

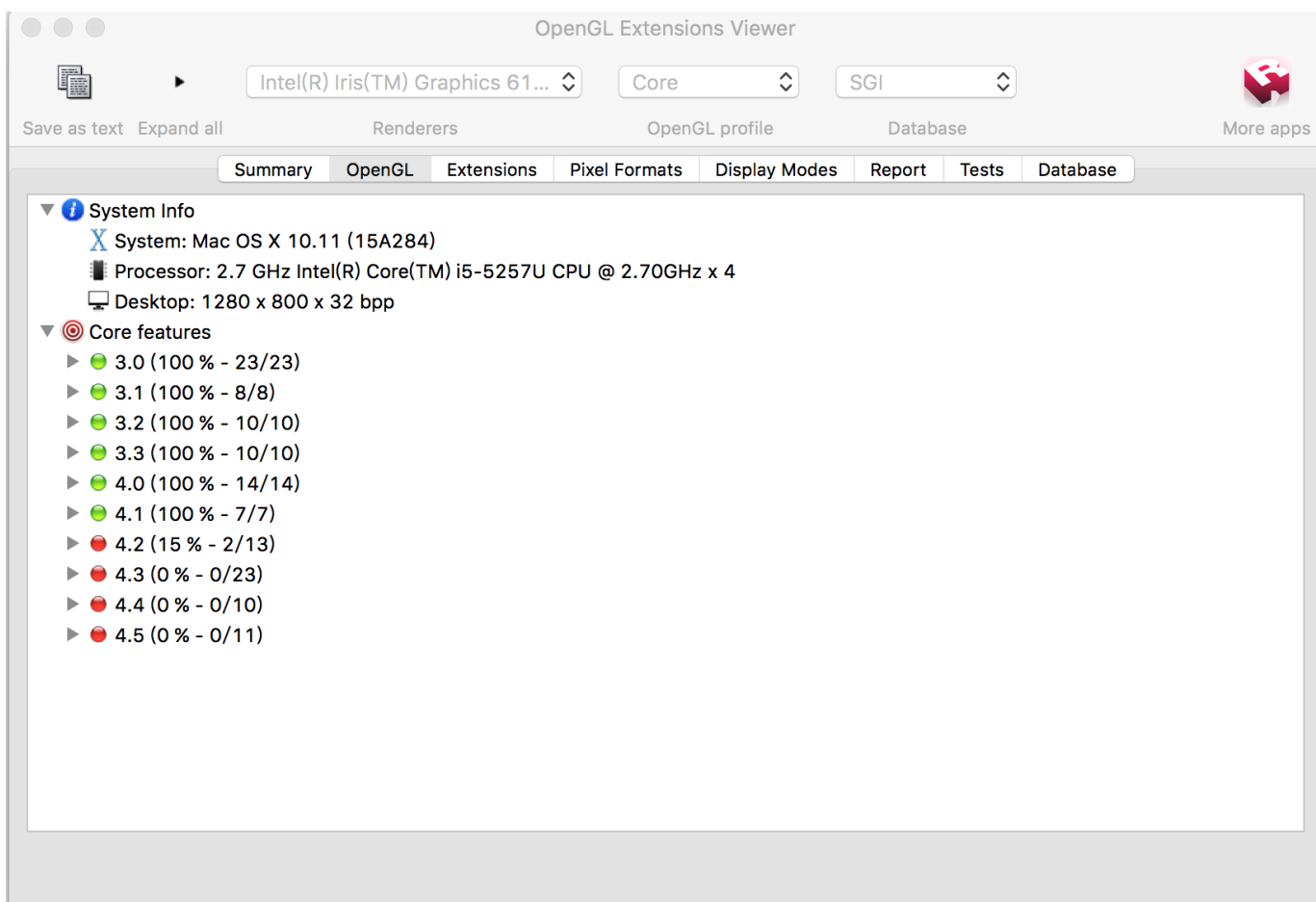
# 在OSX中进行OpenGL开发

本文介绍了如何将《OpenGL编程指南》中第一章的示例程序移植到OSX中运行。

原书中的样例基于OpenGL 4.3版本，并使用了GLUT组件进行开发。在OSX中，GLUT在OSX10.9就已经被废弃了，系统提供了NSOpenGLView组件用于OpenGL绘图。要将样例程序在OSX中运行起来，需要做一些修改。

本文开发环境基于OSX10.11，开发工具Xcode 7.

