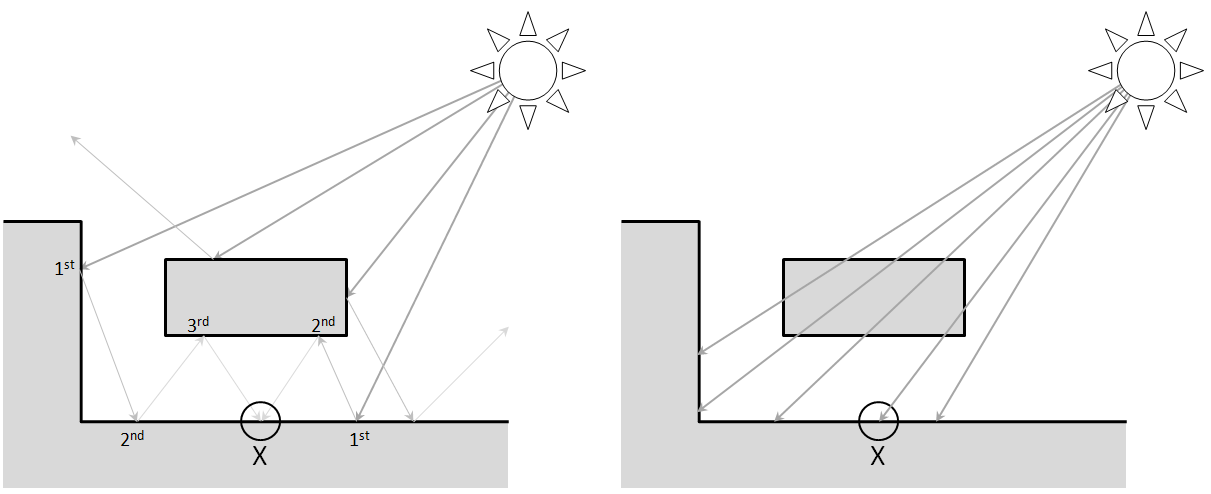
实验3.3 Phong反射模型（1）

1. 实验内容
2. 了解OpenGL中基本的光照模型
3. 掌握OpenGL中实现基于顶点的光照计算
4. 掌握法向量的计算
5. 理论背景

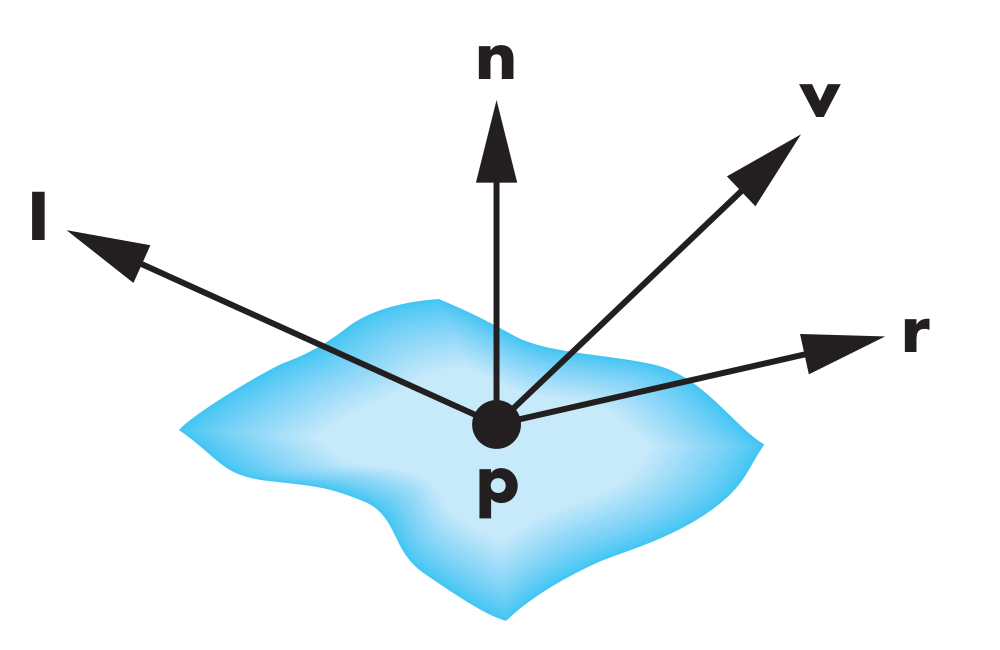
通过仔细观察前面章节中所有绘制的例子可以发现，所得到的图像基本上是平的，没有深度感，从而不能显示出模型的三维特性，不是很符合真实场景中人类视觉对三维物体的感知。这是因为我们假设物体表面只有单独一种颜色，在这样的假设下，球面的正投影将是均匀着色的圆。类比一下真实场景，我们能看到物体具体不同的颜色是因为眼睛接收了特定波长的光，不透明物体的颜色取决于它反射的色光，而透明物体的颜色则取决于透过它的色光。简单来说，物体只有在光照条件下才能呈现出不同的颜色，并且由于光照可以产生不均匀着色和明暗度的效果。在本周我们将考虑简单的光源模型和几种描述光线-材质之间相互作用的模型，目的是在图形绘制流水线中增加明暗处理功能。在现代OpenGL绘制管线中，我们可以将光照模型加在顶点着色器中，或者在片元着色器中。

