**实 验 报 告 三**

**（2021-2022学年第一学期）**

**计算机图形学**

**（Computer Graphics）**

|  |  |  |
| --- | --- | --- |
| 学生姓名：陈卓文 | 学号：201936380215 | 年级，班级：19级2班 |
| 成绩： | 任课教师签名： | 日期：2021/10/8 |
| 实验题目：OpenGL中的模型纹理贴图 | | |
| 目录：   1. 实验内容描述，使用模型定义顶点的属性信息：纹理映射坐标 2. 实验功能算法描述，即纹理材质是如何建立、如何载入、如何使用的 3. 实验shader程序描述，即vertex shader和fragment shader的程序代码及说明 4. 实验结果，有光照和纹理贴图的模型实验结果图 5. 心得体会 | | |
| 实验报告：   1. 实验内容描述，使用模型定义顶点的属性信息：纹理映射坐标   本次实验中使用的模型为Three.js中内置的Utah teapot模型，模型自带生成的纹理映射坐标。  相关代码详见：  https://github.com/mrdoob/three.js/blob/dev/examples/jsm/geometries/TeapotGeometry.js   1. 实验功能算法描述，即纹理材质是如何建立、如何载入、如何使用的   本次实验使用的纹理下载自互联网，通过Three.js内置的TextureLoader载入，通过GLSL的uniforms传入shader。   1. 实验shader程序描述，即vertex shader和fragment shader的程序代码及说明   vertex shader：  varying vec3 Normal;  varying vec3 Position;  varying vec2 vUv;  void main() {  Normal = normalize(normalMatrix \* normal);  Position = vec3(modelViewMatrix \* vec4(position, 1.0));  vUv = uv;  gl\_Position = projectionMatrix \* modelViewMatrix \* vec4(position, 1.0);  }  fragment shader:  varying vec3 Normal;  varying vec3 Position;  varying vec2 vUv;  uniform vec3 Ka;  uniform vec3 Kd;  uniform vec3 Ks;  uniform vec4 LightPosition;  uniform vec3 LightIntensity;  uniform float Shininess;  uniform sampler2D tt;  vec3 phong() {  vec3 n = normalize(Normal);  vec3 s = normalize(vec3(LightPosition) - Position);  vec3 v = normalize(vec3(-Position));  vec3 r = reflect(-s, n);  vec3 ambient = Ka;  vec3 diffuse = Kd \* max(dot(s, n), 0.0);  vec3 specular = Ks \* pow(max(dot(r, v), 0.0), Shininess);  return LightIntensity \* (ambient + diffuse + specular);  }  void main() {  gl\_FragColor = texture2D(tt, vUv) + vec4(phong(), 1.0);  }  Shader中与纹理相关的代码有：  vs中通过vUv保存uv并传至fs。  fs中通过texture2D(tt, vUv) 计算当前点的贴图颜色。   1. 实验结果，有光照和纹理贴图的模型实验结果图      1. 心得体会   本次实验中，我学会如何通过shader进行贴图的渲染，学会了纹理贴图坐标的相关知识。 | | |