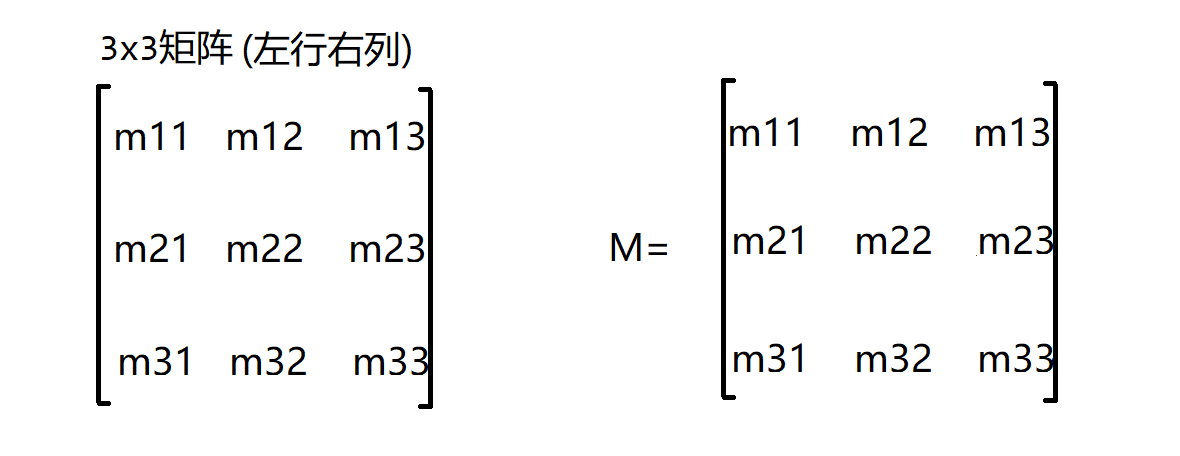
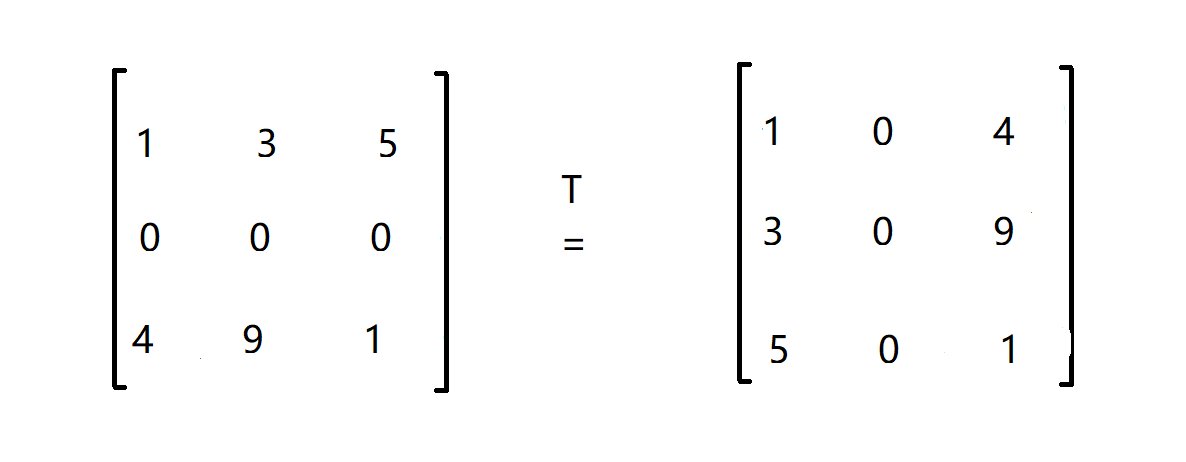
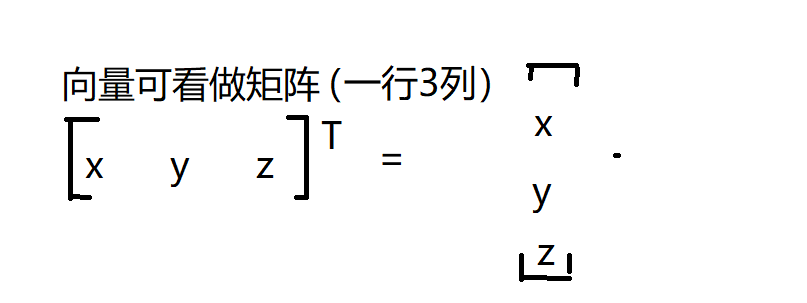
### 矩阵

