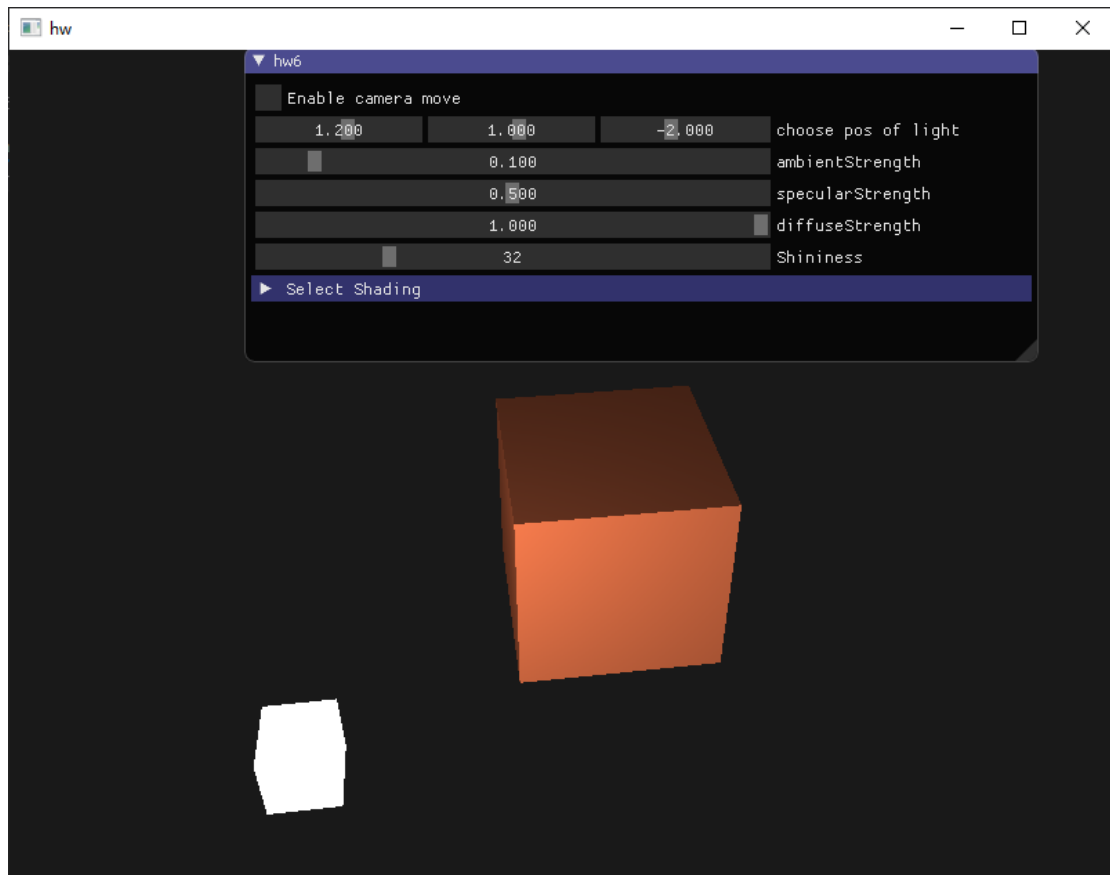


HW6

胡嘉鹏 16340076

实现 Phong Shading



实现原理

在片段着色器中实现：

1. 环境光：

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

光照乘以一个环境光系数

2. 漫反射

```
vec3 norm = normalize(Normal);
```

```
vec3 lightDir = normalize(lightPos - FragPos);
```

```
float diff = max(dot(norm, lightDir), 0.0);
```

```
vec3 diffuse = diffuseStrength * diff * lightColor;
```

