

# Homework 6

此次作业需要自己编写shader，在场景中添加局部光照（Phong Shading/Gouraud Shading），同时通过调节参数来观察不同的光照效果。

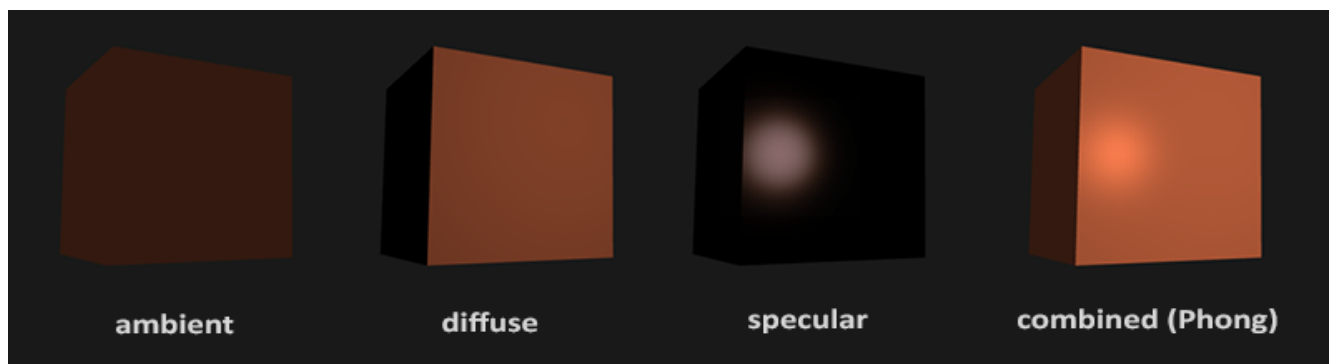
## Phong Lighting Model

Phong Shading与Gouraud Shading都属于冯氏光照模型（Phong Lighting Model）中的着色方式。其中，Phong Shading是对每个片元（fragment）或者说每个点计算一次光照，而Gouraud Shading是每个顶点（vertex）计算一次光照，所以Phong Shading的效果更加真实，当然计算开销也更高。

不管是Phong Shading还是Gouraud Shading，计算光照时都要进行三种计算：环境光照（Ambient Lighting）、漫反射光照（Diffuse Lighting）与镜面光照（Specular Lighting）。其中：

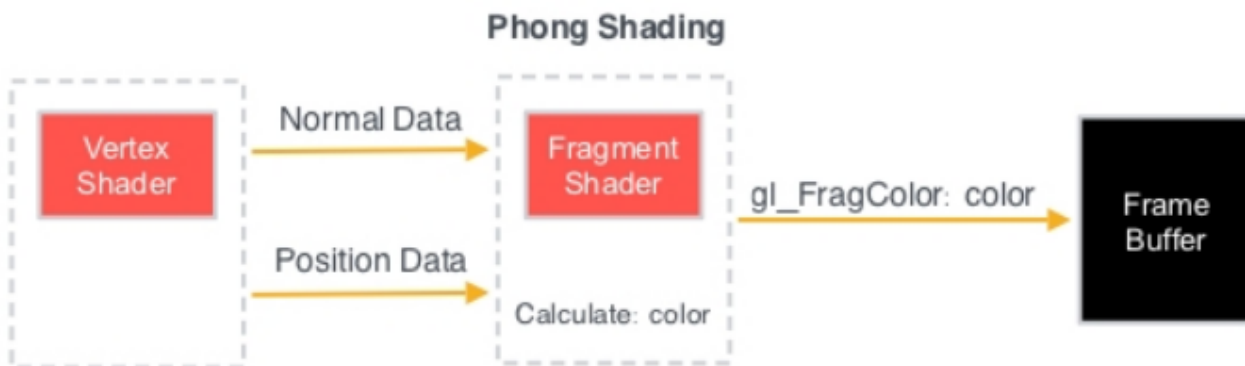
- 环境光照：环境光照永远会给物体一些颜色，其模拟的是即使在黑暗的情况下也存在的一些环境光。
- 漫反射光照：模拟光源对物体的方向性影响，物体正对光源的部分更亮。
- 镜面光照：模拟光源照射到有光泽物体上出现的亮点，镜面光照的颜色更倾向于光的颜色。

