

# 计算机图形学：实验练习三

Edited by 李民选

Edited on 2021 年 3 月 28 日

李洋 2021 *Spring*



华东师范大学  
计算机科学与技术学院

李民选

10185102208

## 目录

1	实验目的	3
2	实验内容与实验步骤	3
3	实验环境	3
4	实验过程与分析	3
5	实验结果总结	5
6	附录	6

## 1 实验目的

熟悉 OpenGL 与 GLFW 的基本框架以及 shader 基本操作。

## 2 实验内容与实验步骤

1. 编辑 build.bat 文件，生成与本机 Visual Studio 版本相匹配的工程；  
注意：这里要修改脚本文件为 `cmake .. -G "Visual Studio 16 2019" -A Win32`
2. 进入上一个步骤生成的目录下，点击 \*.sln 文件进入项目；
3. 将 Hello Triangle 设置为启动项目；
4. 显示 OpenGL 的版本号，并在控制台窗口中回显；
5. 绘制三角形。

## 3 实验环境

1. OS: Windows 10 Home Edition
2. IDE: Visual Studio 16 2019
3. Libiary: 3.3.0 - Build 26.20.100.7262

## 4 实验过程与分析

### 1. cmake 创建项目 & 配置 OpenGL 运行环境

修改 build.bat 脚本文件：

```
1 @echo off
2 mkdir vs2010
3 cd vs2010
4 .\..\cmake\bin\cmake.exe .. -G "Visual Studio 16 2019" -A Win32
5 cd ..
```

### 2. 使用 glGetString() 函数获取 OpenGL 的版本号

获取版本号之后，使用 printf() 函数在控制台窗口中显示即可。

```
1 const GLubyte* version = glGetString(GL_VERSION);
2 printf("%s\n", version);
```

### 3. 绘制三角形

根据实验 ppt, 要通过 helper.cpp 来完成对两个 shader 的调用、编译以及链接。但是我阅读了一下 helper.cpp 的源码, 发现只是封装好了两个 shader 的调用, 只需要填进 shader 对应的路径即可 (同文件夹下可使用相对路径)。

在此我并没有调用 LoadShader 函数, 而是按照 ppt 给出的参考链接中的实现方法, 在 main 里添加以下代码, 即可完成三角形的绘制。

#### 1) 创建 vertex\_shader

```
1 const char* vertex_shader =  
2     "#version 400\n"  
3     "in vec3 vp;"  
4     "void main() {"  
5     "    gl_Position = vec4(vp, 1.0);"  
6     "}";
```

#### 2) 创建 fragment\_shader

```
1 const char* fragment_shader =  
2     "#version 400\n"  
3     "out vec4 frag_colour;"  
4     "void main() {"  
5     "    frag_colour = vec4(0.5, 0.0, 0.5, 1.0);"  
6     "}";
```

#### 3) 编译 & 链接 vertex\_shader 和 fragment\_shader

```
1 GLuint vs = glCreateShader(GL_VERTEX_SHADER);  
2 glShaderSource(vs, 1, &vertex_shader, NULL);  
3 glCompileShader(vs);  
4 GLuint fs = glCreateShader(GL_FRAGMENT_SHADER);  
5 glShaderSource(fs, 1, &fragment_shader, NULL);  
6 glCompileShader(fs);  
7  
8 GLuint shader_programme = glCreateProgram();  
9 glAttachShader(shader_programme, fs);  
10 glAttachShader(shader_programme, vs);  
11 glLinkProgram(shader_programme);
```

#### 4) 创建 Vertex Array

```
1 GLuint vao = 0;  
2 glGenVertexArrays(1, &vao);  
3 glBindVertexArray(vao);  
4 glEnableVertexAttribArray(0);
```

## 5) 创建 Vertex Buffer

```
1 GLuint vbo = 0;
2 glGenBuffers(1, &vbo);
3 glBindBuffer(GL_ARRAY_BUFFER, vbo);
4 glBufferData(GL_ARRAY_BUFFER, 9 * sizeof(float), points, GL_STATIC_DRAW);
5 glBindBuffer(GL_ARRAY_BUFFER, vbo);
6 glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, NULL);
```

## 5 实验结果总结

首先是在控制台显示 OpenGL 版本号



图 1: 显示 OpenGL 版本号

然后是绘制三角形

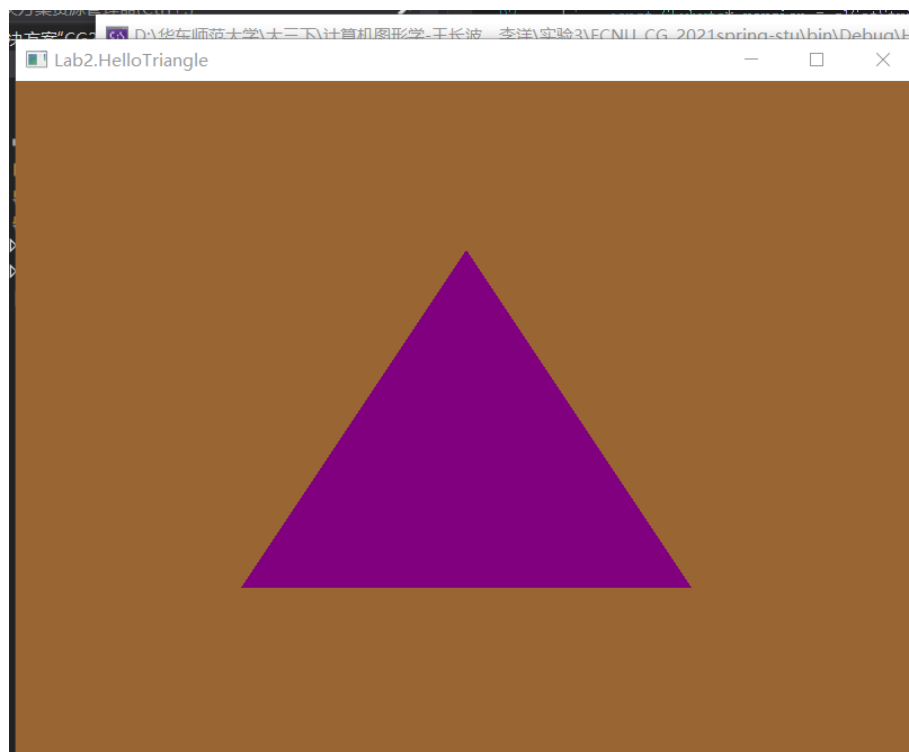


图 2: 绘制三角形

## 6 附录

1. <https://antongerdelan.net/opengl/hellotriangle.html>
2. <https://learnopengl.com/Getting-started/Hello-Triangle>
3. <https://www.glfw.org/documentation.html>