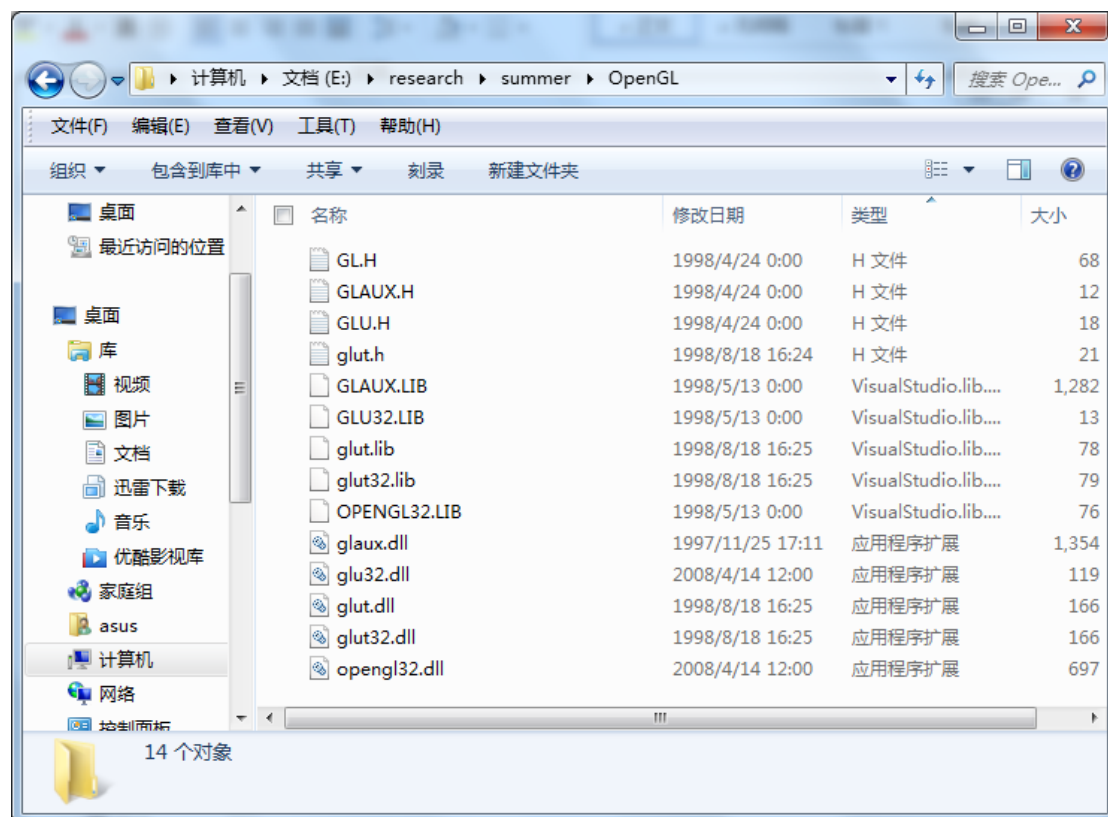


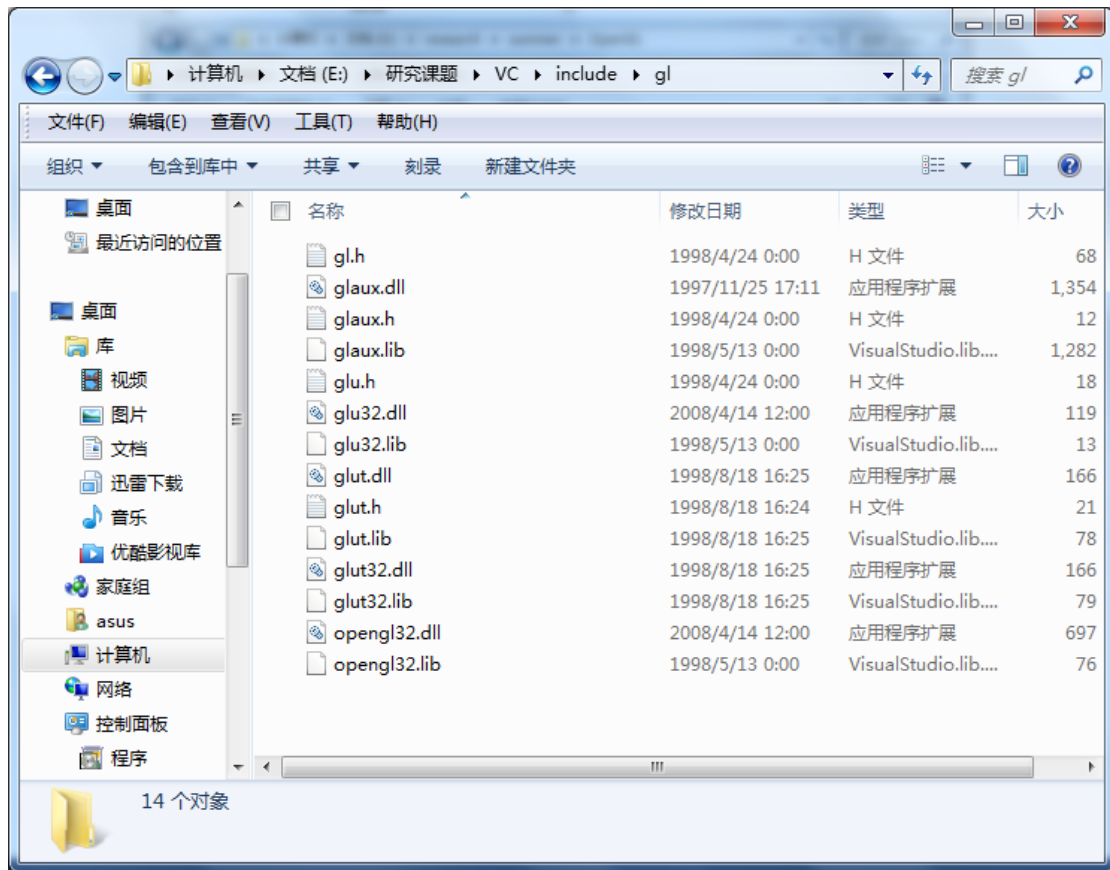
OPENGL 配置指南

