# BASIC：

**动态效果见附视频文件hw7.mp4。**

1. 实现方向光源的Shadowing Mapping:

* 要求场景中至少有一个object和一块平面(用于显示shadow)

一块平面加三个cube。

* 光源的投影方式任选其一即可

实现的是光源的正交投影

* 在报告里结合代码，解释Shadowing Mapping算法

阴影映射的主要实现思路是：首先以光的位置为视角进行渲染，此时能看到的部分都将被点亮，看不到的部分处于阴影中。此时把深度缓冲中的度值（摄像机视角下，对应于一个fragment的一个0到1之间的值表示深度）储存到纹理中，从而对光源的透视图所见的最近的深度值进行采样，表示从光源的透视图下见到的第一个fragment，生成的纹理叫深度贴图。之后渲染物体场景的时候，把每个点变换到光源空间与深度贴图比较深度，从而决定这个点是否在阴影内。

1. 创建深度缓冲区并绑定

const GLuint SHADOW\_WIDTH = 1024, SHADOW\_HEIGHT = 1024;

static GLuint depthMapFBO, depthMap;

if (depthMapFBO == 0 && depthMap == 0)

{

glGenFramebuffers(1, &depthMapFBO);

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);

}

1. 从光源角度进行渲染

simpleDepthShader.use();

lightProjection = glm::ortho(-10.0f, 10.0f, -10.0f, 10.0f, 1.0f, 7.5f);

lightView = glm::lookAt(lightPos, glm::vec3(0.0f), glm::vec3(0.0, 1.0, 0.0));

lightSpaceMatrix = lightProjection \* lightView;

‘’’

Set uniforms

‘’’

glViewport(0, 0, SHADOW\_WIDTH, SHADOW\_HEIGHT);

glBindFramebuffer(GL\_FRAMEBUFFER, depthMapFBO);

glClear(GL\_DEPTH\_BUFFER\_BIT);

RenderScene(simpleDepthShader);

glBindFramebuffer(GL\_FRAMEBUFFER, 0);

1. 场景渲染

glViewport(0, 0, SCR\_WIDTH, SCR\_HEIGHT);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

or\_shader.use();

projection = glm::perspective(camera.Zoom, (float)SCR\_WIDTH / (float)SCR\_HEIGHT, 0.1f, 100.0f);

view = camera.GetViewMatrix();

‘’’

Set uniforms

‘’’

glActiveTexture(GL\_TEXTURE0);

glBindTexture(GL\_TEXTURE\_2D, boxTexture);

glActiveTexture(GL\_TEXTURE1);

glBindTexture(GL\_TEXTURE\_2D, depthMap);

RenderScene(or\_shader);

1. 各顶点着色器代码实现，见代码文件hw7.cpp
2. 修改GUI

可以修改ambient, diffuse, specular因子，以及显示优化阴影的效果

# Bonus

1. 实现光源在正交/透视两种投影下的Shadowing Mapping

没有实现光源透视投影下的Shadowing Mapping

1. 优化Shadowing Mapping (可结合References链接，或其他方法。优化方式越多越好，在报告里说明，有加分)

* 使用了阴影偏移量来改善阴影失真的情况
* 检查远平面，并将深度贴图限制为一个手工指定的边界颜色，改善深度贴图采样超出的问题
* 使用PCF去锯齿化

# 代码

附代码文件（本次作业的核心代码在hw7.cpp文件内）