逻辑教育
Logic education

Hello CC

OpenGL ES 主题[5]

视觉班—OpenGL ES 着色器与光照计算

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

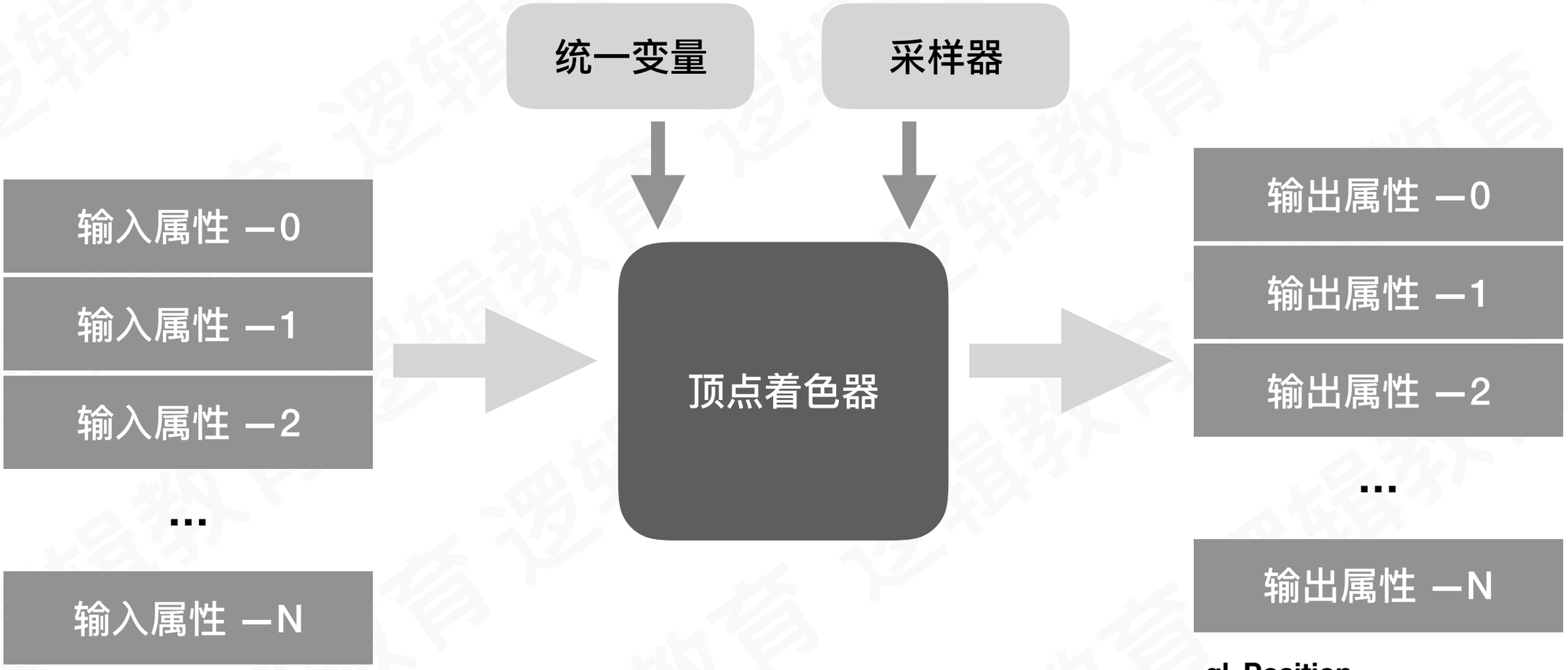
课程目标

1. 顶点着色器详讲
2. 片元着色器详讲
3. 光照的基本概念
4. 光照的种类
5. 光的种类
6. 材质的种类
7. 光照计算
8. 案例

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

OpenGL ES 3.0 顶点着色器



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

顶点着色业务:

顶点着色器 业务:

1. 矩阵变换位置

2.计算光照公式生成逐顶点颜色

3.生成/变换纹理坐标

总结: 它可以用于执行自定义计算,实施新的变换,照明或者传统的固定功能所不允许的基于顶点的效果.

课程研发:CC老师
课程授课:CC老师

顶点着色代码案例:

```
attribute vec4 position;
attribute vec2 textCoordinate;
uniform mat4 rotateMatrix;
varying lowp vec2 varyTextCoord;
void main()
{
    varyTextCoord = textCoordinate;
    vec4 vPos = position;
    vPos = vPos * rotateMatrix;
    gl_Position = vPos;
}
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

顶点着色器 内建特殊变量

gl_VertexID

gl_InstanceID

gl_Position

gl_PointSize

gl_FrontFacing

课程研发:CC老师
课程授课:CC老师

顶点着色器 内建Uniform

```
struct gl_DepthRangeParameters
{
    highp float near; //near z
    highp float far;  //near far
    highp float diff; //far - near
}

uniform gl_DepthRangeParameters gl_DepthRange;
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

顶点着色器 内建常量

```
const mediump int gl_MaxVertexAttribs = 16;
const mediump int gl_MaxVertexUniformVectors = 256;
const mediump int gl_MaxVertexOutputVectors = 16;
const mediump int  gl_MaxVertexTextureImageUnits = 16;
const mediump int gl_MaxCombinedTextureImageUnits = 32;
```

课程研发:CC老师
课程授课:CC老师

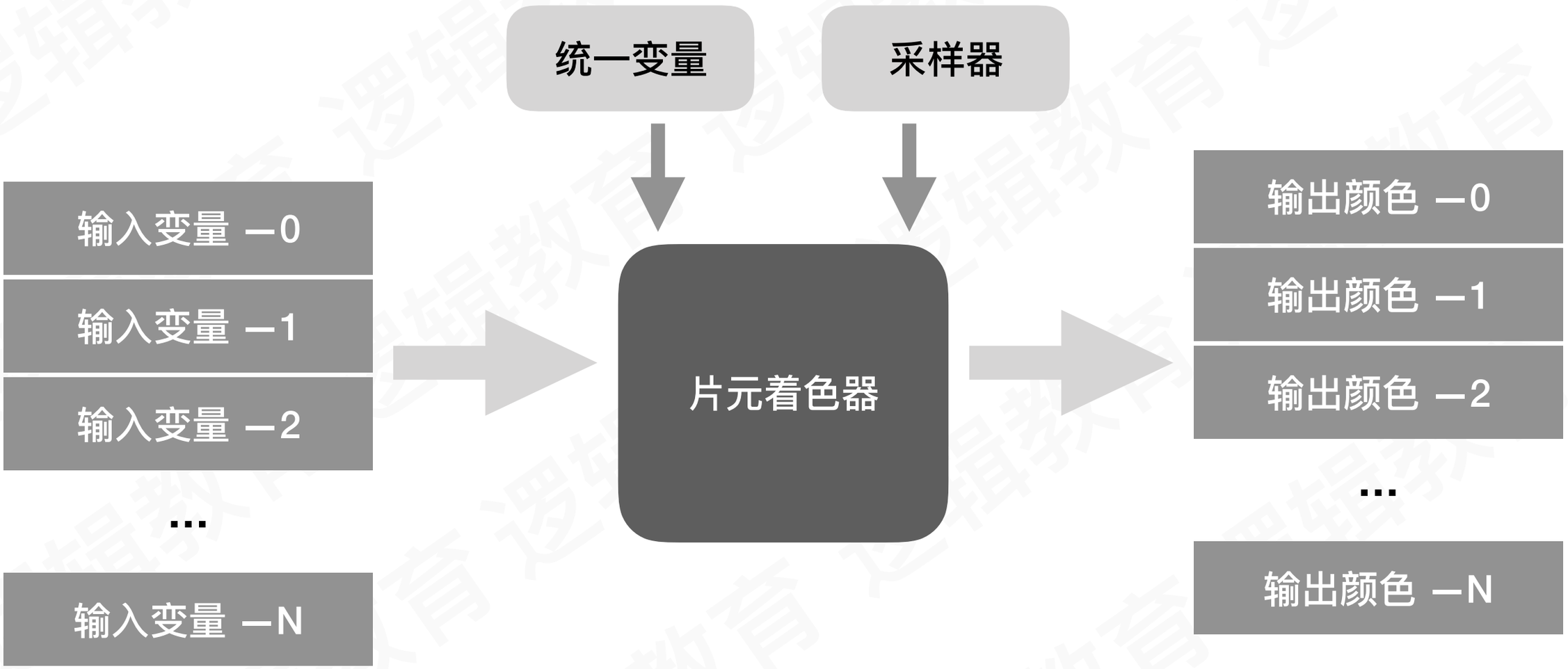
逻辑教育
Logic education

顶点着色器 矩阵变换

MVP(模型->视图->投影)矩阵变换

课程研发:CC老师
课程授课:CC老师

OpenGL ES 3.0 片段着色器/片元着色器



gl_FragColor

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

片元着色器业务:

片元着色器 业务:

1. 计算颜色

2. 获取纹理值

3. 往像素点中填充颜色值(纹理值/颜色值);

总结: 它可以用于图片/视频/图形中每个像素的颜色填充(比如给视频添加滤镜,实际上就是将视频中每个图片的像素点颜色填充进行修改.)

