一、在 Activity 中

```
package com.example.administrator.createopenglview;//此处包名为你的项目包名

import android.opengl.GLSurfaceView;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;

/**
 * 构造 OpenGL ES View
 * 类似    平常创建项目显示的 Hello world 一样
 *
 * 编译运行后 屏幕是一个绿色界面
 */
public class CreateOpenGLActivity extends AppCompatActivity {//class 的名称为文件名称

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        GLSurfaceView glSurfaceView = new GLSurfaceView(this);
        glSurfaceView.setRenderer(new OpenGLRenderer());
        setContentView(glSurfaceView);
    }
}
```

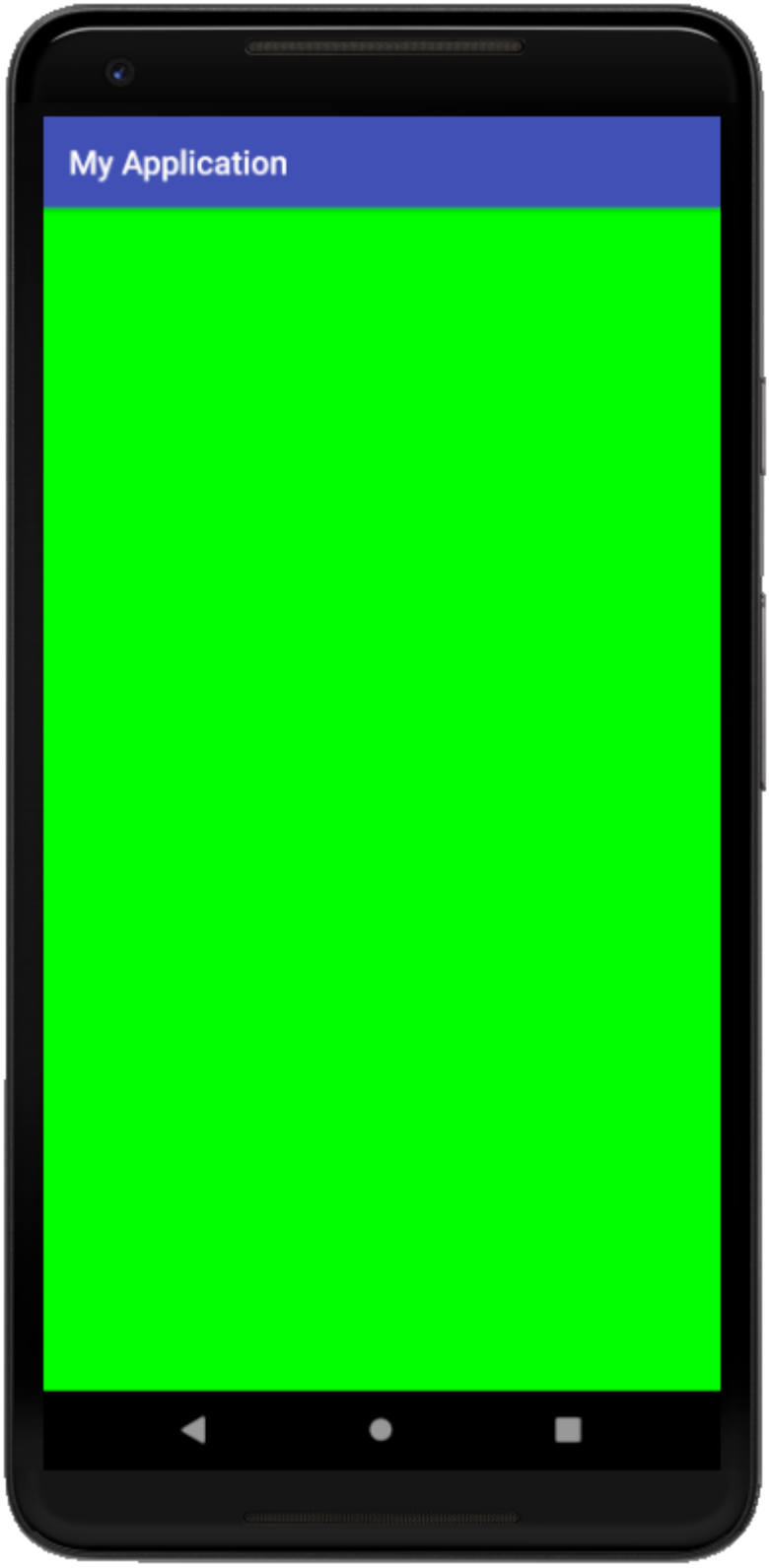
二、在 OpenGLRenderer 中（即在上述的 activity 所在的同个包目录下新建一个 OpenGLRenderer.java 文件）

```
package com.example.administrator.createopenglview; //应与 activity 处在同一个包下

import android.opengl.GLSurfaceView;
import android.opengl.GLU;

import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;

/**
 * Created by Administrator on 2016/9/26.
 * <p>
 * 定义一个统一图形绘制的接口
```

```
 */

public class OpenGLRenderer implements GLSurfaceView.Renderer {
    /**
     * 主要用来设置一些绘制时不常变化的参数，例如：背景色，是否打开 Z-buffer(去除隐藏面)等
     *
     * @param gl
     * @param config
     */
    @Override
    public void onSurfaceCreated(GL10 gl, EGLConfig config) {
        //设置背景的颜色
        gl.glClearColor(0f, 1f, 0f, 0.5f);
        //使光滑的材质,默认不需要。
        gl.glShadeModel(GL10.GL_SMOOTH);
        //深度缓冲设置。
