

光源

2018年5月9日 15:25

平行光：

DirLight = 环境光+漫反射+镜面反射

点光源：

在近处通常会非常亮，但随着距离的增加光源的亮度一开始会下降非常快，但在远处时剩余的光强度就会下降的非常缓慢了

```
float distance = length(light.position - fragPos);
```

```
float attenuation = 1.0 / (light.constant + light.linear * distance + light.quadratic * (distance * distance));
```

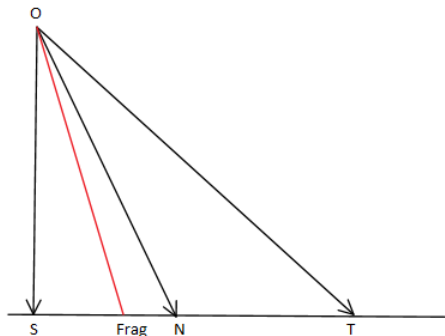
```
PointLight = DirLight * attenuation;
```

聚光灯：

为了创建一种看起来边缘平滑的聚光，我们需要模拟聚光有一个内圆锥(Inner Cone)和一个外圆锥(Outer Cone)。我们可以将内圆锥设置为上一部分中的那个圆锥，但我们也需要一个外圆锥，来让光从内圆锥逐渐减暗，直到外圆锥的边界。

如果一个片段处于内外圆锥之间，将会给它计算出一个0.0到1.0之间的强度值。如果片段在内圆锥之内它的强度就是1.0，如果在外圆锥之外强度值就是0.0。

这样可使边缘延外圆锥平滑变暗。如果使内圆锥等于外圆锥则到边缘强度直接从1.0变为0.0。



$\alpha = \angle SON$ 内圆锥角度

$\theta = \angle SOF$ 片段指向光源的角度

$\varphi = \angle SOT$ 外圆锥角度

$I = (\cos\theta - \cos\varphi) / (\cos\alpha - \cos\varphi)$

如果 $\theta < \alpha$ 在内圆锥内， $I > 1$

如果 $\alpha < \theta$ && $\theta < \varphi$ 在内圆锥和外圆锥之间 $1 > I$ && $I > 0$

如果 $\theta > \varphi$ 则 $I < 0$

```
float theta = dot(lightDir, normalize(-light.direction)); // 聚光灯方向light.direction, 片段指向光源方向lightDir
```

```
float epsilon = light.cutOff - light.outerCutOff; // 内圆锥和外圆锥的余弦差
```

```
float intensity = clamp((theta - light.outerCutOff) / epsilon, 0.0, 1.0);
```

```
SpotLight = PointLight * intensity;
```