本篇使用的是 VS2012 下的 OPENGL 配置

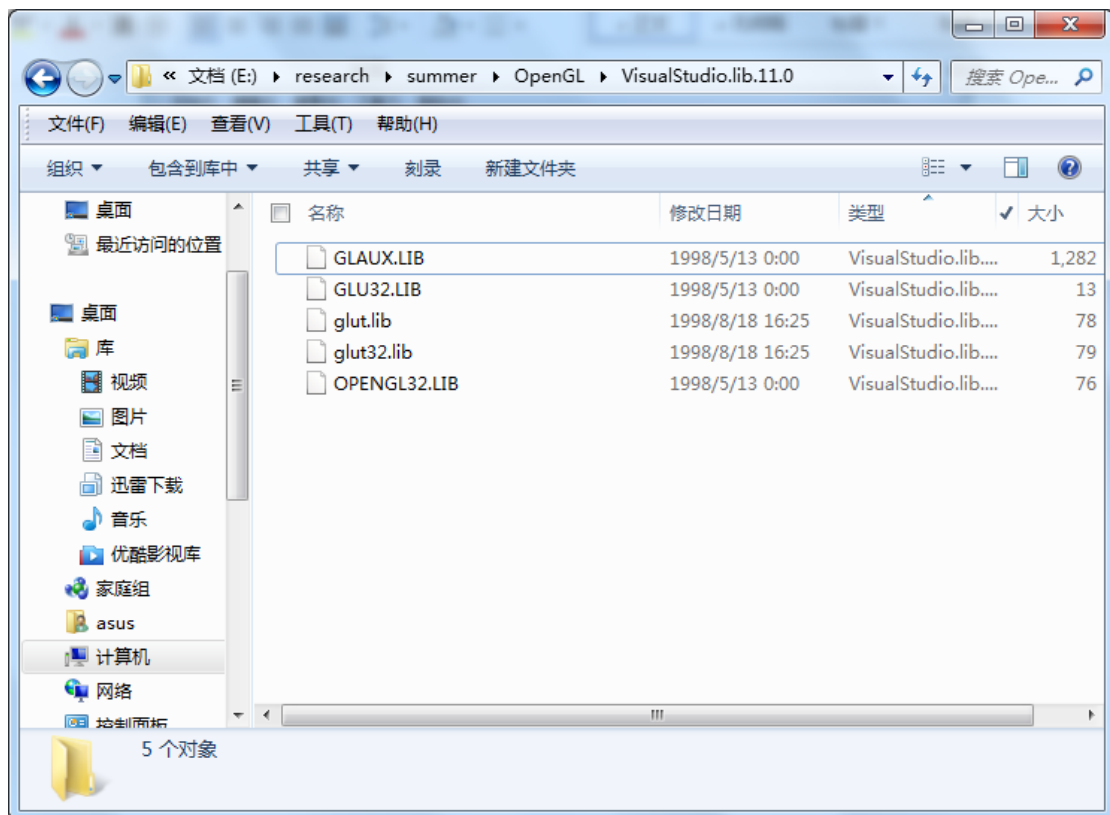
首先是进行解压，发现一共包含以下内容：