光照模型包括局部光照和全局光照。局部光照指物体表面上一点的颜色只取决于表面的材质属性、表面的局部集合性质以及光源的位置和属性，而与场景中其他的表面无关。而全局光照则需要考虑场景中所有表面和光源相互作用的照射效果。如下图所示，左图中x点接收到周围环境的光线照射表示全局光照，右图中点x接收来自光源的直接照射表示局部光照。在本课程中我们只考虑局部光照。



在局部光照模型中，我们考虑从光源发出的光线，并对这些光线和场景中反射光线的表面之间的相互作用进行建模。这个问题包括两个独立的部分，首先是对场景中的光源建模，其次是必须构建一个描述物体表面和光线之间相互作用的模型。本周我们主要讨论Phong反射模型，它是对基于物理模型绘制方法的一种近似，并且具有很高的效率，能支持各种各样的光照条件和材质属性。

Phong反射模型中涉及到了4个基本向量，如下图所示。其中p为三维物体表面上的一点，l是从点p指向光源位置的向量，n表示p点的法向量，v是从p点指向相机（观察者）的向量，r是沿着l方向入射光线按照反射定律的出射方向。注意，在下面的计算中，我们假定所有的向量都已经归一化。



Phong反射模型考虑了光线和材质之间的三种相互作用：环境光反射、漫反射和镜面反射，最终在三维物体表面上的每个点的颜色由这三个成分组合表示如下。



环境光反射：在物体表面所有点处的环境光强度都是相同的。环境光一部分被表面吸收，一部分被表面反射，被反射部分由环境光反射系数决定，所以最终得到的环境光部分可以由如下公式得到：

