# OpenGL ES 3.0能用就行笔记

因为我是个记性不太好的人，很多东西久了不用就会大面积遗忘，我觉得这样很可惜，很多东西像白学了一遍的样子。所以我打算特意写一份OpenGL笔记，只为满足以下几个要求：

1. 随时遗忘了就可以翻阅一下，可以通过温习做过各种例子实现自己想要的东西。
2. 对比书本成体系的知识，例如从渲染管线的概念和流程，到各种变量类型逐一介绍不同，这个笔记主要是通过例子来说明，如果想做什么，可以怎样做，而不是成体系地讲解知识。对比官方的详细知识讲解，提出近似而容易理解的说法，让我自己和读者随时需要时可以快速上手，在需要更详细的知识介绍时，通过关键字和链接提供补充知识，使得读者既可以快速应用OpenGL实现需求，也可以满足好奇心。

目录

[Open能用就行笔记 1](#_Toc1129969382)

[一、 坐标系与坐标系之间的转换 3](#_Toc266266759)

[二、 顶点与线条 3](#_Toc588687436)

[三、 JNI与OpenGL 3](#_Toc1523345237)

[四、 移动、缩放与旋转 3](#_Toc188737164)

[五、 贴图 3](#_Toc1416789835)

[六、 光照 3](#_Toc772917974)

[七、 高级玩法 3](#_Toc128228994)

[特效 3](#_Toc912215276)

## openGL的hello world

1. 首先创建一个安卓工程，继承类GLSurfaceView，例子如下：

import android.content.Context;  
import android.opengl.GLSurfaceView;  
  
public class PanelView extends GLSurfaceView {  
  
 public PanelView(Context context) {  
 super(context);  
 this.setEGLContextClientVersion(3);  
 setRenderer(new GLRenderer());  
 setRenderMode(GLSurfaceView.*RENDERMODE\_CONTINUOUSLY*);  
 }  
}

依赖于安卓SDK实现的OpenGL渲染画面，需要依托于GLSurfaceView进行显示。其中setEGLContextClientVersion用于设定当前GLSurfaceView使用的EGLContext版本，EGLContext可以暂时理解为保存OpenGL渲染状态、shader程序、依附于系统窗口的内存指针地址等的上下文工具类（希望详细了解EGLContext是什么的，可以点击//todo 这里），例如OpenGLES 3.0就是3（//todo 存疑）。

setRender就是设置渲染回调方法，填充GLSurfaceView创建时、状态更改时、以及渲染时实际要做什么的内容。

setRenderMode有2种，*RENDERMODE\_CONTINUOUSLY*代表按照GLSurfaceView能达到的最大刷新率，例如大部分设备的60Hz一直回调，*RENDERMODE\_WHEN\_DIRTY*啊代表用户发起请求渲染，也就是调用GLSurfaceView的requestRedner方法时，GLSurfaceView才会调用用户实现的渲染回调。一边用户交互操作使用后者居多。

1. 然后建立一个实现GLSurfaceView.Renderer的类，然后实现三个接口：

onSurfaceCreated(GL10 gl, EGLConfig config)、

onSurfaceChanged(GL10 gl, int width, int height)、

onDrawFrame(GL10 gl)

onSurfaceCreated方法的作用是使得GLSura

1. 阿萨德
2. 阿萨德

## 一些数学思想

0、//todo 比例与权值的作用，以及线性代数在图形图像中的应用，对应的初等数学例子。

## 坐标系与坐标系之间的转换

//todo 如何变换投影后的顶点为安卓控件顶点。

## 顶点与线条

## JNI与OpenGL

## 移动、缩放与旋转

## 贴图

## 光照

## 高级玩法

## 注意事项：

1、所有opengl操作，包括shader创建等必须要在glthread中进行，否则将出错。

2、视点z轴位置和近平面z轴位置太近可能会发生物体y轴翻转时只能看到部分的物体，因为有一部分已经在翻转过程中到了视点的背后

视频播放：

纹理片元特效：

使用FrameBufferObject实现渲染后结果的保留和持续处理