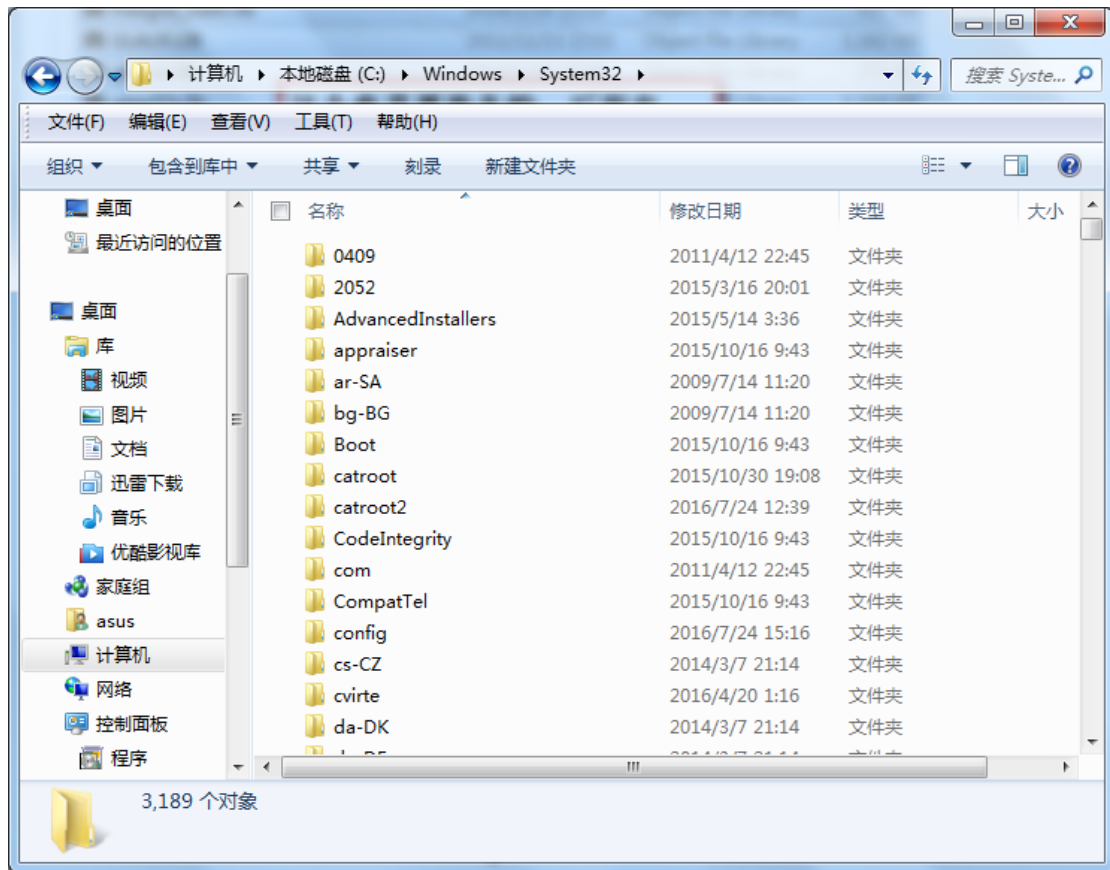
然后，在 VS 的安转目录中找到 VC，Include，新建文件夹 gl，将我们解压的内容全部复制过去：