类似二维数组，可以使用二维数组记录矩阵



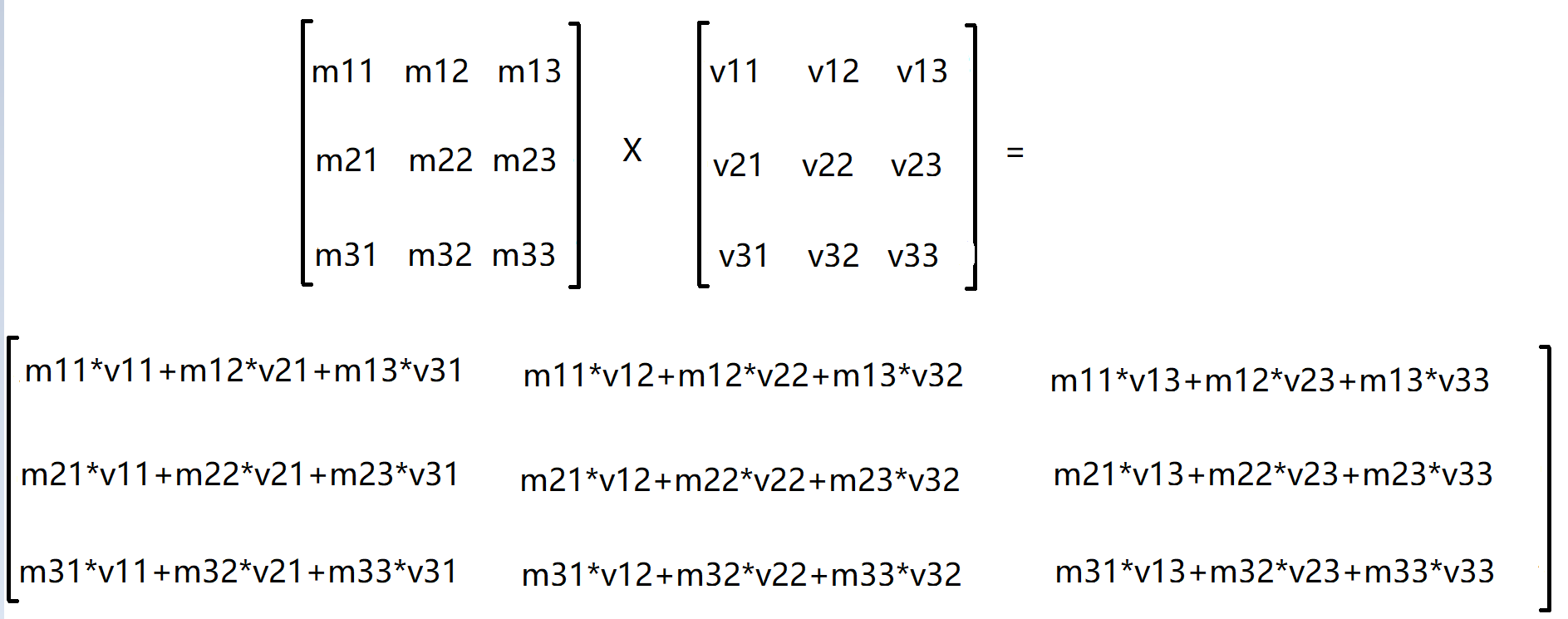
矩阵转置 行变成列，列变成行

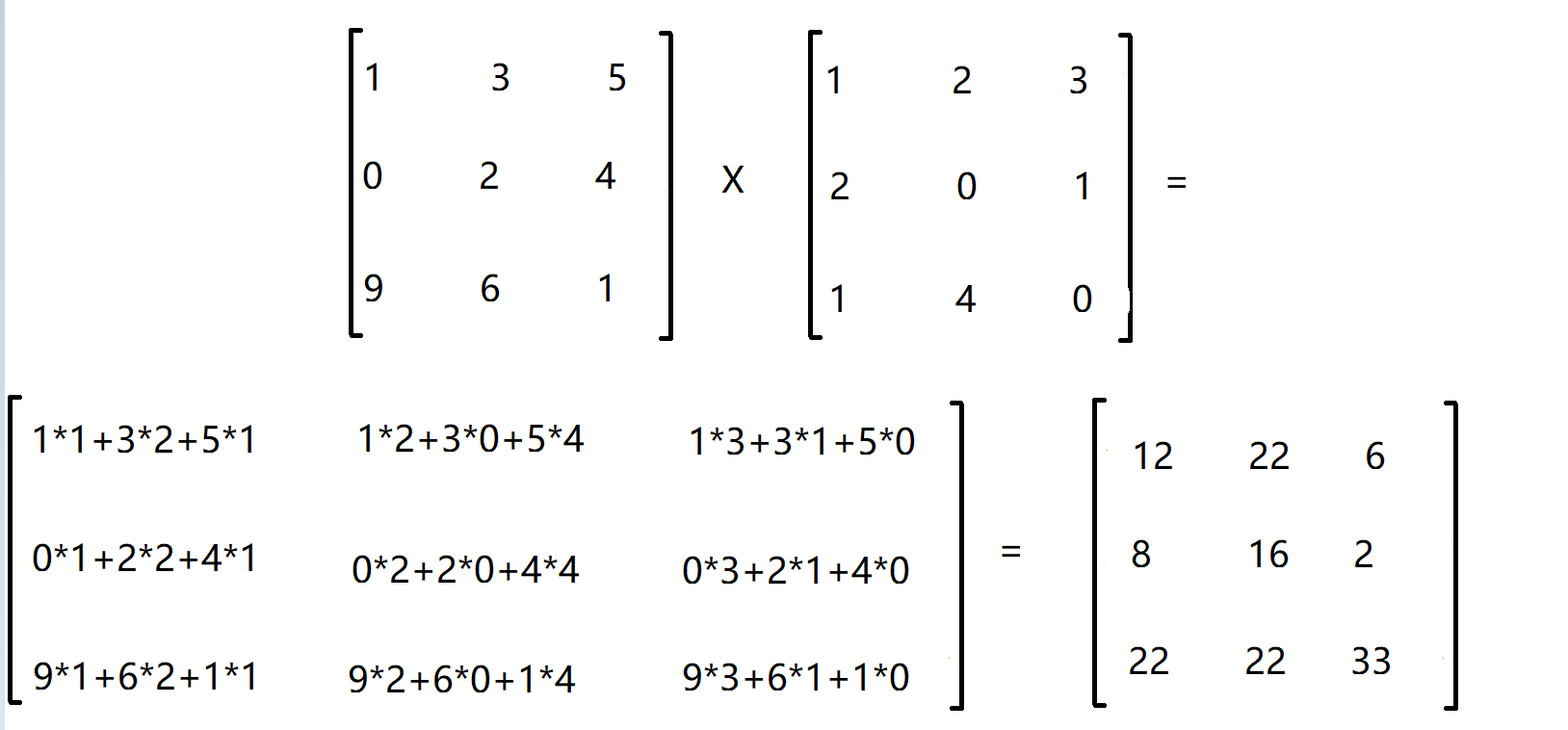




矩阵x标量=矩阵当中每一个元素x标量

矩阵x矩阵