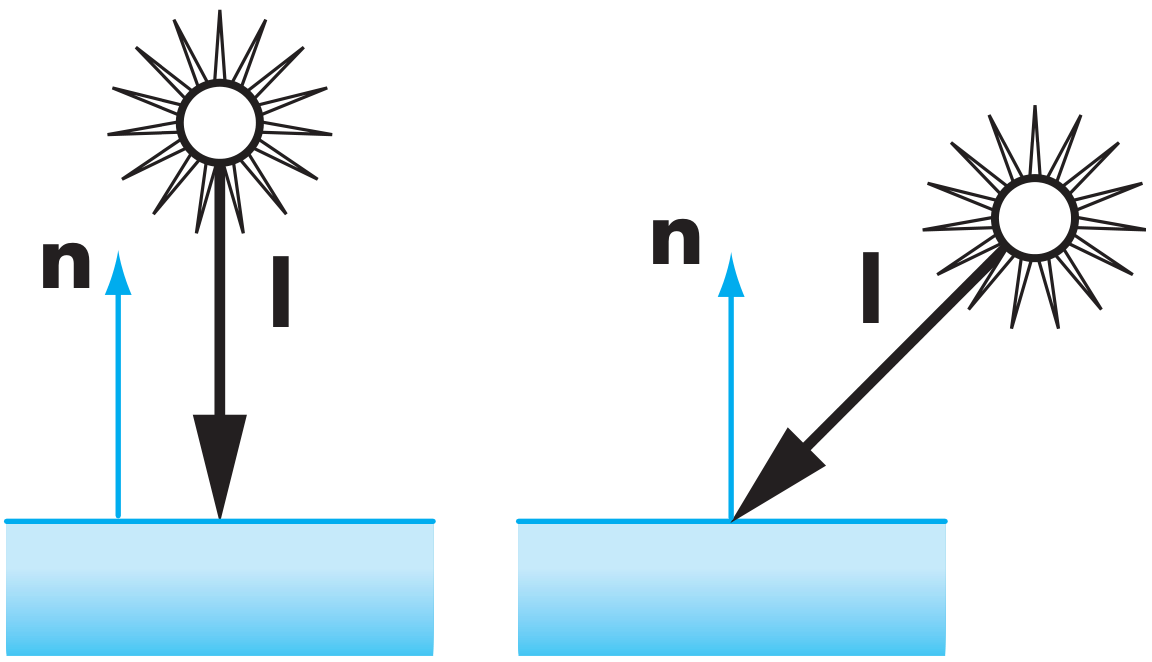


其中可以表示任何单独的光源，也可以代表全局环境光。

在实际场景中，当一个光源发出的一部分光线直接照射到物体表面的时候，通常对最终渲染图像有贡献的部分大可通过漫反射和镜面反射进行建模而近似表示，但是考虑场景的复杂性，通常光线会被场景中其他物体反射多次后重新射入我们正在考虑的这个表面上，为了降低计算的复杂性同时又更加逼真的表示场景的光照效果，我们通过加入环境光分量来近似这部分贡献。

漫反射：理想的漫反射表面会把光线向所有方向均匀散射，这样的表面称之为Lambert表面（Lambertian Surface），根据Lambert定律，只有入射光线的垂直分量才对照明起到作用，也就是说漫反射部分的光照效果与法向量n和光源向量l的夹角大小有关。如下图所示。同时考虑漫反射系数kd表示表面对漫反射光的反射程度，最终结果如下所示：

****

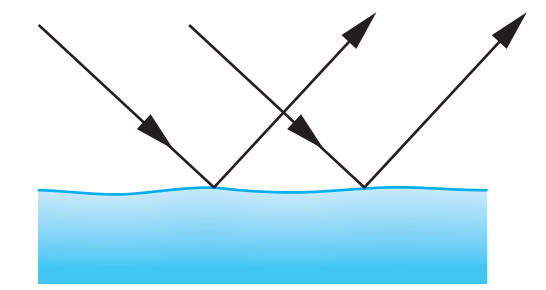


其中表示漫反射光线的强度。另外需要注意的地方，如果光源位于物体表面以下，那么上述公式中值将取负数，但在实际情况下这时物体表面并没有受到漫反射光的照射，所以当取负数时应该直接取0而消除此时漫反射光对最终结果的影响。此外，从物理角度来看，光照的强度会随着光源与物体间距离变长而逐渐衰减，综合考虑上述因素得到漫反射成分的计算公式如下：



其中d表示光源和三维物体表面上点的距离，a，b和c表示距离衰减系数。

镜面反射：在漫反射光照模型中，我们假设了物体表面是均匀粗糙的。而镜面反射成分是为了模拟光滑表面，而且表面越光滑，反射出去的光线越集中在一个角度附近，越接近真实的镜子（不透明镜子），如下图所示。