然后，找到 VC 目录下的 Lib 文件夹，将 lib 全部放进去：

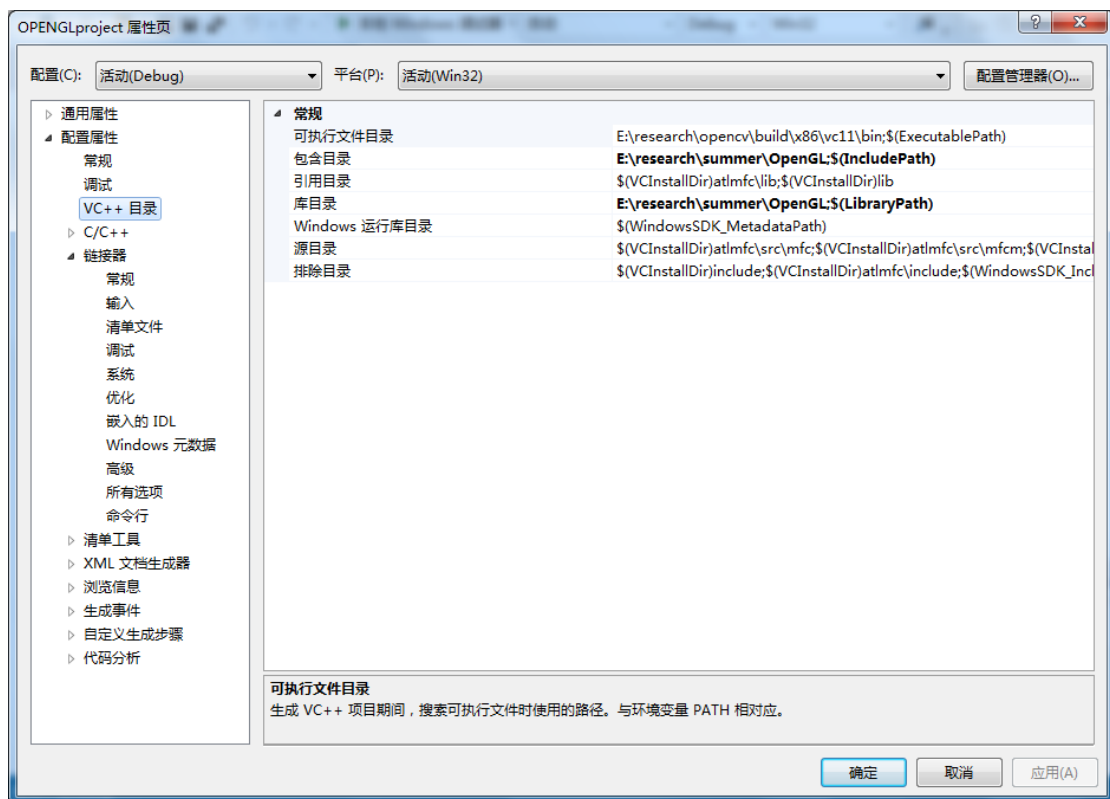


同样的道理，将 DLL 全部放在 c 盘里面，如果位，选择是 32 位，选择 System32，如果是 64 位选择 C:\Windows\SysWOW64

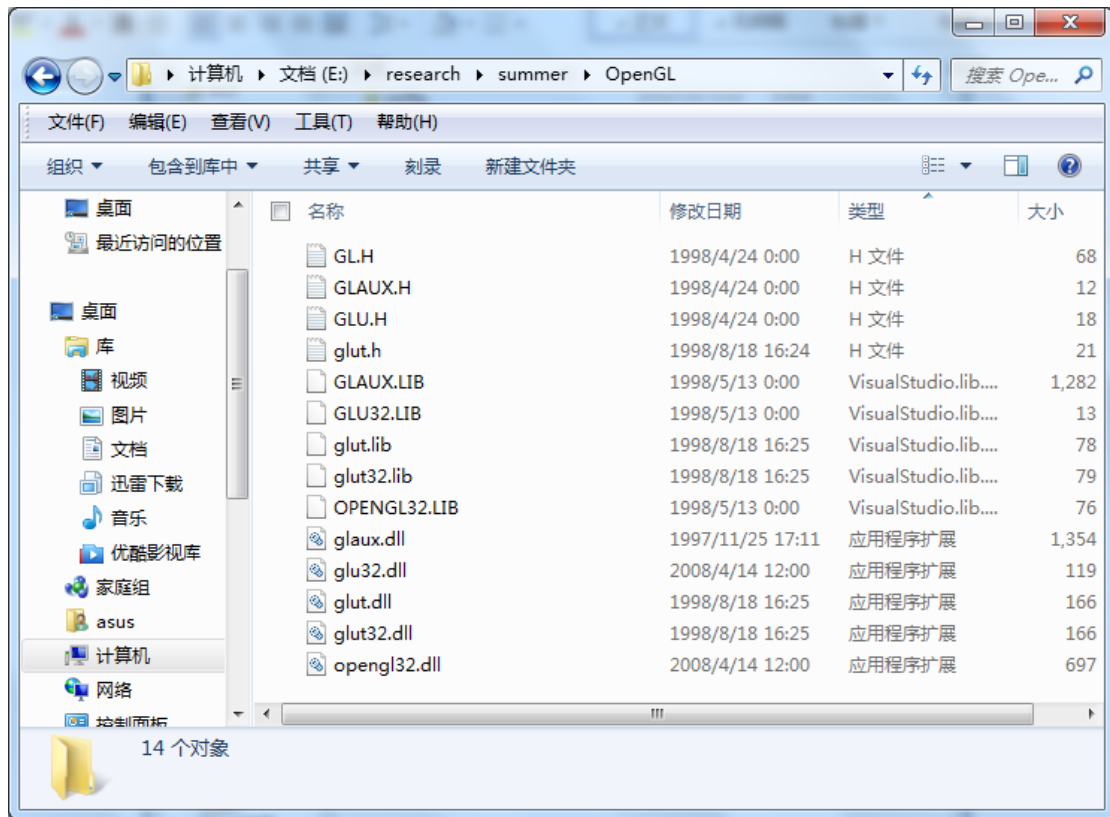


在进行了这些处理之后，我们可以进行环境的配置。

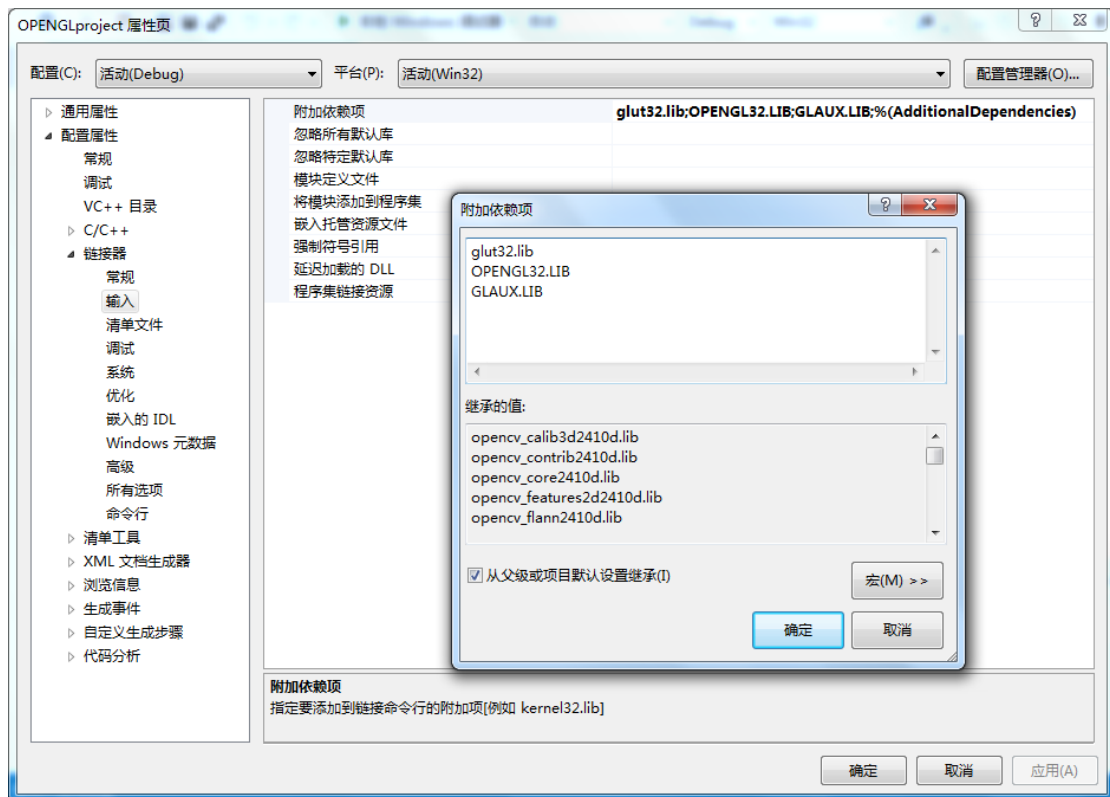
新建工程，找到对应的 VC++ 目录，将一下路径加入到包含目录和库目录中：



上述路径下有以下内容：



然后，我们进行依赖项的输入：



这样，我们就配置好了文件。

接下来用代码进行验证：

```
#include<windows.h>
```

```
#include<GL/glew.h>

#include<GL/gl.h>

#include<GL/glut.h>