在世界坐标系下入射光和片段的法线的余弦值乘以光照和漫反射系数。

3. 镜面反射

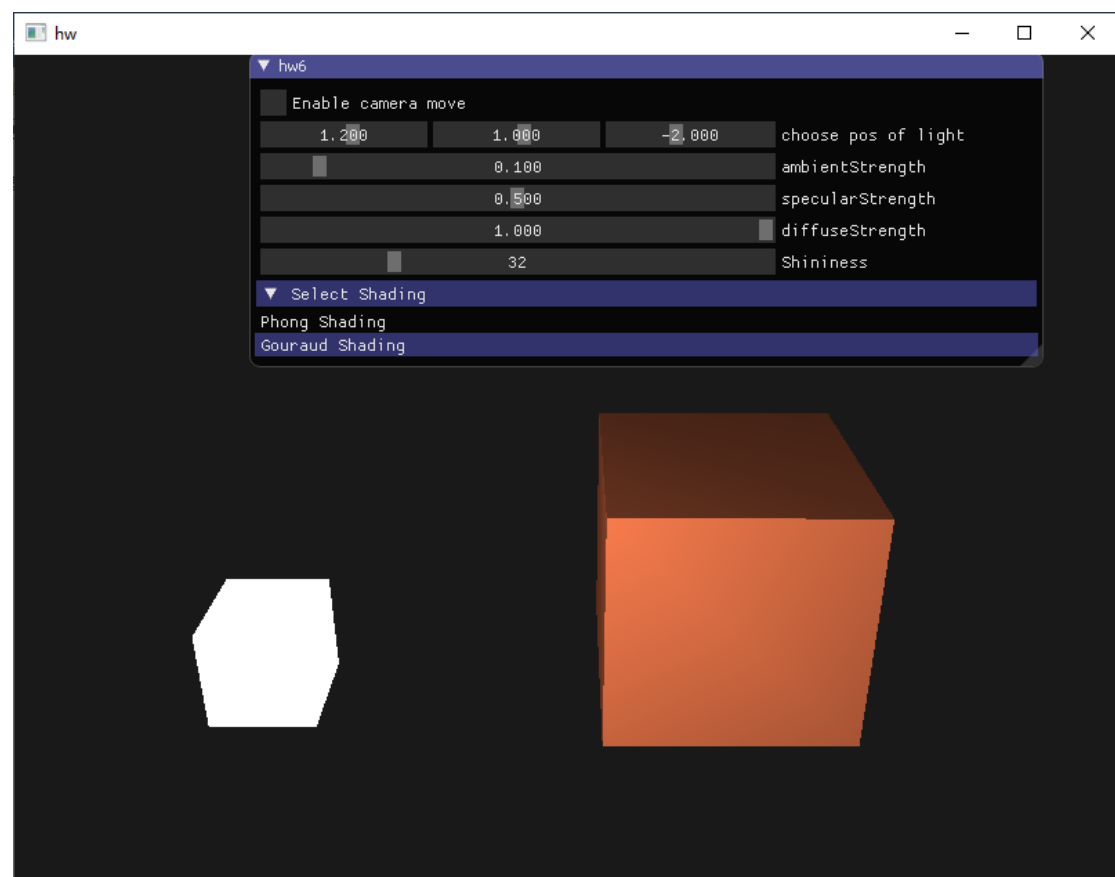
```
vec3 viewDir = normalize(viewPos - FragPos);  
vec3 reflectDir = reflect(-lightDir, norm);  
float spec = pow(max(dot(viewDir, reflectDir), 0.0), Shininess);  
vec3 specular = specularStrength * spec * lightColor;
```

在世界坐标系下，反射光和视角的夹角余弦值取反光度的幂。

最后对所有的光照效果求和即得到片段的颜色

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

实现 Gouraud Shading

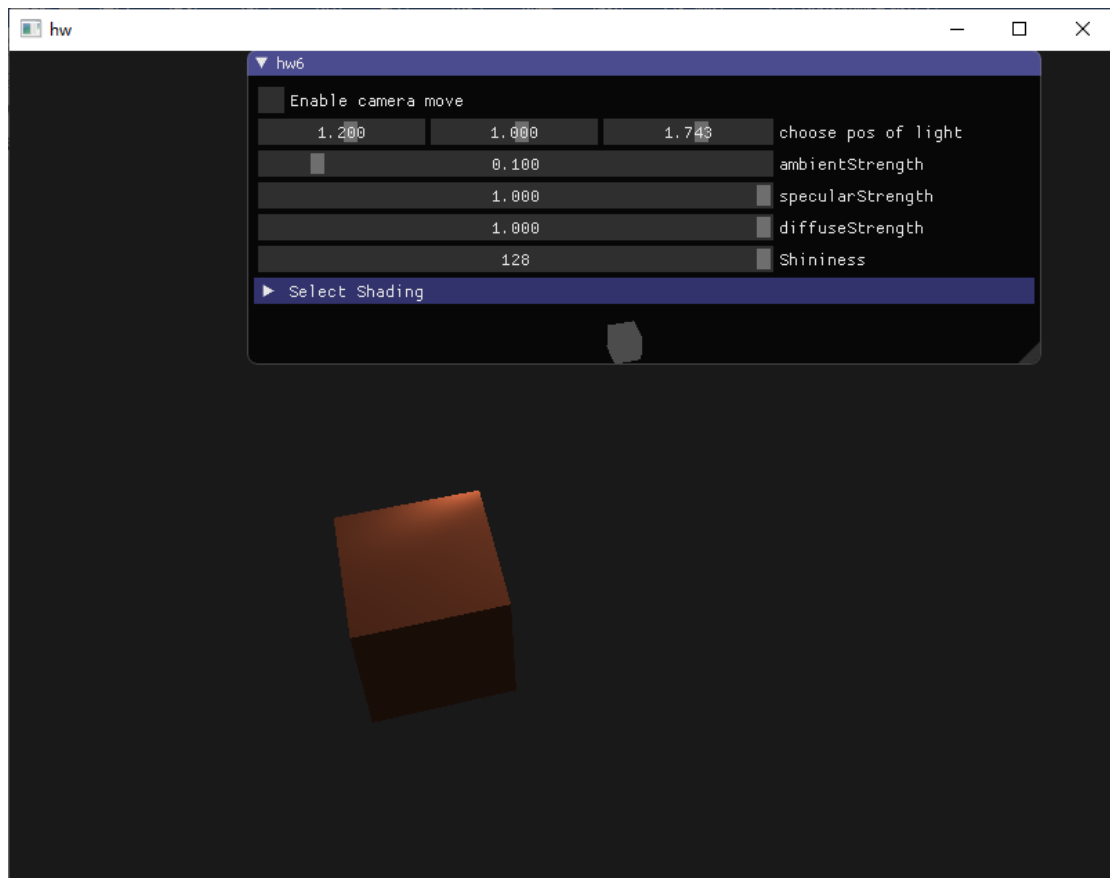


Gouraud Shading 就是将 Phong Shading 中的操作提前到顶点着色器中执行。

这样做的好处是：顶点的数量比片段少，计算速度快。

缺点是：顶点之间的颜色会使用顶点颜色进行插值，真实感降低。

添加 GUI & Bonus



在镜面反射系数为 1，反光系数为 128 时，明显看到正方体顶部的镜面反射现象。