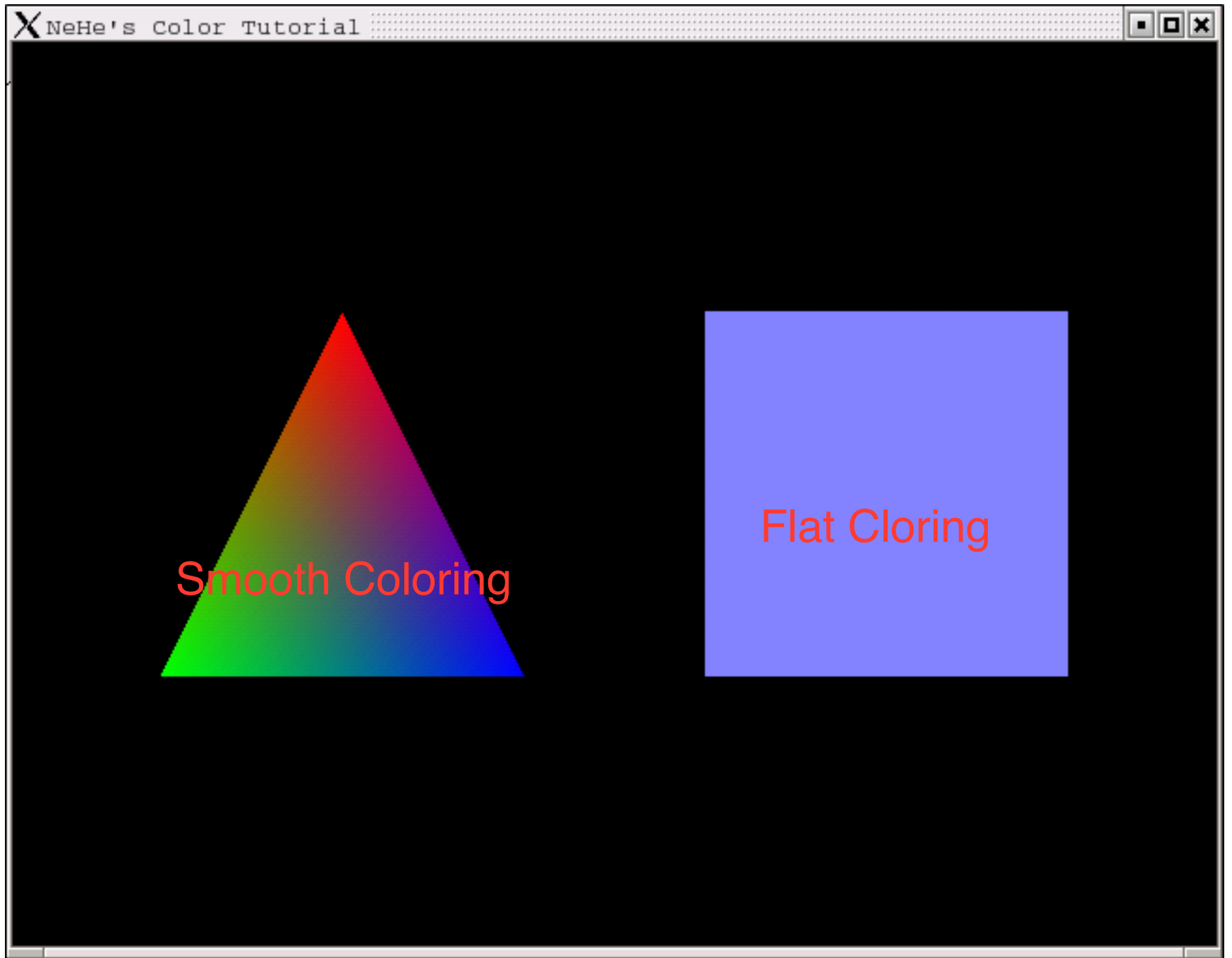


上色



上一课中我教给您三角形和四边形的绘制方法。这一课我将教您给三角形和四边形添加两种不同类型的着色方法。使用单调着色（Flat coloring）给四边形涂上固定的一种颜色。使用平滑着色（Smooth coloring）将三角形的三个顶点的不同颜色混合在一起，创建漂亮的色彩混合。