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

片元着色代码案例:

```
varying lowp vec2 varyTextCoord;
uniform sampler2D colorMap;
void main()
{
    gl_FragColor = texture2D(colorMap, varyTextCoord);
}
```

课程研发:CC老师
课程授课:CC老师

内建特殊变量

- gl_FragCoord
- gl_FrontFacing
- gl_PointCoord
- gl_FragDepth

课程研发:CC老师
课程授课:CC老师

内建常量

```
const mediump int gl_MaxFragmentInputVectors = 15;
const mediump int gl_MaxTextureImageUnits = 16;
const mediump int gl_MaxFragmentUniformVectors = 224;
const mediump int gl_MaxDrawBuffers = 4;
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

多个纹理单元渲染(服务端)

```
//片元着色器代码
attribute vec2 v_texCoord;
uniform sampler2D s_baseMap;
uniform sampler2D s_SecondMap;
void main()
{
  vec4 baseColor;
  vec4 secondColor;

  baseColor = texture(s_baseMap ,v_texCoord);
  secondColor = texture(s_SecondMap ,v_texCoord);

  gl_FragColor = baseColor * secondColor;

}
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

多个纹理单元渲染(客户端)

```
//客户端代码：将各个纹理对象绑定到纹理单元0和1,为采样器设置数值,将采集器绑定到对应的纹理单元
glActiveTexutre(GL_TEXTURE0);
glBindTeture(GL_TEXTURE_2D ,baseMapTexId);
glUniformli(baseMapTexId,0);

glActiveTexutre(GL_TEXTURE1);
glBindTeture(GL_TEXTURE_2D ,secondMapTexId);
glUniformli(secondMapTexId,1);
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

内建函数

常用内建函数:
dot :点乘
cross :叉乘
texture2D :用于对纹理采样
normalize :对一个向量规格化
clamp :将一个向量固定在一个最小值和最大值之间

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

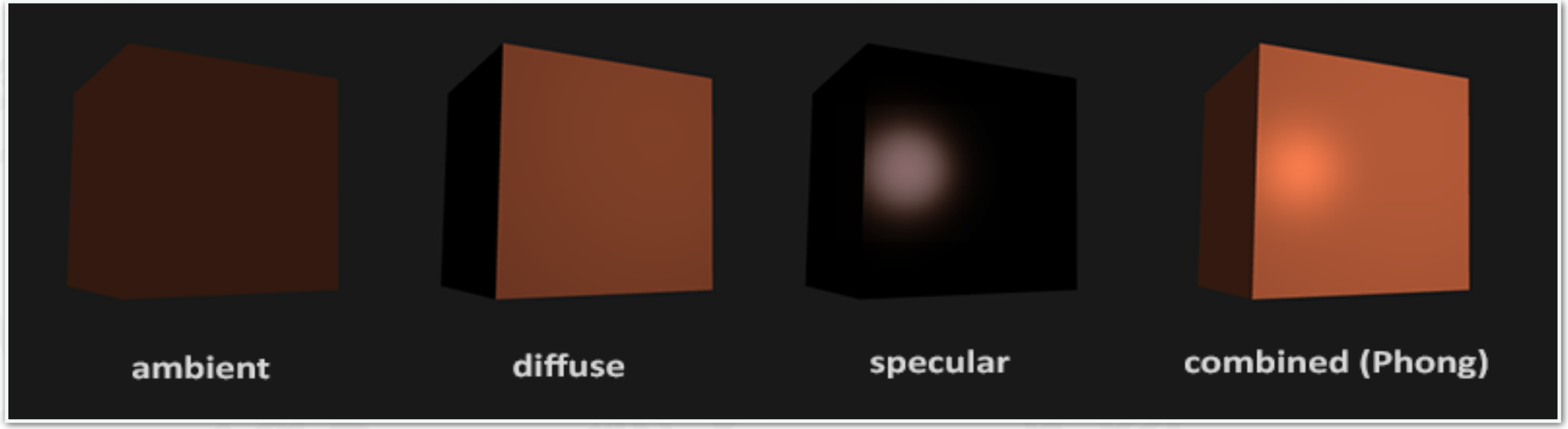
内建函数

函　数	说　明
pow()	幂函数（对矢量和标量同样有效，下同）
exp(), log()	指数函数，对数函数
abs()	绝对值
sqrt()	平方根
max(), min()	最值
ceil(), floor()	取大于实参的最小整数，取小于实参的最大整数
sin(), cos(), tan()	三角函数
asin(), acos(), atan()	反三角函数
sinh(), cosh(),<a>tanh ()	双曲正弦，双曲余弦，双曲正切（以及相应的反函数）
length()	向量长度
distance()	两个向量的距离
dot(), cross()	数乘，叉乘
matrixCompMult()	矩阵对应元素分别相乘
transpose(), determinant(), inverse()	矩阵的转置，行列式，逆
lessThan(), greaterThan(), equal()	小于，大于，等于（对实参向量对应位置的每个分量做大小比较，生成布尔向量）

课程研发:CC老师
课程授课:CC老师

- 光照基础

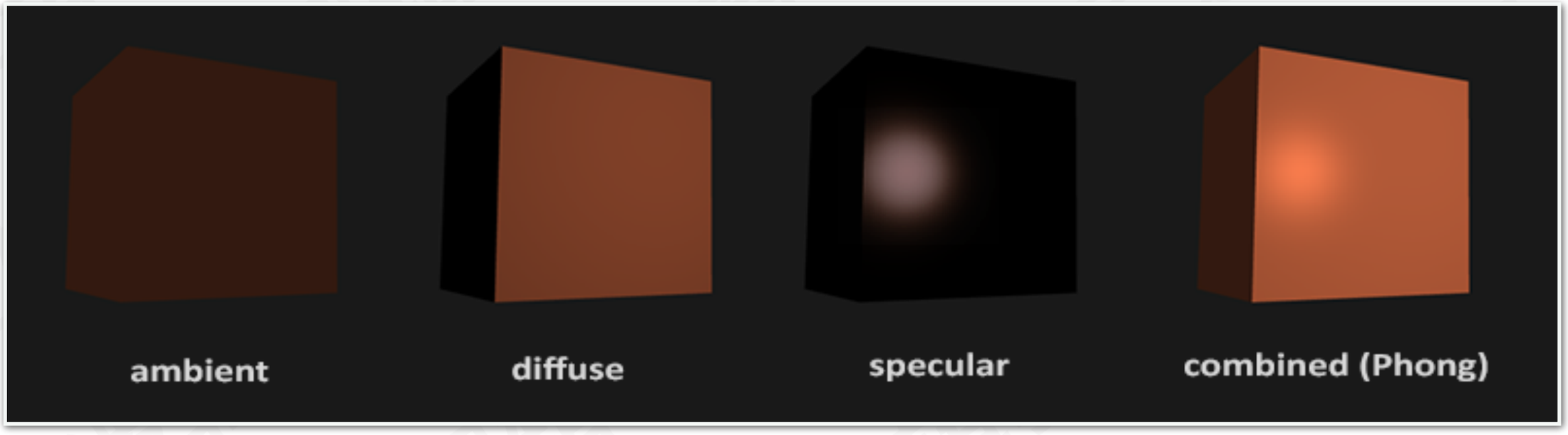
1. 环境光照
2. 漫反射光照
3. 镜面光照



课程研发:CC老师
课程授课:CC老师

- 光照特性

1. 发射光：由物体自身发光
2. 环境光：就是在环境中充分散射的光，而且无法分辨它的方向
3. 漫反射光：光线来自某个方向，但在物体上各个方向反射。
4. 镜面高光：光线来自一个特定的方向，然后在物体表面上以一个特定的方向反射出去



课程研发:CC老师
课程授课:CC老师

- 材质属性

1. 泛射材质
2. 漫反射材质
3. 镜面反射材质
4. 发射材质

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 光照计算

1. 环境光的计算

环境光 = 光源的环境光颜色 * 物体的材质颜色

课程研发:CC老师
课程授课:CC老师

- 光照计算

1. 环境光的计算(亮度)

$$\begin{Bmatrix} R_p \\ G_p \\ B_p \end{Bmatrix} = \text{Pixel Color} \qquad \begin{Bmatrix} R_l \\ G_l \\ B_l \end{Bmatrix} = \text{Light Color}$$

A(亮度) = Ambient intensity

$$\begin{Bmatrix} R_p \\ G_p \\ B_p \end{Bmatrix} \times \begin{Bmatrix} R_l \\ G_l \\ B_l \end{Bmatrix} \times \begin{Bmatrix} A \\ A \\ A \end{Bmatrix} = \text{Final pixel color}$$

课程研发:CC老师
课程授课:CC老师

环境光的GLSL实现

