3D坐标系中 点 的 平移、旋转和缩放

原创 白白皎皎 于 2018-08-11 20:10:44 发布 阅读量1.6w 收藏 63 点赞数 15 版权
分类专栏： 游戏编程

游戏编程 专栏收录该内容 1 订阅 5 篇文章 订阅专栏

基础概念

矩阵

一个m*n矩阵 是 一个m行、n列的矩形数组。

如果一个矩阵只包含单行 或 单列， 这样的矩阵 为 行矩阵或 列矩阵，又叫行向量或列向量

矩阵的乘法：

$$\mathbf{AB}=\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\end{bmatrix}\begin{bmatrix}b_{11}&b_{12}\\b_{21}&b_{22}\\b_{31}&b_{32}\end{bmatrix}=\begin{bmatrix}a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31}&a_{11}b_{12}+a_{12}b_{22}+a_{13}b_{32}\\a_{21}b_{11}+a_{22}b_{21}+a_{23}b_{31}&a_{21}b_{12}+a_{22}b_{22}+a_{23}b_{32}\end{bmatrix}=\mathbf{C}$$

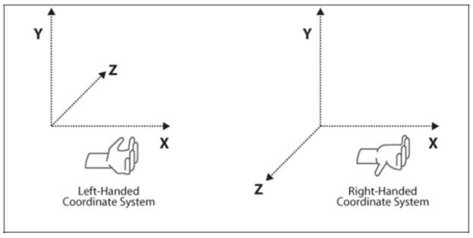
https://blog.csdn.net/qq_20828983

点

空间中的点，我们通常用一个 行向量表示： p = [x, y, z]

坐标系

用笛卡尔坐标系表示3D空间，我们按习惯可分为：左手坐标系 和 右手坐标系。如图：



左手坐标系Z轴正方向向里，右手坐标系Z轴正方向向外。在DirectX3D中我们使用左手坐标系。

平移

在3D空间中，我们经常需要将一个点 平移到另一个位置。假设空间中的一点P，其用行向量表示为$P=[x,\ y,\ z]$；将其向 x方向平移 dx，向y方向平移 dy，向z方向平移 dz后，得到的一点$P_1=[x+dx,\ y+dy,\ z+dz]$。

但是在数学计算或游戏编程中，我们都希望可以运用矩阵的乘法来表示点的平移。我们希望有一个公式：$\mathbf{P_1}=\mathbf{PT}$， 其中$\mathbf{T}$为一个矩阵

为了方便这种矩阵运行，我们引入齐次坐标，即将3D空间中的点P表示为$P=[x,\ y,\ z,\ w]$，其中w为齐次化坐标。 在游戏编程中，我们都用1表示w。 这样我们就能得到一个4*4 的平移矩阵：

$$\mathbf{P_1}=\mathbf{PT}=[x,\quad y,\quad z,\quad 1]\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\dx&dy&dz&1\end{bmatrix}=[x+dx,\quad y+dy,\quad z+dz,1]$$

其中：

$$\mathbf{T}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\dx&dy&dz&1\end{bmatrix}$$

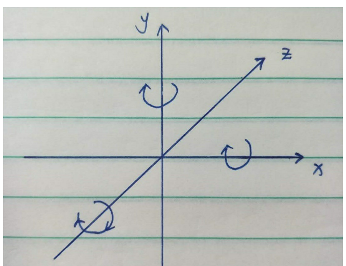
这样通过3D空间中的 任意一点 都可以通过 与平移矩阵向乘， 进行平移。

旋转

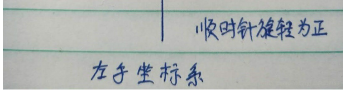
平移能够通过矩阵乘法来表示，同样，我们也希望3D空间中点的旋转也能用矩阵乘法来表示。