#include<GL/glaux.h>

void background(void)

{

//设置背景颜色为黑色

glClearColor(0.0,0.0,0.0,0.0);

}

void myDisplay(void)

{

//buffer设置为颜色可写

glClear(GL_COLOR_BUFFER_BIT);

//开始画三角形

glBegin(GL_TRIANGLES);

//设置为光滑明暗模式

glShadeModel(GL_SMOOTH);

//设置第一个顶点为红色

glColor3f(1.0,0.0,0.0);

//设置第一个顶点的坐标为(-1.0,-1.0)

glVertex2f(-1.0,-1.0);

//设置第二个顶点为绿色
```

```
glColor3f(0.0, 1.0, 0.0);

//设置第二个顶点的坐标为 (0.0, -1.0)

glVertex2f(0.0, -1.0);

//设置第三个顶点为蓝色

glColor3f(0.0, 0.0, 1.0);

//设置第三个顶点的坐标为 (-0.5, 1.0)

glVertex2f(-0.5, 1.0);

//三角形结束

glEnd();

//强制OpenGL函数在有限时间内运行

glFlush();

}

void myReshape(GLsizei w, GLsizei h)

{

glViewport(0, 0, w, h);

//设置视口

glMatrixMode(GL_PROJECTION);

//指明当前矩阵为GL_PROJECTION

glLoadIdentity();

//将当前矩阵置换为单位阵

if(w <= h)

gluOrtho2D(-1.0, 1.5, -1.5, 1.5*(GLfloat)h/(GLfloat)w);
```

```
//定义二维正视投影矩阵

else

gluOrtho2D(-1.0, 1.5*(GLfloat)w/(GLfloat)h, -1.5, 1.5);

glMatrixMode(GL_MODELVIEW);