这三种光照组合，就模拟了真实的光照效果：



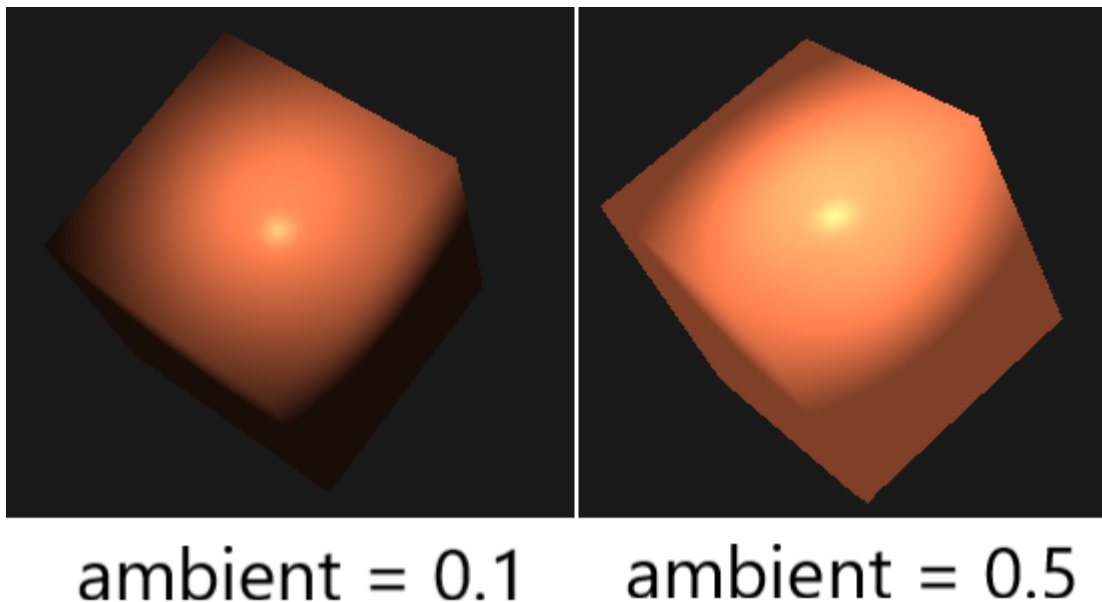
## Phong Shading

Phong Shading每次对片元计算光照，故应该在片段着色器中实现shading：

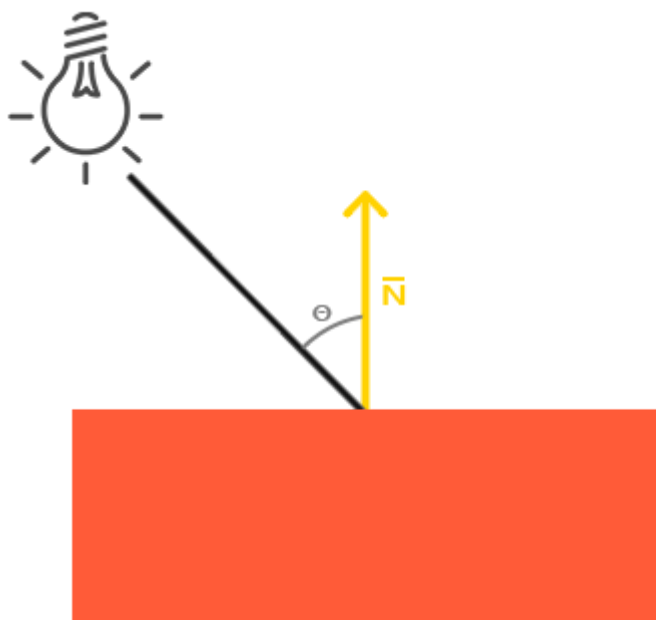


首先设置环境光，环境光实际上是对全局照明（十分复杂）的简化模拟，其实现就是用光的颜色乘以一个很小的常量环境因子。这里将环境光因子设置为uniform类型，以在ImGui中得以改变其值观察光照效果变化。