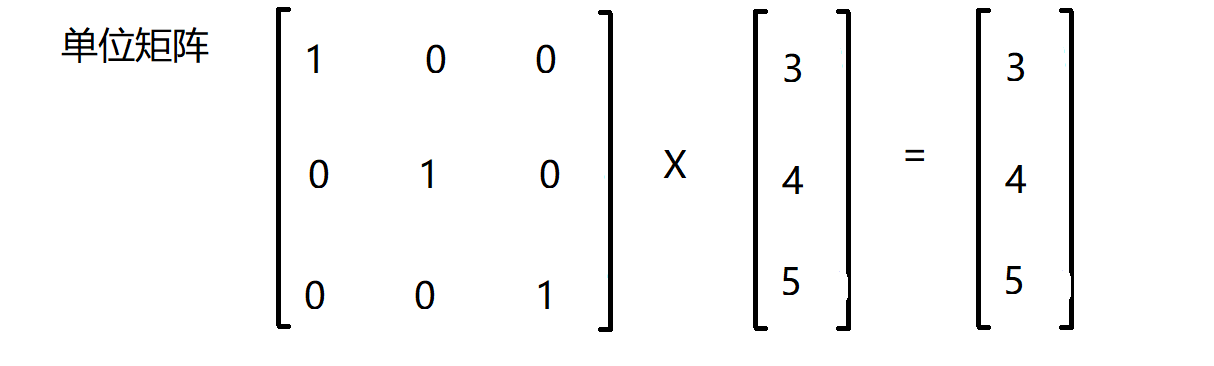


矩阵相乘的位置很关键，将上述两个矩阵转置后交互位置再相乘，结果为上述结果矩阵的转置矩阵。

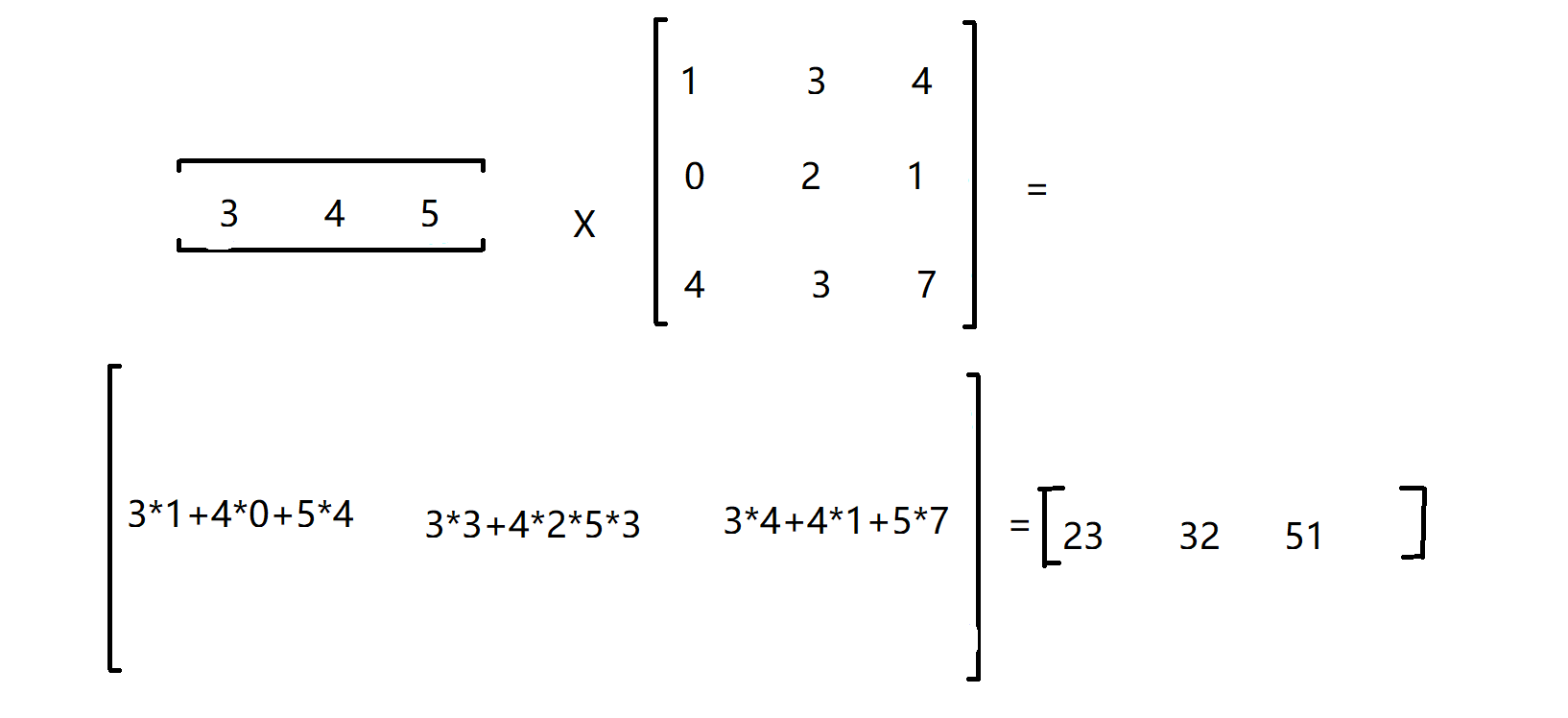
两个矩阵相乘两个矩阵必须前一矩阵的列必须与后一矩阵的行维度相同

