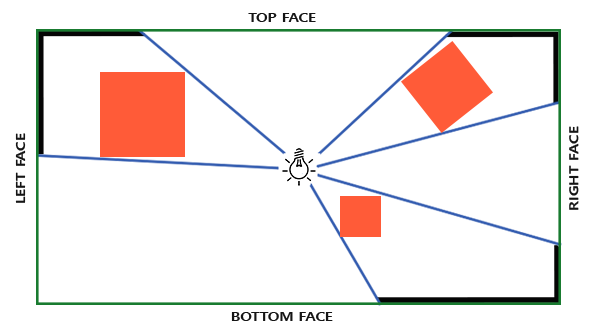
[**https://learnopengl-cn.github.io/05%20Advanced%20Lighting/03%20Shadows/02%20Point%20Shadows/**](https://learnopengl-cn.github.io/05%20Advanced%20Lighting/03%20Shadows/02%20Point%20Shadows/)

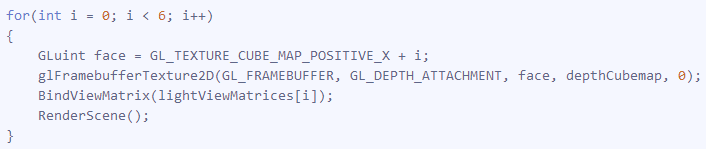
**点光源阴影**

点光源的阴影技术又叫万向贴图技术，算法和平行光定向阴影映射差不多，我们从光的透视图生成一个深度贴图，基于当前片元位置来对深度贴图采样，餐后用储存的深度值和每个片元进行比较，看看它是否在阴影中，万向阴影映射和定向阴影映射的主要不同在于深度贴图的使用上。

对于深度贴图，我们需要从一个点光源的所有渲染场景，普通2D深度贴图不能工作，我们使用立方体贴图。



生成后的深度立方体贴图被传递到光照像素着色器，它会用一个方向向量来采样立方体贴图，从而得到当前的fragment的深度（从光的透视图）为了创建一个光周围的深度值的立方体，我们必须渲染场景6次，每次一个面，显然渲染场景6次需要6个不同的视图矩阵，每次把一个不同的立方体贴图面附加到帧缓冲对象上。这看起来是这样的：

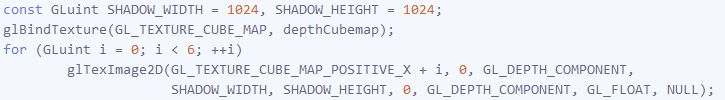


这会很耗费性能，因为一个深度贴图下需要进行很多渲染调用，这里我们使用几何着色器，几何着色器允许我们使用一次渲染过程来建立深度立方体贴图。

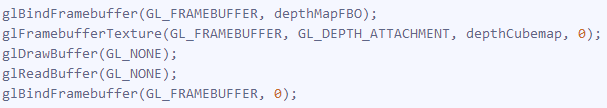
首先，我们需要创建一个立方体：



然后生成立方体贴图的每个面，将它们作为2D深度值纹理图像

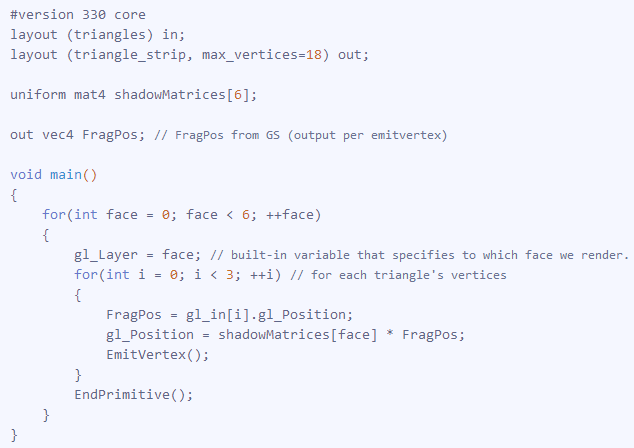


正常情况下，我们把立方体贴图纹理的一个面附加到帧缓冲对象上，渲染场景6次，每次将帧缓冲目标改成不同立方体贴图面，由于我们将使用一个几何着色器，它允许我们把所有面在一个过程渲染，，我们可以使用glFramebufferTexture直接把立方体贴图附加成帧缓冲的深度附件：



万向阴影贴图有两个渲染阶段：首先我们生成深度贴图，然后我们正常使用深度贴图渲染，在场景中创建阴影。

几何着色器：



几何着色器比较简单，我们输入一个三角形，输出总共6个三角形。