

# 华东师范大学计算机科学与技术实验报告

实验课程：计算机图形学	年级：18级	实验成绩：
实验名称：Phong光照	姓名：董辰尧	实验日期：2021/5/11
实验编号：10	学号：10185102144	实验时间：13:00~14:40
指导教师：李洋、王长波	组号：	

## 一、实验目的

利用HLSL实现基本Phong光照。

## 二、实验环境

VS2017

## 三、实验内容

1. 阅读代码
2. 在shader中实现平行光源下的phong光照
3. 在shader中实现点光源下的phong光照

## 四、实验过程与分析

- **HLSL语言**：高阶着色器语言（High Level Shader Language，简称HLSL），由微软拥有及开发的一种语言，HLSL独立的工作在Windows平台上，只能供微软的Direct3D使用。HLSL的主要作用为将一些复杂的图像处理，快速而又有效率地在显示卡上完成，与组合式或低阶Shader Language相比，能降低在编写复杂特殊效果时所发生编程错误的机会。HLSL已经整合到了DirectX 9中。
- **阅读代码**：本次实验需要使用HLSL语言完成，首先阅读代码发现shaders.hlsl文件的上半部分主要是一些结构体，后半部分是实现的函数。并且我需要完成的函数的输出分为了三个部分，分别是漫反射、镜面反射、环境光。这三个部分可以分开计算。
- **在shader中实现平行光源下的phong光照**：漫反射光、镜面反射光、环境光结合起来就成了结果。每种光线的计算公式ppt中有，需要了解的是传入的参数怎么使用的问题。主要代码如下：

```
1 void calc_directional_light(Material mat, DirectionalLight L,
2   float3 normal, float3 to_eye,
3   out float4 ambient,
4   out float4 diffuse,
5   out float4 spec)
6 {
7   ambient = float4(0.0f, 0.0f, 0.0f, 0.0f);
8   diffuse = float4(0.0f, 0.0f, 0.0f, 0.0f);
9   spec = float4(0.0f, 0.0f, 0.0f, 0.0f);
10
11   // 光线方向
12   float3 lightVec = -L.direction;
13   // 环境光计算
14   ambient = mat.ambient * L.ambient;
15   // 计算漫反射系数
```

```

16     float diffuseFactor = dot(lightVec, normal);
17
18     [flatten]
19     if (diffuseFactor > 0.0f)
20     {
21
22         float3 v = reflect(-lightVec, normal);
23         float specFactor = pow(max(dot(v, to_eye), 0.0f),
mat.specular.w);
24         //计算漫反射光
25         diffuse = diffuseFactor * mat.diffuse * L.diffuse;
26         //计算高光
27         spec = specFactor * mat.specular * L.specular;
28     }
29 }

```

- **在shader中实现点光源下的phong光照:**点光源和平行光源类似，需要计算顶点到光源的距离以及归一化光源的方向，最后还需要进行衰减。主要代码如下：

```

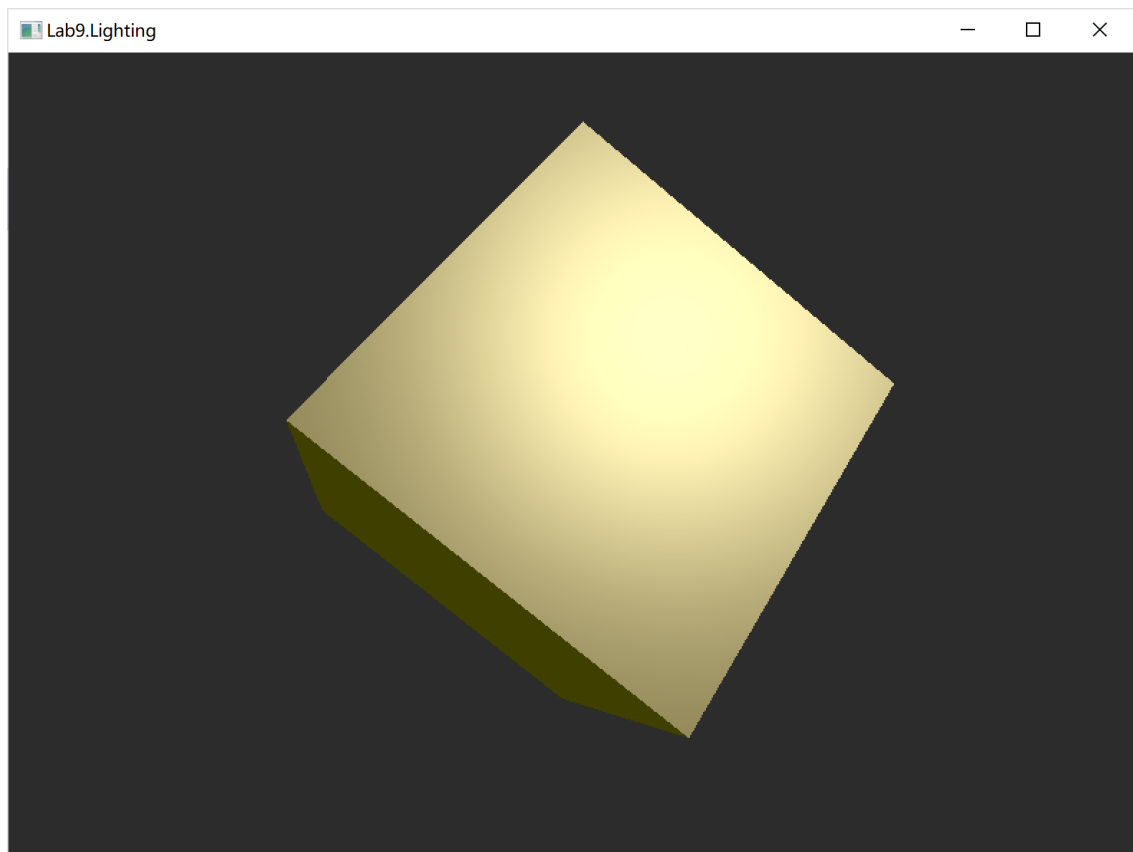
1 void calc_point_light(Material mat, PointLight L, float3 pos, float3
normal, float3 to_eye,
2     out float4 ambient, out float4 diffuse, out float4 spec)
3 {
4     ambient = float4(0.0f, 0.0f, 0.0f, 0.0f);
5     diffuse = float4(0.0f, 0.0f, 0.0f, 0.0f);
6     spec = float4(0.0f, 0.0f, 0.0f, 0.0f);
7     //光照方向: 顶点到光源
8     float3 lightVec = L.position - pos;
9
10    //顶点到光源距离
11    float d = length(lightVec);
12
13    //归一化光照方向
14    lightVec /= d;
15
16    //计算环境光
17    ambient = mat.ambient * L.ambient;
18
19    //漫反射系数
20    float diffuseFactor = dot(lightVec, normal);
21
22    [flatten]
23    if (diffuseFactor > 0.0f)
24    {
25        float3 v = reflect(-lightVec, normal);
26        float specFactor = pow(max(dot(v, to_eye), 0.0f),
mat.specular.w);
27        //计算漫反射光
28        diffuse = diffuseFactor * mat.diffuse * L.diffuse;
29        //计算高光
30        spec = specFactor * mat.specular * L.specular;
31    }
32
33    // 计算衰减
34    float att = 1.0f / dot(0.01f, float3(1.0f, d, d*d));