```
varying vec3 objectColor;

void main()
{
    //至少有%10的光找到物体所有面
    float ambientStrength = 0.1;

    //环境光颜色
    vec3 ambient = ambientStrength * lightColor;

     //最终颜色 = 环境光颜色 * 物体颜色
    vec3 result = ambient * objectColor;
    gl_FragColor = vec4(result, 1.0);
}
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 光照计算

1. 发射光的计算

发射颜色 = 物体的反射材质颜色

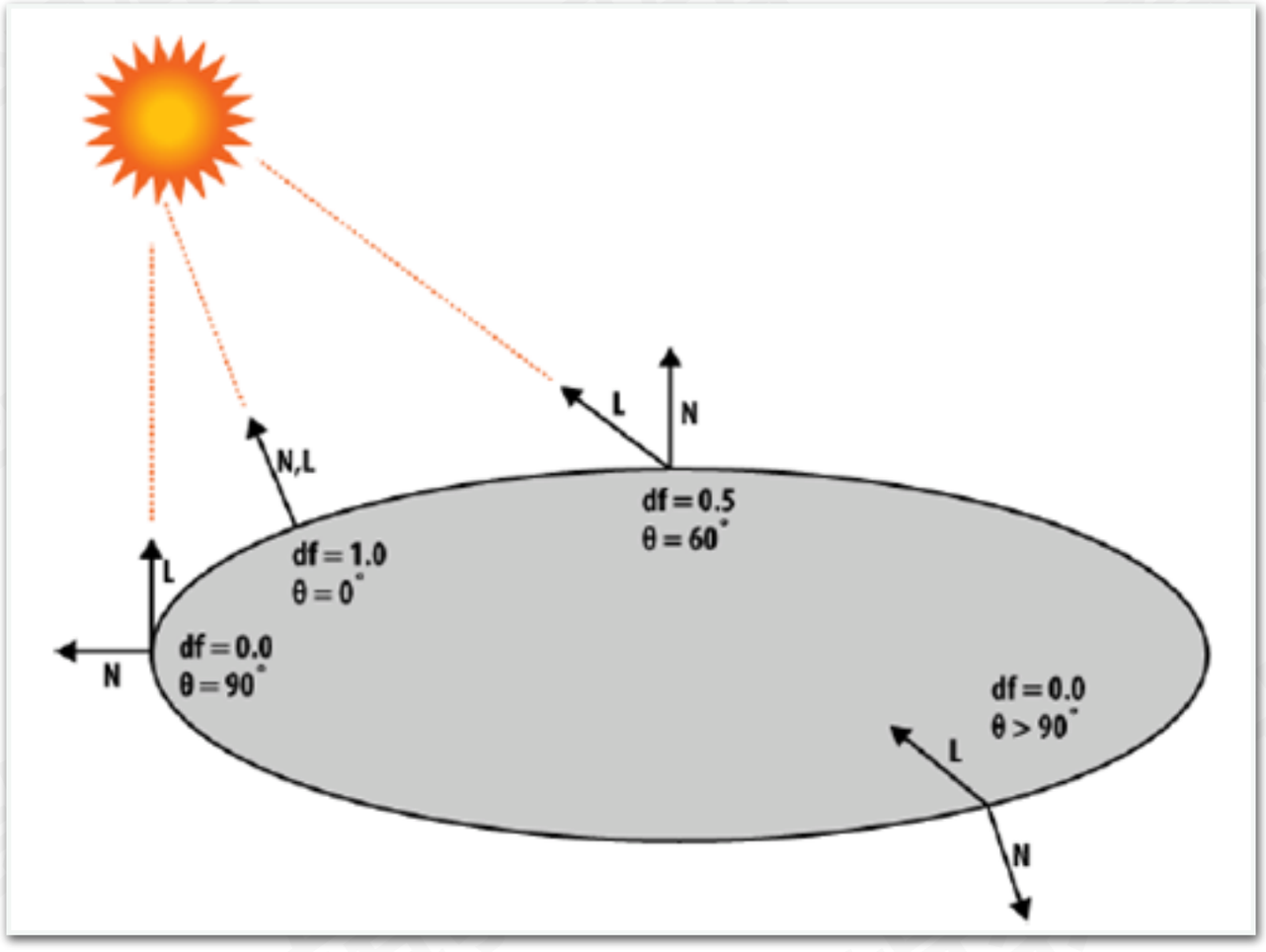
课程研发:CC老师
课程授课:CC老师

- 光照比较

课程研发:CC老师
课程授课:CC老师

- 漫反射光照计算

1. 漫反射光计算



课程研发:CC老师
课程授课:CC老师

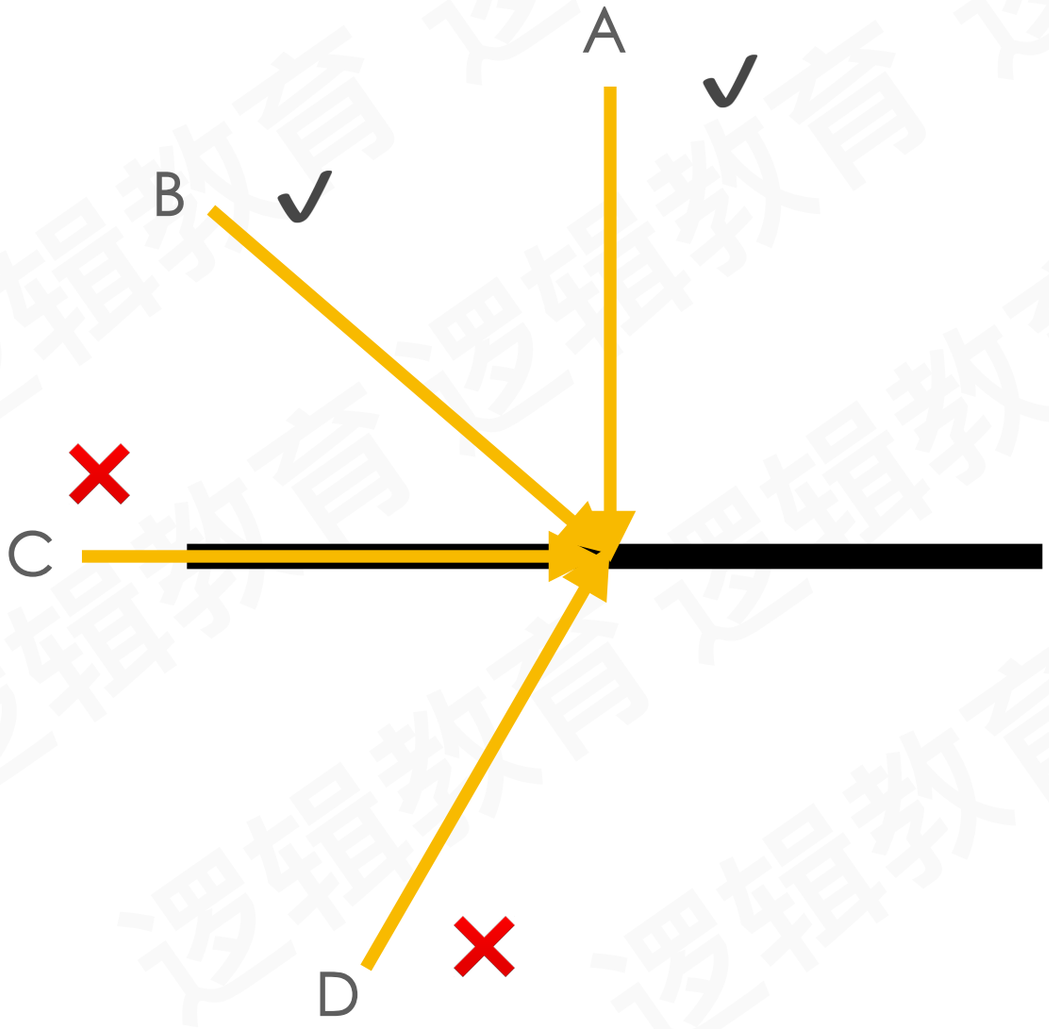
- 漫反射光照计算

光线照射到物体表面,决定物体表面光照的强度,用下图表示

课程研发:CC老师
课程授课:CC老师

- 漫反射光照计算

光照强度是光本身强度和光线与物体表面法线夹角COS的乘积.



结论:

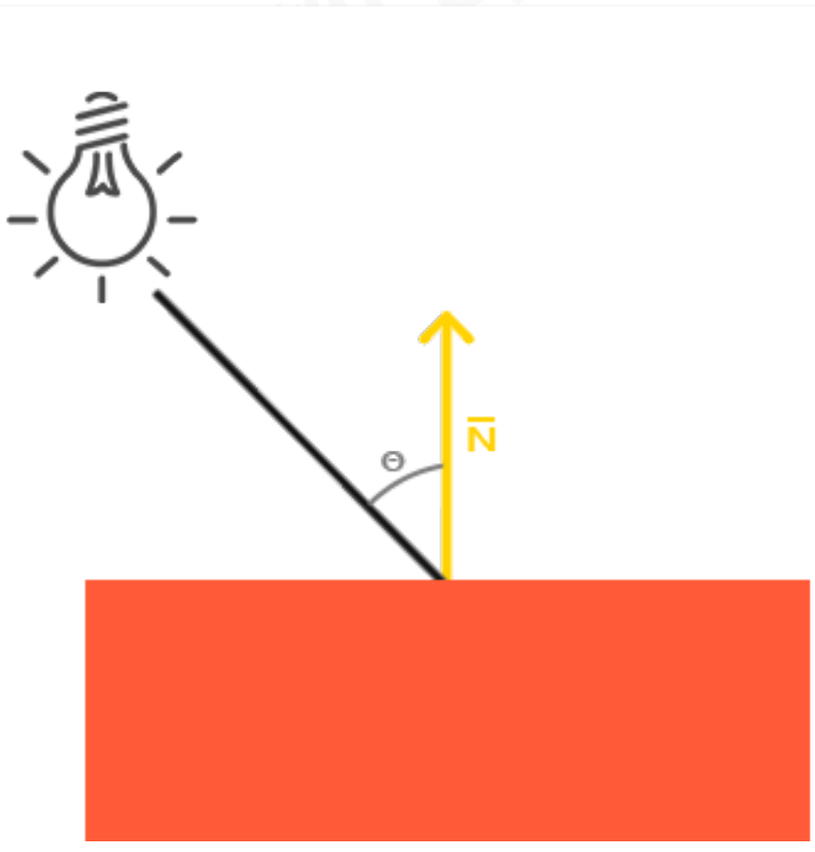
有效的光照方向是与物体表面法线夹角在0~90度之间的

- 漫反射光计算