即MxV的矩阵与MxS的矩阵 V必须等于M 最后等于的维度为MxS 不符合条件的可以将向量转置之后再进行乘法运算

单位矩阵：单位矩阵形式，对角线为1，其他元素为0。与矩阵相乘之后结果不变

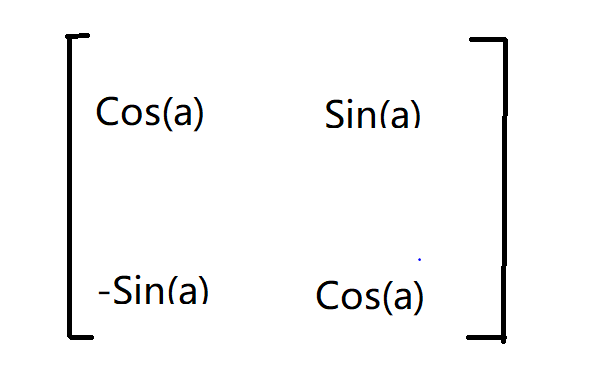


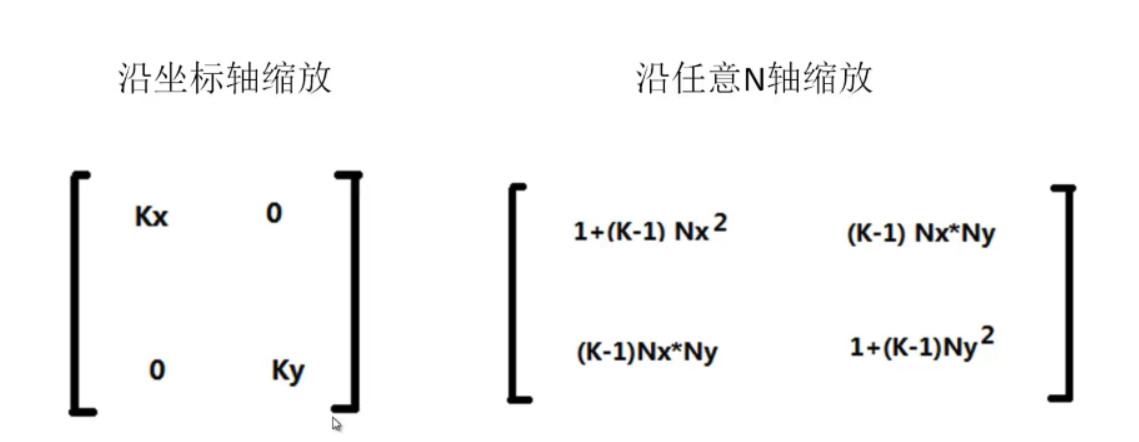
矩阵与向量相乘：



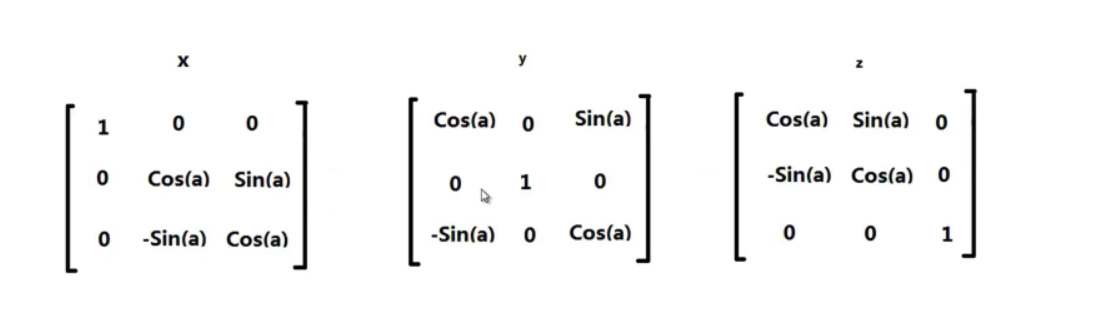
2D旋转矩阵

饶坐标中心旋转a角度 进行计算的时候只需要将原始的顶点2D向量与当前矩阵相乘即可

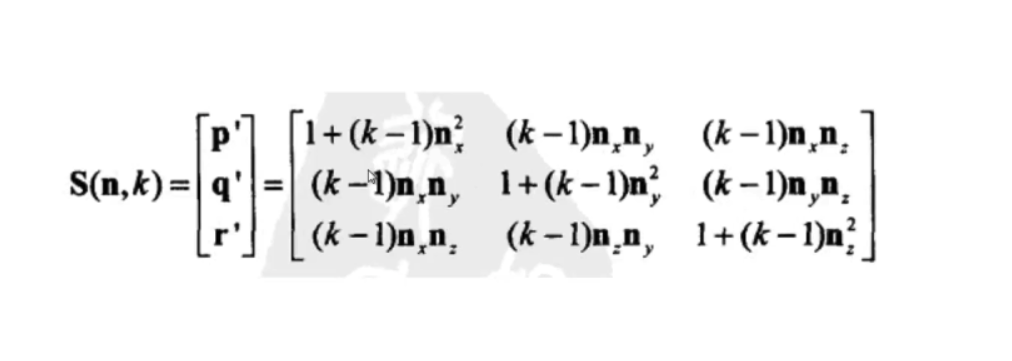




