

计算机图形学

Assignment 6 Lighting and shading

16340063 巩泽群

本次完成的是光照模型，首先完成的是Phong光照模型，然后还完成了Gouraud光照模型。

算法原理

Phong 光照模型

Phong模型采用了三种不同的光照合成为最终的光照，分别是环境光，漫反射，镜面反射三个部分，如下

```
1 | vec3 result = (ambient + diffuse + specular) * objectColor;
```

环境光就是简单的给物体加上一定的亮度，让物体不是处于完全的黑暗中，非常简单，给物体本来的颜色乘上一个因子就好了。

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

漫反射则需要一定的计算，首先需要物体的法向量，我们通过顶点矩阵直接输入物体法向量的值。

```
1 | std::vector<float> cubeVertices {  
2 |     // 顶点位置           法向量方向  
3 |     -0.5f, -0.5f, -0.5f,  0.0f,  0.0f, -1.0f,  
4 |     ...  
5 | }
```

读入法向量之后，首先在顶点着色器里进行处理，将Model去掉位移的部分之后，得到仅含方向的法向量。

```
1 | Normal = mat3(transpose(inverse(model))) * aNormal;
```

在片段着色器中使用 `normalize(Normal)` 处理，然后使用 `dot` 函数计算出漫反射的因子的大小，然后计算出 `diffuse` 得到漫反射的光照。

```
1 | // diffuse  
2 | vec3 norm = normalize(Normal);  
3 | vec3 lightDir = normalize(lightPos - FragPos);  
4 | float diff = max(dot(norm, lightDir), diffuseStrength);  
5 | vec3 diffuse = diff * lightColor;
```

镜面反射也需要一定的计算，得到通过 `pow` 函数计算得到镜面反射的强度因子。

```
1 // specular
2 vec3 viewDir = normalize(viewPos - FragPos);
3 vec3 reflectDir = reflect(-lightDir, norm);
4
5 float spec = pow(max(dot(viewDir, reflectDir), 0.0), 32);
6 vec3 specular = specularStrength * spec * lightColor;
7
8 vec3 result = (ambient + diffuse + specular) * objectColor;
```

Gouraud 光照模型

Gouraud模型只需要简单的将 Phong 光照模型中放在片段着色器里的部分放在顶点着色器里就好了。