        gl.glClearDepthf(1.0f);
        //启用深度测试。
        gl.glEnable(GL10.GL_DEPTH_TEST);
        //深度测试类型
        gl.glDepthFunc(GL10.GL_LEQUAL);
        //最好的的角度计算。
        gl.glHint(GL10.GL_PERSPECTIVE_CORRECTION_HINT, GL10.GL_NICEST);
    }

    /**
     * 如果设备支持屏幕的横向和纵向切换，
     * 这个方法将发生在横向<==>纵向互换时，
     * 此时可以重新设置绘制的纵横比率。
     *
     * @param gl
     * @param width
     * @param height
     */
    @Override
    public void onSurfaceChanged(GL10 gl, int width, int height) {

        //将当前视图端口设置为新的大小。
        gl.glViewport(0, 0, width, height);
        //选择投影矩阵
        gl.glMatrixMode(GL10.GL_PROJECTION);
        //重置投影矩阵
        gl.glLoadIdentity();
```

```
        /**
         * 建立一个透视投影矩阵
         *
         *   gl ： GL10 接口
         * fovy ： 指定领域的视角,在 Y 轴方向。指定方面定量 determin 领域在 x 方向上的
看法。
         *           高宽比的比例是 x(宽度)y(高度)。
         * zNear ： 指定观众的距离不远的剪裁平面的(总是正数)。
         * zFar ： 指定了与观众的距离遥远的剪裁平面的(总是正数)。
         */
        //计算窗口的长宽比
        GLU.gluPerspective(gl, 45.0f, (float) width / (float) height, 0.1f, 100.0f);
        //选择 modelview 矩阵
        gl.glMatrixMode(GL10.GL_MODELVIEW);
        //重置投影矩阵
        gl.glLoadIdentity();

    }

    /**
     * 定义实际的绘图操作
     *
     * @param gl
     */
    @Override
    public void onDrawFrame(GL10 gl) {
        //清除屏幕和深度缓冲。
        gl.glClear(GL10.GL_COLOR_BUFFER_BIT | GL10.GL_DEPTH_BUFFER_BIT);
    }
}
```