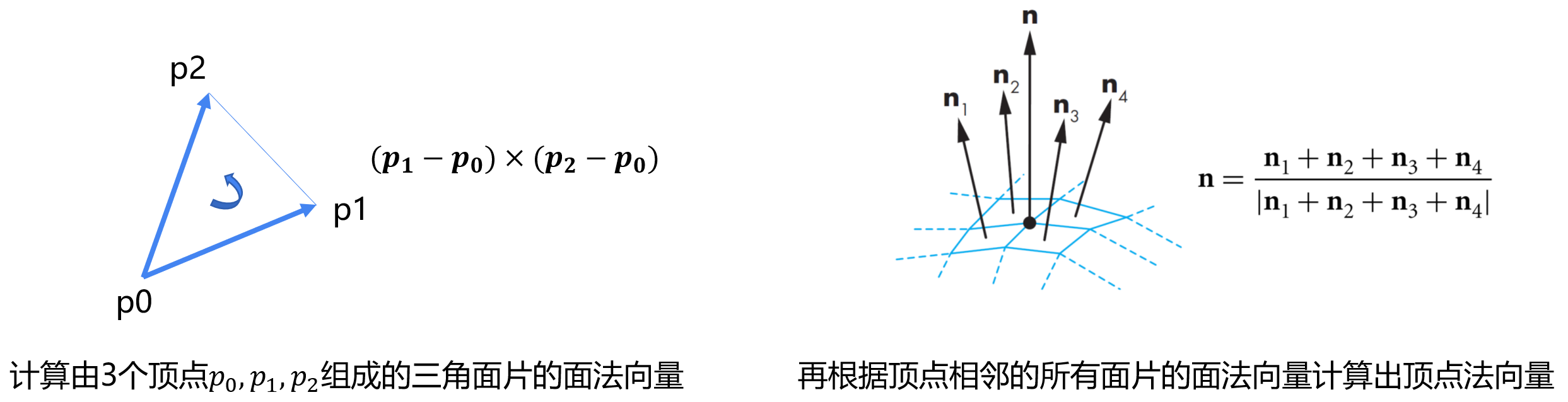
Phong提出了一个近似模型考虑镜面反射光部分，将表面看成是光滑的，观察者看到的光线强度取决于物体表面反射光的方向r和观察方向v这两者之间的夹角，用公式表示如下。

类似于漫反射，我们可以像计算漫反射分量那样加上一个距离因子。

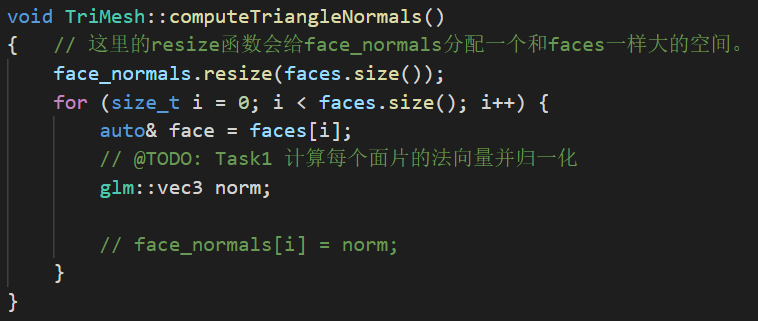
综合上述几种不同的光照情况，我们得到最终的Phong模型公式如下。

1. 实验内容
2. 法向量计算

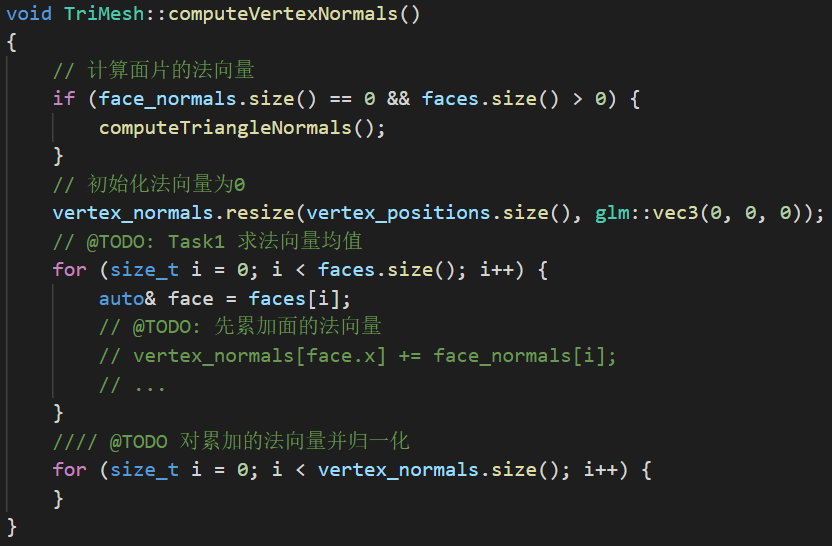
法向量对光照计算有着重要作用，如果法向量计算有误，会造成许多绘制结果的缺陷。Task1中我们使用球作为模型，球的法向量可以精确求解，而对于其他的模型，我们需要先计算面片的法向量，然后再通过面片法向量计算顶点的法向量均值。