漫反射颜色 = 光源的漫反射颜色 * 物体的漫发射材质颜色 * DiffuseFactor

DiffuseFactor = max(0,dot(N,L))

漫反射因子DiffuseFactor 是光线 与顶点法线向量的点积



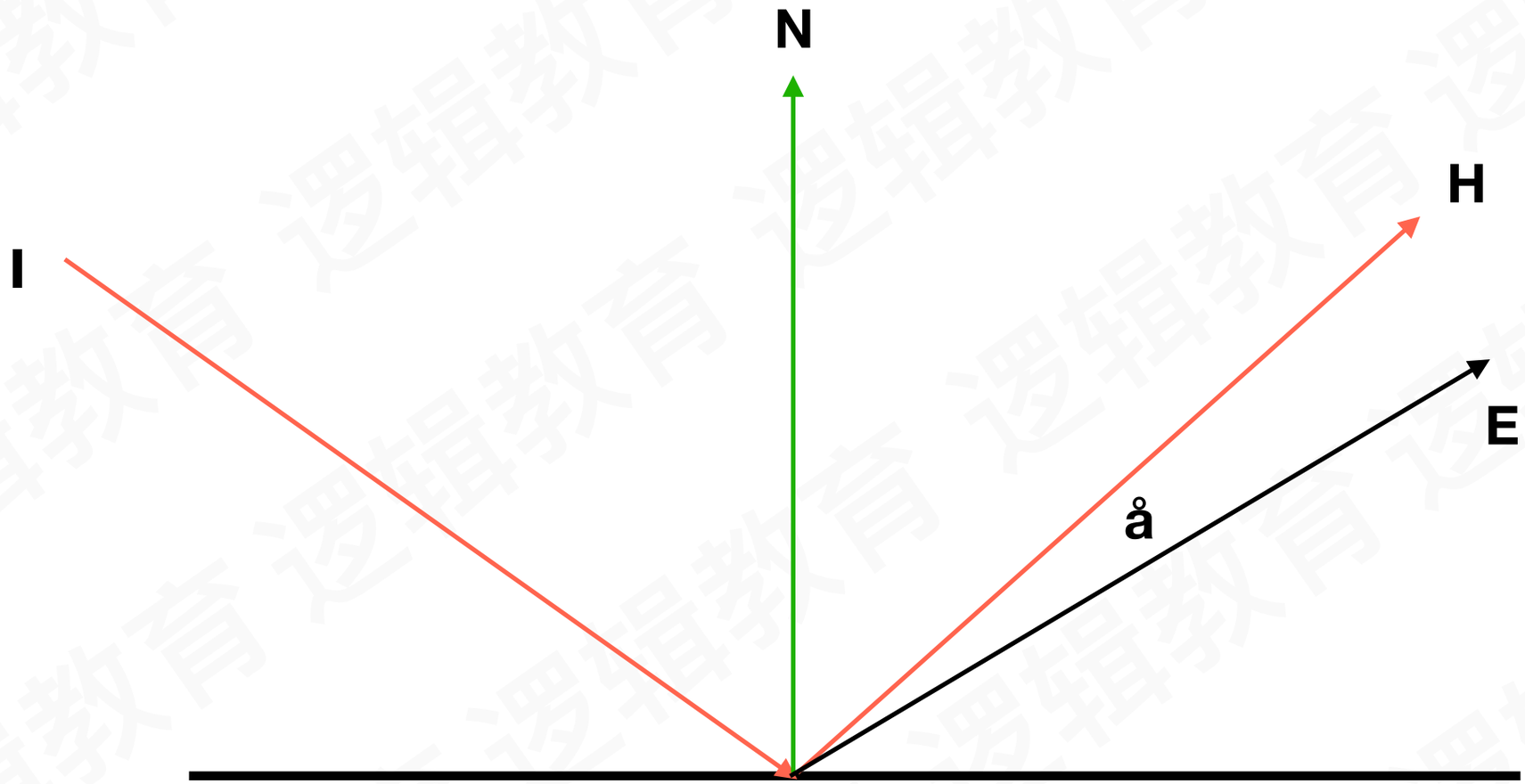
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 漫反射光计算

```
uniform vec3 lightColor;          //光源色
uniform vec3 lightPo;             //光源位置
uniform vec3 objectColor;         //物体色
uniform vec3 viewPo;              //物体位置
varying vec3 outNormal;           //传入当前顶点平面的法向量

//确保法线为单位向量
vec3 norm = normalize(outNormal);
//顶点指向光源 单位向量
vec3 lightDir = normalize(lightPo - FragPo);
//得到两向量的cos值 小于0则则为0
float diff = max(dot(norm, lightDir),0.0);
//得到漫反射收的光源向量
vec3 diffuse = diff * lightColor;

vec3 result  = diffuse * ojbectColor;
gl_FragColor = vec4(result,1.0);
```

