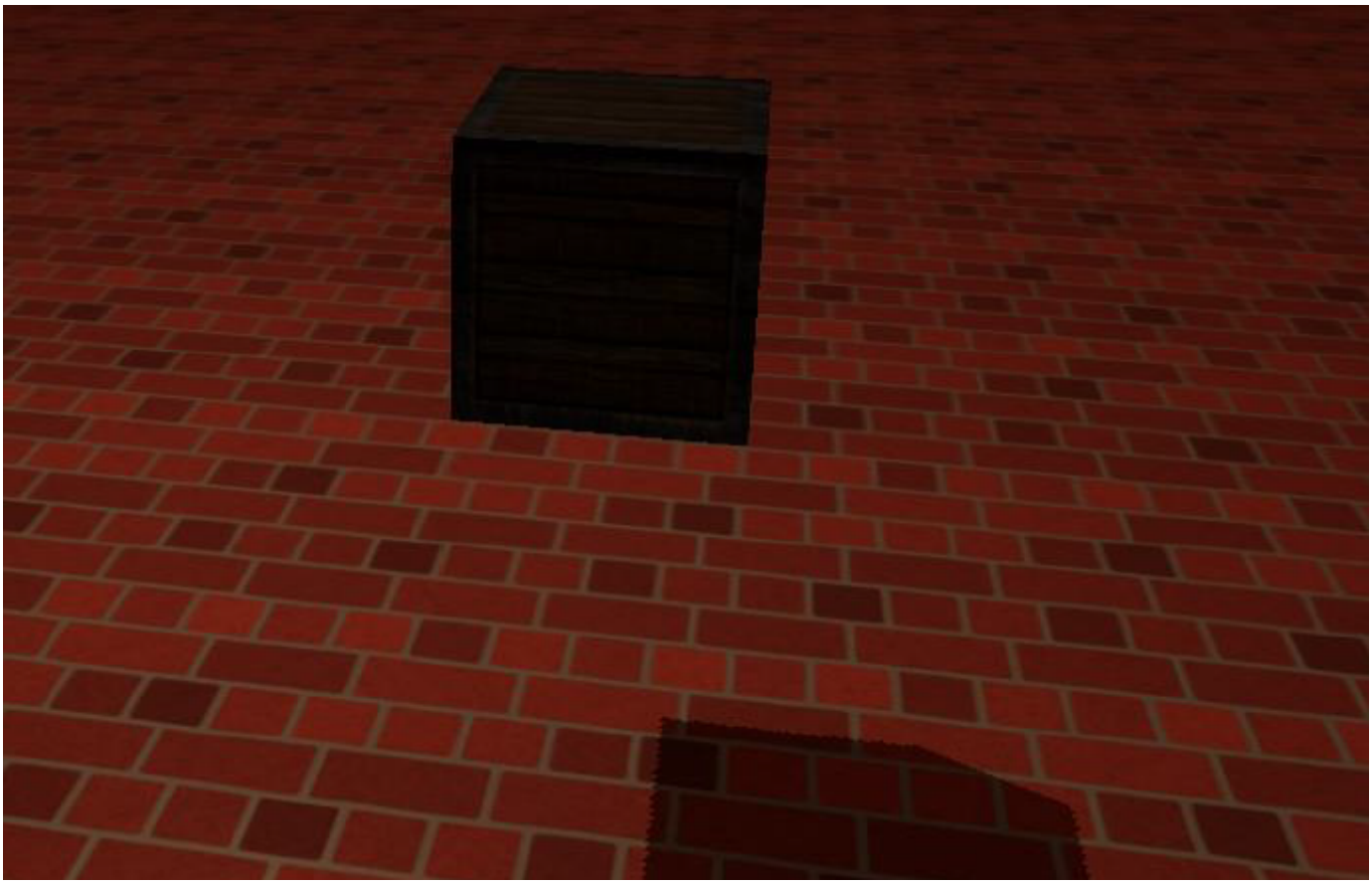
hw7

16340272 杨淼

Basic

shadowing mapping



```
vec3 ambient = ambientStrength * lightColor;
```

物体在灯光坐标轴系统下的深度信息存储在一个2D的texture

```
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);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
GL_TEXTURE_2D, depthMap, 0);
glDrawBuffer(GL_NONE);
glReadBuffer(GL_NONE);
glBindFramebuffer(GL_FRAMEBUFFER, 0);
```

```
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);
```

光空间下将渲染顶点的的深度值和该位置最小的深度值比较

```
 vuniform sampler2D shadowMap;
...
float ShadowCalculation(vec4 fragPosLightSpace)
{
...

float shadow = currentDepth - bias > closestDepth ? 1.0 : 0.0;
return shadow;
}
...

vec3 lighting = (ambient + (1.0 - shadow) * (diffuse + specular)) *
color;
```