

# 计算机图形学第六次作业

## 一、Phong Shading

Phong Shading 由三个部分组成，分别是环境光照（Ambient）、漫反射光照（Diffuse）和镜面光照（Specular）。

1、环境光照：使用一个很小的常量（光照）颜色，添加到物体片段的最终颜色中，使得没有直接的光源也能看起来存在一些发散的光。

```
" vec3 ambient = ambientStrength * lightColor;\n"
```

2、漫反射光照：漫反射光照使物体上与光线方向越接近的片段能从光源处获得更多的亮度。添加一个法向量，用来计算光线照射物体表面的角度，角度越大，光对该片段的影响就越小。

```
" vec3 norm = normalize(Normal);\n" vec3 lightDir = normalize(lightPos - FragPos);\n" float diff = max(dot(norm, lightDir), 0.0);\n" vec3 diffuse = diffStrength * diff * lightColor;\n"
```

3、镜面光照：根据观察角度的不同，镜面反射光照有不同的结果。因此，我们需要将摄像机的位置添加到片段着色器中。

```
" vec3 viewDir = normalize(viewPos - FragPos);\n" vec3 reflectDir = reflect(-lightDir, norm);\n" float spec = pow(max(dot(viewDir, reflectDir), 0.0), reflectance);\n" vec3 specular = specularStrength * spec * lightColor;\n"
```

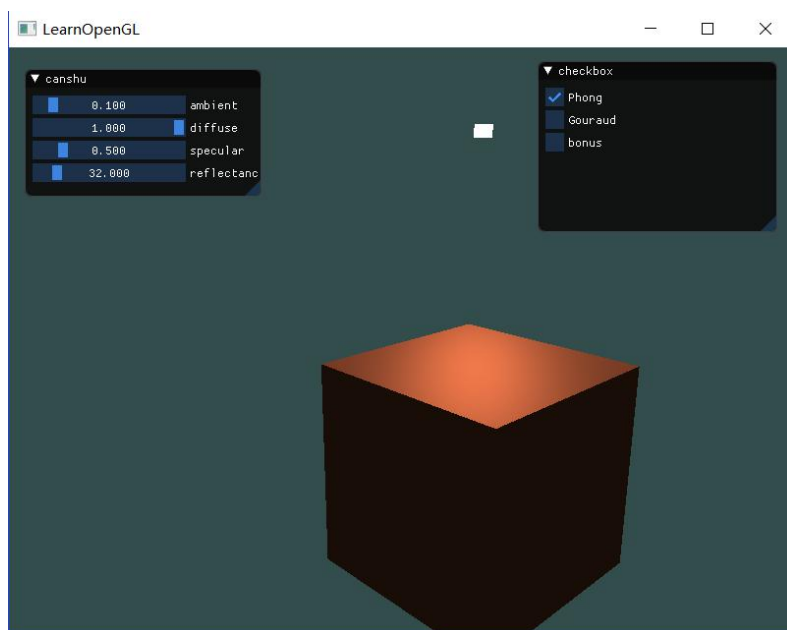
4、将三个光照分量相加，用结果乘以物体的颜色。

```
" vec3 result = (ambient + diffuse + specular) * objectColor;\n" FragColor = vec4(result, 1.0);\n"
```

5、添加 GUI，调节各个参数。

```
ImGui::Begin("canshu");  
ImGui::SliderFloat("ambient", (float*)&ambientStrength, 0.0f, 1.0f);  
ImGui::SliderFloat("diffuse", (float*)&diffStrength, 0.0f, 1.0f);  
ImGui::SliderFloat("specular", (float*)&specularStrength, 0.0f, 3.0f);  
ImGui::SliderFloat("reflectance", (float*)&reflectance, 0.0f, 256.0f);  
ImGui::End();
```

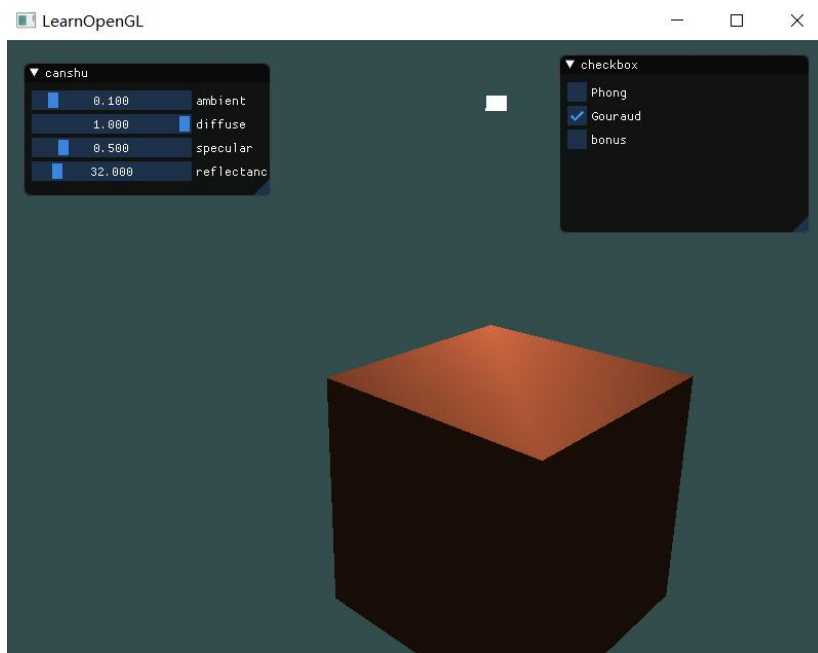
6、运行效果



## 二、Gouraud Shading

1、Gouraud 光照模型与 Phong 光照模型类似，也是由环境光照、漫反射光照、镜面光照三个分量组成。区别在于，Gouraud 光照模型的计算过程放在了顶点着色器中，在顶点着色器中计算出了颜色值，再传进片段着色器中，这种方法效率较高，因为对每个顶点只做一次计算，但是根据像素周围顶点来插值计算像素颜色时，会导致不平滑。而 Phong 光照模型是在片段着色器中基于每个像素的光照来计算，效果连续而平滑。

### 2、运行效果



与 Phong 光照模型的结果对比，可以看到，上图的正方体上表面的对角线有明显的平滑情况，对角线恰好就是绘制正方体时两个三角形的分界线。

## 三、Bonus

添加两行代码，让光源位置发生改变

```
if (bonus)
{
    lightPos.x = 1.0f + sin glfwGetTime() * 2.0f;
    lightPos.y = sin glfwGetTime() / 2.0f * 1.0f;
}
```