调整环境光因子的效果变化：



接着是漫反射光照，它使得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，如图：



其中的N为法向量，它是一个垂直于顶点表面的向量，这里的法向量在顶点数据中手动设置。因为需要得到的是一个漫反射影响因子，这里对顶点数据中设置的法向量和光的方向向量都进行标准化：

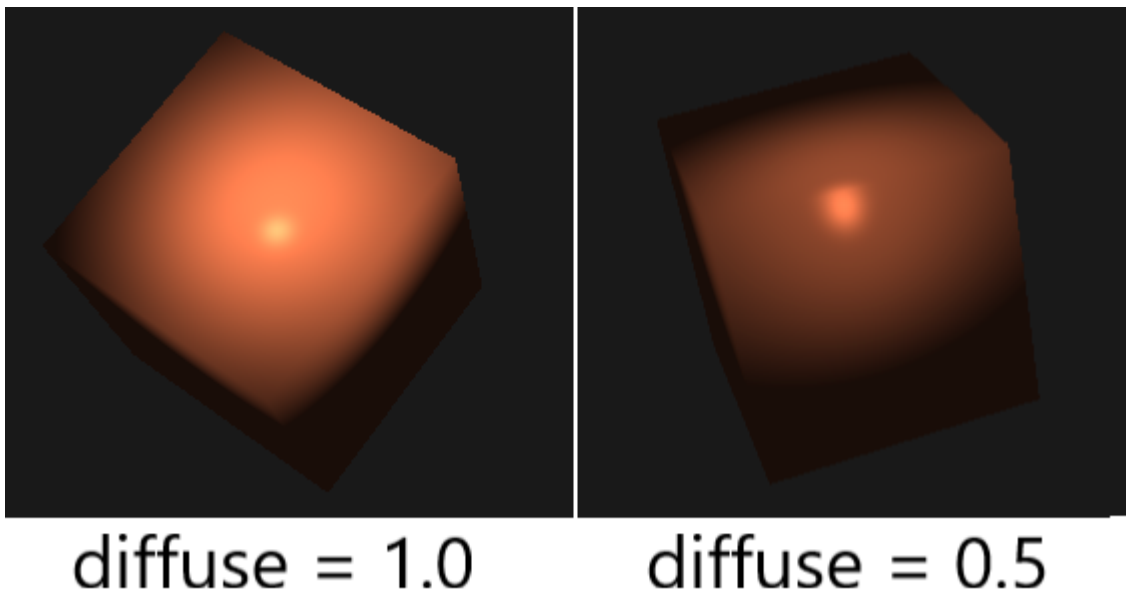
```
vec3 norm = normalize(Normal);  
vec3 lightDir = normalize(lightPos - FragPos);
```

