张吉祺 16340286

Basic:

1. 解释Shadowing Mapping算法

Shadow Mapping：以光的位置为视角进行渲染，我们能看到的东西都将被点亮，看不见的认为在阴影之中。通过光的位置为视角下深度缓冲里对应于一个片元的一个0到1之间的深度值的结果储存到纹理中，对光源的透视图所见的最近的深度值进行采样，深度值就会显示从光源的透视图下见到的第一个片元，储存在纹理中的深度值叫做深度贴图（depth map）。使用一个来自光源的视图和投影矩阵结合在一起成为一个T变换将任何三维位置转变到光源的可见坐标空间来渲染场景创建深度贴图。然后像往常一样渲染场景，使用深度贴图来计算片元是否在阴影之中。

2. 实现方向光源的Shadowing Mapping

先生成深度贴图。`glGenFramebuffers(1, &depthMapFBO)`创建帧缓冲，然后`glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT,SHADOW_WIDTH, SHADOW_HEIGHT, 0, GL_DEPTH_COMPONENT, GL_FLOAT, NULL)`得到2D纹理深度，`glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT, GL_TEXTURE_2D, depthMap, 0)`将深度纹理作为帧缓冲的深度缓冲。

然后渲染深度贴图。通过投影矩阵`lightProjection`、`glm::lookAt`生成的视图矩阵`lightView`结合成光空间的变换矩阵`lightSpaceMatrix`，将每个世界空间坐标变换到光源空间的坐标渲染深度贴图。

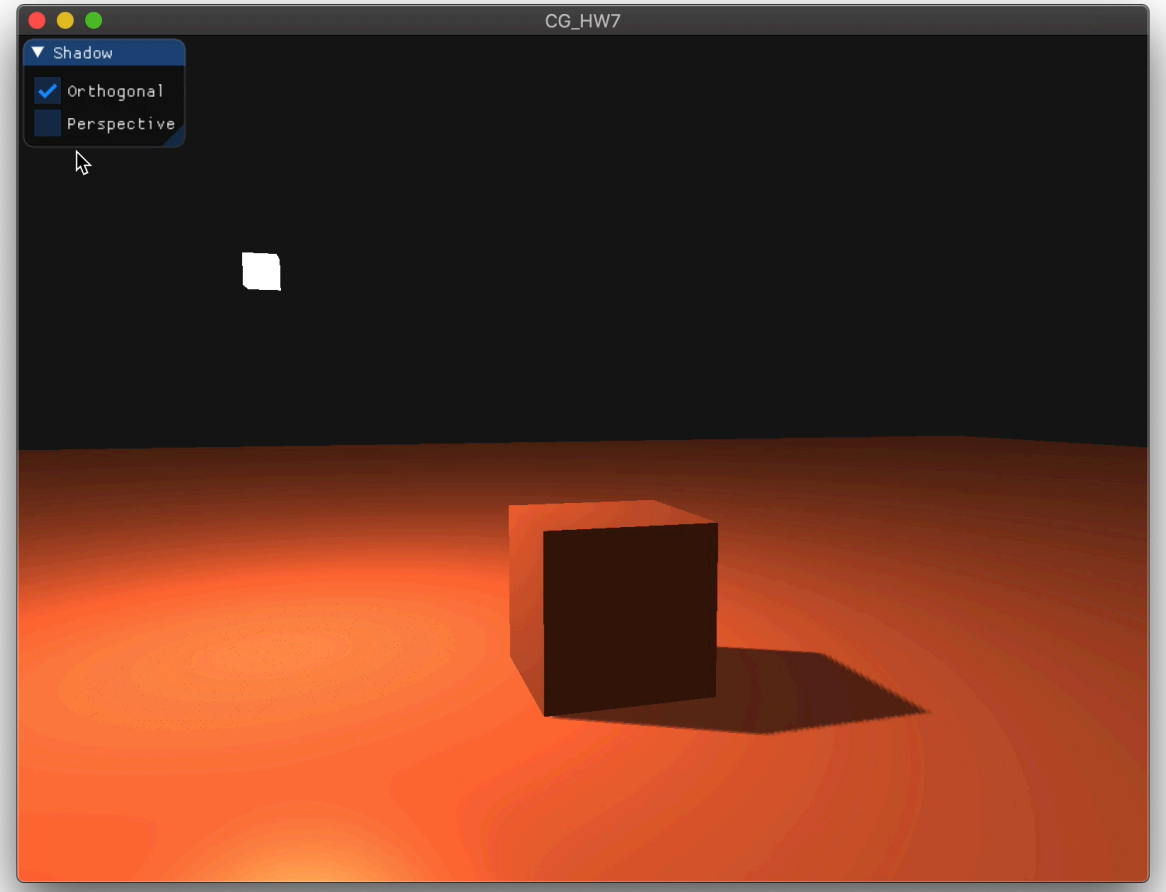
然后使用深度贴图生成阴影。在正常场景的片段着色器中传入片段在光源空间的位置`FragPosLightSpace`，并进行透视除法`projCoords = fragPosLightSpace.xyz / fragPosLightSpace.w`，再`projCoords = projCoords * 0.5 + 0.5`变换到`[0,1]`，再得到相应位置深度贴图的结果得到深度`closestDepth = texture(shadowMap, projCoords.xy).r`，用当前`currentDepth = projCoords.z`与其对比确定片段是否在阴影中`shadow = currentDepth > closestDepth ? 1.0 : 0.0`，将`shadow`加入光照模型`(ambient + (1.0 - shadow) * (diffuse + specular)) * color`渲染出阴影。