我们只要修改第二课中的NeHeWidget类中的paintGL()函数就可以了。

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 );
```

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

红色。

如果您还记得上节课的内容，这段代码在屏幕的左半部分绘制三角形。这一行代码是我们第一次使用命令`glColor3f(r, g, b)`。括号中的三个参数依次是红、绿、蓝三色分量。取值范围可以从0.0到1.0。类似于以前所讲的清除屏幕背景命令。

我们将颜色设为红色（纯红色，无绿色，无蓝色）。

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

上顶点。

接下来的一行代码设置三角形的第一个顶点（三角形的上顶点），并使用当前颜色（红色）来绘制。从现在开始所有的绘制的对象的颜色都是红色，直到我们将红色改变成别的什么颜色。

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

绿色。

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

左下顶点。

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

蓝色。

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

右下顶点。

```
glEnd();
```

`glEnd()`出现后，三角形将被填充。但是因为每个顶点有不同的颜色，因此看起来颜色从每个角喷出，并刚好在三角形的中心汇合，三种颜色相互混合。这就是平滑着色。

```
glTranslatef( 3.0, 0.0, 0.0 );
```

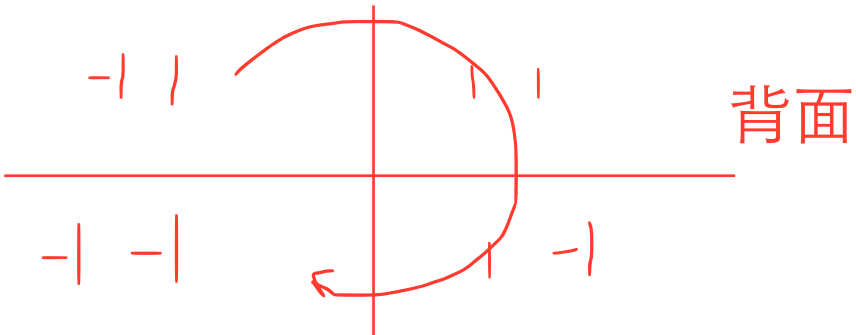
```
glColor3f( 0.5, 0.5, 1.0 );
```

一次性将颜色设置为蓝色。

现在我们绘制一个单调着色——蓝色的正方形。最重要的是要记住，设置当前色之后绘制的所有东东都是当前色的。以后您所创建的每个工程都要使用颜色。即便是在完全采用纹理贴图的时候，`glColor3f`仍旧可以用来调节纹理的色调。等等...，以后再说吧。

我们必须要做的事只需将颜色一次性的设为我们想采用的颜色（本例采用蓝色），然后绘制场景。每个顶点都是蓝色的，因为我们没有告诉OpenGL要改变顶点的颜色。最后的结果是.....全蓝色的正方形。再说一遍，顺时针绘制的正方形意味着我们所看见的是四边形的背面。

```
glBegin( GL_QUADS );
glVertex3f( -1.0,  1.0,  0.0 );
glVertex3f(  1.0,  1.0,  0.0 );
glVertex3f(  1.0, -1.0,  0.0 );
glVertex3f( -1.0, -1.0,  0.0 );
glEnd();
}
```



在这一课中，我试着尽量详细的解释如何为您的OpenGL多边形添加单调和平滑的着色效果的步骤。改改代码中的红绿蓝分量值，看看最后有什么样的结果。

本课程的[源代码](#)。

[\[上一课：你的第一个多边形\]](#) [\[Qt OpenGL教程主页\]](#) [\[下一课：旋转\]](#)