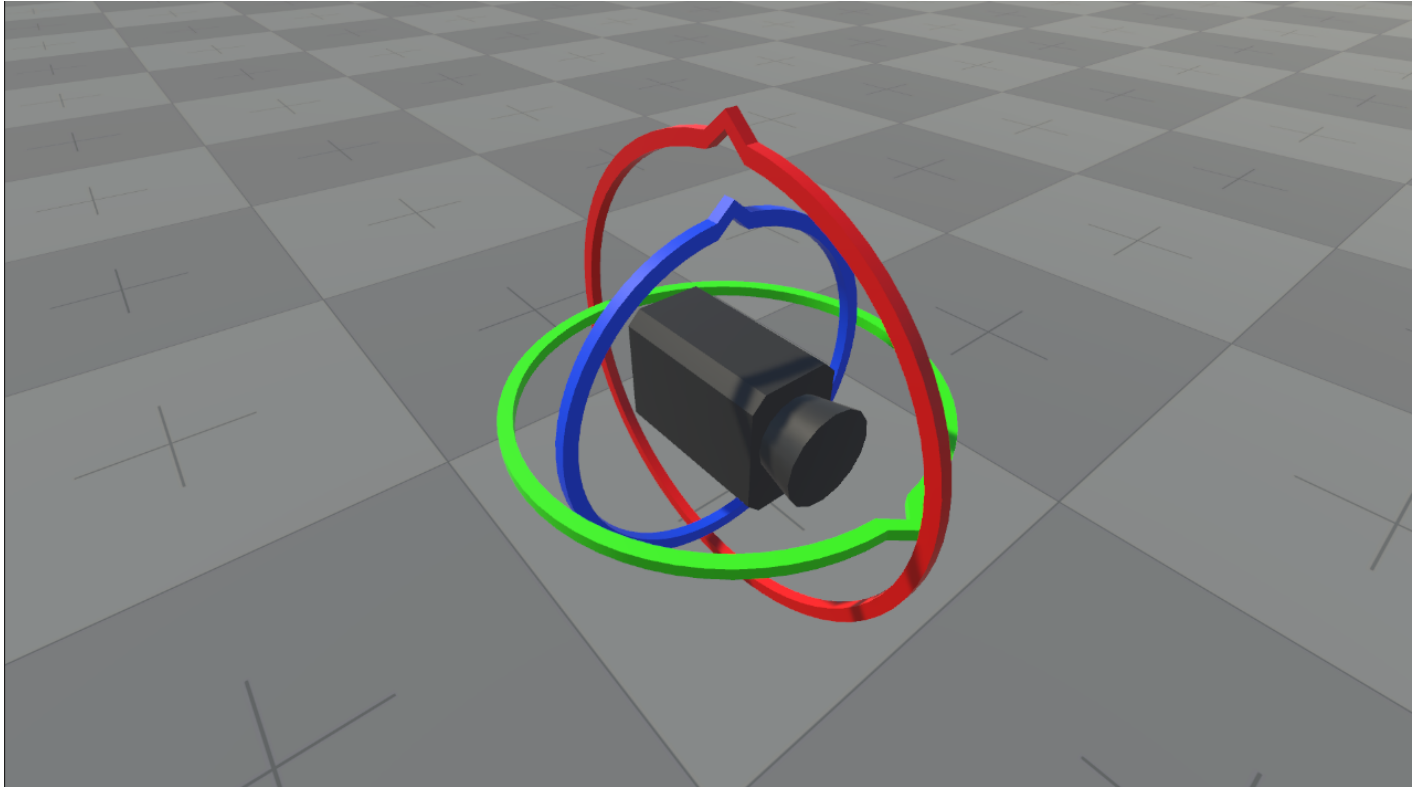


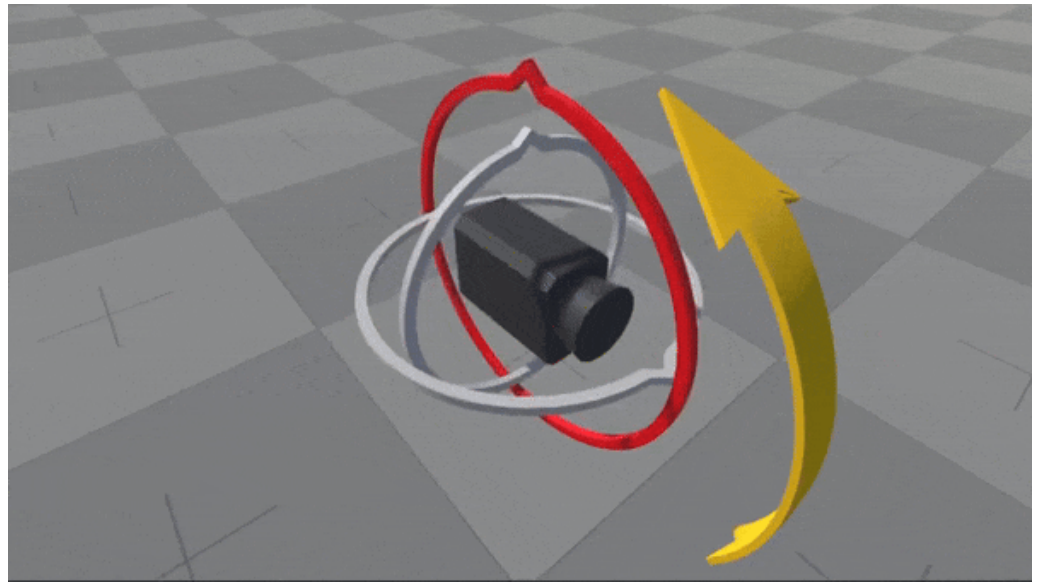
?????Euler????????????????

...

