**作业二：第一部分**

一、矩阵栈

二、输入文件

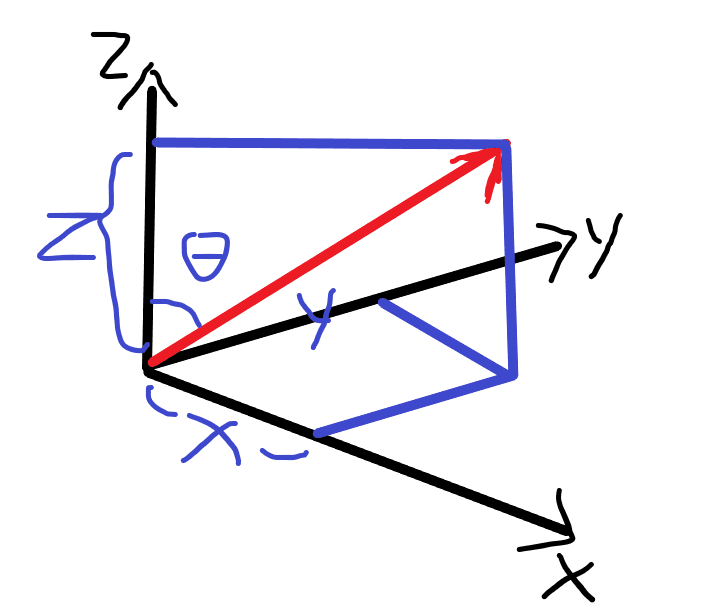
三、点与骨骼

最开始写的时候是没有用矩阵栈的，因为注意到有一个bindWorldToJointTransform矩阵。

用其存储当前坐标系矩阵，也就是父节点所有矩阵相乘的结果。然后就可以模拟栈的做法，得到相同结果。

对于画出骨骼，应该是花费最多时间的地方。

首先是解法，由于需要用递归，因此想了多种解法：最终为，使用遍历点的递归函数格式，然后以当前关节为起点，以其子节点为终点进行划线。而由于全为相对位置的矩阵，因此可以将当前关节矩阵看做原点，以另一个矩阵的三维坐标看作为一个点，那么我们可以得到如图的一个红色向量。而距离（也就是我们要求的伸缩比例length）就是此向量的长度。

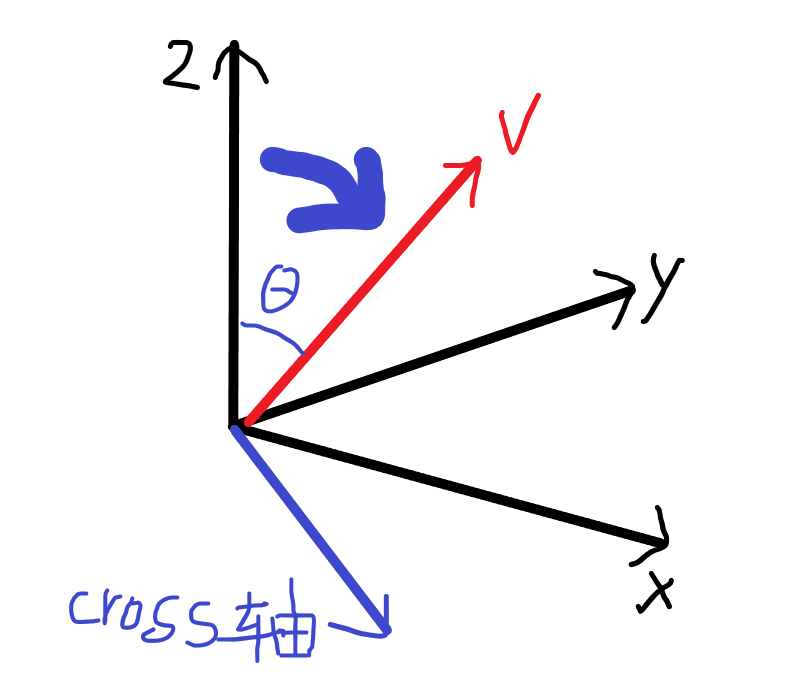


那么我们需要的是将一个指向z轴的方块旋转到此方向。那么旋转轴即为两个向量的叉乘得到的轴，再旋转一个角度即可。而这个角度可以把向量单位化，向量长度为1，而向量与z轴的夹角的cos值为 z/1，单位化便于计算，所以角度为arccos（z）。那么我们可以根据已提供的函数传入这些参数就可以得到结果。

其中有许多需要注意的细节。以下是一些个性问题：

循环中的变量（矩阵）初始化，防止由于当前关节有多个子节点，由于计算使得当前节点的矩阵改变。

对于旋转角度，由于叉积方位未知，虽然角度一定是在180以内，但是对于之后的骨骼平移的方向还是有关系的。由于骨骼最开始位置是以当前节点为原点，那么它平移的方向是需要考虑的。可通过判断旋转角度是否是钝角来得到方向。



四、用户拖动关节旋转滑块

最初的方法是直接将当前的关节（joint）的矩阵（transform） 自身连乘三个变换矩阵，rotate x，y，z。但是转第一个关节可能有好的结果，当转多个关节会出现旋转叠加，即当前关节在转，其他动过的关节也在转。分析结果，认为应该是操作过的关节始终是相对于之前的状态进行旋转一定角度的，而滑动模块应该表示的是以某一默认位置进行旋转的一个角度。

因此第一种解决方案是，记录原始的矩阵，即将joint中的transform再记录到bindWorldToJointTransform 中，旋转进行的矩阵操作为bindWorldToJointTransform矩阵乘以Transform矩阵，再将结果赋值给transform即可，那么每次都是相对于初始进行旋转，而不是再叠加。

另一种方式是参考同学的方法。使用Matrix4f自带的得到子矩阵的一种方法。将当前关节的矩阵的左上角的3\*3子矩阵变为三个叠加旋转得到的矩阵的同样位置的子矩阵。我个人理解的是，因为第四列的值除了w（最后一个值）前三个分别为三个坐标的位置。而那个3\*3子矩阵是表示的是当前坐标轴方向。此坐标轴是把当前节点看作一个原点坐标轴，此关节子树就是相对于此坐标系的。那么将关节坐标系旋转最终显示的是一部分的旋转。