3D旋转矩阵

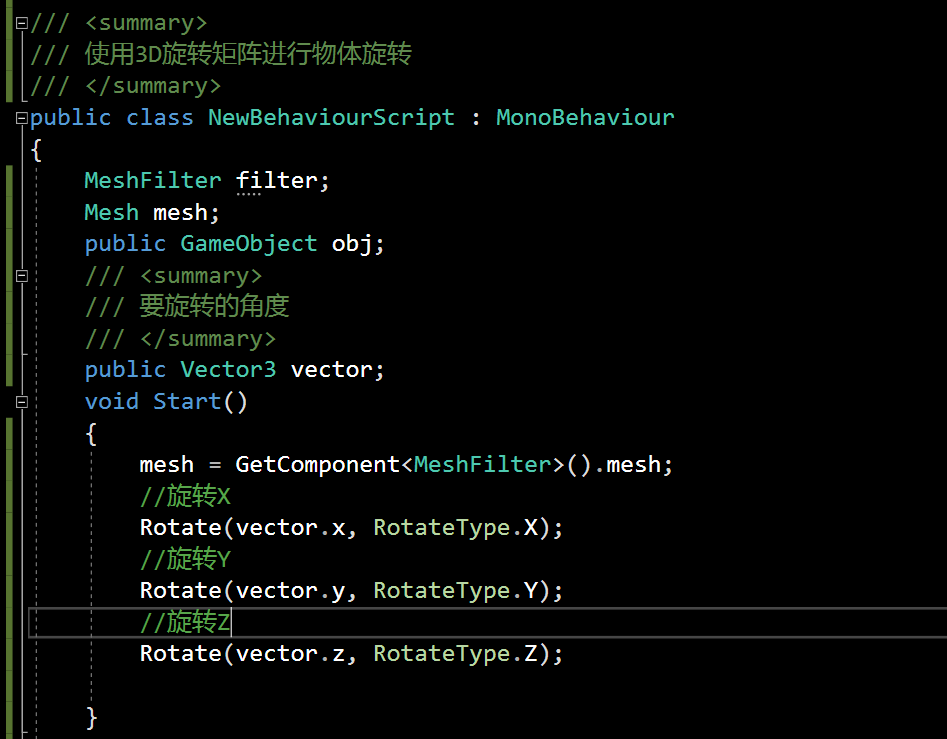


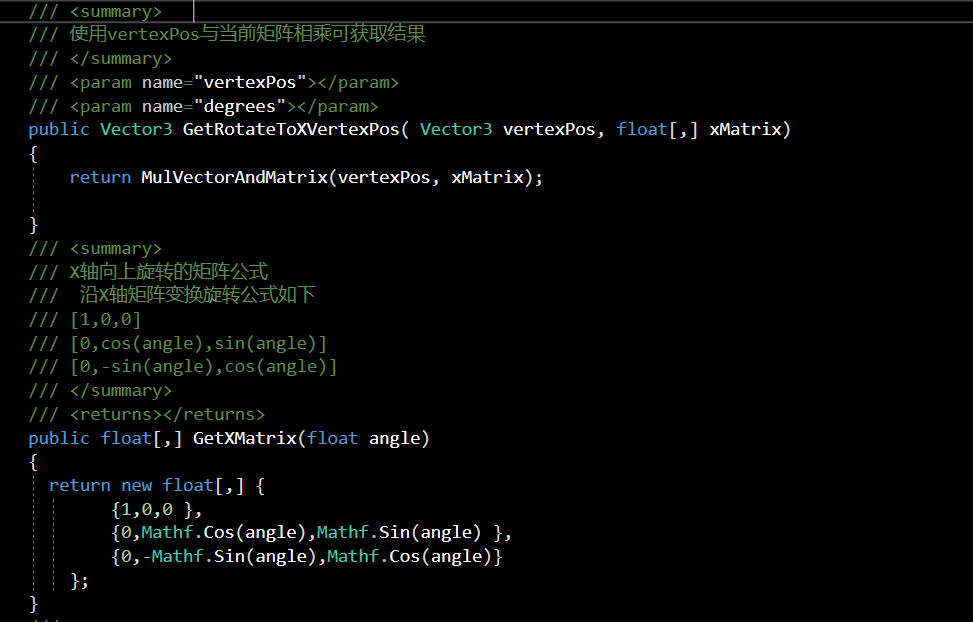
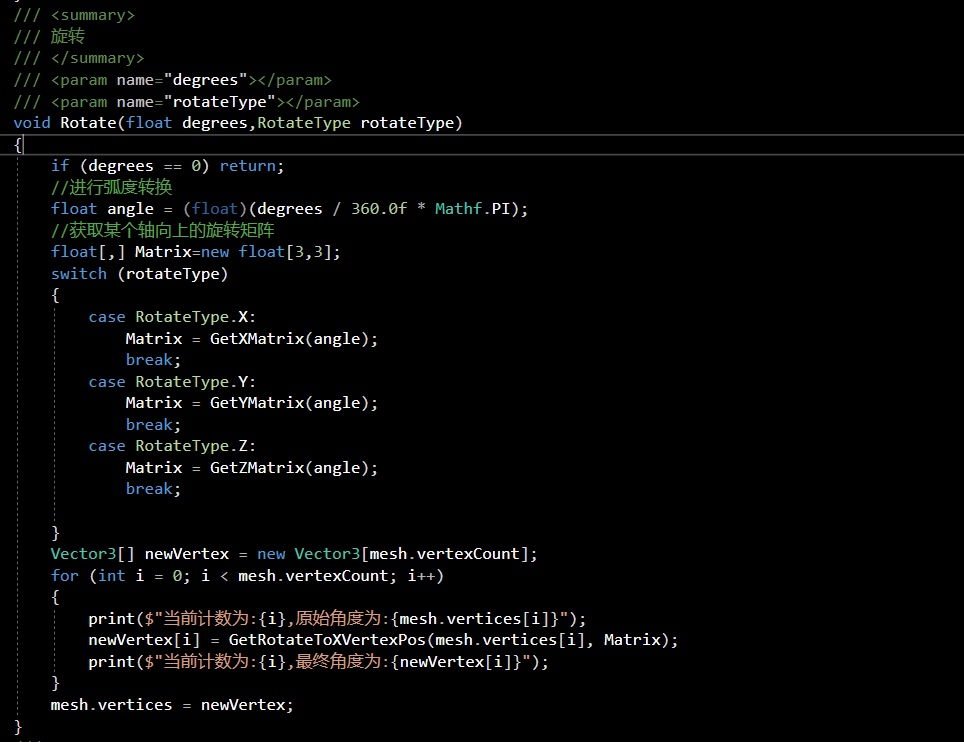
3D缩放矩阵