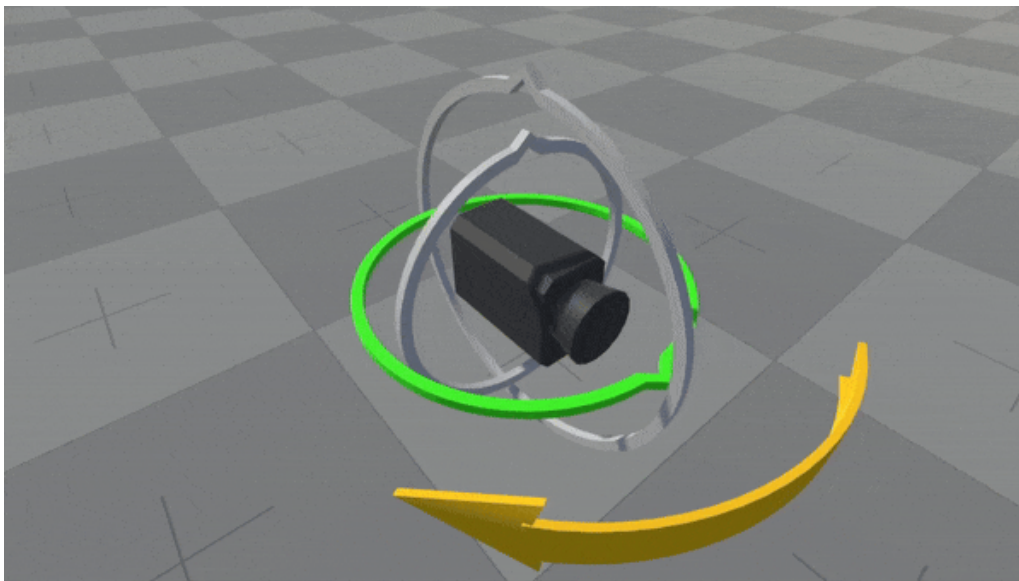


$\rightarrow Z$

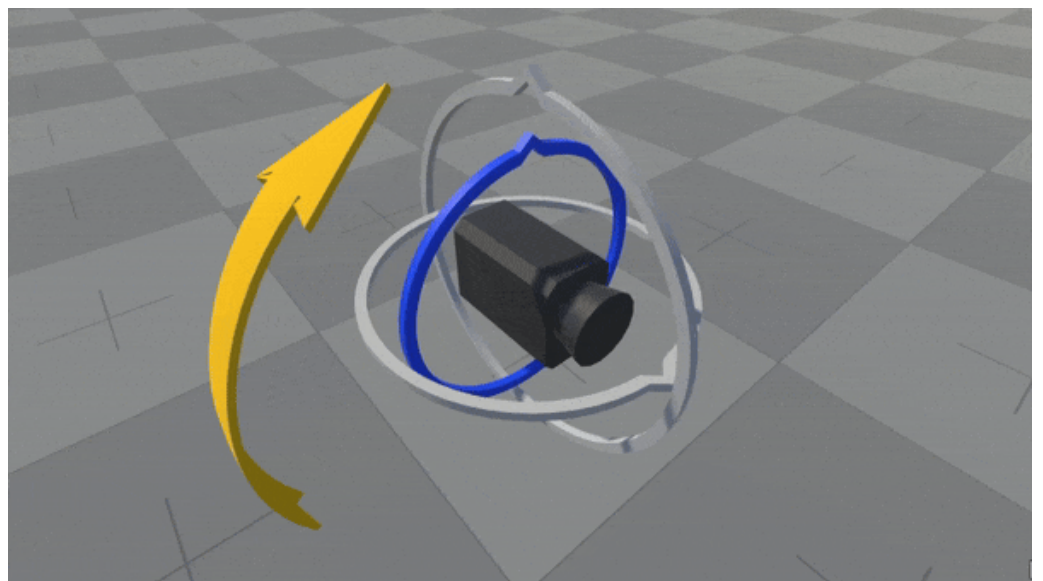
X



Y



Z



Y

90

-

??????Angle-Axis??-????

[illegible]



$$R_v(\theta) = \begin{bmatrix} tv_x^2 + c & tv_xv_y - v_zs & tv_xv_z - v_ys & 0 \\ tv_xv_y + v_zs & tv_y^2 + c & tv_yv_z + v_xs & 0 \\ tv_xv_z + v_ys & tv_yv_z - v_xs & tv_z^2 + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

--	--	--	--

$$R_v(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Unity Z

Unity

Unity Quaternion.AngleAxis

```
// X 45
Vector3 axis = new Vector3(1, 0, 0); //
float angle = 45; //

transform.localRotation = Quaternion.AngleAxis(angle, axis);