张吉祺 16340286

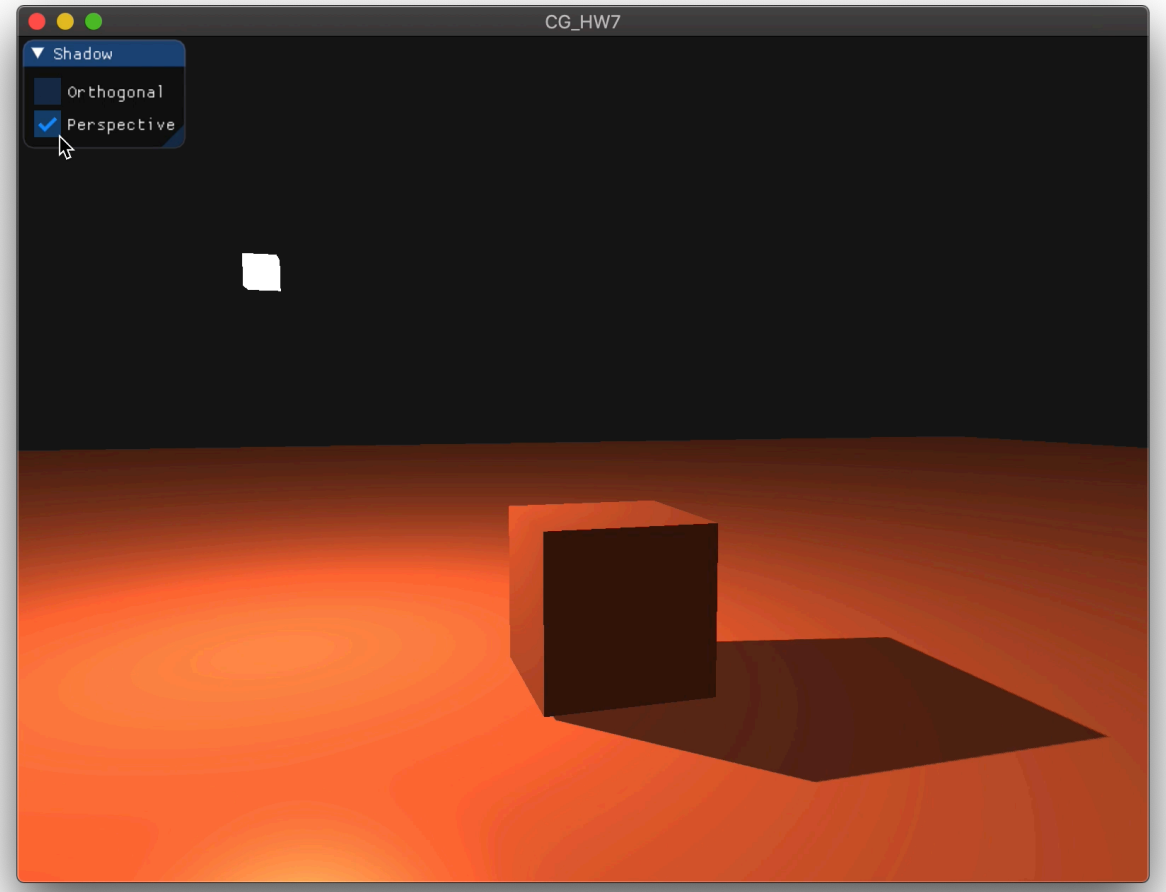
Bonus:

1. 实现光源在正交/透视两种投影下的Shadowing Mapping

在渲染深度贴图时，指定投影矩阵`lightProjection`为正交
`lightProjection = glm::ortho(-10.0f, 10.0f, -10.0f, 10.0f, near_plane, far_plane)`



或透视`lightProjection = glm::perspective(45.0f, (GLfloat)SHADOW_WIDTH / (GLfloat)SHADOW_HEIGHT, near_plane, far_plane)`。



2. 优化Shadowing Mapping

(1) 阴影偏移

因为阴影贴图受限于解析度，多个片元从同一个深度值进行采样，当光源以一个角度朝向表面的时候会出问题，多个片元就会从同一个斜坡的深度纹理像素中采样，有些在地板上面，有些在地板下面，有些片元被认为是在阴影之中，有些不在，产生条纹样式。

简单的对深度贴图根据表面朝向光线的角度应用一个偏移量`bias = max(0.05 * (1.0 - dot(normal, lightDir)), 0.005)`，`shadow = currentDepth - bias > closestDepth  ? 1.0 : 0.0`，这样片元就不会被错误地认为在表面之下了。为了避免悬浮，渲染深度贴图时使用正面剔除`glCullFace(GL_FRONT)`。

(2) 采样过多

光的视锥不可见的区域（超出深度贴图的大小）一律被认为是处于阴影中。

解决方法是储存一个边框颜色，然后把深度贴图的纹理环绕选项设置为`GL_CLAMP_TO_BORDER`，让超出深度贴图的坐标的深度范围是1.0，这样超出

的坐标将永远不在阴影之中。

而当一个点比光的远平面还要远时，它的投影坐标的z坐标大于1.0，这种情况下，GL_CLAMP_TO_BORDER环绕方式不起作用，因为把坐标的z元素和深度贴图的值进行了对比，它总是为大于1.0的z返回true，可以在投影向量的z坐标大于1.0时把shadow的值强制设为0.0解决这个问题。

(3) PCF

深度贴图有一个固定的解析度，多个片元对应于一个纹理像素，从同一个深度值进行采样得到同一个阴影，会产生锯齿边。

一个解决方法是从深度贴图中多次采样，每一次采样的纹理坐标都稍有不同，每个独立的样本可能在也可能不在阴影中，再将采样平均化得到柔和阴影。

一个简单的PCF的实现是简单的从纹理像素四周对深度贴图采样，然后把结果平均起来。