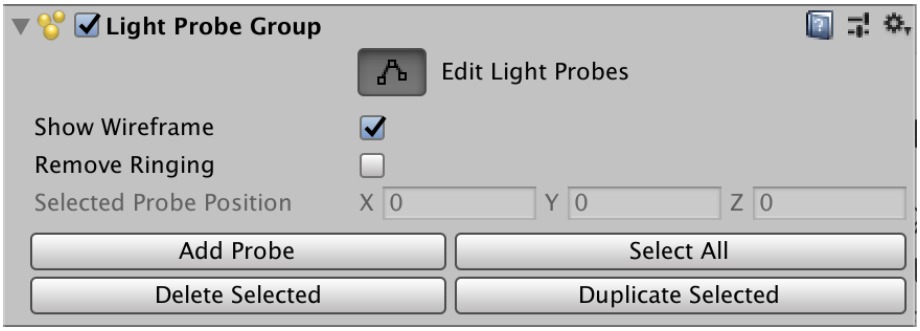
- 首要用途是为场景中运动的对象(moving objects)提供高质量的光照(包括间接反射的光，indirect bounced light)
- 次要用途是为使用了Unity的LOD system的静态场景提供光照信息。

> *注：LOD即Level of Detail，当场景中的GameObject离Camera很远时，已经无法看到许多细节了，此时可以使用LOD优化渲染，即当GameObject离Camera很远时减少GameObject的Mesh的三角面片数量。*

Light Probe Groups

为了把Light Probes放到场景中，必须使用一个带有**Light Probe Group**组件的GameObject。可以通过菜单:**Component >Rendering>Light Probe Group**来添加**Light Probe Group**组件。

但更好的做法是在场景中新建一个empty GameObject(通过菜单:**GameObject > Create Empty**)，然后把Light Probe Group组件添加到其上面。这样可以减少意外删除该组件的可能性。



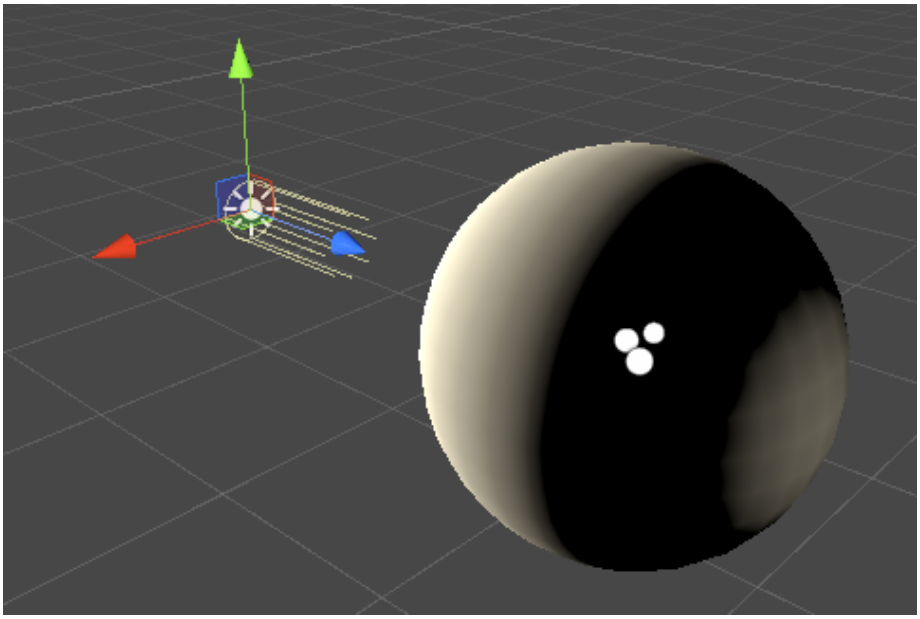
上图是Light Probe Group组件。

属性	功能
Edit Light Probes	点击此按钮，可以编辑Light Probe Group。这样使得Unity会只允许移动和编辑LightProbes。故编辑完一点要Distable此按钮。
Show Wireframe	此选项开启后，Unity在场景视图中显示连接Light Probes的线(wireframe)。关闭时，只显示代表Light Probe的点(points)
Remove Ringing	此选项开户后，Unity自动去除场景中的Light Probe ringing(见下文)
Selected Probe Postion	这是只读项。显示选中的Light Probe的x,y和z坐
Add Probe	点此按钮，添加一个Light Probe到Light Probe Group中
Select All	选中此Light Probe Group中所有的Light Probes
Delete Selected	从Light Probe Group中删除选中的Light Probes
Duplicate Selected	复制选中的Light Probes