首先，使用OpenGL Extensions Viewer可以看到系统显卡支持OpenGL 4.1版本的core功能。



在Xcode中建立Cocoa工程，并在Targets的配置中添加OpenGL.framework。

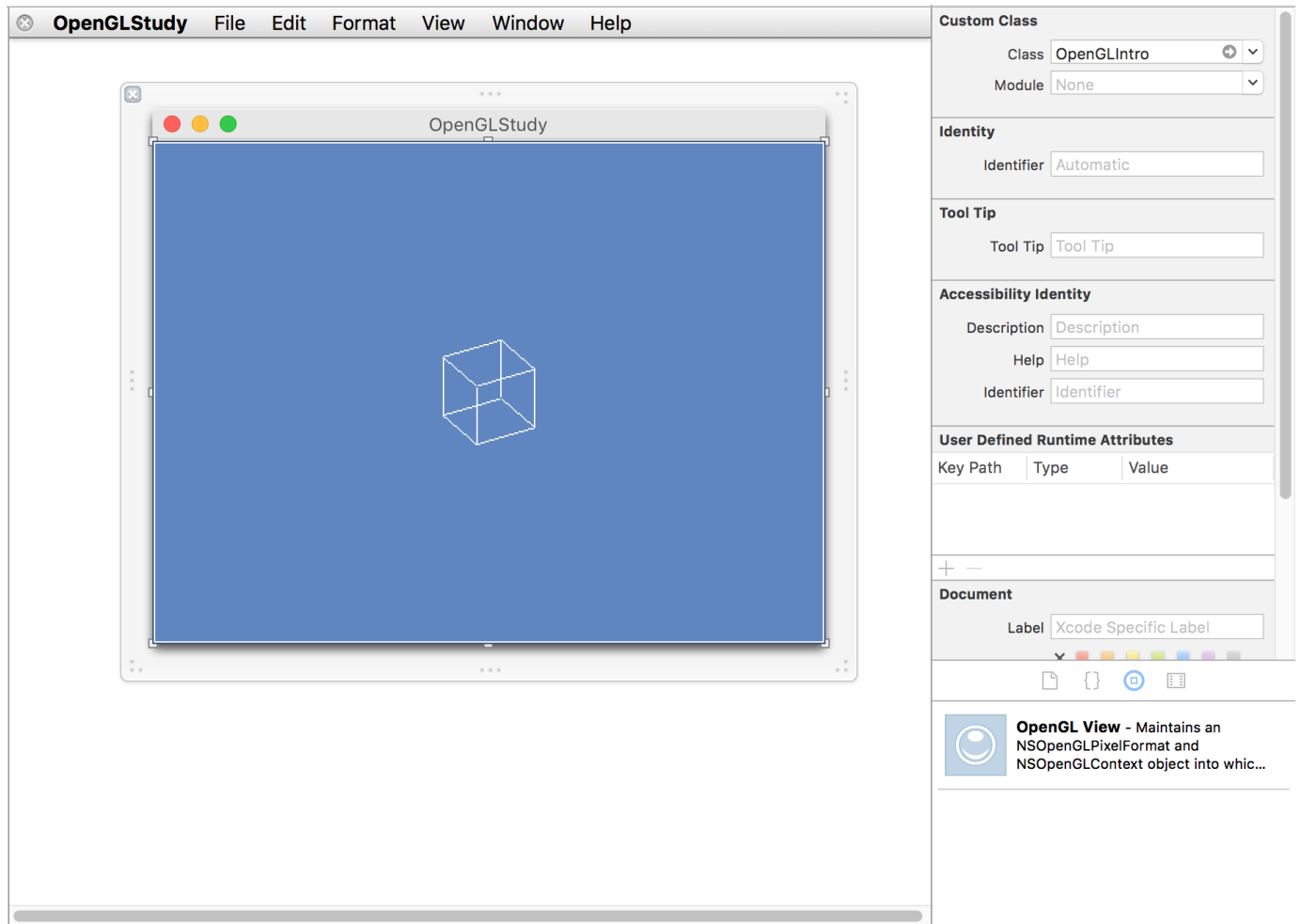
在OpenGL.framework中，提供了gl.h和gl3.h两个头文件，gl.h提供了到OpenGL2.1版本的支持，而gl3.h则提供了到OpenGL4.1的支持。

## ▼ Link Binary With Libraries (1 item)

Name	Status
 OpenGL.framework	Required ⇅
+ — Drag to reorder frameworks	

新建一个Cocoa Class，注意创建的时候选择继承NSOpenGLView。创建完成后可以看到自动生成的文件中给我们添加了drawRect方法，通过重写drawRect方法即可完成OpenGL绘图，

这里我创建的类名叫OpenGLIntro，接下来我们在视图中添加一个OpenGL View。打开MainMenu.xib文件，拖一个OpenGL View到窗口中，拉伸到窗口大小，并在属性窗口的Custom Class中将Class修改成OpenGLIntro。



先不急着写绘图代码，在drawRect代码中添加以下代码，测试我们使用的GL版本和GLSL版本。

```
const GLubyte *version = glGetString( GL_VERSION );
const GLubyte *glslVersion = glGetString( GL_SHADING_LANGUAGE_VERSION );

NSLog(@"GL Version : %s", version);
NSLog(@"GLSL Version : %s", glslVersion);
```

运行输出如下，可以看到这里用的还是OpengGL 2.1的版本，无论我们的头文件用gl.h还是gl3.h都没有任何作用。

```
2015-10-03 12:50:03.105 OpenGLStudy[2959:297347] GL Version : 2.1 INTEL-10.8.77
2015-10-03 12:50:03.105 OpenGLStudy[2959:297347] GLSL Version : 1.20
```