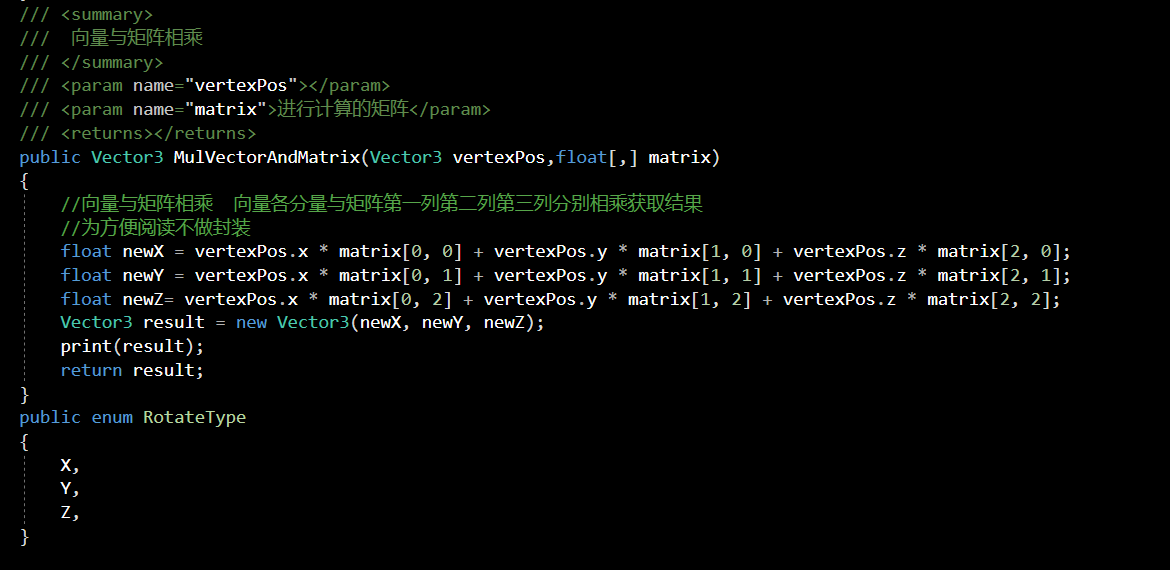
公式使用方法，以3D旋转矩阵为例，在unity当中实现自己的简易Rotate的方法：

