1. OpenGL ES (OpenGL for Embedded System)为适用于嵌入式系统的一个免费二维和三维图形库。是桌面版OpenGL的一个子集
2. OpenGL和OpenGL ES的关系：

OpenGL ES 1.0基于OpenGL 1.3

OpenGL ES 1.1基于OpenGL 1.5

OpenGL ES 2.0基于OpenGL 2.0

1. 管道的工序大致可分为Transformation Stage 和Tasterization Stage两大步。

OpenGL ES支持的基本图形为点Point、线Line和三角形Triangle，其他所有复制图形都是通过这几种基本图形组合而成。

在发出绘图指令后，会对顶点（Vertices）数组进行指定的坐标变换或光照处理。

顶点处理完成后，通过Rasterizer来生成像素信息，称为“Fragments”。对于Fragment在经过Texture Processing，Color Sum，Fog等处理并将最终处理结果存放在内存中（FrameBuffer）。

1. OpenGL ES提供了两类方法来绘制一个空间几何图形：

Public abstract void glDrawArrays(int mode, int first, int count)使用VertexBuffer来绘制，顶点的顺序由VertexBuffer中的顺序指定。

Public abstract void glDrawElements(int mode, int count, int type, Buffer indices)可以重新定义顶点的顺序，顶点的顺序由indices Buffer指定。

Mode列表：绘制独立的点、GL\_LINE\_STRIP绘制一条线段、GL\_LINE\_LOOP绘制一条封闭的线段（首尾相连）、GL\_lINES绘制多条线段、

GL\_TRIANGLES绘制多个三角形（两两不相邻）、（每三个顶点绘制三角形）

GL\_TRIANGLE\_STRIP绘制多个三角形（两两相邻）、（以v0v1v2,v1v2v3,v2v3v4的形式绘制）

GL\_TRIANGLE\_FAN以一个点为顶点绘制多个相邻的三角形（以v0v1v2,v0v2v3,v0v3v4的形式绘制三角形）

参数2：从数组缓存中的哪一位开始绘制，一般定义为0

参数3：顶点的数量

OpenGL坐标系：

右手坐标系：从左到右，x递增；从下到上，y递增；从远到近，z递增。

OpenGL坐标可分为：世界坐标系和当前绘图坐标系：

世界坐标系：以屏幕中心为原点，面对屏幕，右边是x正轴，屏幕指向你的为z轴正轴，长度单位这样来定：窗口范围按此单位恰好是（-1，-1）到（1，1）

当前绘图坐标系：是绘制物体时的坐标系。程序刚初始化时，世界坐标系和当前的绘图坐标系是重合的。当用glTranslatef()、glScalaf()、glRotatef()对当前绘图坐标系进行平移、伸缩、旋转变换后，世界坐标系和当前绘图坐标系不再重合。改变以后再用glVertex3f()等绘图函数绘图时，都是对当前绘图坐标系进行绘图，所有的函数参数也都是相对当前绘图坐标系来讲的

OpenGL的六种坐标系

1. Object or model coordinates
2. World coordinates
3. Eye(or Camera) coordinates
4. Clip coordinates
5. Normalized device coordinates
6. Window(or screen) coordinates

从object coordainates到world coordinates再到camera coordinate的变换，在OpenGL中统一称为model-view转换，初始化的时候，object coordinates和world coordinates还有camera coordinates坐标重合在原点，变换矩阵都为Identity。Model-view matix转换points、vectorsd到camera坐标系。

其中四种坐标系经常在程序中用到：世界坐标，物体坐标，设备坐标，眼坐标。

1. 世界坐标：是OpenGL中描述场景的坐标。我们用这个坐标系来描述物体及光源的位置。将物体放到场景中也就是将物体平移到特定位置、旋转一定角度，这就是坐标变换。OpenGL提供了glTranslate\*/glRotate\*/glScale\*三条坐标变换命令，利用OpenGL的矩阵运算命令，则可以实现任意复杂的坐标变换。
2. 物体坐标：是以物体为某一点为原点而建立的“世界坐标”，该坐标仅对物体适用，用来简化对物体各部分坐标的描述。物体放到场景中时，各部分经历的坐标变换相同，相对位置不变，所以可以看作一个整体。
3. 眼坐标：是以视点为原点，以视线的方向为Z+轴的正方向的坐标系中的方向。OpenGL管道会将世界坐标系先变换到眼坐标，然后进行裁剪，只有在视线范围（视见体）之内的场景才会进入下一阶段的计算。
4. 设备坐标：OpenGL的重要功能之一就是将三维的世界坐标经过变换、投影等计算，最终算出他在显示设备上对应的位置，这个位置就称为设备坐标。在屏幕、打印机等设备上的坐标是二维坐标。值得一提的是，OpenGL可以只使用设备的一部分进行绘制，这个部分称为视区或视口（Viewport）。投影得到的是视区的坐标（投影坐标），从投影坐标到设备坐标的计算过程就是设备变换了。

