

iOS开发-OpenGL ES入门教程4



作者 落影loyinglin (/u/815d10a4bdce) + 关注

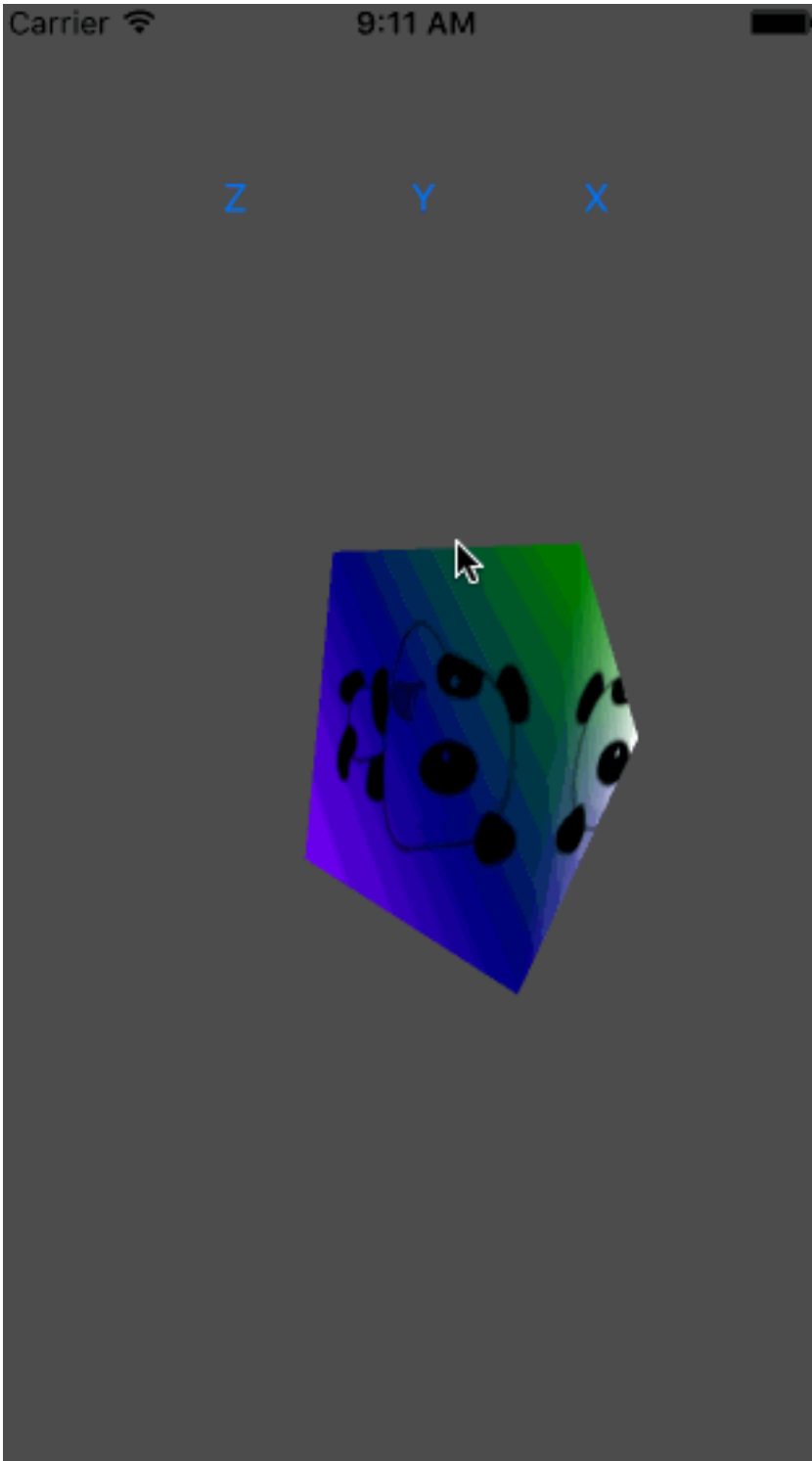
2016.04.06 10:05* 字数 1167 阅读 3991 评论 13 喜欢 15 赞赏 1 阅读 3991 评论 13 喜欢 15 赞赏 1 (/u/815d10a4bdce)

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>)
这次我们用GLKit，更简单的实现图形变换、纹理贴图、着色、深度测试(代码在这
(<https://github.com/loyinglin/LearnOpenGLES/tree/master/Tutorial04-GLKit%E8%BF%9B%E9%98%B6>))。

OpenGL ES系列教程在这里 (<http://www.jianshu.com/notebooks/2135411/latest>)。
OpenGL ES系列教程的代码地址 (<https://github.com/loyinglin/LearnOpenGLES>) - 你的star和fork是我的源动力，你的意见能让我走得更远。