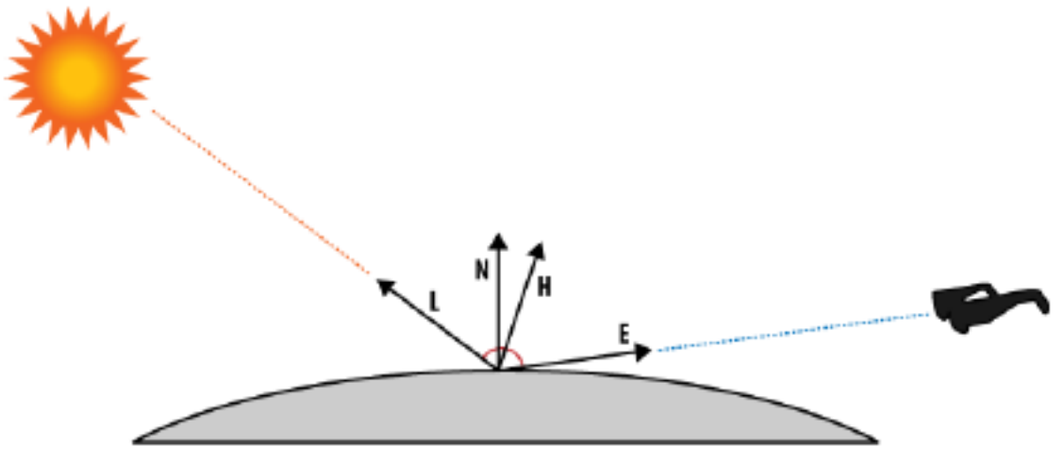
- 镜面光计算

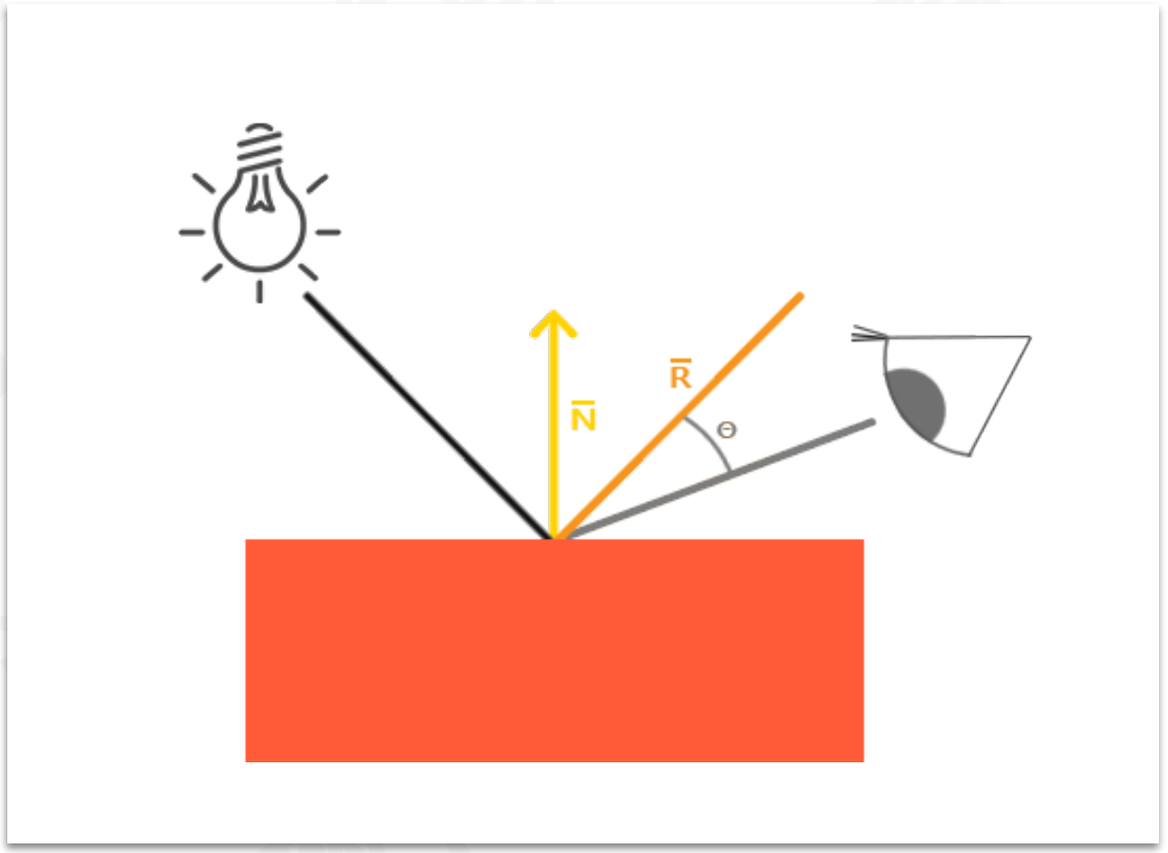


N：平面法线
I ： 入射光线
H：反射光线
E：视线
å：视点与反射光的夹角

镜面光强度与视点位置相关

逻辑教育
Logic education





课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 镜面光计算

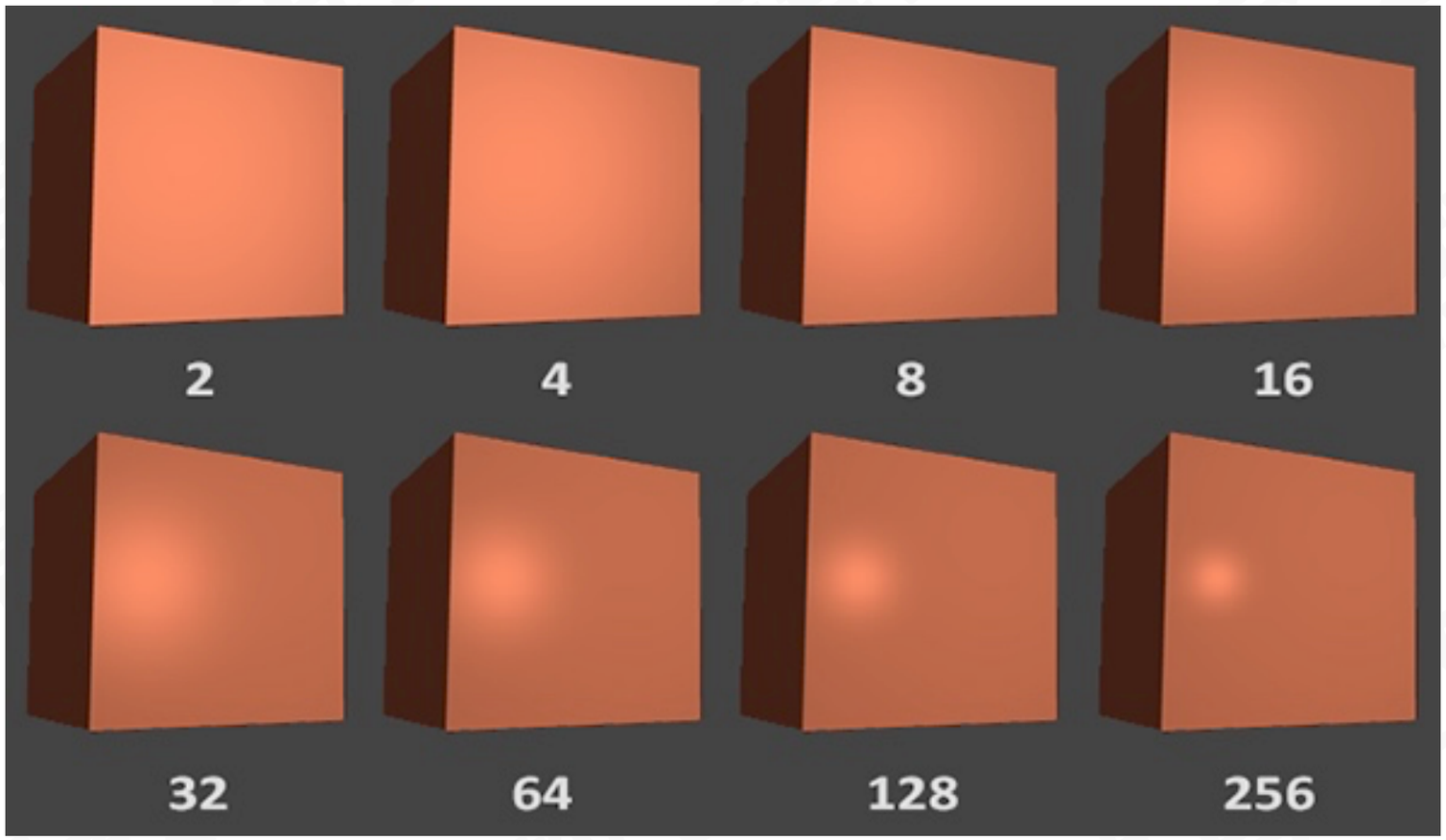
镜面反射颜色 = 光源的镜面光的颜色 * 物体的镜面材质颜色 * SpecularFactor

