郭隆邦 2023-06-29

4. 矩阵乘法multiply

下面给大家介绍矩阵的**乘法运算**，查看文档，你可看到Three.js矩阵`Matrix4`类直接提供的矩阵乘法运算方法，比如`.multiply()`、`.multiplyMatrices()`、`.premultiply()`，这三个方法功能相同，只是语法细节有差异而已。

- `c.multiplyMatrices(a,b)`:axb，结果保存在c
- `a.multiply(b)`:axb，保存在a
- `a.premultiply(b)`:bxa，保存在a

回顾矩阵创建知识

先回顾下，上节课创建平移、旋转、缩放矩阵的知识点。

- 平移矩阵`.makeTranslation(Tx,Ty,Tz)`
- 缩放矩阵`.makeScale(Sx,Sy,Sz)`
- 绕x轴的旋转矩阵`.makeRotationX(angleX)`
- 绕y轴的旋转矩阵`.makeRotationY(angleY)`
- 绕z轴的旋转矩阵`.makeRotationZ(angleZ)`

```
// 空间中p点坐标
const p = new THREE.Vector3(50,0,0);

const T = new THREE.Matrix4();
T.makeTranslation(50,0,0);//平移矩阵
const R = new THREE.Matrix4();
R.makeRotationZ(Math.PI/2);//旋转矩阵
// p点矩阵变换
p.applyMatrix4(T);//先平移
p.applyMatrix4(R);//后旋转

mesh.position.copy(p);//用小球可视化p点位置
```

矩阵乘法`.multiply()`含义

关于平移矩阵、旋转矩阵和缩放矩阵乘法的含义，前面给大家讲解过，咱们这里再强调一遍。

比如两个矩阵，一个是平移矩阵`T`、一个是旋转矩阵`R`，两个矩阵相乘的结果，就表示旋转和平移的复合变换。

下面代码先把旋转矩阵和平移矩阵相乘，然后再对坐标进行变换，你可以看到结果上面代码相同。

```
const T = new THREE.Matrix4();
T.makeTranslation(50,0,0);//平移矩阵
const R = new THREE.Matrix4();
R.makeRotationZ(Math.PI/2);//旋转矩阵

// p点矩阵变换
// p.applyMatrix4(T);//先平移
// p.applyMatrix4(R);//后旋转

// 旋转矩阵和平移矩阵相乘得到一个复合模型矩阵
const modelMatrix = R.clone().multiply(T);
p.applyMatrix4(modelMatrix);
```

矩阵乘法顺序

矩阵乘法除特殊情况外，一般不满足交换律，`R.clone().multiply(T)`和`T.clone().multiply(R)`表示的结果不同，也就是`R * T`和`T * R`计算结果不同。

`R * T * p`表示p点先平移、后旋转

```
// R * T * p:先平移、后旋转
const modelMatrix = R.clone().multiply(T);
p.applyMatrix4(modelMatrix);
```

`T * R * p`表示p点先旋转、后平移

```
// T * R * p: 先旋转、后平移
const modelMatrix = R.clone().multiply(T);
p.applyMatrix4(modelMatrix);
```

哪个几何变换先发生，在矩阵乘法公式中，对应矩阵更靠近p点，或者说根据矩阵变换顺序，从右往左写。

← 3. Three.js矩阵Matrix4

5. 模型本地矩阵、世界矩阵→