效果展示



核心思路

使用GLKit来进行图形变换、纹理贴图加载、深度测试，用GLKBaseEffect来管理纹理贴图和进行着色。

具体细节

1、顶点属性

```
typedef NS_ENUM(GLint, GLKVertexAttrib)
{
    GLKVertexAttribPosition,
    GLKVertexAttribNormal,
    GLKVertexAttribColor,
    GLKVertexAttribTexCoord0,
    GLKVertexAttribTexCoord1
} NS_ENUM_AVAILABLE(10_8, 5_0);
```

GLKEffects用到的顶点属性已经定义好，使用时直接用枚举量赋值。如下图，直接对顶点的位置、颜色、纹理坐标进行赋值。

```
glEnableVertexAttribArray(GLKVertexAttribPosition);
glVertexAttribPointer(GLKVertexAttribPosition, 3, GL_FLOAT, GL_FALSE, sizeof(
GLfloat) * 8, (GLfloat *)NULL);
//顶点颜色
glEnableVertexAttribArray(GLKVertexAttribColor);
glVertexAttribPointer(GLKVertexAttribColor, 3, GL_FLOAT, GL_FALSE, 4 * 8, (GL
float *)NULL + 3);

glEnableVertexAttribArray(GLKVertexAttribTexCoord0);
glVertexAttribPointer(GLKVertexAttribTexCoord0, 2, GL_FLOAT, GL_FALSE, 4 * 8,
(GLfloat *)NULL + 6);
```

2、纹理

在自定义shader中使用纹理，需要用CoreGraphics把图像转换成bitmapdata，再申请纹理内存，把图像数据传进去，最后还要释放bitmapdata。

在GLKit中，仅仅需要如下三行代码，就可以完成纹理的加载。

```
//纹理
NSString* filePath = [[NSBundle mainBundle] pathForResource:@"for_test" ofType:@"png"];
NSDictionary* options = [NSDictionary dictionaryWithObjectsAndKeys:@(1), GLKTextureLoaderOriginBottomLeft, nil];
GLKTextureInfo* textureInfo = [GLKTextureLoader textureWithContentsOfFile:filePath options:options error:nil];
```

GLKTextureLoaderOriginBottomLeft 参数是避免纹理上下颠倒，原因是纹理坐标系和世界坐标系的原点不同。

最后创建着色器，启用纹理，把刚刚创建的textureInfo的name赋值给着色器。

```
//着色器
self.mEffect = [[GLKBaseEffect alloc] init];
self.mEffect.texture2d0.enabled = GL_TRUE;
self.mEffect.texture2d0.name = textureInfo.name;
```

3、图形变换

在OpenGL ES里面，图形变换的表现形式就是矩阵操作，GLKit也提供了很多矩阵操作函数。

```
//初始的投影
CGSize size = self.view.bounds.size;
float aspect = fabs(size.width / size.height);
GLKMatrix4 projectionMatrix = GLKMatrix4MakePerspective(GLKMathDegreesToRadians(90.0), aspect, 0.1f, 10.f);
projectionMatrix = GLKMatrix4Scale(projectionMatrix, 1.0f, 1.0f, 1.0f);
self.mEffect.transform.projectionMatrix = projectionMatrix;

GLKMatrix4 modelViewMatrix = GLKMatrix4Translate(GLKMatrix4Identity, 0.0f, 0.0f, -2.0f);
self.mEffect.transform.modelviewMatrix = modelViewMatrix;
```

GLKMatrix4MakePerspective 是透视投影变换

GLKMatrix4Translate 是平移变换

```
/**
 * 场景数据变化
 */
- (void)update {
    GLKMatrix4 modelViewMatrix = GLKMatrix4Translate(GLKMatrix4Identity, 0.0f, 0.0f, -2.0f);

    modelViewMatrix = GLKMatrix4RotateX(modelViewMatrix, self.mDegreeX);
    modelViewMatrix = GLKMatrix4RotateY(modelViewMatrix, self.mDegreeY);
    modelViewMatrix = GLKMatrix4RotateZ(modelViewMatrix, self.mDegreeZ);

    self.mEffect.transform.modelviewMatrix = modelViewMatrix;
}
```