//指明当前矩阵为GL_MODELVIEW

}

int main(int argc, char*argv[])

{

//初始化

glutInit(&argc, argv);

glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);

glutInitWindowSize(400, 400);

glutInitWindowPosition(200, 200);

//创建窗口

glutCreateWindow("Triangle");

//绘制与显示

background();

glutReshapeFunc(myReshape);

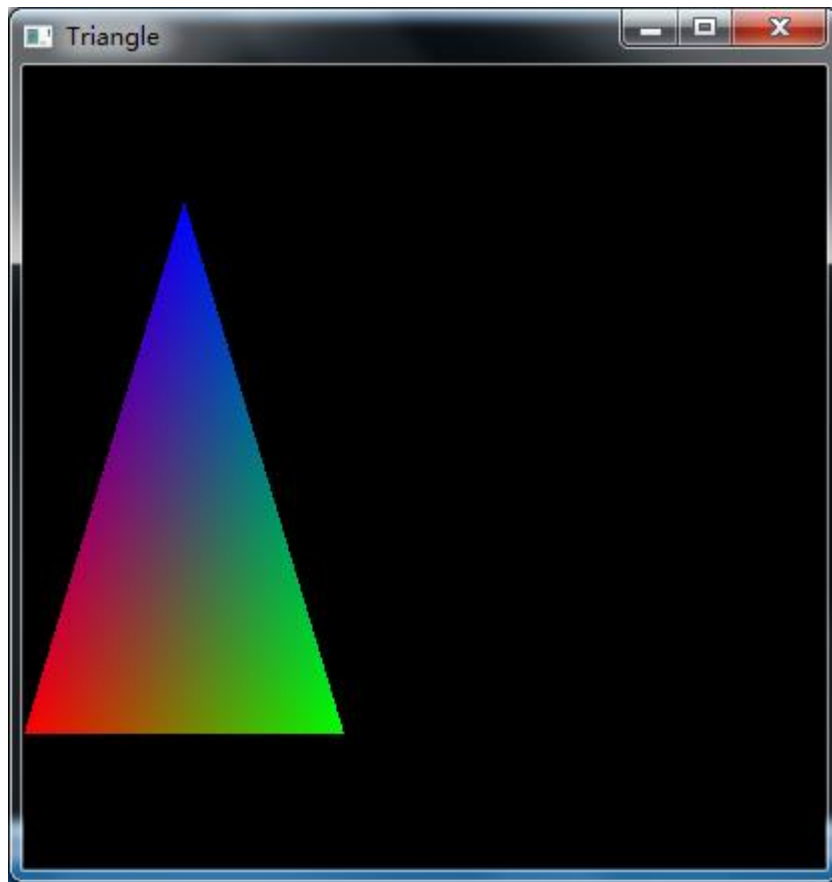
glutDisplayFunc(myDisplay);

glutMainLoop();

return(0);

}
```

得到以下结果说明，配置没有问题：



注意：运行中可能会需要加入一些dll库，按照电脑提示加入就好！