接下来就可以计算光源对当前片段的实际漫反射影响：

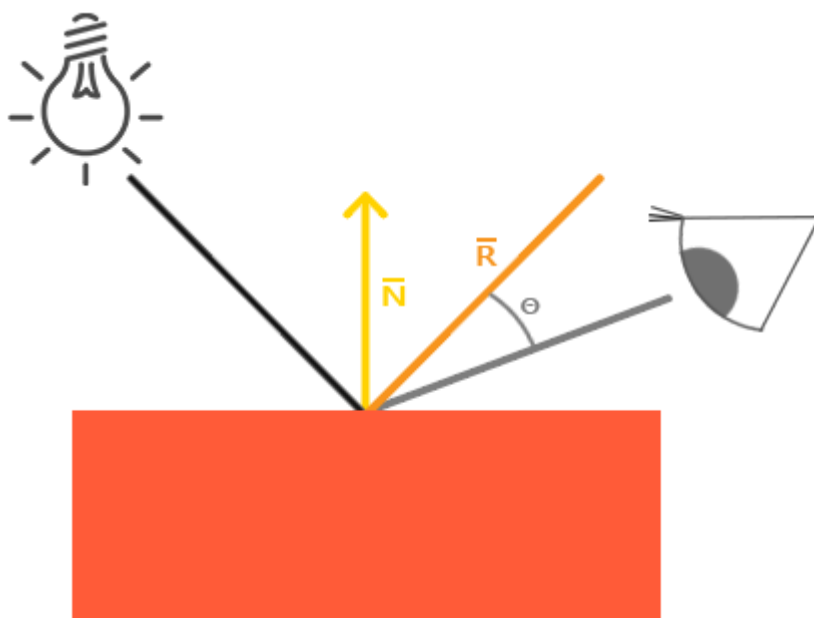
```
float diff = max(dot(norm, lightDir), 0.0);  
vec3 diffuse = diff * lightColor;
```

可见光的方向向量与法向量间的角度越大，漫反射影响就越小。光源越是正对片段，其影响就越大。

调整漫反射因子的效果变化：



最后是镜面光照，镜面光照也是根据光的方向向量和物体的法向量来决定的，但它也依赖于观察方向（这里即为摄像机的位置），如下图：



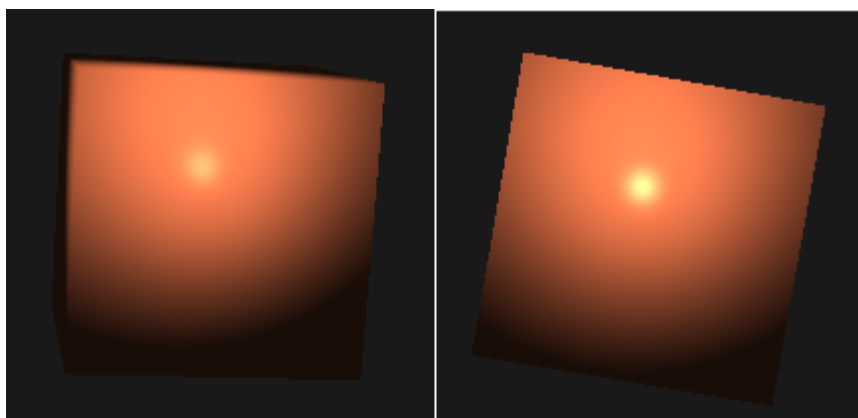
首先计算出反射向量R，接着计算反射向量与视线方向间的角度差。角度越大，镜面光的影响就越小。镜面光的效果是在物体表面呈现的高光。为了计算视线方向，需要给着色器传入摄像机位置坐标再进行计算：

```
uniform vec3 viewPos;  
shader.setVec3("viewPos", camera.Position);
```

接下来，在计算镜面光照时，有两个参数可以设置。第一个是镜面强度，它反映了镜面高光的程度，初始设置为0.5：

```
float specularStrength = 0.5;
```

第二个参数是高光的反光度，反光度越高则物体的反射光的能力越强，光的散射越少，即高光点就会越小。反光度普遍可从2到256，它对视觉效果影响很大。调整这两个参数的效果变化如下：

