变量 g_last 和 now 都是 Date 对象的 now() 方法的返回值，其单位是毫秒（1/1000 秒）。所以，如果你想让三角形以 ANGLE_STEP（度 / 秒）来旋转，你还需要将 ANGLE_STEP 乘以 elapsed/1000 来计算旋转角度。我们先将 ANGLE_STEP 与 elapsed 相乘，然后再除以 1000，其结果是相同的（第 121 行）。

最后，在返回这一帧的旋转角 newAngle 的同时，还需要保证它始终小于 360 度（第 122 行）并返回结果。

在浏览器中运行 RotatingTriangle.html，可以看到，三角形正在按照恒定的速度进行旋转。在接下来的几章中，我们还会使用这个方式来实现动画，所以本节的代码值得仔细研究，你应该确保掌握了其中的每个细节。

用示例程序做实验

在这一小节中，我们来实现一个包含了复合变换的动画 RotatingTranslatedTriangle。该程序首先将三角形平移至 X 轴正半轴 0.35 单位处，然后以 45 度 / 秒的速度旋转该三角形。

如果你还记得，复合变换可以使用变换矩阵相乘来实现（参见第 3 章），那么这一小节的内容就再简单不过了。

我们可以插入一个平移操作来实现，modelMatrix 变量已经包含了旋转矩阵，我们直接在它上面调用 translate() 方法（而非 setTranslate() 方法），就可以使 modelMatrix 乘以平移矩阵（第 102 行）：

```
 99  function draw(gl,n, currentAngle, modelMatrix, u_ModelMatrix){
100    // 设置旋转矩阵
101    modelMatrix.setRotate(currentAngle, 0, 0, 1);
102    modelMatrix.translate(0.35, 0, 0);
103    // 将旋转矩阵传输给顶点着色器
104    gl.uniformMatrix4fv( u_ModelMatrix, false, modelMatrix.elements);
```

运行示例程序，你会看到如图 4.11 所示的动画效果。