思路如下：获取物体的MeshFilter当中的Mesh中每一个顶点，然后将Mesh当中每一个顶点与3D旋转矩阵当中对应轴向的旋转矩阵相乘，代码如下：

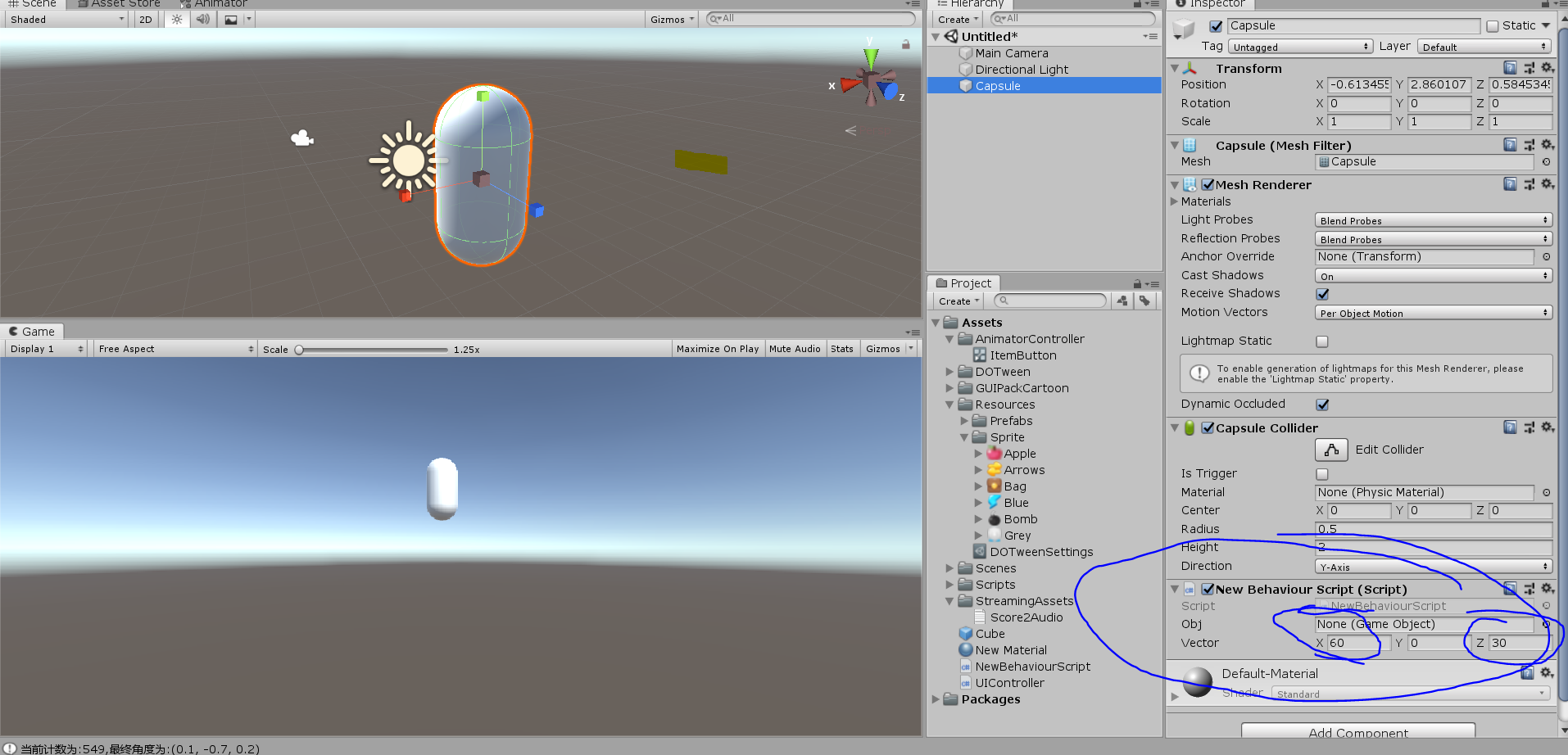




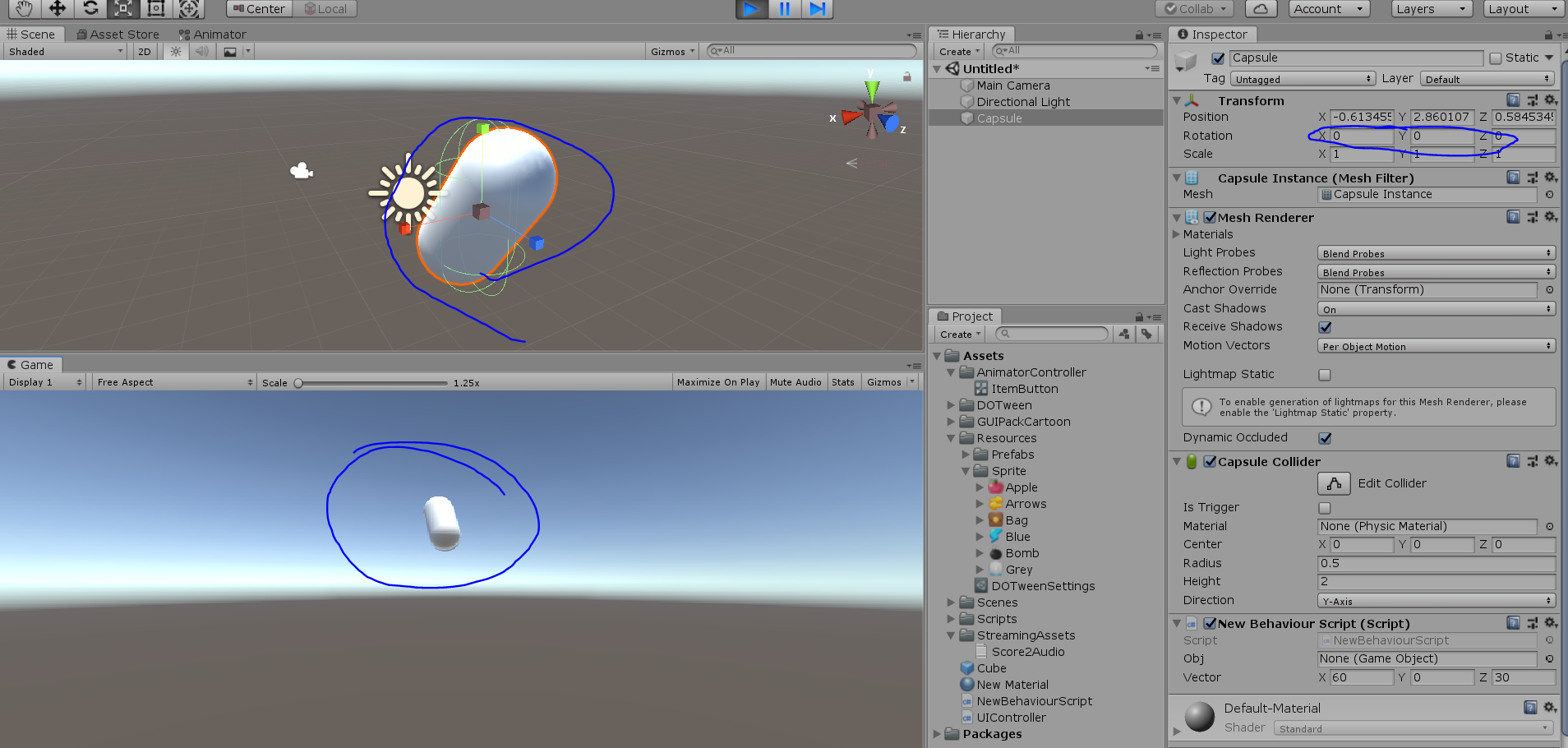




效果如下：挂载脚本分别让当前物体沿X轴旋转60°，z轴旋转30°



运行结果如下：Transform的组件角度未变，直接修改顶点为设置的角度



其他公式的使用方式相同

平移如下

