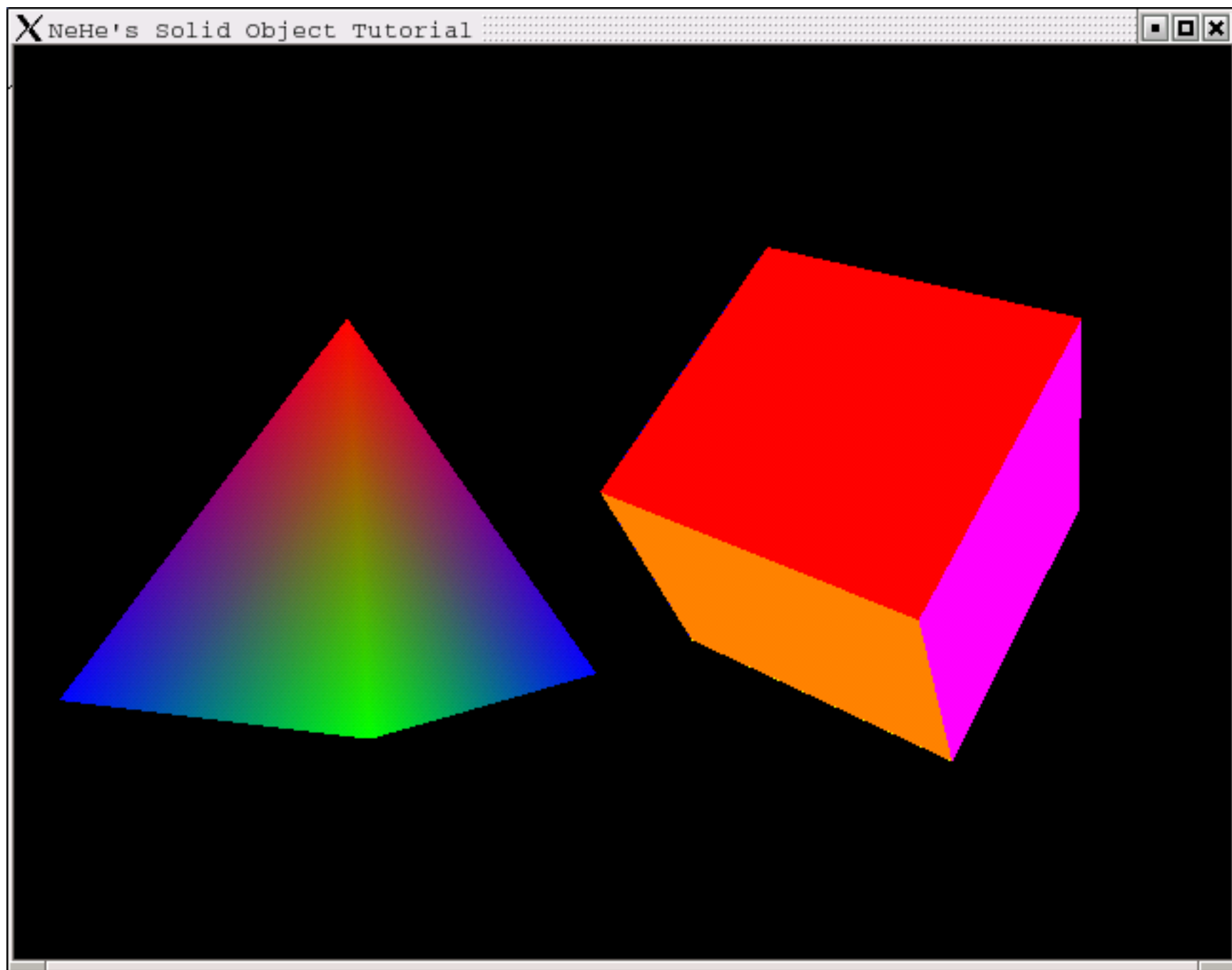


## 向三维进军



在上一课的内容上作些扩展，我们现在开始生成真正的三维对象，而不是象前两节课中那样在三维世界中的二维对象。我们给三角形增加一个左侧面，一个右侧面，一个后侧面来生成一个金字塔（四棱锥）。给正方形增加左、右、上、下及背面生成一个立方体。

我们混合金字塔上的颜色，创建一个平滑着色的对象。给立方体的每一面则来个不同的颜色。

其实只需在上节课的代码上增加几行就可以了。

### NeHeWidget类

（由nehewidget.cpp展开。）

```
void NeHeWidget::paintGL()
{
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
```

```
glLoadIdentity();
glTranslatef( -1.5, 0.0, -6.0 );
glRotatef( rTri, 0.0, 1.0, 0.0 );
```

