

# iOS开发-OpenGLES进阶教程2



作者 落影loyinglin (/u/815d10a4bdce) [+ 关注](#)

2016.04.18 09:40 字数 877 阅读 1573 评论 1 喜欢 5 阅读 1573 评论 1 喜欢 5 (/u/815d10a4bdce)

## 教程

- OpenGL ES入门教程1-Tutorial01-GLKit (<http://www.jianshu.com/p/750fde1d8b6a>)
- OpenGL ES入门教程2-Tutorial02-shader入门 (<http://www.jianshu.com/p/ee597b2bd399>)
- OpenGL ES入门教程3-Tutorial03-三维变换 (<http://www.jianshu.com/p/87c5413c1fc7>)
- OpenGL ES入门教程4-Tutorial04-GLKit进阶 (<http://www.jianshu.com/p/ed7fb9555839>)
- OpenGL ES进阶教程1-Tutorial05-地球月亮 (<http://www.jianshu.com/p/a82f3f66ddddd>)

这一次的的内容是光照。

## 概念准备

所谓的光照，是GPU为每个三角形的顶点进行**光线计算**，再把结果进行插值，得出每个片元的最终颜色。

OpenGL ES的灯光模拟包括：环境光、漫反射光、镜面反射光。

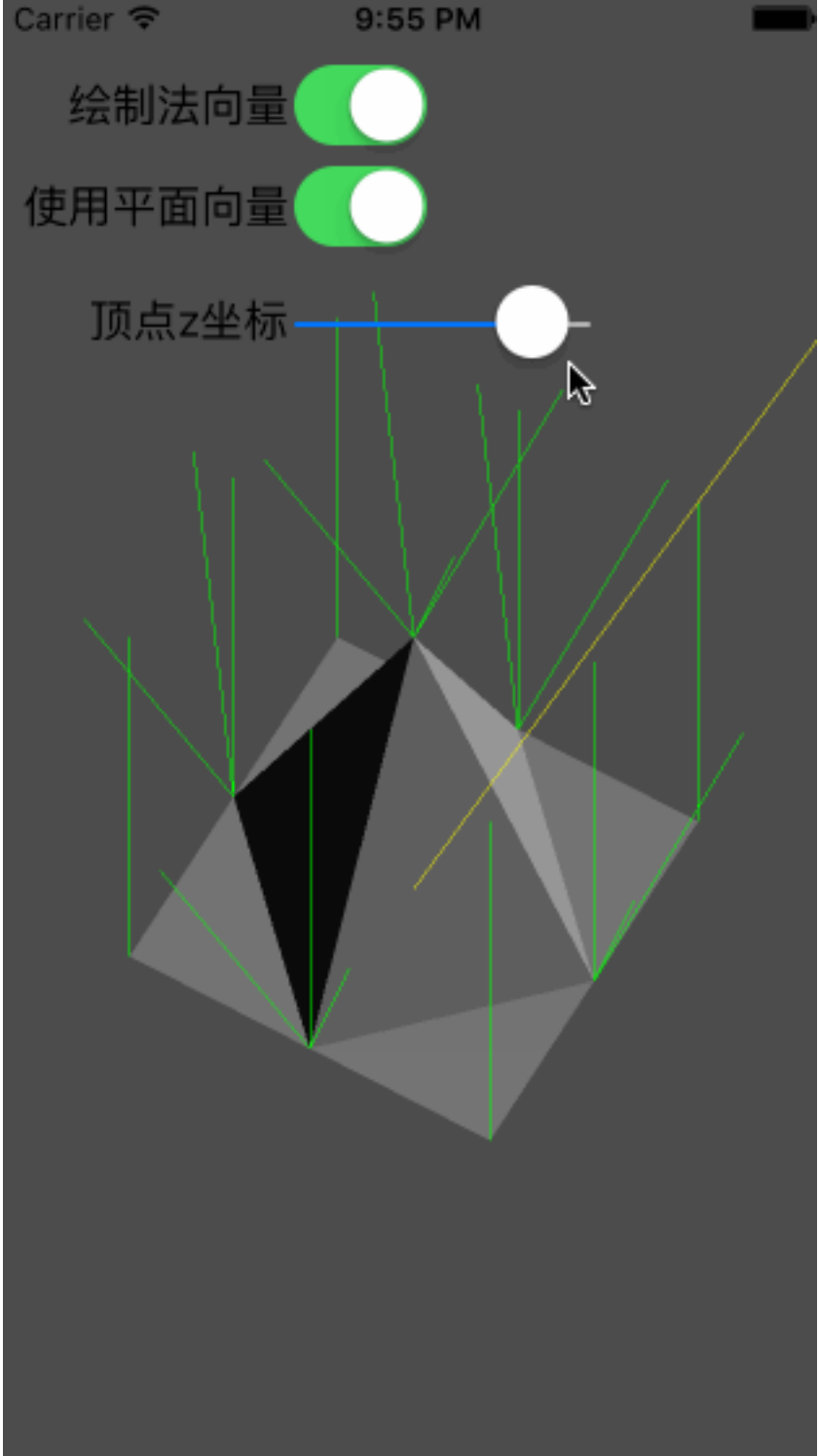
光线与几何图形相互作用的关键：**计算出每个物体照射和发散出来多少光线。**

光线计算依赖于**表面法向量**。法向量也是**单位向量**。

表面法向量可以通过平面内两个点的叉积（矢量积）来计算。

光线计算过程还包括材质、聚光灯效果、衰减因子等，但是**GLKit**简化了这一过程。

## 效果展示



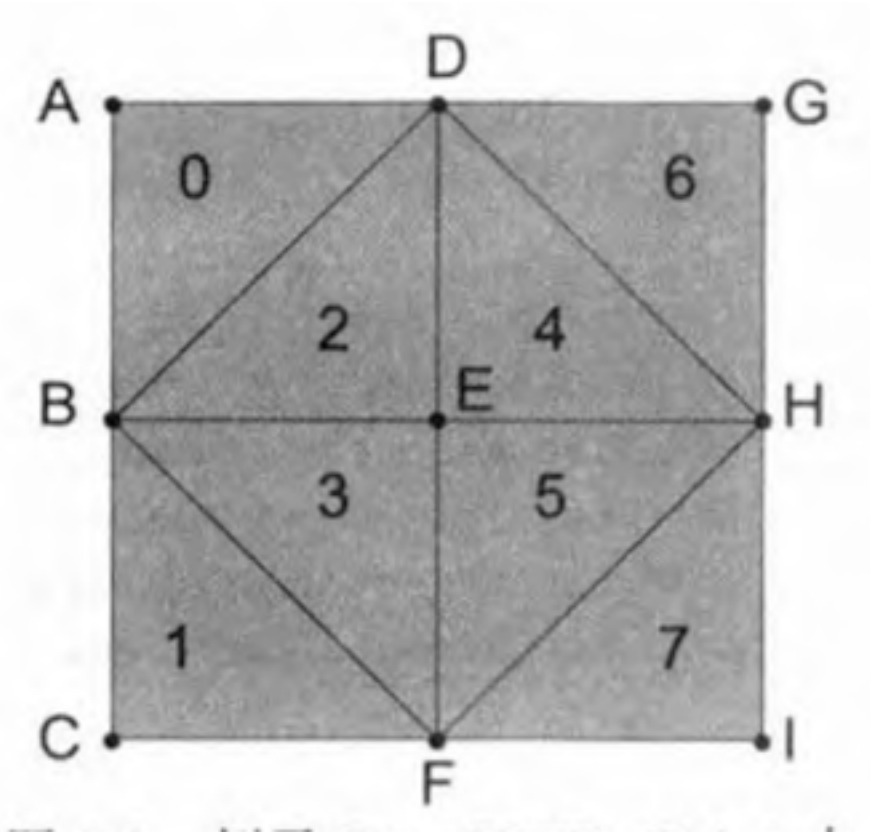
光照

## 核心思路

如下图，总共A~I九个点，0~7八个面。E的z坐标是可变的，随着UISlider的数值由**-1到0**变化。

为了方便观察，绕**X轴和Z轴**旋转了一定角度。

灯光用GLKBaseEffect类。



顶点-平面图

## 具体细节

- 顶点

总共有9个顶点，具体坐标如下。前面为顶点坐标，后面为法线坐标。

```
static const SceneVertex vertexA = {{-0.5, 0.5, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexB = {{-0.5, 0.0, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexC = {{-0.5, -0.5, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexD = {{ 0.0, 0.5, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexE = {{ 0.0, 0.0, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexF = {{ 0.0, -0.5, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexG = {{ 0.5, 0.5, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexH = {{ 0.5, 0.0, -0.5}, {0.0, 0.0, 1.0}};
static const SceneVertex vertexI = {{ 0.5, -0.5, -0.5}, {0.0, 0.0, 1.0}};
```

重新缓存顶点数组

- 平面

SceneVertex是顶点的数据结构

SceneTriangle是平面(三角形)的数据结构

```
//顶点
typedef struct {
    GLKVector3 position; //
    GLKVector3 normal;
}
SceneVertex;
//三角形
typedef struct {
    SceneVertex vertices[3];
}
SceneTriangle;
```