SpecularFactor = power(max(0,dot(N,H)),shininess)

H： 视线向量E 与 光线向量L 的半向量
dot(N,H):H,N的点积几何意义,平方线与法线夹角的cos值
shiniess：高光的反光度;

课程研发:CC老师
课程授课:CC老师

- 镜面光计算



课程研发:CC老师
课程授课:CC老师

- 镜面光计算

```
//镜面强度
float specularStrength = 0.5;

//顶点指向观察点的单位向量
vec3 viewDir = normalize(viewPo - FragPo);

//求得光线 在 顶点的反射线(传入光源指向顶点的向量)
vec3 reflectDir = reflect(-lightDir ,outNormal);

// 求得夹角cos值 取256次幂 注意 pow(float,float)函数参数类型
float spec = pow(max(dot(viewDir, reflectDir),0.0),256.0);

vec3 specular = specularStrength * spec * lightColor;
```

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 光照计算

光照颜色 =(环境颜色 + 漫反射颜色 + 镜面反射颜色)* 衰减因子

课程研发:CC老师
课程授课:CC老师

- 衰减因子

$$F_{att} = \frac{1.0}{K_c + K_l * d + K_q * d^2}$$

衰减因子 = 1.0/(距离衰减常量 + 线性衰减常量 * 距离 + 二次衰减常量 * 距离的平方)

距离衰减常量，线性衰减常量和二次衰减常量均为常量值

- 注意

环境光，漫反射光和镜面光的强度都会受距离的增大而衰减，只有发射光和全局环境光的强度不会受影响

- 衰减因子计算

```
//距离衰减常量
float constantPara = 1.0f;
//线性衰减常量
float linearPara = 0.09f;
 //二次衰减因子
 float quadraticPara = 0.032f;
 //距离
 float LFDistance = length(lightPo - FragPo);
//衰减因子
 float lightWeakPara = 1.0/(constantPara + linearPara
* LFDistance + quadraticPara * (LFDistance*LFDistance));
```

课程研发:CC老师
课程授课:CC老师

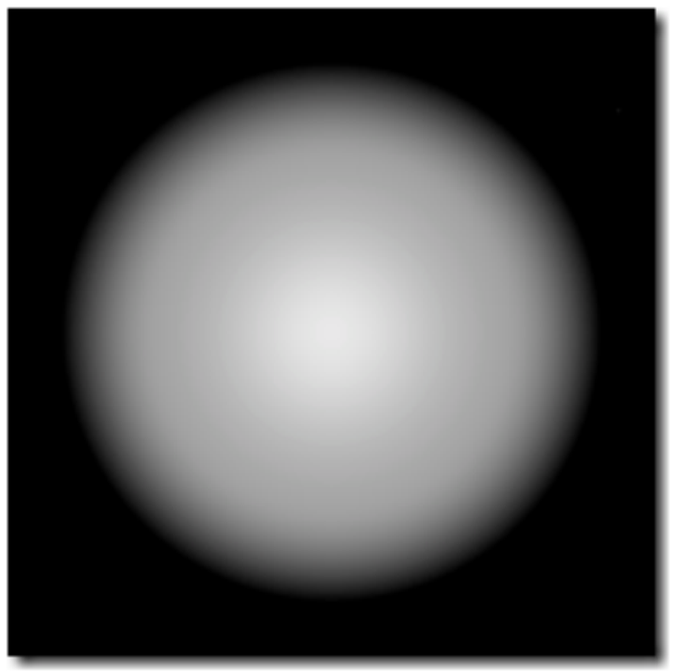
- 聚光灯因子

聚光灯夹角cos值 = power(max(0,dot(单位光源位置，单位光线向量)),聚光灯指数);

