实验2 OpenGL几何变换

1. 实验目的：

理解掌握一个OpenGL程序平移、旋转、缩放变换的方法。

1. 实验内容：

（1）阅读实验原理，运行示范实验代码，掌握OpenGL程序平移、旋转、缩放变换的方法；

1. 根据示范代码，尝试完成实验作业；
2. 实验原理：
3. OpenGL下的几何变换

在OpenGL的核心库中，每一种几何变换都有一个独立的函数，所有变换都在三维空间中定义。

平移矩阵构造函数为glTranslate<f,d>(tx, ty, tz)，作用是把当前矩阵和一个表示移动物体的矩阵相乘。tx, ty，tz指定这个移动物体的矩阵，它们可以是任意的实数值，后缀为f（单精度浮点float）或d（双精度浮点double），对于二维应用来说，tz=0.0。

旋转矩阵构造函数为glRotate<f,d>(theta, vx, vy, vz)，作用是把当前矩阵和一个表示旋转物体的矩阵相乘。theta, vx, vy, vz指定这个旋转物体的矩阵，物体将绕着(0,0,0)到(x,y,z)的直线以逆时针旋转，参数theta表示旋转的角度。向量v=(vx, vy，vz)的分量可以是任意的实数值，该向量用于定义通过坐标原点的旋转轴的方向，后缀为f（单精度浮点float）或d（双精度浮点double），对于二维旋转来说，vx=0.0，vy=0.0，vz=1.0。

缩放矩阵构造函数为glScale<f,d>(sx, sy, sz)，作用是把当前矩阵和一个表示缩放物体的矩阵相乘。sx, sy，sz指定这个缩放物体的矩阵，分别表示在x,y,z方向上的缩放比例，它们可以是任意的实数值，当缩放参数为负值时，该函数为反射矩阵，缩放相对于原点进行，后缀为f（单精度浮点float）或d（双精度浮点double）。

注意这里都是说“把当前矩阵和一个表示移动<旋转, 缩放>物体的矩阵相乘”，而不是直接说“这个函数就是旋转”或者“这个函数就是移动”，这是有原因的，马上就会讲到。

假设当前矩阵为单位矩阵，然后先乘以一个表示旋转的矩阵R，再乘以一个表示移动的矩阵T，最后得到的矩阵再乘上每一个顶点的坐标矩阵v。那么，经过变换得到的顶点坐标就是((RT)v)。由于矩阵乘法满足结合率，((RT)v) = R(Tv))，换句话说，实际上是先进行移动，然后进行旋转。即：实际变换的顺序与代码中写的顺序是相反的。由于“先移动后旋转”和“先旋转后移动”得到的结果很可能不同，初学的时候需要特别注意这一点。

1. OpenGL下的各种变换简介

我们生活在一个三维的世界——如果要观察一个物体，我们可以：

1. 从不同的位置去观察它（人运动，选定某个位置去看）。（视图变换）
2. 移动或者旋转它，当然了，如果它只是计算机里面的物体，我们还可以放大或缩小它（物体运动，让人看它的不同部分）。（模型变换）
3. 如果把物体画下来，我们可以选择：是否需要一种“近大远小”的透视效果。另外， 我们可能只希望看到物体的一部分，而不是全部（指定看的范围）。（投影变换）
4. 我们可能希望把整个看到的图形画下来，但它只占据纸张的一部分，而不是全部（指定在显示器窗口的那个位置显示）。（视口变换）

这些，都可以在OpenGL中实现。

从“相对移动”的观点来看，改变观察点的位置与方向和改变物体本身的位置与方向具有等效性。在OpenGL中，实现这两种功能甚至使用的是同样的函数。

由于模型和视图的变换都通过矩阵运算来实现，在进行变换前，应先设置当前操作的矩阵为“模型视图矩阵”。设置的方法是以GL\_MODELVIEW为参数调用glMatrixMode函数，像这样：

glMatrixMode(GL\_MODELVIEW);

该语句指定一个4×4的建模矩阵作为当前矩阵。

通常，我们需要在进行变换前把当前矩阵设置为单位矩阵。把当前矩阵设置为单位矩阵的函数为：

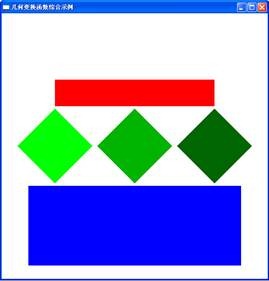
glLoadIdentity();

我们在进行矩阵操作时，有可能需要先保存某个矩阵，过一段时间再恢复它。当我们需要保存时，调用glPushMatrix（）函数，它相当于把当前矩阵压入堆栈。当需要恢复最近一次的保存时，调用glPopMatrix（）函数，它相当于从堆栈栈顶弹出一个矩阵为当前矩阵。OpenGL规定堆栈的容量至少可以容纳32个矩阵，某些OpenGL实现中，堆栈的容量实际上超过了32个。因此不必过于担心矩阵的容量问题。