- 光源

配置漫反射光的颜色，还有光源的位置

```
self.baseEffect = [[GLKBaseEffect alloc] init];
self.baseEffect.light0.enabled = GL_TRUE;
self.baseEffect.light0.diffuseColor = GLKVector4Make(
    0.7f, // Red
    0.7f, // Green
    0.7f, // Blue
    1.0f); // Alpha

self.baseEffect.light0.position = GLKVector4Make(
    1.0f,
    1.0f,
    0.5f,
    0.0f);

self.extraEffect = [[GLKBaseEffect alloc] init];
self.extraEffect.useConstantColor = GL_TRUE;
```

- 变换

先旋转，后平移

```
GLKMatrix4 modelViewMatrix = GLKMatrix4MakeRotation(GLKMathDegreesToRadians(-60.0f), 1.0f, 0.0f, 0.0f);
modelViewMatrix = GLKMatrix4Rotate(
    modelViewMatrix,
    GLKMathDegreesToRadians(-30.0f), 0.0f,
    0.0f, 1.0f);
modelViewMatrix = GLKMatrix4Translate(
    modelViewMatrix,
    0.0f, 0.0f, 0.25f);

self.baseEffect.transform.modelviewMatrix = modelViewMatrix;
self.extraEffect.transform.modelviewMatrix = modelViewMatrix;
```

- 法线绘制

先设置光源颜色为绿色，画顶点法线

再设置光源颜色为黄色，画光源线

```
self.extraEffect.useConstantColor = GL_TRUE;
self.extraEffect.constantColor =
GLKVector4Make(0.0, 1.0, 0.0, 1.0);

[self.extraEffect prepareToDraw];

[self.extraBuffer drawArrayWithMode:GL_LINES
    startVertexIndex:0
    numberOfVertices:NUM_NORMAL_LINE_VERTS];

self.extraEffect.constantColor =
GLKVector4Make(1.0, 1.0, 0.0, 1.0);

[self.extraEffect prepareToDraw];

[self.extraBuffer drawArrayWithMode:GL_LINES
    startVertexIndex:NUM_NORMAL_LINE_VERTS
    numberOfVertices:(NUM_LINE_VERTS - NUM_NORMAL_LINE_VERTS)];
```

关键函数

- 求法向量函数

```
GLKVector3 SceneTriangleFaceNormal(const SceneTriangle triangle);
```

- 通过叉积求单位法向量函数

```
GLKVector3 SceneVector3UnitNormal(  
    const GLKVector3 vectorA,  
    const GLKVector3 vectorB)
```

总结

进阶教程不只是图形学知识的进阶，代码的规范也很重要，能避免一部分错误。

光照原理的内容可以参考[这里](#)

([http://www.cnblogs.com/kesalin/archive/2012/12/29/light\\_theroy.html](http://www.cnblogs.com/kesalin/archive/2012/12/29/light_theroy.html)), 讲解非常详细，但是本次使用的GLKit，所以简化了许多。

附上源码 (<https://github.com/loyinglin/LearnOpenGLS/tree/master/Tutorial06-%E5%85%89%E7%85%A7>)



落影loyinglin (/u/815d10a4bdce)

写了 171405 字，被 4763 人关注，获得了 2744 个喜欢

(/u/815d10a4bdce) 写了 171405 字，被 4763 人关注，获得了 2744 个喜欢

+ 关注

工程师一枚，喜欢思考，喜欢游戏，喜欢运动。做过什么已经不重要，未来的方向以及当下的准备是生活的...

♡ 喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button)

5








更多分享


被以下专题收入，发现更多相似内容




iOS Dev... (/c/3233d1a249ca?utm\_source=desktop&utm\_medium=notes-included-collection)




iOS学习 (/c/1332c736fe39?utm\_source=desktop&utm\_medium=notes-included-collection)




游戏开发 (/c/9336cf7430c4?utm\_source=desktop&utm\_medium=notes-included-collection)




JZ专题 (/c/0c10836ef5a8?utm\_source=desktop&utm\_medium=notes-included-collection)



iOS 开发 (/c/2ffaa203eb6a?utm\_source=desktop&utm\_medium=notes-included-collection)




OpenGL ... (/c/044a5240577d?utm\_source=desktop&utm\_medium=notes-included-collection)




iOS开发专题 (/c/c258bc0ea6bd?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多 ▾

 登录/注册

为你个性化推荐内容

(/sign\_in?utm\_source=desktop&utm\_medium=note-app-download)

 下载简书App

随时随地发现和创作内容

(/apps/download?utm\_source=desktop&utm\_medium=click-note-bottom-bind)