- 单位光线向量是从光源指向顶点的单位向量
- 聚光灯指数，表示聚光灯的亮度程度
- 公式解读：单位光源位置 * 单位光线向量 点积 的 聚光灯指数次方。

课程研发:CC老师
课程授课:CC老师

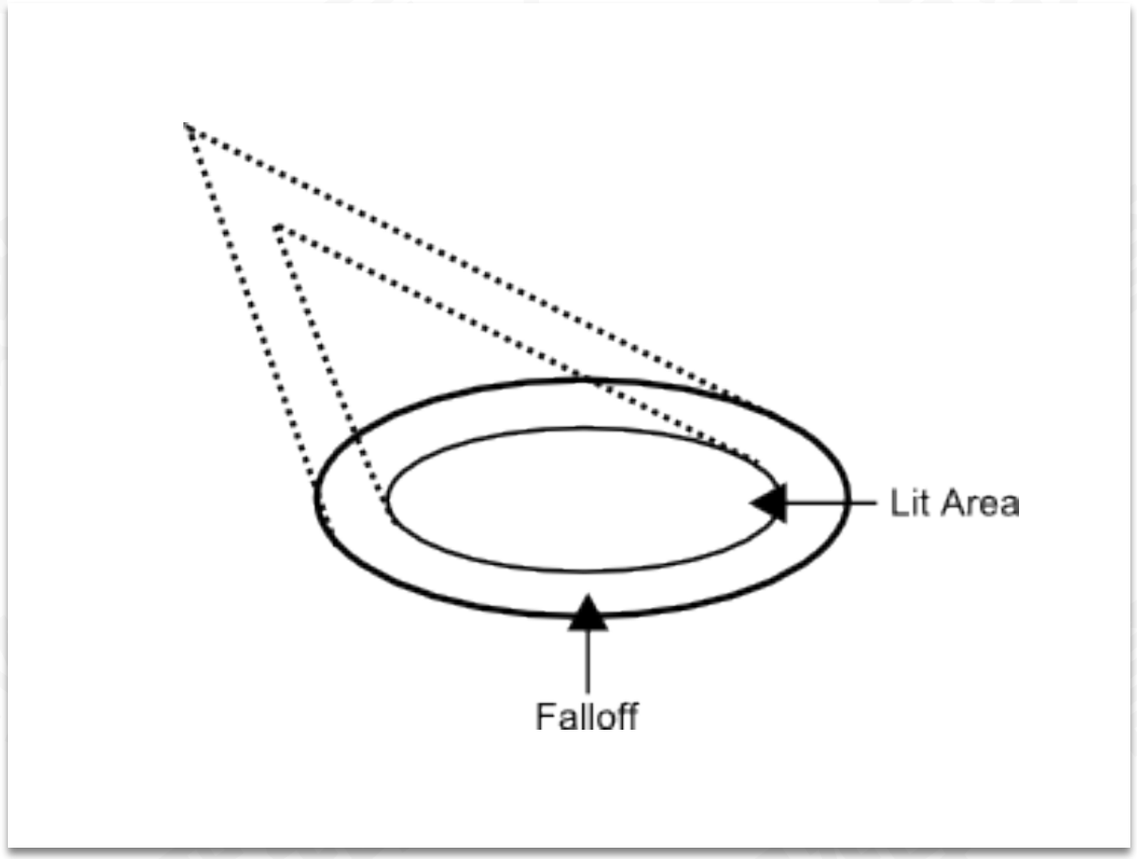
- 聚光灯无过渡 与 有过渡处理

无过渡	有过渡
	

课程研发:CC老师
课程授课:CC老师

- 增加过渡计算

聚光灯因子 = clamp((外环的聚光灯角度cos值 - 当前顶点的聚光灯角度cos值)/(外环的聚光灯角度cos值- 内环聚光灯的角度的cos值),0,1);



课程研发:CC老师
课程授课:CC老师

• 聚光灯过渡计算

```
//(一些复杂的计算操作 应该让CPU做,提高效率,不变的量也建议外部传输,避免
重复计算)
//内锥角cos值
float inCutOff = cos(radians(10.0f));
//外锥角cos值
float outCutOff = cos(radians(15.0f));
//聚光朝向
 vec3 spotDir = vec3(-1.2f,-1.0f,-2.0f);

//光源指向物体的向量 和 聚光朝向的 cos值
float theta = dot(lightDir ,normalize(-spotDir));
//内外锥角cos差值
float epsilon  = inCutOff - outCutOff;

//clamp(a,b,c);若b<a<c 则函数返回值为a 若不是,则返回值最小为b 最大
为c
// (theta - outCutOff)/epsilon 若theta的角度小于内锥角 则其值
>=1 若theta的角度大于外锥角 则其值<=0 这样光线就在内外锥角之间平滑变
化.
float intensity = clamp((theta - outCutOff)/epsilon,
0.0,1.0);
```

