

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。

a) 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);
}

b) 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);

c) 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);

d) 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
2. 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）