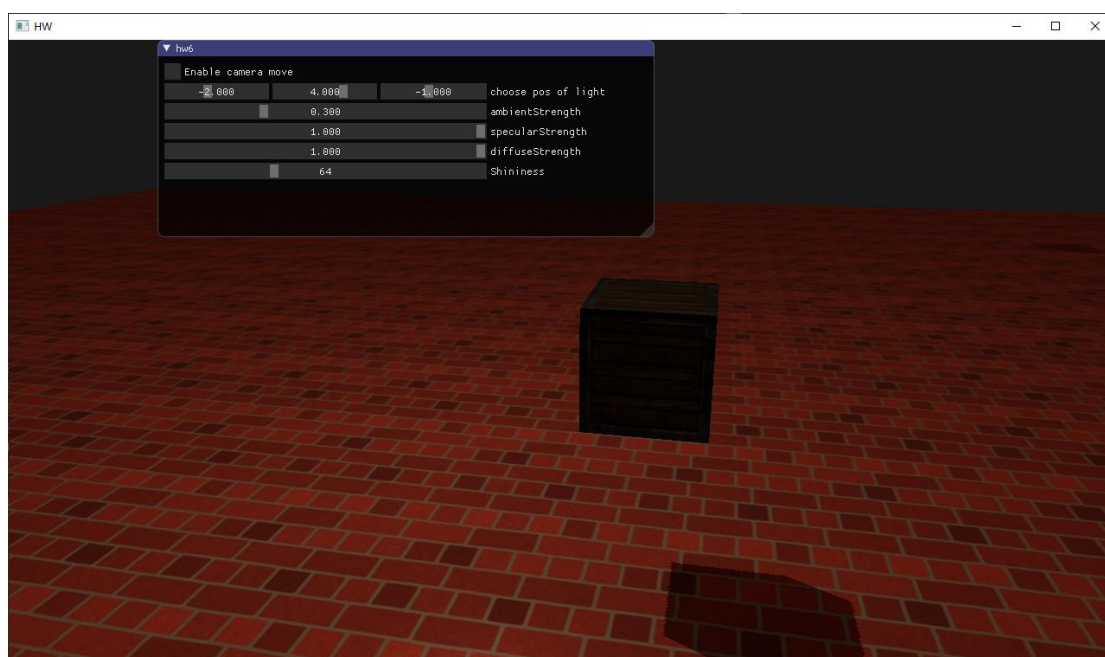


HW7

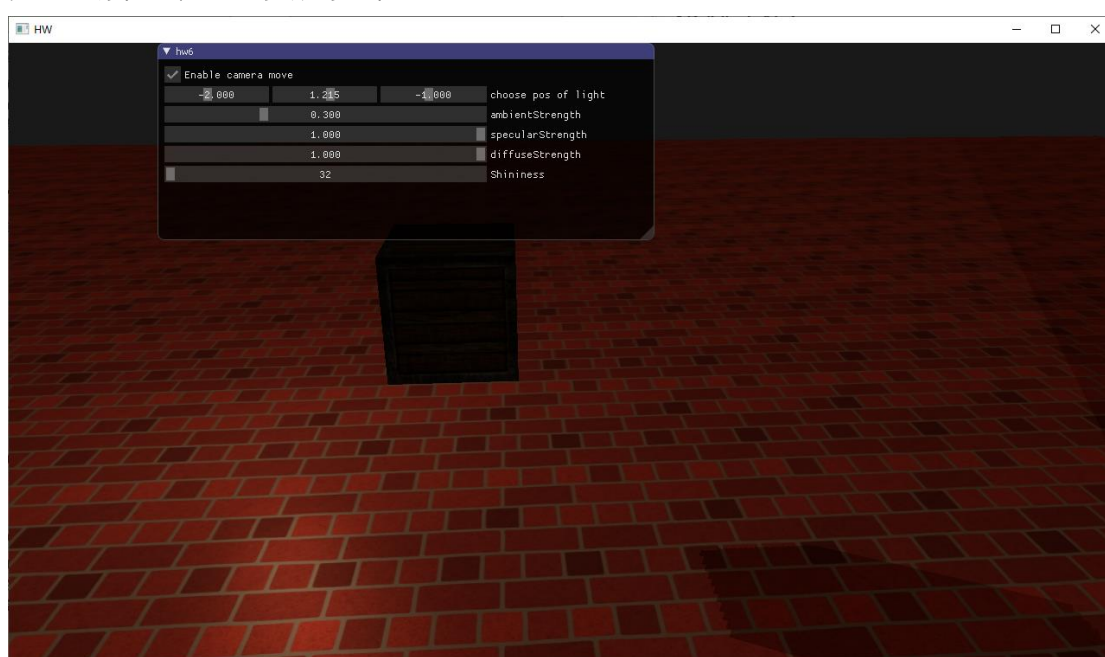
胡嘉鹏 16340076

Basic 1 & 2

使用正交投影实现 shadowing mapping



更改光源位置，光照参数等属性



shadowing mapping 算法解释:

简单来说就是: 1. 求出所有点在灯光视角下的深度信息。2. 在同一个位置, 将所有深度值大于最小深度值的点的漫反射和镜面反射置为 0。

将物体在灯光坐标轴系统下的深度信息存储在一个 2D 的 texture 中

```
// 准备帧缓冲和深度纹理
const unsigned int SHADOW_WIDTH = 1024, SHADOW_HEIGHT = 1024;
unsigned int depthMapFBO;
glGenFramebuffers(1, &depthMapFBO);
unsigned int depthMap;
glGenTextures(1, &depthMap);
glBindTexture(GL_TEXTURE_2D, depthMap);
glTexImage2D(GL_TEXTURE_2D, 0, GL_DEPTH_COMPONENT, SHADOW_WIDTH,
SHADOW_HEIGHT, 0, GL_DEPTH_COMPONENT, GL_FLOAT, NULL);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glBindFramebuffer(GL_FRAMEBUFFER, depthMapFBO);
// 深度纹理作为帧缓冲的深度缓冲
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
GL_TEXTURE_2D, depthMap, 0);
glDrawBuffer(GL_NONE);
glReadBuffer(GL_NONE);
glBindFramebuffer(GL_FRAMEBUFFER, 0);
```

```
// 在light的视角坐标下渲染
lightProjection = glm::ortho(-10.0f, 10.0f, -10.0f, 10.0f,
near_plane, far_plane);
lightView = glm::lookAt(lightPos, glm::vec3(0.0f), glm::vec3(0.0,