有些人可能早已在上节课中的代码上尝试自行创建3D对象了。但经常有人来信问我：“我的对象怎么不会绕着其自身的轴旋转？看起来总是在满屏乱转。”要让您的对象绕自身的轴旋转，您必须让对象的中心坐标总是( 0.0, 0.0, 0.0 )。

下面的代码创建一个绕着其中心轴旋转的金字塔。金字塔的上顶点离中心一个单位，底面离中心也是一个单位。上顶点在底面的投影位于底面的中心。

注意所有的面- 三角形都是逆时针次序绘制的。这点十分重要，在以后的课程中我会作出解释。现在，您只需明白要么都逆时针，要么都顺时针，但永远不要将两种次序混在一起，除非您有足够的理由必须这么做。

```
glBegin( GL_TRIANGLES );
glColor3f( 1.0, 0.0, 0.0 );
glVertex3f( 0.0, 1.0, 0.0 );
glColor3f( 0.0, 1.0, 0.0 );
glVertex3f( -1.0, -1.0, 1.0 );
glColor3f( 0.0, 0.0, 1.0 );
glVertex3f( 1.0, -1.0, 1.0 );
```

上面是我们绘制的金字塔的前侧面。因为所有的面都共享上顶点，我们将这点在所有的三角形中都设置为红色。底边上的两个顶点的颜色则是互斥的。前侧面的左下顶点是绿色的，右下顶点是蓝色的。这样相邻右侧面的左下顶点是蓝色的，右下顶点是绿色的。这样四边形的底面上的点的颜色都是间隔排列的。

还应注意到后面的三个侧面和前侧面处于同一个glBegin( GL\_TRIANGLES )和glEnd()语句中间。因为我们是通过三角形来构造这个金字塔的。OpenGL知道每三个点构成一个三角形。当它画完一个三角形之后，如果还有余下的点出现，它就以为新的三角形要开始绘制了。OpenGL在这里并不会将四点画成一个四边形，而是假定新的三角形开始了。所以千万不要无意中增加任何多余的点。

```
glColor3f( 1.0, 0.0, 0.0 );
glVertex3f( 0.0, 1.0, 0.0 );
glColor3f( 0.0, 0.0, 1.0 );
glVertex3f( 1.0, -1.0, 1.0 );
glColor3f( 0.0, 1.0, 0.0 );
glVertex3f( 1.0, -1.0, -1.0 );
```

绘制右侧面。注意其底边上的两个顶点的X坐标位于中心右侧的一个单位处。顶点则位于Y轴上的一个单位处，且Z坐标正好处于底边的两顶点的Z坐标中心。右侧面从上顶点开始向外侧倾斜至底边上。

这次的左下顶点用蓝色绘制，以保持与前侧面的右下顶点的一致。蓝色将从这个角向金字塔的前侧面和右侧面扩展并与其他颜色混合。

```
glColor3f( 1.0, 0.0, 0.0 );
glVertex3f( 0.0, 1.0, 0.0 );
glColor3f( 0.0, 1.0, 0.0 );
glVertex3f( 1.0, -1.0, -1.0 );
glColor3f( 0.0, 0.0, 1.0 );
```

```
glVertex3f( -1.0, -1.0, -1.0 );
```

后侧面。再次切换颜色。左下顶点又回到绿色，因为后侧面与右侧面共享这个角。

```
glColor3f( 1.0, 0.0, 0.0 );
glVertex3f( 0.0, 1.0, 0.0 );
glColor3f( 0.0, 0.0, 1.0 );
glVertex3f( -1.0, -1.0, -1.0 );
glColor3f( 0.0, 1.0, 0.0 );
glVertex3f( -1.0, -1.0, 1.0 );
```

最后画左侧面。又要切换颜色。左下顶点是蓝色，与后侧面的右下顶点相同。右下顶点是蓝色，与前侧面的左下顶点相同。

到这里金字塔就画完了。因为金字塔只绕着Y轴旋转，我们永远都看不见底面，因而没有必要添加底面。如果您觉得有经验了，尝试增加底面（正方形），并将金字塔绕X轴旋转来看看您是否作对了。确保底面四个顶点的颜色与侧面的颜色相匹配。

```
glEnd();
```

接下来开始画立方体。他由六个四边形组成。所有的四边形都以逆时针次序绘制。就是说先画右上角，然后左上角、左下角、最后右下角。您也许认为画立方体的背面的时候这个次序看起来好像顺时针，但别忘了我们从立方体的背后看背面的时候，与您现在所想的正好相反。（译者注：您是从立方体的外面来观察立方体的。）

```
glLoadIdentity();
glTranslatef( 1.5, 0.0, -7.0 );
```