* 1. 计算面片法向量：给定三角形的三个顶点，和，法向量为：

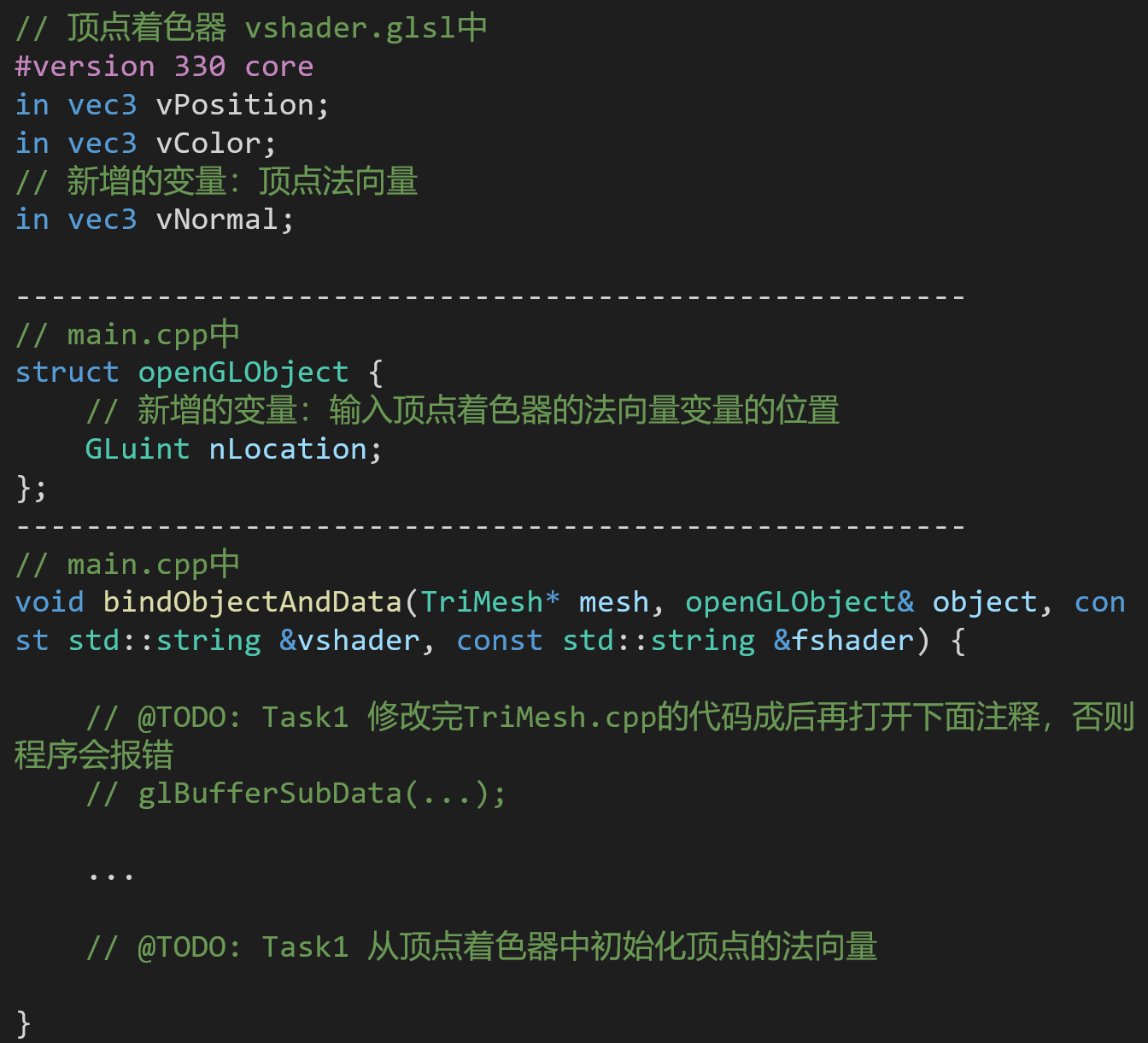


* 1. 计算顶点法向量：给定顶点所在面片的法向量，顶点的平均法向量为法向量的和。



上面的函数将会在storeFacesPoints函数中调用。

* 1. 我们计算好法向量后，和顶点坐标类似，我们需要将其数据传递给着色器，为此我们在顶点着色器文件内新加了一个法向量变量。同时我们给保存缓存对象的 openGLObject结构体内新加了一个 nLocation。参照顶点坐标vPosition的写法，在bindObjectAndData函数中将传递法向量的代码补全，注意BUFFER\_OFFSET的数值在bindObjectAndData函数中我们需要将法向量数据传递给着色器，具体写法参考vPosition的代码。



