## 这篇文章是综合得讲了下目前的欧拉旋转

<https://glooow1024.github.io/2024/04/16/rotation/>

我们逐一说明，首先，欧拉旋转角表示的是一个过程，而不是一个状态，

A close up of text

Description automatically generated

这个过程中，我们最难以理解的其实是 内旋 和 外旋，可以参考这个文章了解什么是内外旋转

<https://zhuanlan.zhihu.com/p/85108850>

简而言之就是相对旋转的到底是局部坐标轴，还是全局坐标轴。

## 外旋和内旋

外旋转的公式推导，我们都知道，直接三个矩阵相乘就完事了，但是内乘却是反过来，这是为什么？

<https://dominicplein.medium.com/extrinsic-intrinsic-rotation-do-i-multiply-from-right-or-left-357c38c1abfd>

这篇文章讲的特别好， 其实和数学无关，就是一个概念性的问题，我们要回答一个问题，那就是，我们算旋转的时候究竟在算什么？用最容易理解的就是，我们想知道一个全局坐标点P，经过多次旋转后，P在全局坐标系下，他的坐标是多少？

内乘计算基于一个事实，我们每次内旋，本质上是在旋转一个坐标系，假设P点最开始在全局坐标系下是 (x, y, z)，那么无论我们怎么旋转这个全局坐标系，旋转多少次，最后，在旋转后的坐标系下，P点它的局部坐标还是那个值(x, y, z)，而这个时候，我们想要知道的，其实是 P点它现在相对全局坐标系的值是多少，很简单，往回推，P点相对于上一次旋转的坐标系的值是多少？链式推导即可（见下文）。

A diagram of a diagram of a rotation

Description automatically generated with medium confidence

最后一个公式就是，Origin是全局坐标系，A是Origin经过Rotation 1 后得到的坐标系，而 B 是 A经过 Rotation 2 后得到的坐标系，因此从右往左看，先是用P在B的局部坐标值，乘上 A\_R\_B 得到 P在A的局部坐标值，然后乘上 O\_R\_A 得到 P在O的局部坐标值（也就是全局坐标系）

这就是为什么

A screenshot of a computer

Description automatically generated

## 而关于万向锁，

比较简单直观的解释看这个地方

<https://www.zhihu.com/question/433208062/answer/3008859970>

A screenshot of a computer

Description automatically generated

就是欧拉角这一个过程中的三个参数，中因为第二个参数变为了某个值，导致第一和第三参数相对于全局坐标系，效果是一样，都不会产生绕全局Z轴旋转的可能，相当于丢了个自由度

## 四元数

通过这个回答，就能知道设计的用意，除了万向锁，除了插值，还有 歧义 这个原因才导致要用到四元数

<https://www.zhihu.com/question/47736315/answer/236808639>

## 在引擎中如何测试？

A screenshot of a computer

Description automatically generated

比如我们现在的需求是，让摄像机就固定某个距离，绕着Player走，你可以通过给 Player下挂一个cube当作Player局部坐标下的某个点，这样你旋转Player的时候，就可能看到它坐标系下的某个点的变化是怎样的。