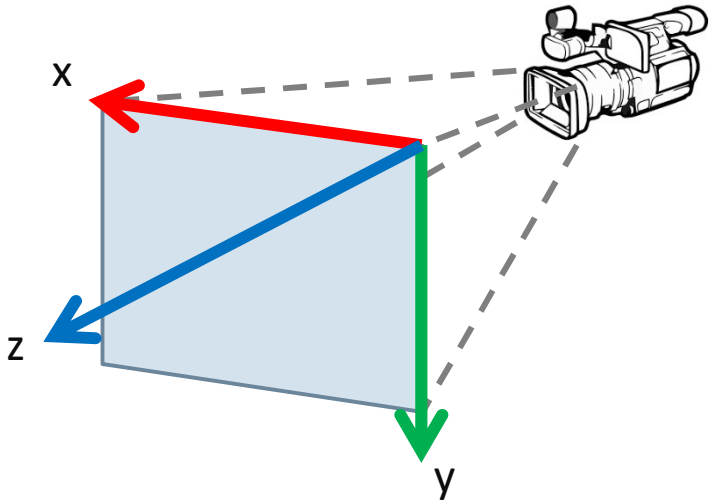
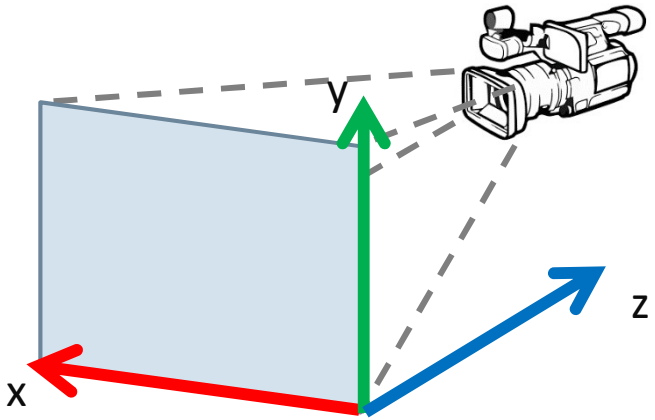
Coordinate systems

- JSARToolKit cameras use a right-handed, Y-down system
- Three.js cameras use a right-handed, Y-up system

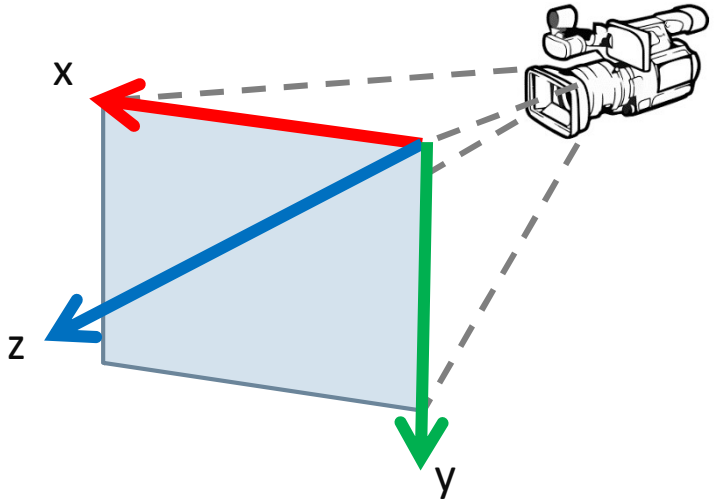


JSARToolKit cameras



Standard Three.js cameras

Coordinate systems



Our camera, after changing the camera matrix

Screen coordinates computed by our new camera matrix

Screen coordinates expected by the three.js renderer

- By changing the camera matrix, we forced three.js to use the JSARToolKit system
- However, the **renderer** still expects to draw 2D images where the Y grows upwards, and this causes the Y-flipping problem

Fixing the camera matrix

- Solution: flip the Y coordinates directly in the camera matrix by multiplying them by -1
- Flipping-Y homogeneous matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$