通常，用这种先保存后恢复的措施，比先变换再逆变换要更方便，更快速。 注意：模型视图矩阵和投影矩阵都有相应的堆栈。使用glMatrixMode来指定当前操作的究竟是模型视图矩阵还是投影矩阵。

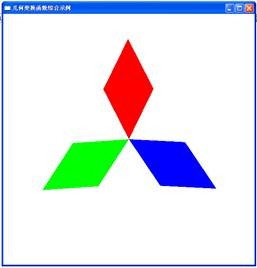
1. 示范代码：(略)

程序运行结果：

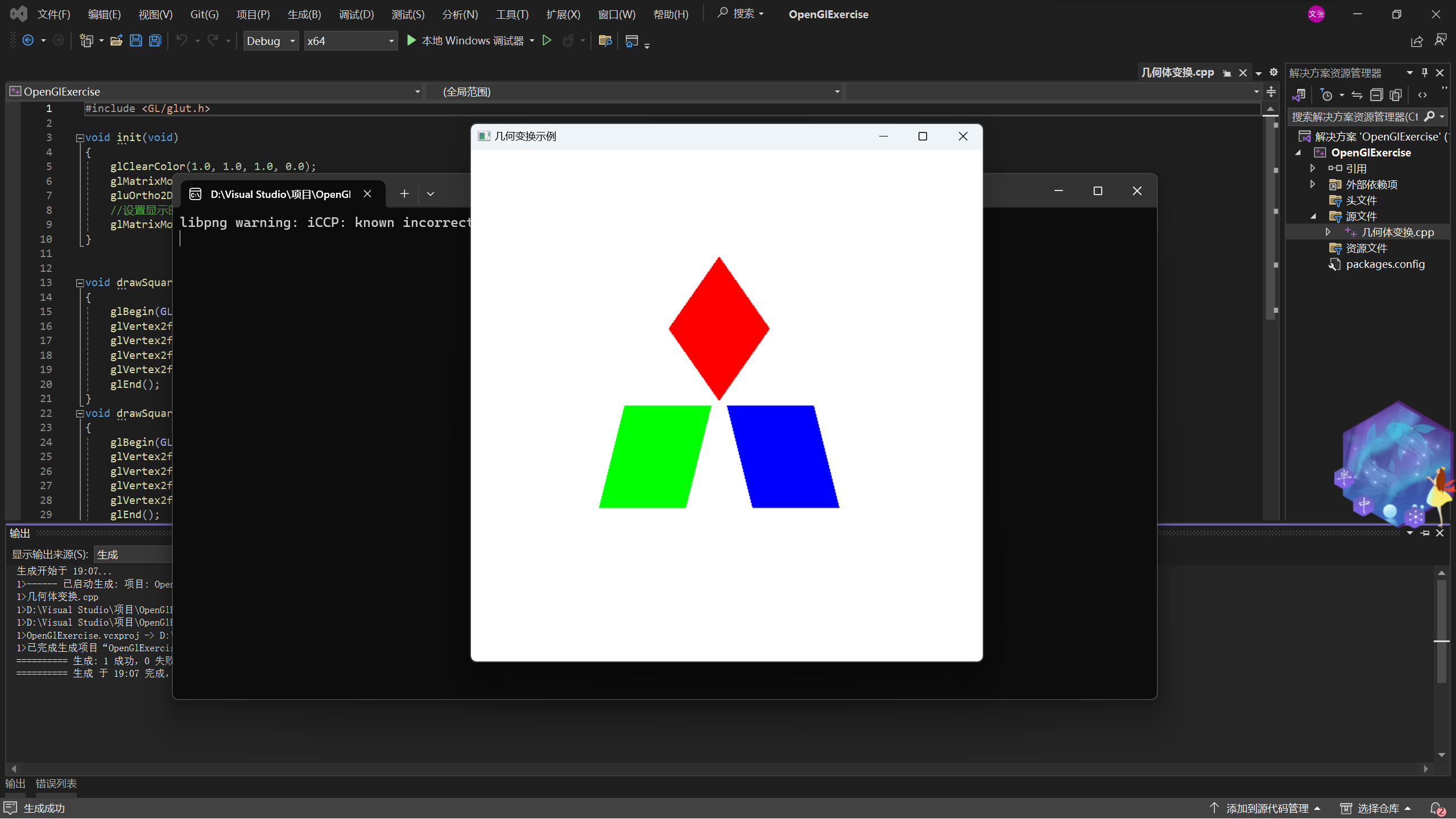


1. 实验作业：

绘制如下图形：



结果演示：



代码：  
