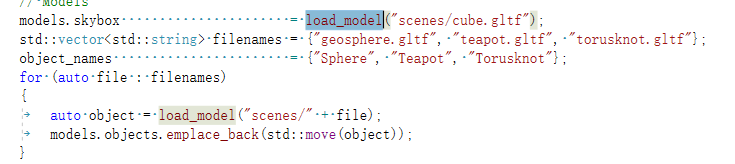
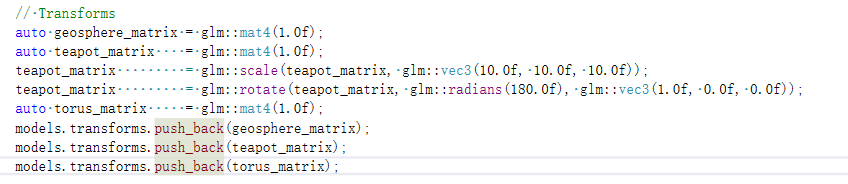
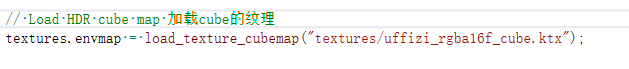
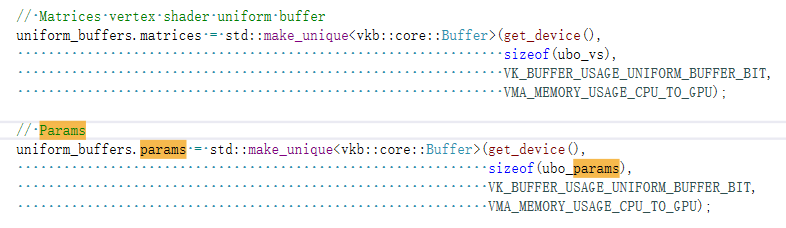
1.load\_assets

从 gltf文件中加载模型数据,一共4个分别是一组天空盒数据 和3个物体模型数据  
  


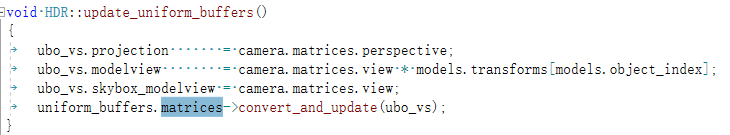
定义了3中物体模型的变换矩阵  
  
使用采样器加载纹理

2. prepare\_uniform\_buffers

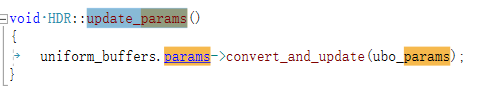
创建了两个一致变量缓冲区，分别是顶点数据和曝光参数



更新/初始化uniform\_buffer的对象缓冲区。填充视角矩阵的信息，并把数据写入一致缓冲区



把当前的曝光数据写入一致缓冲区



3. prepare\_offscreen\_buffer()

准备一个离屏缓冲区，呈现镜像场景<https://blog.csdn.net/qq_35312463/article/details/107424330>

4. setup\_descriptor\_set\_layout

为不同的pipeline构造不同的不同的描述集，怎么渲染不同的东西

5. prepare\_pipelines

准备渲染管线

6 setup\_descriptor\_pool

初始化描述集池，为分配描述集做准备

7. setup\_descriptor\_sets

分配描述集

8.

构建命令缓冲区，为最终的提交渲染做准备

9draw

VkResult result = render\_context->get\_swapchain().acquire\_next\_image(current\_buffer, semaphores.acquired\_image\_ready, VK\_NULL\_HANDLE);

获取交换链中的当前帧索引

VK\_CHECK(vkQueueSubmit(queue, 1, &submit\_info, VK\_NULL\_HANDLE));

提交命令缓冲

ApiVulkanSample::submit\_frame();