在场景变换函数里面， GLKMatrix4RotateY 是绕Y轴旋转，其他的分分别是绕X、Z轴旋转。

4、深度测试

在前面的教程介绍过，开启深度测试需要分配深度测试的缓冲区，并挂载到相应的帧缓冲区。

在GLKit代码中，深度测试的开启十分简单。

在新建上下文时调用 glEnable(GL_DEPTH_TEST); 开启深度测试。

```
//新建OpenGL ES 上下文
self.mContext = [[EAGLContext alloc] initWithAPI:kEAGLRenderingAPIOpenGLES2];

GLKView* view = (GLKView *)self.view;
view.context = self.mContext;
view.drawableColorFormat = GLKViewDrawableColorFormatRGBA8888;
view.drawableDepthFormat = GLKViewDrawableDepthFormat24;

[EAGLContext setCurrentContext:self.mContext];
glEnable(GL_DEPTH_TEST);
```

在渲染场景时， glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); glClear参数加入 GL_DEPTH_BUFFER_BIT 即可。

```
/**
 * 渲染场景代码
 */
- (void)glkView:(GLKView *)view drawInRect:(CGRect)rect {
    glClearColor(0.3f, 0.3f, 0.3f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    [self.mEffect prepareToDraw];
    glDrawElements(GL_TRIANGLES, self.mCount, GL_UNSIGNED_INT, 0);
}
```

总结

OpenGL ES的入门已经差不多，后面还有很多很多知识，需要时再学习即可。

总结这几篇教程花了一个星期左右。

学习过程中有几部分最难受：

- 第一部分是OpenGL ES的顶点属性、纹理贴图，对顶点到图形的过程不理解，对OpenGL ES的数据缓存机制不了解， **无知容易使人知难而退**；
- 第二部分是shader和glsl，glsl无法调试、编译信息不会查看、语法不懂等等，一个1.0 + 1都会报错，自己却莫名其妙，只能通过二分注释代码来定位问题，特别让人泄气；
- 第三部分是OpenGL ES的三维图形变换和光照等，这部分更多的是数学知识和物理知识，代码一般都很简单；这部分比前面的都难，但是因为知道哪里缺失，虽然不懂，但是不慌，只是有点难受。

最后的感想，**计算机图形学和线性代数**毕竟是基础，自己多花点时间学习，培养为核心竞争力非常不错。



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) | 15

   更多分享

(http://cwb.assets.jianshu.io/notes/images/3479955

被以下专题收入，发现更多相似内容

-  首页投稿 (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS Dev... (/c/3233d1a249ca?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)
-  OPenGL ... (/c/408442c9c764?utm_source=desktop&utm_medium=notes-included-collection)
-  IOS个人开发 (/c/b76f9973bf0f?utm_source=desktop&utm_medium=notes-included-collection)
-  OpenGL (/c/c2cd8d71b0c0?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

推荐阅读

更多精彩内容 > (/)

使用AudioToolbox播放AAC (/p/279a9e5b36b5?utm_campaign=maleskine&utm_content=note&utm_source=desktop)

前言 使用VideoToolbox硬编码H.264使用VideoToolbox硬解码H.264使用AudioToolbox编码AAC在上一篇中，介绍了如何从麦克风采集声音并用

落影loyinglin (/u/815d10a4bdce?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

GPUImage详细解析（二） (/p/1eea8bf8451e?utm_campaign=maleskine&utm_content=note&utm_source=desktop)

解析GPUImage详细解析（一） 上一篇介绍的是GPUImageFramebuffer和GPUImageFilter。简单回顾一下：GPUImageFilter就是用来接收源图像，通过

落影loyinglin (/u/815d10a4bdce?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

学了这个技能，秒杀90%的人！ (/p/6ca0b98076fd?utm_campaign=maleskine&utm_content=note&utm_source=desktop)

先抛出一个观点：每个人都应该看一看行业趋势报告！ 2011年我在一个上市公司实习，有一周的时间领导都只让我做一件事：找地产行业报告和数据 那是我

骁哥能量站 (/u/5bf74a1ac836?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

还不会写读书笔记？怪不得你的书都白看了 (/p/b346c377e2e6?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

关于读书笔记，我一直在钻研，在研究，以至于找出最好看，最好用，实力与美貌并存的读书手帐。在这个过程中，我发现干货类的书和文学类的书是有很大不

萌薇 (/u/c20b62e8e2ba?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

有没有哪一瞬间，你的人生充满了无力感 (/p/a6bc9f30ec0c?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)


-1- 前段时间坐火车回家，对面坐了一位大约四十出头的大叔，抱着一个三四岁的小女孩，她靠在他的怀里，拿着一包旺仔小馒头静静地吃着，不哭也不闹。

共央君 (/u/8c84a932666e?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

 登录/注册

为你个性化推荐内容

(/sign_in?utm_source=desktop&utm_medium=note-bottom-bind)

 下载简书App

随时随地发现和创作内容

(/app/download?utm_source=desktop&utm_medium=click-note-bottom-bind)