```

??

- Moon
 -
 -
 -
- Inspector
- Quaternion.AngleAxis (transform.up)
- Scene

??????

Quaternion -

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

+

$$q = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$



- 
- 
- 
-  ↔ 



```
// UnityQuaternion
Quaternion rot = transform.rotation;
```

??????? Quaternion ???

UnityXYZQuaternion

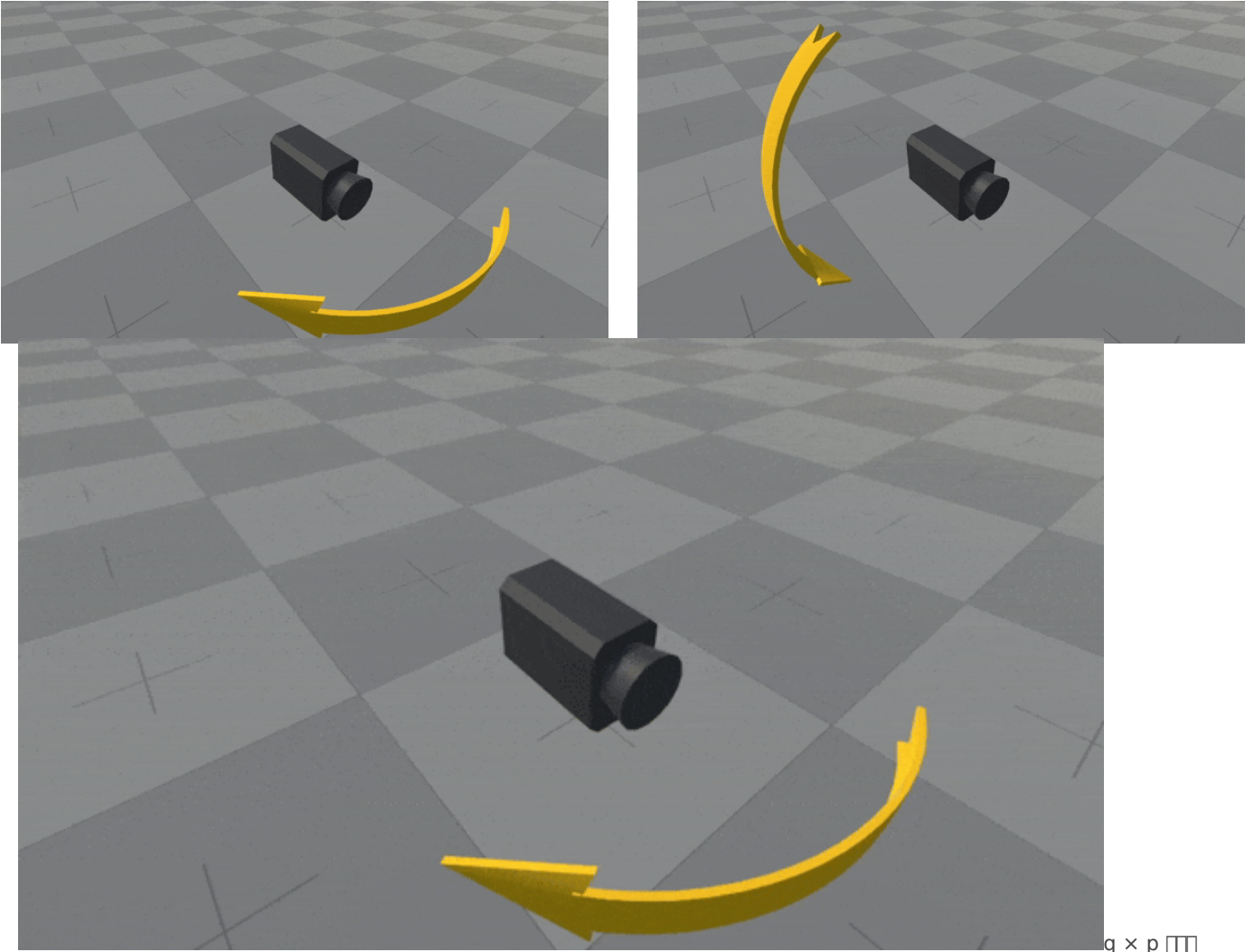
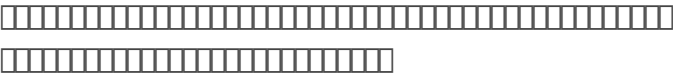
```
// Quaternion
Quaternion rotA = transform.rotation;

// 
Vector3 angles = rot.eulerAngles;

// Quaternion
Quaternion rotB = Quaternion.Euler(x, y, z); // x, y, z

// 
transform.rotation = rotB;
```

