1.

在《OpenGL编程指南》中谈到了全局固定坐标系和局部移动坐标系。看到这两个名词，我们可能会问：“那么什么时候使用全局固定坐标系,什么时候使用局部移动坐标系？又如何设定当前使用的坐标系呢？”。事实上，（本人觉得）这两个表面上有所区别的概念只是我们在绘制图形时的不同思考方式而已，根本不存在有个函数可以让我们选择使用的是全局坐标系或局部坐标系。

1.全局固定坐标系：在我们按全局固定坐标系去考虑场景中几个图形的绘制时，我们把对几个图形的模型视图变换操作都认为是相对于固定坐标系的原点的。例如，若我们要绘制一个太阳与其八大行星的场景，我们可以把太阳绘制在固定坐标系的原点，然后依次绘制八大行星。对每一个行星，在绘制之前我们使用glPushMatrix()保存视图变换，然后进行模型视图变换，再绘制，绘制完成调用glPopMatrix()恢复视图变化，然后再进行下一个行星的绘制。这样我们的绘制每一个行星进行的模型视图变化都是相对于一个固定的坐标原点（太阳所在处）进行的。

2.局部移动坐标系：按局部移动坐标系考虑场景中几个图形的绘制，我们以绘制日，地，月系统为例。我们首先绘制太阳，然后进行模型视图变化，再绘制地球，然后我们把认为坐标原点此时移动到了地球所在位置（移动坐标系），并不调用glPushMatrix()保存变换矩阵，而是在原来基础上再进行模型视图变换，在相对于地球的某个位置绘制月球，这是符合实际的，因为月球是地球的卫星。

3.通过上面的分析，有时我们看到在一个程序中可能同时使用了这两种方式考虑场景的绘制也就不奇怪了。

上面是本人暂时的看法，记录下来，不知道对不对，欢迎指正，不胜感激。

2.

OPENGL坐标系可分为：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(0, 0, 0)。你面对屏幕，你的右边是x正轴，上面是y正轴，屏幕指向你的为z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，世界坐标系和当前绘图坐标系是重合的。当用glTranslatef()，glScalef(), glRotatef()对当前绘图坐标系进行平移、伸缩、旋转变换之后， 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，都是在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。

这段话是csdn中的

3.

一）坐标系统

　　在现实世界中，所有的物体都具有三维特征，但计算机本身只能处理数字，显示二维的图形，将三维物体及二维数据联系在一起的唯一纽带就是坐标。

　　为了使被显示的三维物体数字化，要在被显示的物体所在的空间中定义一个坐标系。这个坐标系的长度单位和坐标轴的方向要适合对被显示物体的描述，这个坐标系称为世界坐标系。世界坐标系是始终固定不变的。

　　OpenGL还定义了局部坐标系的概念，所谓局部坐标系，也就是坐标系以物体的中心为坐标原点，物体的旋转或平移等操作都是围绕局部坐标系进行的，这时，当物体模型进行旋转或平移等操作时，局部坐标系也执行相应的旋转或平移操作。需要注意的是，如果对物体模型进行缩放操作，则局部坐标系也要进行相应的缩放，如果缩放比例在案各坐标轴上不同，那么再经过旋转操作后，局部坐标轴之间可能不再相互垂直。无论是在世界坐标系中进行转换还是在局部坐标系中进行转换，程序代码是相同的，只是不同的坐标系考虑的转换方式不同罢了。

　　计算机对数字化的显示物体作了加工处理后，要在图形显示器上显示，这就要在图形显示器屏幕上定义一个二维直角坐标系，这个坐标系称为屏幕坐标系。这个坐标系坐标轴的方向通常取成平行于屏幕的边缘，坐标原点取在左下角，长度单位常取成一个象素。

从三维物体到二维图象，就如同用相机拍照一样，通常都要经历以下几个步骤：

　　1、将相机置于三角架上，让它对准三维景物，它相当于OpenGL中调整视点的位置，即视点变换（Viewing Transformation）。

　　2、将三维物体放在场景中的适当位置，它相当于OpenGL中的模型变换（Modeling Transformation），即对模型进行旋转、平移和缩放。

　　3、选择相机镜头并调焦，使三维物体投影在二维胶片上，它相当于OpenGL中把三维模型投影到二维屏幕上的过程，即OpenGL的投影变换（Projection Transformation），OpenGL中投影的方法有两种，即正射投影和透视投影。为了使显示的物体能以合适的位置、大小和方向显示出来，必须要通过投影。有时为了突出图形的一部分，只把图形的某一部分显示出来，这时可以定义一个三维视景体（Viewing Volume）。正射投影时一般是一个长方体的视景体，透视投影时一般是一个棱台似的视景体。只有视景体内的物体能被投影在显示平面上，其他部分则不能。

　　4、冲洗底片，决定二维相片的大小，它相当与OpenGL中的视口变换（Viewport Transformation）（在屏幕窗口内可以定义一个矩形，称为视口（Viewport），视景体投影后的图形就在视口内显示）规定屏幕上显示场景的范围和尺寸。

　　通过上面的几个步骤，一个三维空间里的物体就可以用相应的二维平面物体表示了，也就能在二维的电脑屏幕上正确显示了。