矩阵栈切换：glMatrixMode();

矩阵栈操纵命令：glPushMatrix();glPopMatrix();glLoadIdentity();glMultMatrix(M);

7 OpenGL中的几种变换

OpenGL中的各种转换是通过矩阵运算实现的，具体的说，就是当发出一个转换命令时，该命令就会产生一个4\*4阶的转换矩阵（OpenGL中的物体坐标一律采用齐次坐标，即（x, y, z, w），故所有变换矩阵都采用4\*4矩阵），当前矩阵与这个矩阵相乘，从而参数新的当前矩阵。例如，对于顶点坐标v，转换命令通常在坐标命令之前发出，若当前矩阵为C，转换命令构成的矩阵为M，则发出转换命令后，生成的新的当前矩阵为CM，这个矩阵在乘以顶点坐标v，从而构成新的顶点坐标CMv。上述过程说明，程序中绘制顶点前的最后一个变换命令先作用于顶点之上。这同时也说明，OpenGL编程中，实际的变换顺序与指定的顺序是相反的。

1. 视点变换：void gluLookAt（GLdouble eyex， GLdouble eyey， GLdouble eyez， GLdouble centerx， GLdouble centery， GLdouble upx， GLdouble upy， GLdouble upz）该函数定义了视点矩阵，并用该矩阵乘以当前矩阵。Eyex、eyey、 eyez定义了视点的位置；centerx、centery、centerz变量指定了参考点的位置，该点通常为相机所瞄准的场景中心轴线上的点；upx、upy、upz变量指定了向上向量的方向。
2. 模型变换
3. 模型平移：glTranslate{fd}（TYPE x, TYPE y, TYPE z）；该函数指定的x,y,z值沿着x轴、y轴、z轴平移物体（或按照相同的量值移动局部坐标系）。
4. 模型旋转：glRotate{fd}（TYPE angle, TYPE x, TYPE y, TYPE z）；该函数中的第一个angle定制模型旋转的角度，单位为度，后三个变量表示以原点（0，0，0）到（x，y，z）的连线为轴逆时针旋转的物体。
5. 模型缩放：glScale{fd}（TYPE x, TYPE y, TYPE z）；该函数可以对物体沿x，y，z轴分别进行放大缩小。函数中的三个参数分别是x，y，z轴方向的比例变换因子。缺省时都为1.0，即物体没变化。程序中物体Y轴比例为2.0，其余都是1.0，就是说将立方体变为长方体。
6. 投影变换：目的就是定义一个视景体，使得视景体外多余的部分裁剪掉，最终进入图像的只是视景体内的有关部分。投影包括透视投影（Perspective Projection）和正视投影（Orthographic Projection）两种。

透视投影函数有两个：void glFrustum（GLdouble left， GLdouble Right， GLdouble bottom，GLdouble top， GLdouble near， GLdouble far）；它创建一个透视景体。其操作是创建一个透视投影矩阵，并且用这个矩阵乘以当前矩阵。这个函数的参数只定义近剪裁平面的左下角点和右上角点的三维空间坐标，即（left, bottom, -near）和（right, top, -near）；最后一个参数far是远裁剪平面的z负值，其左下角点和右上角点空间坐标由函数根据透视投影原理自动生成。Near和far表示离视点的远近，总为正值。

另一个函数：void glOrtho2D（GLdouble left， GLdouble Right， GLdouble bottom， GLdouble top）它是一个特殊的正射投影函数，主要用于二维图像到二维屏幕上的投影。它的near和far缺省值分别是-1.0和1.0，所有二维物体的Z坐标都为0.0。因此它的剪裁面是一个左下角点为（left，bottom）、右上角点为（right，top）的矩形。

1. 视口变换：就是将视景体投影的物体显示在二维的视口平面上。

glViewport（GLint x， GLint y， GLsizei width， GLsizei height）；

这个函数定义一个视口。函数参数（x，y）是视口在屏幕窗口坐标系中的左下角点坐标，参数width和height分别是视口的宽度和高度。缺省时，参数值即（0， 0，winWidth， winHeight）指的是屏幕窗口的实际尺寸大小。所有这些值都是以像素为单位，全为整型数。

1. 裁剪变换
2. 矩阵栈的操作

8 有些操作只能放在render中，

问题1：矩阵旋转问题，在create中旋转的矩阵只能在初始化的时候作用一次，所有好多事不能再Create中完成，比如激活使用染色器。

在onNewframe中，由于每调用一次就要调用一次渲染器，所以可以实现一个持续调用的效果，例如持续旋转。

但是如果调用其它初始化函数时不会出现这种情况

9 先平移 再旋转 然后再画物体