Ringing

"Ringing"是light probes在某些环境下表现出来的不想要的行为。这经常发生在Light Probe周围光照有显著差异时。例如，若在一个Light Probe的一面有明亮光(bright light)，在另一面没有光(no light)，则在光强就会在背面“超调”(overshoot)，会在背面(本来是没光的)形成“光斑”(light spot)。

下图是一个Light Probe ringing的例子:



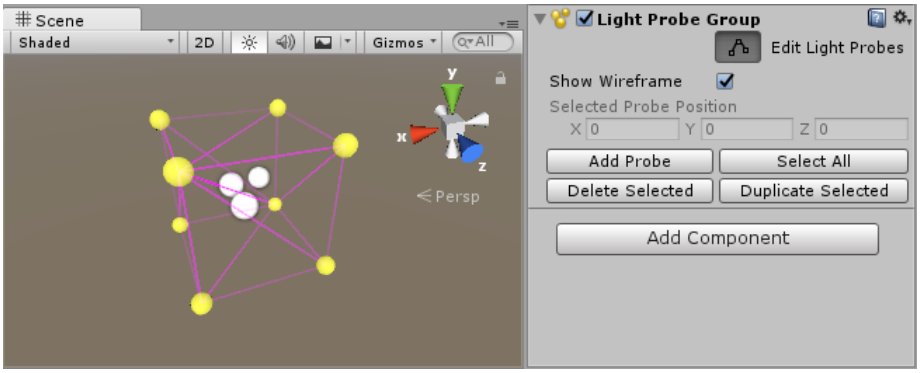
有几个方法可以解决"ringing":

1. 在Light Probe Group组件中，开启**Remove Ringing**。Unity会自动移除不想要的光斑。然而，这通常会造成Light Probes准确性下降，减少光的差异(reduces light contrast，应该是降低了不同的光之间的差异)。
2. 放置一些游戏障碍物，使得玩家无法到达能看到光斑的位置。
3. 避免baking有向光(direct light，或者叫平行光)到Light Probes中。Direct light倾向于有明显的不连续性(sharp discontinuities)(如，阴影边)，这使得它不适合用于Light Probes。仅烘焙间接光(indirect light)时，使用Mixed lighting。对于动态(dynamic)GameObjects 应该使用Realtime GI而不是Light Probes。

Placing Light Probes

在编辑一个Light Probe Group时，你可以用类似处理GameObjects的方式处理单个的Light Probes。然而，Light Probes不是GameObjects，它们是Light Probe Group组件中的一组点(points)。

当开始编辑一个新的Light Probe Group时，会有8个默认的light probes，排放在立方体的8个顶点上，如下图所示：



你可以使用 Light Probe Group inspector面板上的**Add Probe**控件添加 新的Light Probe。Light Probes在场景中呈现为黄色的小球，可以移动它们(就像移动GameObjects一样)。

Choosing Light Probe positions

Lightmaps在对象表面上具有连续的分辨率(resolution)(大概就是分布密度吧？)。与Lightmaps不同，LightProbes的信息分布情况可能是不连续的，这取决于你放置的Light Probes的位置的紧凑程度。

为了优化Light Probes存储的数据量和游戏运行时的计算量，你通常应该尽量减少使用的Light Probes的数量。然而，你又应该放置“足够多”的Light Probes，使得从一个空间到另一个空间的光照变化能够被记录一来，并且达到你要的要求。这意味着你可能需要在光线复杂，光照差异明显的地方多放置点Light Probes(即，在这种光线复杂的环境，应该以更紧凑的方式放置Light Probes)。在光照变化不大的地方，以更分散的方式放置Light probes。

 Light Probes placed with varying density around a simple Scene

如上图，建筑(房屋等)附近及之间放置的Probes比较密集，而在马路上就比较稀疏。

最简单的放置方式是按均匀三维网格的方式放置Light Probes(如，把它们放在网格的顶点)。这种方式很简、快捷，但是会消耗过多的、没必要的存储空间。如上图中，沿着马路，光照变化很少，许多相邻的Light Probes会存储相同的光照数据(lighting data)。在这种情况下，可以像上图那样，通过对少量、分散开来的light probes的lighting data进行插值处理。

参考：

1. Light Probes
2. Light Probe Groups