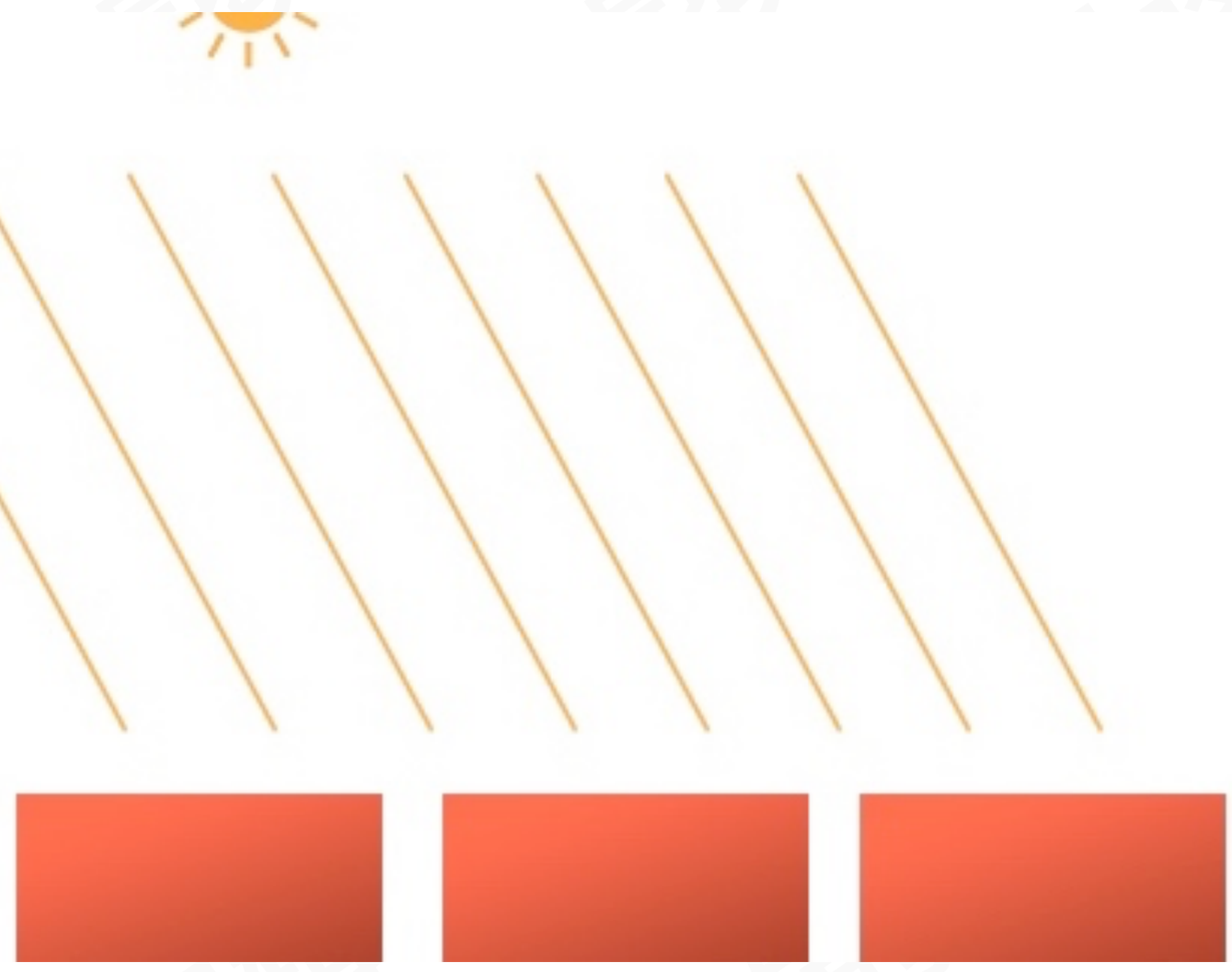
- 光照计算终极公式

光照颜色 = 发射颜色 + 全局环境颜色 + (环境颜色 + 漫反射颜色 + 镜面反射颜色) * 聚光灯效果 * 衰减因子

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

- 平面光终极公式



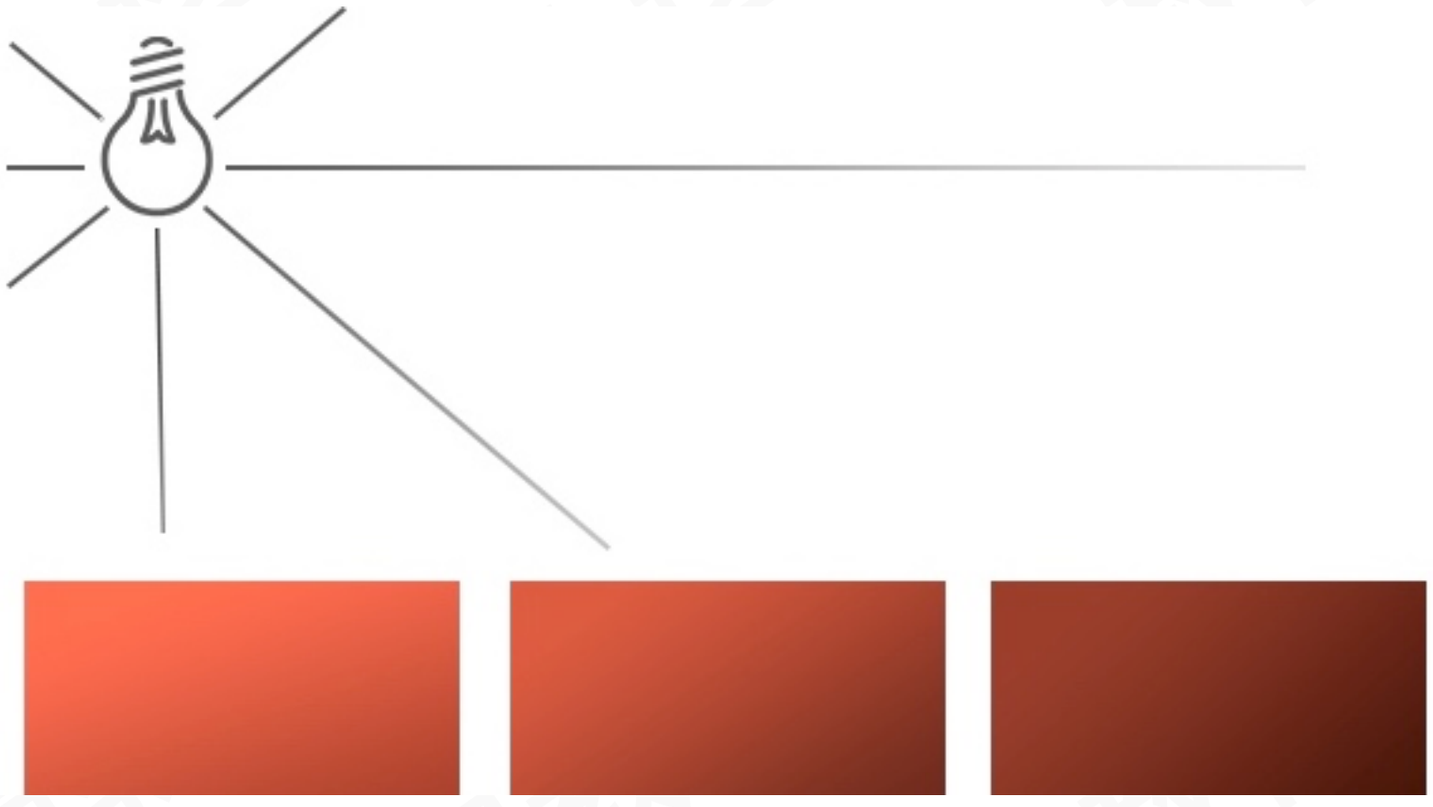
课程研发:CC老师
课程授课:CC老师

• 平面光计算代码实现

```
//环境因子
float ambientStrength = 0.3;
//镜面强度
float specularStrength = 2.0;
//反射强度
float reflectance = 256.0;

//平行光方向
vec3 paraLightDir = normalize(vec3(-0.2,-1.0,-0.3));

//环境光
vec3 ambient = ambientStrength * texture(Texture ,outTexCoord).rgb;

//漫反射
vec3 norm = normalize(outNormal);
vec3 lightDir = normalize(lightPo - FragPo);    //当前顶点 至 光源的的单位向量
float diff = max(dot(norm ,paraLightDir),0.0);
vec3 diffuse = diff * lightColor*texture(Texture ,outTexCoord).rgb;

//镜面反射
vec3 viewDir = normalize(viewPo - FragPo);
vec3 reflectDir = reflect(-paraLightDir ,outNormal);
float spec = pow(max(dot(viewDir, reflectDir),0.0),reflectance);
vec3 specular = specularStrength * spec * texture(specularTexture ,outTexCoord).rgb;

//最终光照颜色
vec3 res = ambient + diffuse + specular;

FragColor = vec4(res,1.0);
```

逻辑教育
Logic education

- 点光源终极公式

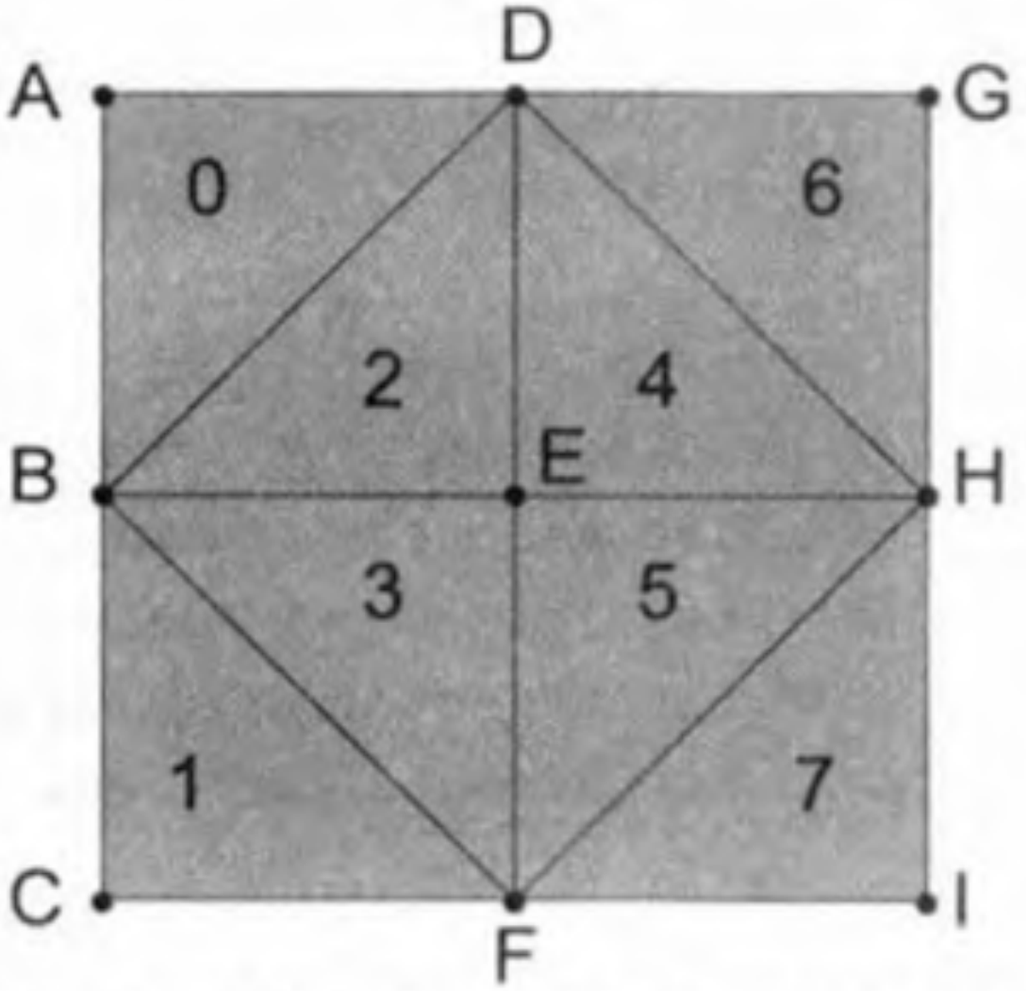


课程研发:CC老师
课程授课:CC老师

- 点光源计算代码实现

```
float ambientStrength = 0.3;    //环境因子
float specularStrength = 2.0;   //镜面强度
float reflectance = 256.0;      //反射强度

float constantPara = 1.0f;    //常亮
float linearPara = 0.09f;     //线性部分因数
float quadraticPara = 0.032f; //二次项部分因数

//环境光
vec3 ambient = ambientStrength * texture(Texture ,outTexCoord).rgb;

//漫反射
vec3 norm = normalize(outNormal);
vec3 lightDir = normalize(lightPo - FragPo);    //当前顶点 至 光源的的单位向量

//点光源
float diff = max(dot(norm ,lightDir),0.0);   //光源与法线夹角
vec3 diffuse = diff * lightColor*texture(Texture ,outTexCoord).rgb;

//镜面反射
vec3 viewDir = normalize(viewPo - FragPo);
vec3 reflectDir = reflect(-lightDir ,outNormal);

float spec = pow(max(dot(viewDir, reflectDir),0.0),reflectance);
vec3 specular = specularStrength * spec * texture(specularTexture ,outTexCoord).rgb;

//光线衰弱
float LFDistance = length(lightPo - FragPo);
float lightWeakPara = 1.0/(constantPara + linearPara * LFDistance + quadraticPara *
(LFDistance*LFDistance));

vec3 res = (ambient + diffuse + specular)*lightWeakPara;

FragColor = vec4(res,1.0);
```

- 案例实战



逻辑教育
Logic education

- 案例总结

参考思维导图

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

课程研发:CC老师
课程授课:CC老师