???????



$q \times p$

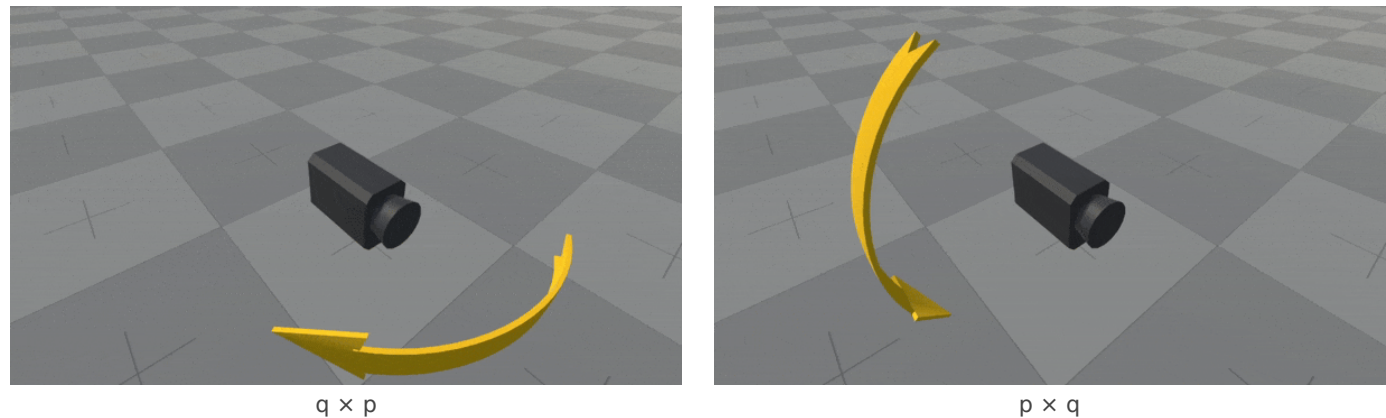
Unity Quaternion

Quaternion *

```
// Quaternion qp = q * p;
```

Progress bar (30 empty boxes)

$p \times q \neq q \times p$



????????

Progress bar (30 empty boxes)
Progress bar (5 empty boxes)

$$R(q) = \begin{bmatrix} 2(a^2 + b^2) - 1 & 2(bc - ad) & 2(bd + ac) & 0 \\ 2(bc + ad) & 2(a^2 + c^2) - 1 & 2(cd - ab) & 0 \\ 2(bd - ac) & 2(cd + ab) & 2(a^2 + d^2) - 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

☐ a, b, c, d