1. 在顶点着色器中实现Phong光照模型

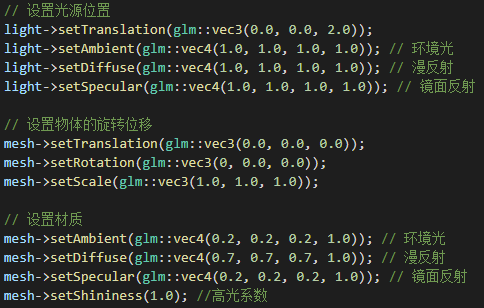
需要说明的是，在OpenGL绘制流水线中，光照计算需要等到三维物体经过相机变换和投影变换之后才能进行，因为此时物体在三维空间中的位置才完全确定下来。另外，通常情况下，我们在相机坐标系下来计算所有向量，因为在相机坐标系下，原点就是相机位置/眼睛位置。

具体实验过程如下：

1. 使用封装在TriMesh.h中的TriMesh类，读入球模型。



1. 设置光源，光源对象由光源位置环境光、漫反射光、镜面反射光组成。物体材质着由高光系数、环境光、漫反射光、镜面反射光参数组成。

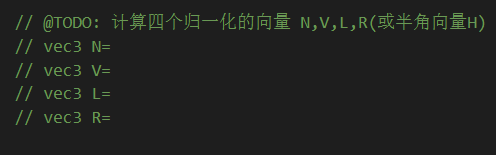


1. 为了传递这些材质数据，我们实现了一个bindLightAndMaterial函数用于传递数据给着色器。

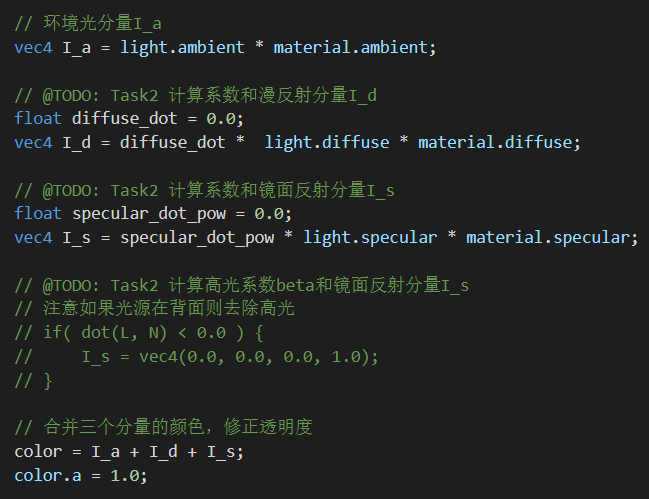


在顶点着色器中执行为每个顶点执行光照计算，为了简单考虑，我们这里假设衰减系数。

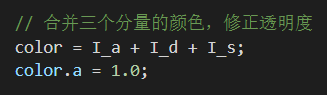
计算Phong反射模型涉及到的四个向量，并归一化，其中使用的normalize(a)函数和reflect(in,norm)函数均为GLSL语言内置函数，分别是归一化向量和依据入射向量和法向量计算反射向量。请根据图示计算N、L、V、R四个分量：



再依据公式，计算漫反射分量和镜面反射分量。代码中会使用到GLSL语言内置函数dot(a,b)，max(x,y)，作用分别是向量点积和取两者最大值，具体使用方法请参考相关文档。