注意到这次我们将立方体移地更远离屏幕了。因为立方体的大小要比金字塔大，同样移入6个单位时，立方体看起来要大的多。这是透视的缘故。越远的对象看起来越小：)。

```
glRotatef( rQuad, 1.0, 1.0, 1.0 );
```

```
glBegin( GL_QUADS );
glColor3f( 0.0, 1.0, 0.0 );
glVertex3f( 1.0, 1.0, -1.0 );
glVertex3f( -1.0, 1.0, -1.0 );
glVertex3f( -1.0, 1.0, 1.0 );
glVertex3f( 1.0, 1.0, 1.0 );
```

先画立方体的顶面。从中心上移一单位，注意Y坐标始终为一单位，表示这个四边形与Z轴平行。先画右上顶点，向右一单位，再屏幕向里一单位。然后左上顶点，向左一单位，再屏幕向里一单位。然后是靠近观察者的左下和右下顶点。就是屏幕往外一单位。

```
glColor3f( 1.0, 0.5, 0.0 );
glVertex3f( 1.0, -1.0, 1.0 );
glVertex3f( -1.0, -1.0, 1.0 );
glVertex3f( -1.0, -1.0, -1.0 );
glVertex3f( 1.0, -1.0, -1.0 );
```

底面的画法和顶面十分类似。只是Y坐标变成了-1。如果我们从立方体的下面来看立方体的话，您会注意到右上角离观察者最近，因此我们先画离观察者最近的顶点。然后是左上顶点最后才是屏幕里面

的左下和右下顶点。

如果您真的不在乎绘制多边形的次序（顺时针或者逆时针）的话，您可以直接拷贝顶面的代码，将Y坐标从1改成-1，也能够工作。但一旦您进入象纹理映射这样的领域时，忽略绘制次序会导致十分怪异的结果。

```
glColor3f( 1.0, 0.0, 0.0 );
glVertex3f( 1.0, 1.0, 1.0 );
glVertex3f( -1.0, 1.0, 1.0 );
glVertex3f( -1.0, -1.0, 1.0 );
glVertex3f( 1.0, -1.0, 1.0 );
```

立方体的前面。保持Z坐标为一单位，前面正对着我们。

```
glColor3f( 1.0, 1.0, 0.0 );
glVertex3f( 1.0, -1.0, -1.0 );
glVertex3f( -1.0, -1.0, -1.0 );
glVertex3f( -1.0, 1.0, -1.0 );
glVertex3f( 1.0, 1.0, -1.0 );
```

立方体后面的绘制方法与前面类似。只是位于屏幕的里面。注意Z坐标现在保持-1不变。

```
glColor3f( 0.0, 0.0, 1.0 );
glVertex3f( -1.0, 1.0, 1.0 );
glVertex3f( -1.0, 1.0, -1.0 );
glVertex3f( -1.0, -1.0, -1.0 );
glVertex3f( -1.0, -1.0, 1.0 );
```

还剩两个面就完成了。您会注意到总有一个坐标保持不变。这一次换成了X坐标。因为我们在画左侧面。

```
glColor3f( 1.0, 0.0, 1.0 );
glVertex3f( 1.0, 1.0, -1.0 );
glVertex3f( 1.0, 1.0, 1.0 );
glVertex3f( 1.0, -1.0, 1.0 );
glVertex3f( 1.0, -1.0, -1.0 );
```

立方体的最后一个面了。X坐标保持为一单位。逆时针绘制。您愿意的话，留着这个面不画也可以，这样就是一个盒子:)

```
glEnd();
```

或者您要是有兴趣可以改变立方体所有顶点的色彩值，象金字塔那样混合颜色。您会看见一个非常漂亮的彩色立方体，各种颜色在它的各个表面流淌。

```
rTri += 0.2;
rQuad -= 0.15;
```

```
}
```

这一课又结束了。到这里您应该已经较好的掌握了在三维空间创建对象的方法。必须将OpenGL屏幕想象成一张很大的画纸，后面还带着许多透明的层。差不多就是个由大量的点组成的立方体。这些点从左至右、从上至下、从前到后的布满了这个立方体。如果您能想象的出在屏幕的深度方向，应该在

设计新三维对象时没有任何问题。

如果您对三维空间的理解很困难的话，千万不要灰心！刚开始的时候，领会这些内容会很难。象立方体这样的对象是您练习的好例子。继续努力吧！

本课程的[源代码](#)。

[\[上一课： 旋转\]](#) [\[Qt OpenGL教程主页\]](#) [\[下一课： 纹理映射\]](#)

---

<http://www.qiliang.net>

<mailto:cavendish@qiliang.net>

2002年12月22日