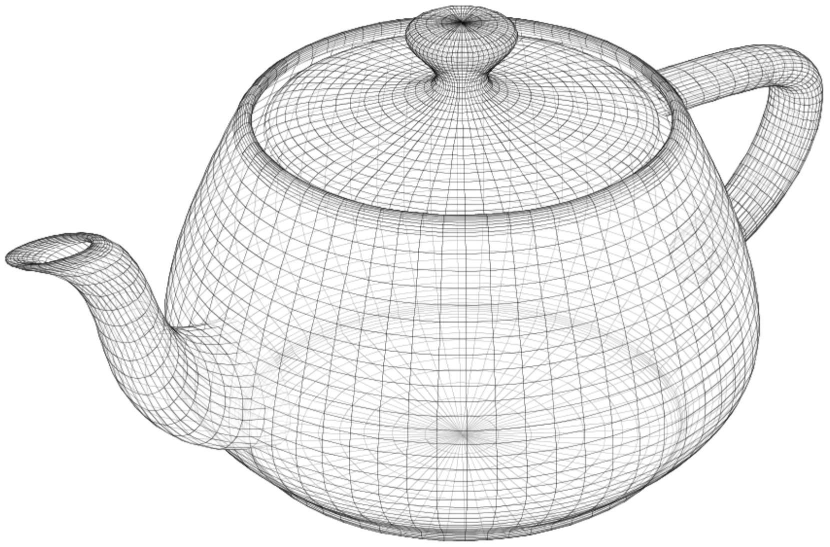
Mathematics for Orientation

Euler Angles



CS 418: Interactive Computer Graphics

Professor Eric Shaffer

If you do not change direction,
you may end up where you are heading.

— Lao Tzu (600–531 BCE)

ILLINOIS

Orientation

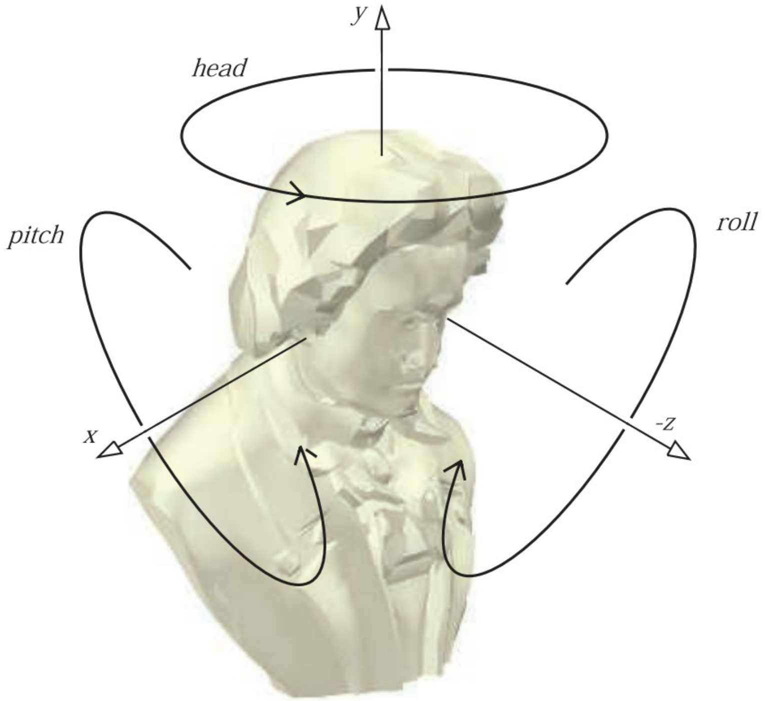
We can position a model in our 3D virtual world easily

- Just 3 values to translate it to desired position

What about orientation?

How will we specify that for a model?

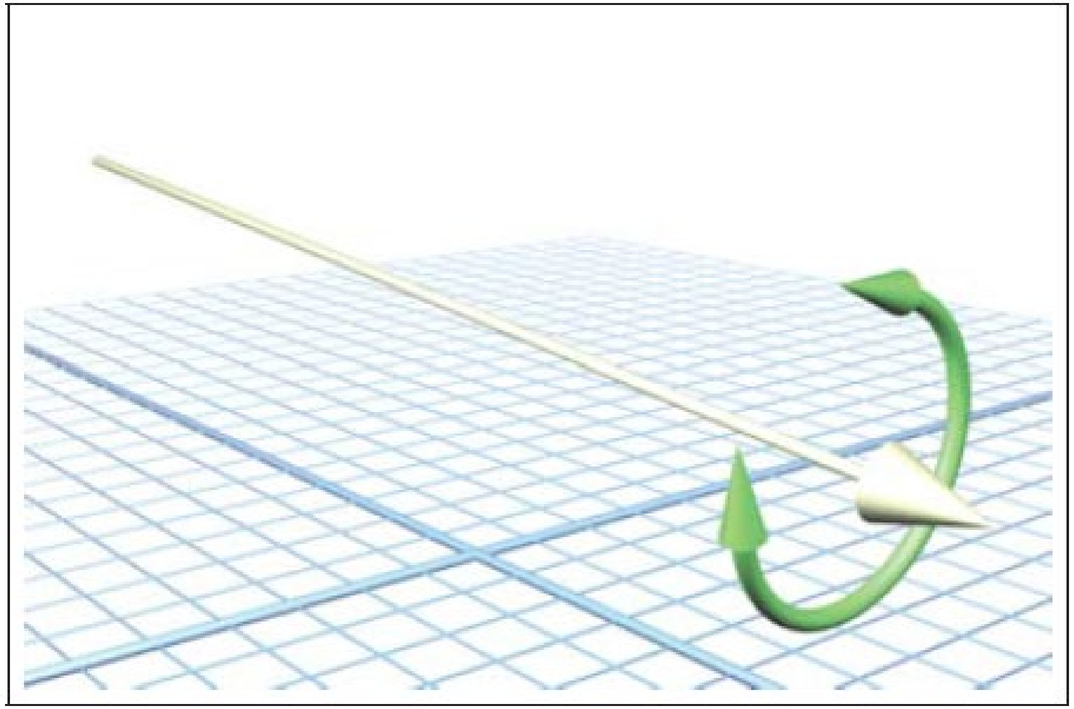
Or for a camera?