最后为累加三个部分的颜色分量，得到每个顶点的颜色如下，颜色相加后最后一维的透明度需要进行修正。



1. 材质应用

曲面上某一点的颜色是光照和材质共同作用的结果，相同颜色的光照射在不同材质的物体上会得到不同的颜色，在代码中创建材质对象并应用到光照的计算中

设置材质对象，材质的属性有环境反射率、漫反射率、镜面反射率、（自发光系数）、高光系数等组成。可以参考这些网页提供的材质参数进行绘制。

<http://www.it.hiof.no/~borres/j3d/explain/light/p-materials.html>

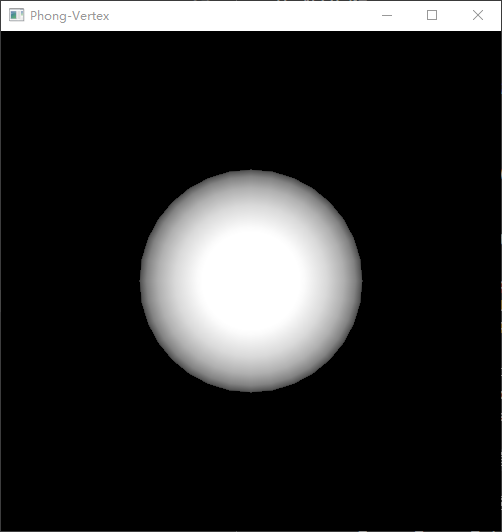
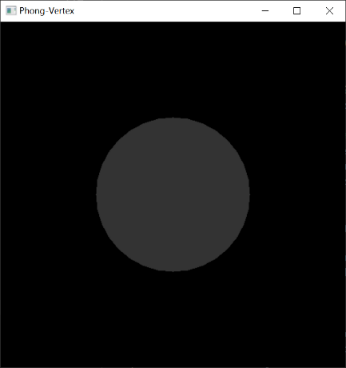
<http://devernay.free.fr/cours/opengl/materials.html>

1. 添加更多交互

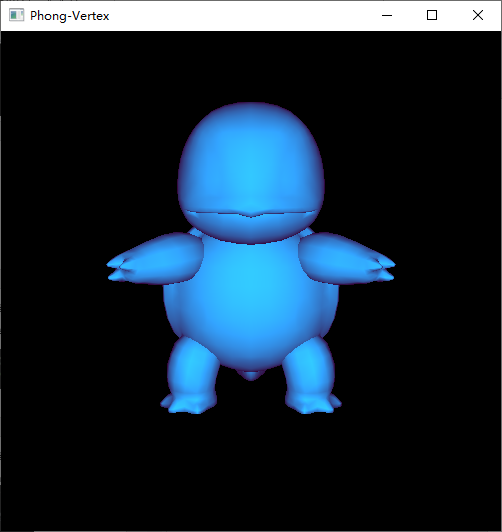
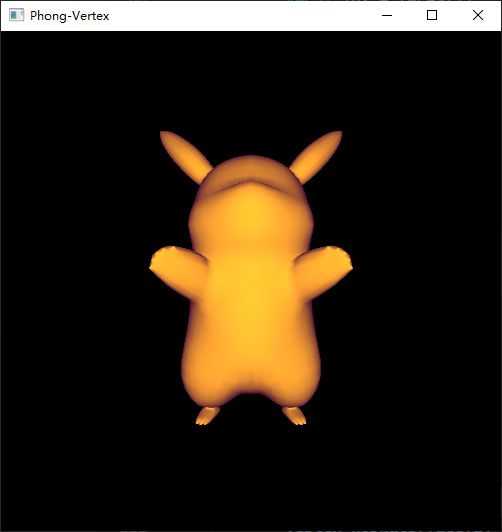
通过交互地改变物体、光源、材质、相机等参数进一步了解使用Phong反射模型实现的光照效果。

* 1. 尝试为光源light设置交互：如鼠标设置光源x,y位置（已实现）
  2. 尝试为材质material设置交互：1~9增减反射系数，0增减高光指数（实现一半）
  3. 尝试旋转或平移物体相机，从不同角度观察光照效果（已实现）

1. 示例结果
2. 具体修改内容如下：
   1. Task-1：在Trimesh.cpp中完善computeTriangleNormals()和computeVertexNormals()，接着在main.cpp中完善bindObjectAndData()
   2. Task-2：在vhsader.glsl中完善main()
   3. Task-3：在main.cpp中init()函数里改变材质参数
   4. Task-4：在main.cpp中mainWindow\_key\_callback ()函数完善键盘交互
3. 下面分别为初始结果、添加漫反射结果、添加镜面反射结果



1. 下面分别为相同光源不同材质的结果



1. 下面为调整光源位置后的结果