为了使用高版本的OpenGL，需要重写awakeFromNib方法，如下

```

- (void) awakeFromNib
{
    NSOpenGLPixelFormatAttribute attributes [] = {
        NSOpenGLPFADepthSize, (NSOpenGLPixelFormatAttribute)24,
        NSOpenGLPFAOpenGLProfile, NSOpenGLProfileVersion3_2Core,
        (NSOpenGLPixelFormatAttribute)0
    };

    NSOpenGLPixelFormat *pf = [[NSOpenGLPixelFormat alloc]
initWithAttributes:attributes];
    [self setPixelFormat:pf];
}

```

运行输出如下，可以看到已经变成了OpenGL 4.1版本

```

2015-10-03 12:47:42.301 OpenGLStudy[2929:292575] GL Version : 4.1 INTEL-
10.8.77
2015-10-03 12:47:42.301 OpenGLStudy[2929:292575] GLSL Version : 4.10

```

接下来就是实现绘图代码了，我这里只是贴一下代码，不再对代码进行解释了。在绘图中用到两个着色器，着色器的version使用410，另外shaders要替换成着色文件的路径。

在代码中用到了LoadShaders.h这个头文件，这个头文件在OpenGL红皮书的配套源码中提供，相信看这本书的同学都已经下载了配套源码了，添加LoadShaders.h和LoadShaders.cpp到工程即可。需要注意的是，LoadShaders.cpp的102~106要注释掉，否则编译会报错。

OpenGLInfo.h

OpenGLInfo.h

```

//
//  OpenGLIntro.h
//  OpenGLStudy
//
//  Created by VimerSu on 15/10/2.
//  Copyright © 2015年 vimersu. All rights reserved.
//

#import <Cocoa/Cocoa.h>

@interface OpenGLIntro : NSOpenGLView

@end

```

```
//
//  OpenGLIntro.m
//  OpenGLStudy
//
//  Created by VimerSu on 15/10/2.
//  Copyright © 2015年 vimersu. All rights reserved.
//

#import "OpenGLIntro.h"
#include <OpenGL/gl3.h>

#include "LoadShaders.h"

enum VAO_IDs {Triangles, NumVAOs};
enum Buffer_IDs {ArrayBuffer, NumBuffers};
enum Atrib_IDs {vPostion = 0};
GLuint VAOs[NumVAOs];
GLuint Buffers[NumBuffers];

const GLuint NumVertices = 6;

static void init() {
    glGenVertexArrays(NumVAOs, VAOs);
    glBindVertexArray(VAOs[Triangles]);

    GLfloat vertices[NumVertices][2] = {
        { -0.90, -0.90 },
        {  0.85, -0.90 },
        { -0.90,  0.85 },
        {  0.90, -0.85 },
        {  0.90,  0.90 },
        { -0.85,  0.90 },
    };

    glGenBuffers(NumBuffers, Buffers);
    glBindBuffer(GL_ARRAY_BUFFER, Buffers[ArrayBuffer]);
    glBufferData(GL_ARRAY_BUFFER, sizeof(vertices), vertices,
GL_STATIC_DRAW);

    ShaderInfo shaders[] = {
        {GL_VERTEX_SHADER, "/path/to/triangles.vert"},
        {GL_FRAGMENT_SHADER, "/path/to/triangles.frag"},
    };
}
```

```

        {GL_NONE, NULL}
    };

    GLuint program = LoadShaders(shaders);
    glUseProgram(program);

    glVertexAttribPointer(vPosition, 2, GL_FLOAT, GL_FALSE, 0, (void *)0);
    glEnableVertexAttribArray(vPosition);

}

static void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glBindVertexArray(VAOs[Triangles]);
    glDrawArrays(GL_TRIANGLES, 0, NumVertices);
    glFlush();
}

@implementation OpenGLIntro

- (void)drawRect:(NSRect)dirtyRect {
    [super drawRect:dirtyRect];

    const GLubyte *version = glGetString( GL_VERSION );
    const GLubyte *glslVersion = glGetString( GL_SHADING_LANGUAGE_VERSION );

    NSLog(@"GL Version : %s", version);
    NSLog(@"GLSL Version : %s", glslVersion);

    init();
    display();
    // Drawing code here.
}

- (void) awakeFromNib
{
    NSOpenGLPixelFormatAttribute attributes [] = {
        NSOpenGLPFADepthSize, (NSOpenGLPixelFormatAttribute)24,
        NSOpenGLPFAOpenGLProfile, NSOpenGLProfileVersion3_2Core,
        (NSOpenGLPixelFormatAttribute)0
    };
};

```

```
        NSOpenGLPixelFormat *pf = [[NSOpenGLPixelFormat alloc]
initWithAttributes:attributes];
        [self setPixelFormat:pf];
    }

@end
```

triangles.frag

triangles.frag

运行结果

