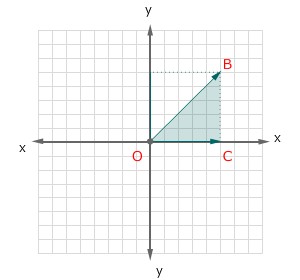
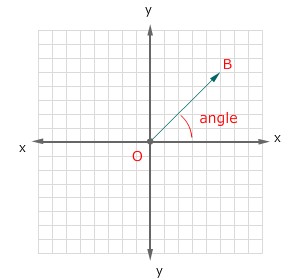
[**Flash与3D编程探秘（四）- 摄像机旋转基础知识**](http://www.cnblogs.com/yangzhou1030/archive/2008/11/05/1326790.html)

日期：2008年10月

#### ****关于三角函数****

现在已经把最基本的移动摄像机技巧介绍完了，并且我相信上篇的几个例子也应该让读者加深了印象。可是你会发现前面我们只是把摄像机沿着x轴，y轴或 者z轴移动摄像机，可是实际中可以把摄像机向左，向右，向上或者向下旋转一定的角度，这样在观看空间时就有更大的自由度。 不过介绍旋转之前，我想介绍一些关于三角法的计算（Oh,No!），如果你还对三角函数不熟悉，那么文章看起来可能会有些费解。不过不要担心，中国人有着 聪明的头脑，这些对你来说很容易。提醒一下：如果下面的内容对你来说太容易了，那么可以跳过这些。不过我想我还是给已经忘记的（Just like me）各位补上一课，毕竟就是这些简单的知识驱使着我们的3D空间旋转着。

三角法是数学的一个分支，我们主要用它来分析三角形的边和角度的关系。为什么要三角形要和角度有关系呢？任意一条在2D空间里有旋转角度的直线，它 都会在x轴和y轴有相应的映射(当然在3D空间里， 我们并不光有x轴和y轴)。假设有一条线段从原点到点B（4,0），当你把这个线段哦OB沿着原点旋转一定角度，过B（旋转后）向x轴做一条垂线 BC，那么x轴和BC,OB所组成的就是一个直角三角形。当然你可以把OC看作是线段OB在x的投影。那么对这个旋转角度使用sin和cos就可以计算得 到我们这条线段在x和y的分量。

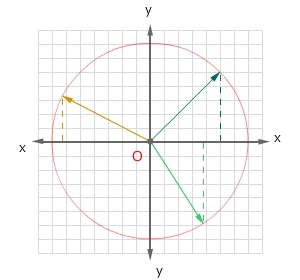
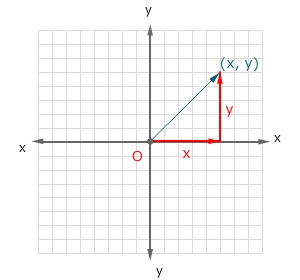
  
**旋转一条直线得到的三角形**

**以原点为中心旋转小P**

以原点为中心旋转只是个例子，这里提到的旋转点不一定是原点，可以把任意一点作为旋转的原点。你会发现在旋转中，当x线段OB旋转一定角度后，它会 与旋转前的线段重合，B在旋转中所过的点的轨迹就是一个圆，那旋转的这条线段就OB就是这个圆的半径。OK，这就是在Flash里需要知道基本三角函数。 下面两个是根据圆的半径得到x和y分量的公式：

object.x = Math.cos(angle) \* radius;  
object.y = Math.sin(angle) \* radius;

[复制代码](javascript:void(0);)



**旋转后的直线在x和y轴的分量**

当然，上面的公式在旋转点为原点的情况下成立，如果旋转点不是原点的话使用：

object.x = origin.x + Math.cos(angle) \* radius;  
object.y = origin.y + Math.sin(angle) \* radius;

[复制代码](javascript:void(0);)

#### ****弧度****

当测量物体小P的旋转角度的时候，可以使用度数，这也是最常用的，它从0到360沿逆时针方面递增。但是Flash并不知道360是什么，它所知道 的只是弧度(当然你可以自己写sin和cos函数，用0到360作为参数)。这里的弧度可以这样理解，360度数是2\*PI弧度，那么就是说360度旋转 是一个整圆的话，2\*PI旋转也是同样效果。把度数转化成弧度的公式是：

randians = degrees \* (Math.PI / 180); PI = 3.1415926535897932384626433832795...

#### ****反三角函数****

在Flash里，可以通过直角三角形的两个边的比率得到角度，用下面的代码即可：

angle = Math.atan2(object.y, object.x);  
angle = Math.atan(y/x);

[复制代码](javascript:void(0);)

还有一个要说明的那就是勾股定理：

hypotenuse = Math.sqrt(x\*x + y\*y);

那么你现在已经具备2D旋转的基本知识了，再看一下3D，3D旋转中加进了z轴，hum，那么就把xy平面，yz平面，zx平面的旋转组合起来，就 得到摄像机的全方位旋转。不过指得注意的是，三角法不能直接运用到摄像机旋转中，可以在横向旋转摄像机的时候保持摄像机的其他轴旋转角度不变。另外你还可 以旋转x，y和z轴已到达目的，比如要旋转180度的话，可以使z旋转到x的位置然后再转到z的位置，同时保持y轴不动。

**旋转x和z轴**

#### ****横向旋转摄像机****

上面讲述了一些旋转摄像机的原理，我们更关心的是如何使用这些原理来解决问题。联想一下实际，对于摄像机的横向旋转来说，摄像机的所在位置的高度 y，深度z和横向x都是保持不变的，唯一改变的就是摄像机的旋转角度。也就是说当一个空间中的物体在位置不变的情况下，摄像机与它的距离是不变的，于是问 题转化成在角度变化的情况下保证它们的距离不变。设旋转后的角度为a，那么物体所在的x和y（以摄像机为原点建立坐标系）就是：

x = distance \* Math.cos(a);  
y = distance \* Math.sin(a);

[复制代码](javascript:void(0);)

如果摄像机所在的坐标为（x0,y0）的话，就可以得到物体所在的坐标（还是上面讲过的公式）：

x = x0 + distance \* Math.cos(a);  
y = y0 + distance \* Math.sin(a);

[复制代码](javascript:void(0);)

以上所述的仅是对一个物体的操作，同样道理对所有舞台上的物体进行操作，那么所看到的就是摄像机旋转后的景象。

**对比旋转物体和旋转摄像机**

在上面的动画中，我们俯瞰整个场景，对比一下旋转摄像机和旋转舞台上所有物体的差异。当然对于其他的平面（纵向旋转摄像机）来说，可以运用相同的理 论对物体进行操作。总而言之那么，你应该有种感觉你已经具备所有摄像机旋转的理论知识了，不过还有一小点，将在下一篇文章进行分析。

http://www.cnblogs.com/yangzhou1030/archive/2008/11/05/1326790.html