在理解旋转时，我们必须向记住几个高中学习的三角函数公式：

首先强调一个规则：在左手坐标系中，旋转角的正方向为顺时针方向；在右手坐标系中，旋转角的正方向为逆时针方向。

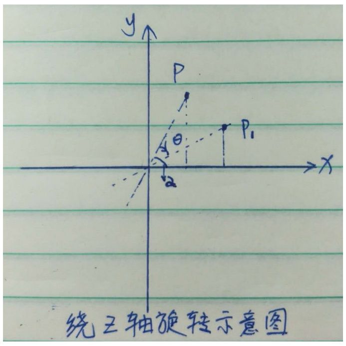


白白皎皎 码龄9年 暂无认证
29 原创 23万+ 周排名 110万+ 总排名 15万+ 访问 等级
1397 积分 73 粉丝 112 获赞 23 评论 454 收藏
私信 关注
DevChat AI编程助手哪家好？体验完 DevChat 就知道了！
发布首篇原创文章，原力分+10分，点亮新秀勋章 去发布
搜博主文章
热门文章
SQL语句多表联合查询 46481
R语言 -- car::scatterplotmatrix散点图矩阵参数详解 25764
3D坐标系中 点 的 平移、旋转和缩放 16376
RegisterRawInputDevices捕获设备消息 6549
MFC对话框里利用CHtmlView加载界面（一）———加载网页 6329
最新评论
MFC对话框里利用CHtmlView加载界面（…
赚够100个亿就收手: 如果web页面显示在主界面中的一个子窗口呢？也闪烁…
R语言 -- car::scatterplotmatrix散点图矩…
demo.ta:
sln、db、opendb、vcxproj、filters、use…
超自然祈祷: sln和suo这两个必要的二进制文件，在多人共同维护的时候非常容易冲…
sln、db、opendb、vcxproj、filters、use…
lql_csdn: user文件有配置环境变量设置，如果需要在工程中的DLL文件则需要配置…
R语言 -- car::scatterplotmatrix散点图矩…
医学和生信笔记: 为了老婆也是拼了
您愿意向朋友推荐“博客详情页”吗？
强烈不推荐 不推荐 一般般 推荐 强烈推荐
最新文章
Release、Debug版本 float 未在拷贝构造函数中初始化问题
R语言 -- car::scatterplotmatrix散点图矩阵 参数详解
DirectX3D--正交矩阵、旋转、平移、DrawPrimitive 综合dome
2019年 2篇 2018年 5篇
2017年 20篇 2016年 6篇
2015年 1篇
写文章 写代码 发动态 提问题 传资源 建项目
他们都在参与话题 围观看看
如何看待unity新的收费模式？
分类专栏
游戏编程 5篇
MFC 9篇
windows 8篇
c++基础 5篇
MFC 和 HTML 交互 5篇
网络编程 3篇
数据库 1篇
R
R语言 1篇



旋转我们可分为绕x轴、y轴、z轴旋转。

以绕z轴旋转为例：当绕z旋转时，其z坐标是不会发生变化的。x坐标和y坐标发生旋转。



由图可得，旋转后P_1的x和y坐标分别可表示为：

$$x_1 = r * \cos(\alpha - \theta) = r * \cos\alpha * \cos\theta + r * \sin\alpha * \sin\theta$$

$$y_1 = r * \sin(\alpha - \theta) = r * \sin\alpha * \cos\theta - r * \cos\alpha * \sin\theta$$

其中r为点到坐标原点的距离。很容易化简可得：

$$x_1 = x\cos\theta + y\sin\theta$$

$$y_1 = -x\sin\theta + y\cos\theta$$

按照矩阵乘法，我们$\mathbf{P_1 = PM_z}$

$$\mathbf{P_1 = PM_z} = [x, \quad y, \quad z, \quad 1]\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

、

其中绕Z轴旋转矩阵$\mathbf{M_z}$为：

$$\mathbf{M_z} = \begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

同样的方法，我们可以求得绕x轴 和 绕y轴 的旋转矩阵分别为：

$$\mathbf{M_x} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & -\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \mathbf{M_y} = \begin{bmatrix}\cos\theta & 0 & -\sin\theta & 0\\ 0 & 1 & 0 & 0\\ \sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

这样，我们就可以通过矩阵的乘法，实现点的任意旋转。

缩放

想要对相对于原点缩放某个点，只需要将x,y,z三个分量分别乘以对应的缩放因子：sx,sy,sz。

同样我们可以用矩阵乘法表示：

$$\mathbf{P_1 = PM_s} = [x, \quad y, \quad z, \quad 1]\begin{bmatrix}s_x & 0 & 0 & 0\\ 0 & s_y & 0 & 0\\ 0 & 0 & s_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix} = [x * s_x, \quad y * s_y, \quad z * s_z, 1]$$

其中缩放矩阵$\mathbf{M_s}$为：

$$\mathbf{M_s} = \begin{bmatrix}s_x & 0 & 0 & 0\\ 0 & s_y & 0 & 0\\ 0 & 0 & s_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

总结

有了以上平移、旋转、缩放矩阵后，我们就可以通过矩阵乘法求得点P任意变化后坐标：

$$\mathbf{P_1 = PTM_xM_yM_zM_s}$$

2 条评论 > zhuangjialo 热评 看看 写评论

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务
中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2023北京创新乐知网络技术有限公司

白白皎皎 关注 15 63 2 专栏目录

Beta Beta 举报