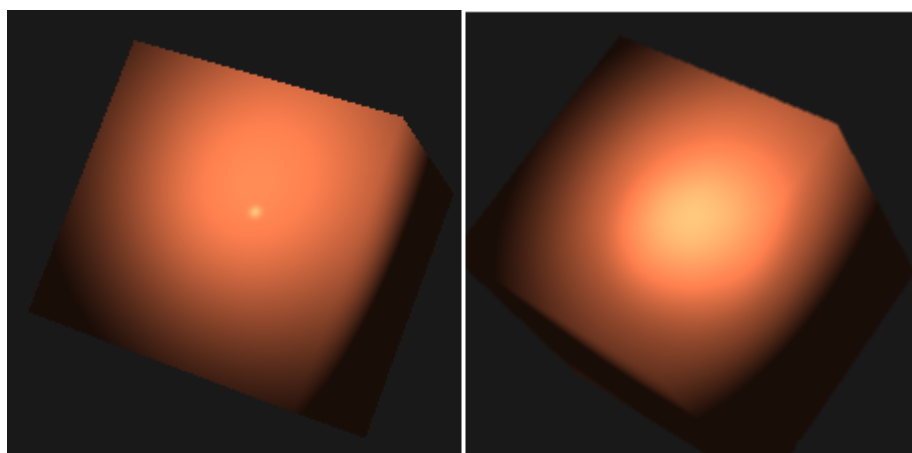


specular = 0.5

反光度 = 32

specular = 1.0

反光度 = 32



specular = 0.5

反光度 = 256

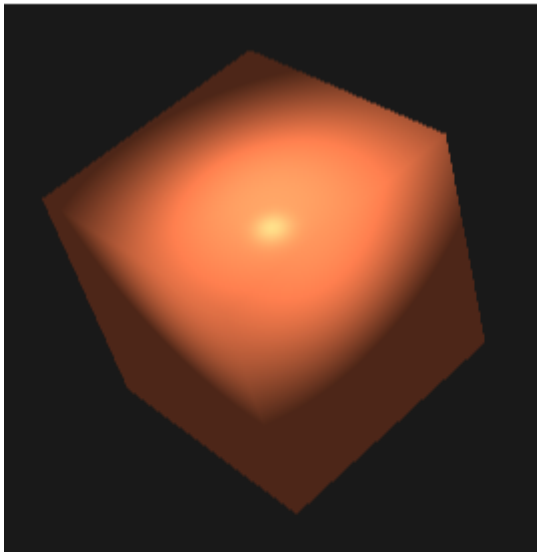
specular = 0.5

反光度 = 2

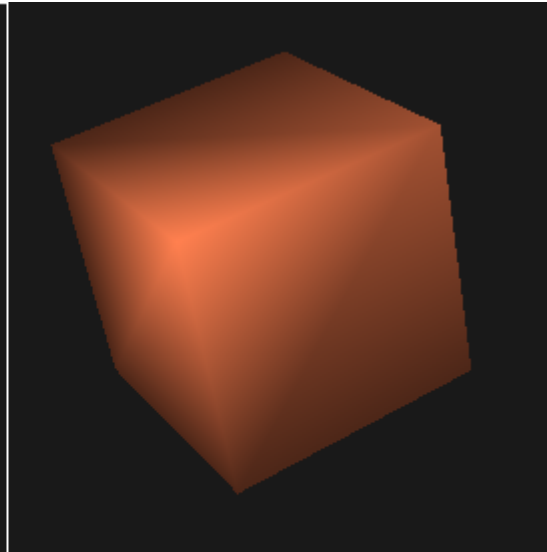
## Gouraud Shading

与Phong Shading不同，Gouraud Shading是对每个vertex计算一次光照，所以开销要小得多。但想像这样一种情况：顶点数组设置的三角形只有中间部分被光照射到，而顶点处没有，此时整个三角形都没有光照效果，这就导致Gouraud Shading的效果没有Phong Shading那么好。

除了Gouraud Shading是在顶点着色器中实现之外，它的实现细节和Phong Shading没有什么不同。其与Phong Shading的不同效果（均为ambient=0.301，diffuse=1.0，specular=0.5，反光度=32）：



Phong Shading



Gouraud Shading