三、显示结果


My Application

GLSurfaceView 基础学习笔记

OpenGL 即 Open Graphics Library(开放的图形库接口)，主要用于三维图形编程。

OpenGL ES：OpenGL 的子集，嵌入式开放图形库接口。OpenGL ES 提供了 GLSurfaceView 组件。

GLSurfaceView 用于显示 3D 图像，本身并不提供绘制 3D 图形的功能，绘制功能由 GLSurfaceView.Renderer 来完成。

使用 OpenGL ES 的步骤如下：

1、创建 GLSurfaceView 组件（也可继承该组件），并使用 Activity 显示 GLSurfaceView 组件；

```
1.  GLSurfaceView glSurfaceView=new GLSurfaceView(this);
```

2、为 GLSurfaceView 创建 GLSurfaceView.Renderer 实例，并实现 GLSurfaceView.Renderer 接口的三个方法：

其中，GL10 代表 OpenGL 的绘制画笔

```
1.  MyRenderer implements Renderer{
2.  //onSurfaceCreated 方法主要用于执行一些初始化操作
3.      @Override
4.      public void onSurfaceCreated(GL10 gl, EGLConfig config) {
5.          gl.glDisable(GL10.GL_DITHER);// gl.glDisable 用于禁用 OpenGL 某方面的特性，该处表示关闭抗抖动，可以提高性能
6.          gl.glHint(GL10.GL_PERSPECTIVE_CORRECTION_HINT, GL10.GL_FASTEST);//该方法用于修正，本处用于设置对透视进行修正
7.          gl.glClearColor(0, 0, 0, 0);//设置 OpenGL 清屏所用的颜色，四个参数分别代表红、绿、蓝和透明度值，范围为 0-1，此处表示黑色
8.          gl.glEnable(GL10.GL_DEPTH_TEST);//启用某方面的性能，此处为启动深度测试，负责跟踪每个物体在 Z 轴上的深度，避免后面的物体遮挡前面的物体
```

```
9.          gl.glDepthFunc(GL10.GL_LEQUAL);//设置深度测试的类型，此处为如果输入的深度值小于或等于参考值，则通过
10.         gl.glShadeModel(GL10.GL_SMOOTH);//设置阴影模式为平滑模式
11. }
12.
13.   //当 SurfaceView 大小改变时回调，通常用于初始化 3D 场景
14.   @Override
15.   public void onSurfaceChanged(GL10 gl, int width, int height) {
16.         gl.glViewport(0, 0, width, height);//设置 3D 视窗的位置与大小
17.         gl.glMatrixMode(GL10.GL_PROJECTION);//设置矩阵模式为投影矩阵，这意味着越远的东西看起来越小，GL10.GL_MODELVIEW：模型视图矩阵：任何新的变换都会影响            该矩阵中的所有物体
18.         float ratio=(float)width/height;
19.         gl.glFrustumf(-ratio, ratio, -1, 1, 1, 10);//设置透视投影的空间大小，前两个参数用于设置 X 轴的最小值与最大值，中间两个参数用于设置 y 轴的最小值最大值            ，后两个参数用于设置 Z 轴的最小值最大值
20.   }
21.   //用于绘制 3D 图形
22.   @Override
23.   public void onDrawFrame(GL10 gl) {
24.         //清楚屏幕缓存和深度缓存(一般为必须设置的)
25.         gl.glClear(GL10.GL_COLOR_BUFFER_BIT|GL10.GL_DEPTH_BUFFER_BIT);
26.         gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);//启用顶点坐标数组
27.         gl.glEnableClientState(GL10.GL_COLOR_ARRAY);//启用顶点颜色数组
28.         gl.glMatrixMode(GL10.GL_MODELVIEW);设置矩阵模式为模型视图矩阵
29.         gl.glLoadIdentity();//相当于 reset()方法，用于初始化单位矩阵
30.         gl.glTranslatef(-0.32f, 0.35f, -1.1f);//移动绘图中心
```

```
<pre code_snippet_id="1634794" snippet_file_name="blog_20160405_2_9523416" name="code" class="java"><span style="font-family: Arial, Helvetica, sans-serif;">                gl.glVertexPointer(3, GL10.GL_FLOAT, 0, triangleFloatBuffer);//设置顶点的位置数据，第一个参数指定多少个元素确定一个顶点，通常为 3；第二个参数指定顶点坐标                    值的    类型，  triangleFloatBuffer  是一维数组，  形如，(x1,y1,z1,x2,y2,z2,...)，用于指定顶点坐标值</span>
```

```
1.  gl.glColorPointer(4, GL10.GL_FIXED, 0, triangleColorBuffer);//设置顶点的颜色数据
```

```
1.  gl.glDrawArrays(GL10.GL_TRIANGLES, 0, 3);//根据顶点绘制平面图形，第一个参数表示绘制图形的类型，GL10.GL_TRIANGLES：绘制简单的三角形，
    GL10.GL_TRIANG            LE_STRIP:沿着给出的顶点数据绘制三角形来形成平面图形。
    OpenGL 只能绘制三角形组成 3D 图形；第二个参数指定从哪个顶点开始，第三个参数表示顶点的数量
2.          gl.glFinish();//绘制结束
3.          gl.glDisableClientState(GL10.GL_VERTEX_ARRAY);
4.   }
5.  }
```

注：输入顶点数组时，数组中每个顶点的顺序不同，可能会导致画出的图形不一样

3、使用 setRenderer 方法为 GLSurfaceView 添加 Renderer。