1.0, 0.0));
lightSpaceMatrix = lightProjection * lightView;
simpleDepthShader.use();
simpleDepthShader.setMat4("lightSpaceMatrix", lightSpaceMatrix);
```

```
#version 330 core
layout (location = 0) in vec3 aPos;

uniform mat4 lightSpaceMatrix;
uniform mat4 model;
```

```

void main()
{
    // 着色器只将顶点变换到光空间中
    gl_Position = lightSpaceMatrix * model * vec4(aPos, 1.0);
}

```

经过上面代码的过程，纹理 depthMap 已经保存了顶点在光空间下的深度信息。

在光空间下将渲染顶点的深度值和该位置最小的深度值比较

```

...
// 传入深度纹理
uniform sampler2D shadowMap;
...
float ShadowCalculation(vec4 fragPosLightSpace)
{
    ...
    // 将当前渲染点的深度值和最近深度值比较
    float shadow = currentDepth - bias > closestDepth ? 1.0 : 0.0;
    return shadow;
}
...
// 如果当前点在阴影中，漫反射和镜面反射就为0
vec3 lighting = (ambient + (1.0 - shadow) * (diffuse + specular)) * color;

```

Bonus

优化 shadowing mapping

因为阴影贴图的解析度有限，光源以一个角度朝向表面的时候，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，得到的阴影就有了差异。

对相对表面深度应用一个偏移量来解决：

```

float bias = max(0.05 * (1.0 - dot(normal, lightDir)), 0.005);
float shadow = currentDepth - bias > closestDepth ? 1.0 : 0.0;

```