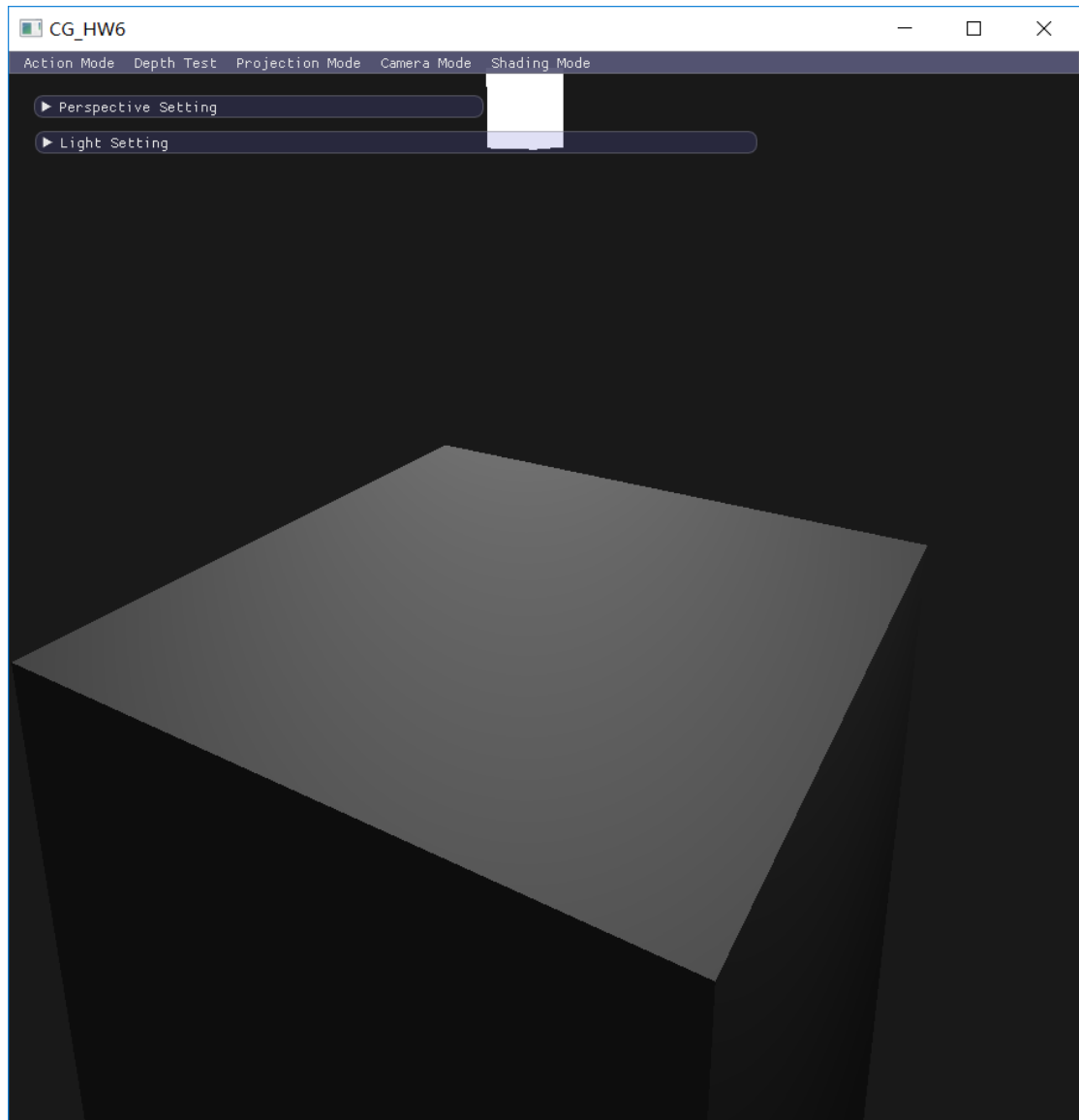


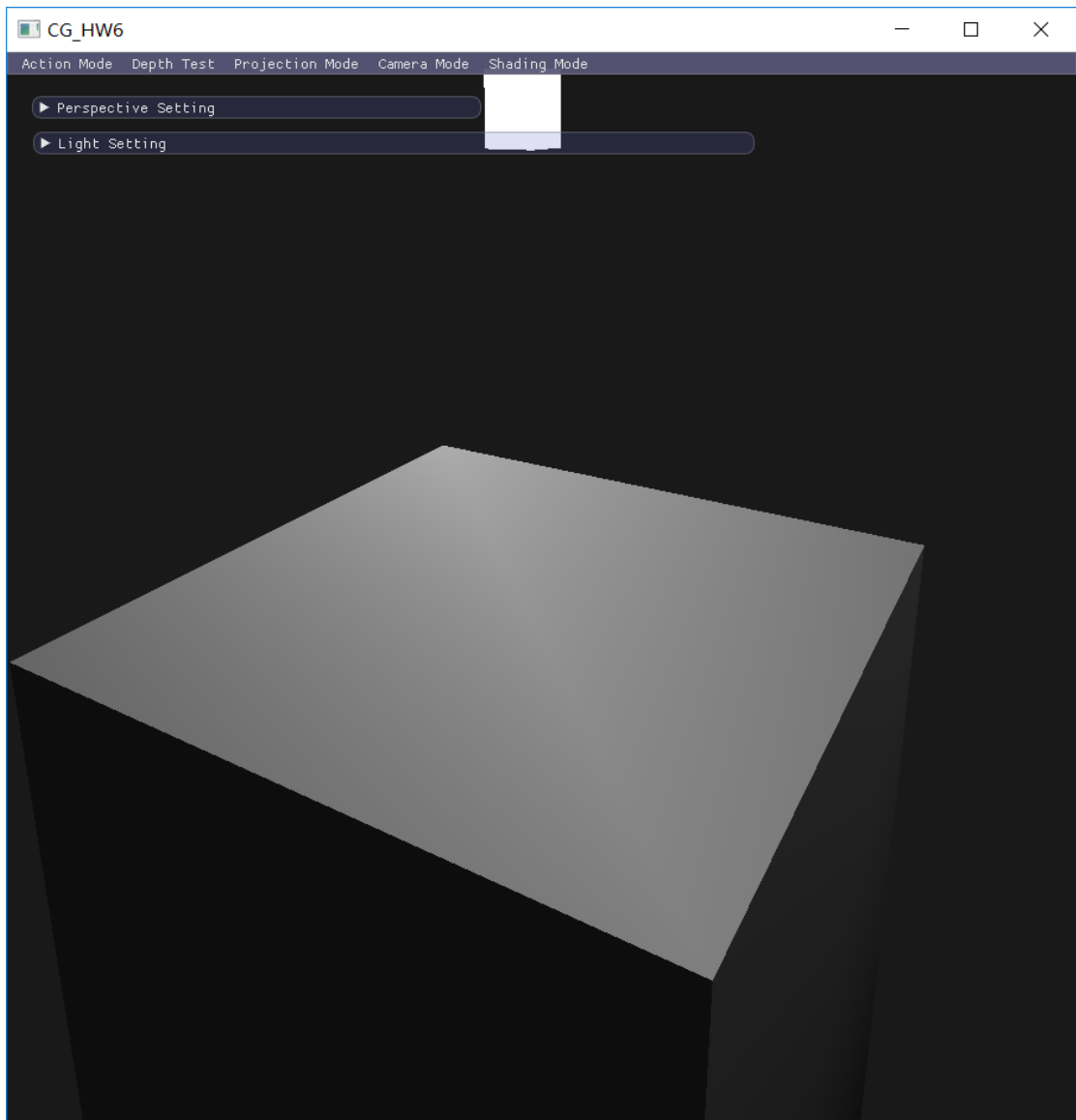
1. 实现 Phong 光照模型:

①场景中绘制一个 cube



②自己写 shader 实现两种 shading: Phong Shading 和 Gouraud Shading, 并解释两种 shading 的实现原理

(①中的截图为 phong Shading 模型的结果)



(Gouraud 模型的结果)

实现原理:

Phong Shading 模型:

每一点的法向量由邻近的已知平面的法向量线性插值得来, 每一点都应用 Phong 光照模型和其法向量来计算其颜色值。

Gouraud Shading 模型:

在每一个顶点处用临接平面的法向量的平均归一化的向量作为该点的法向量, 在顶点处应用光照模型与其法向量来计算顶点的颜色值。而在多边形内部的颜色值则由其顶点的颜色值线性插值得来。

③合理设置视点、光照位置、光照颜色等参数, 使光照效果明显显示

2. 使用 GUI, 使参数可调节, 效果实时更改:

①GUI 里可以切换两种 shading

②使用如进度条这样的控件, 使 ambient 因子、diffuse 因子、specular 因子、反光度等参数可调节, 光照效果实时更改

(见 Gif)

实现思路:

Phong 光照模型的实现:

(1) Phong 光照模型包括三部分:

- ① 环境光: 真实的环境光非常复杂, 这里直接以物体本身颜色值作为环境光的成分。
- ② 漫反射: 利用公式 $I_{diffuse} = I_{light}(\mathbf{n} \cdot \mathbf{l})$ 来计算点的漫反射成分, 其中 \mathbf{n} 为点的法向量, \mathbf{l} 为入射光线的取反。
- ③ 镜面反射: 利用公式 $I_{specular} = I_{light}(\bar{\mathbf{v}} \cdot \bar{\mathbf{r}})^{n_{shiny}}$ 来计算镜面反射成分。

三个成分加权求和即得到最终的光照。

(2) Phong Shading 模型需要对多边形内部的每一点都应用 Phong 光照模型来计算光照, 因此要在片段着色器中进行运算。Gouraud Shading 仅需对顶点应用光照模型, 所以在顶点着色器中进行运算, 随后片段着色器会自动进行线性插值得到多边形内部的颜色。