```

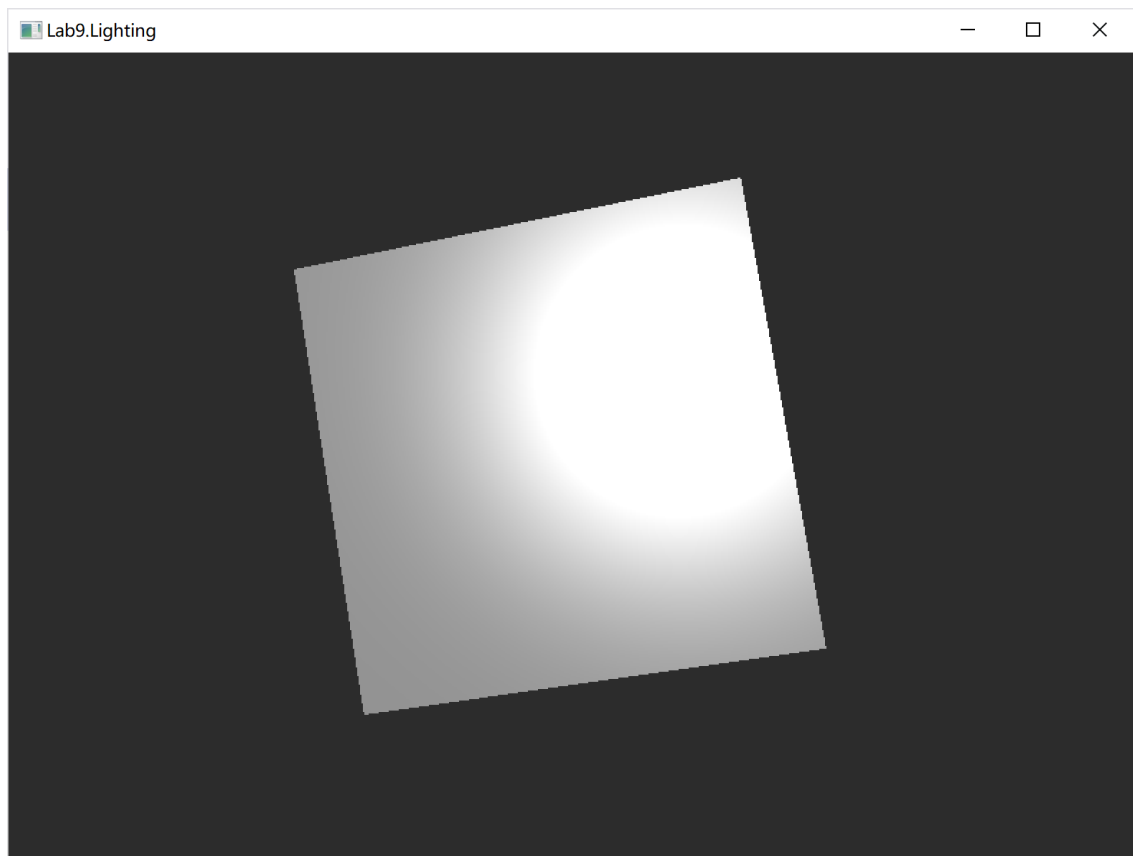
```
35  
36     diffuse *= att;  
37     spec *= att;  
38 }
```

## 五、实验过程总结

- 结果截图：
  - 平行光源



- 点光源



- 总结：这节课成功让一个平面图形3D化了，还是很有成就感，激发了我学习的兴趣。