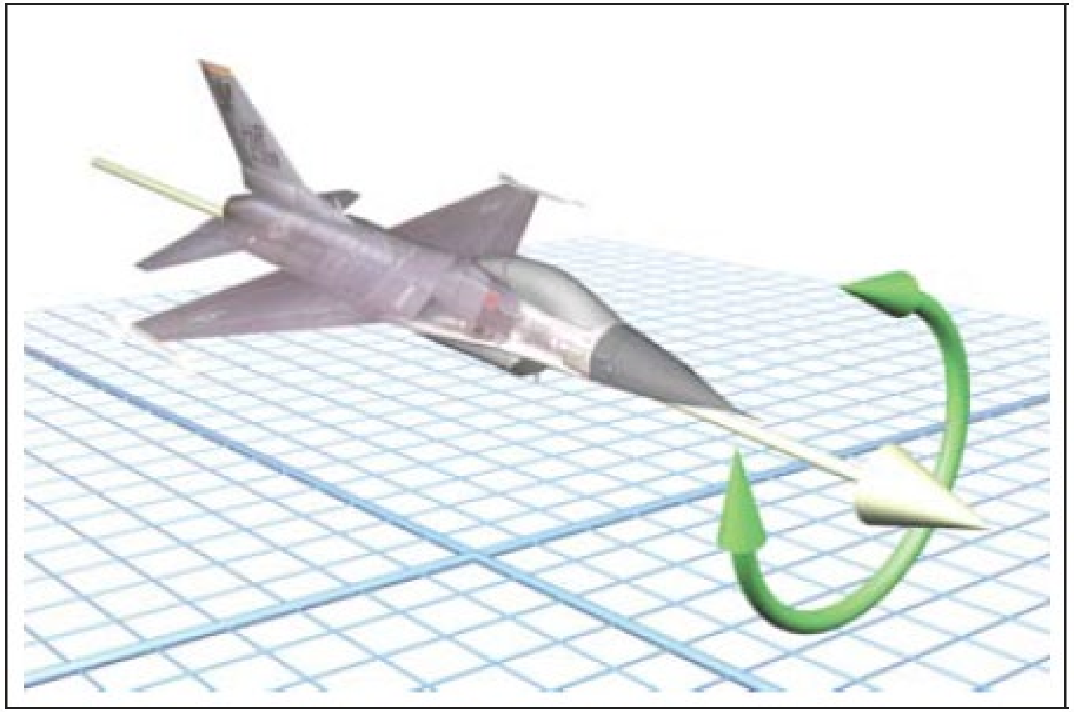


Orientation is not Direction

If you do not change direction,
you may end up where you are heading.

— Lao Tzu (600–531 BCE)





- Rotating a vector around itself does not change it
- Rotating an object around its principal direction changes its orientation

Representing Orientations

In 3D…how much information is needed to represent a direction?

How about an orientation?

There are several popular options to encode orientation:

- Euler angles
- Rotation vectors (axis/angle)
- 3x3 matrices
- Quaternions

Euler Angles

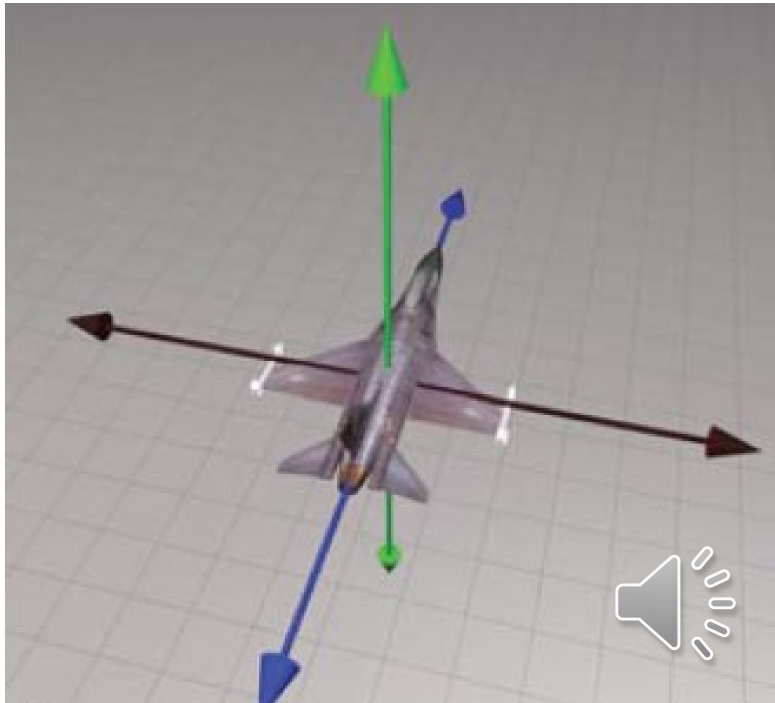
We can represent an orientation with 3 numbers

- A sequence of rotations around principal axes is called an Euler Angle Sequence

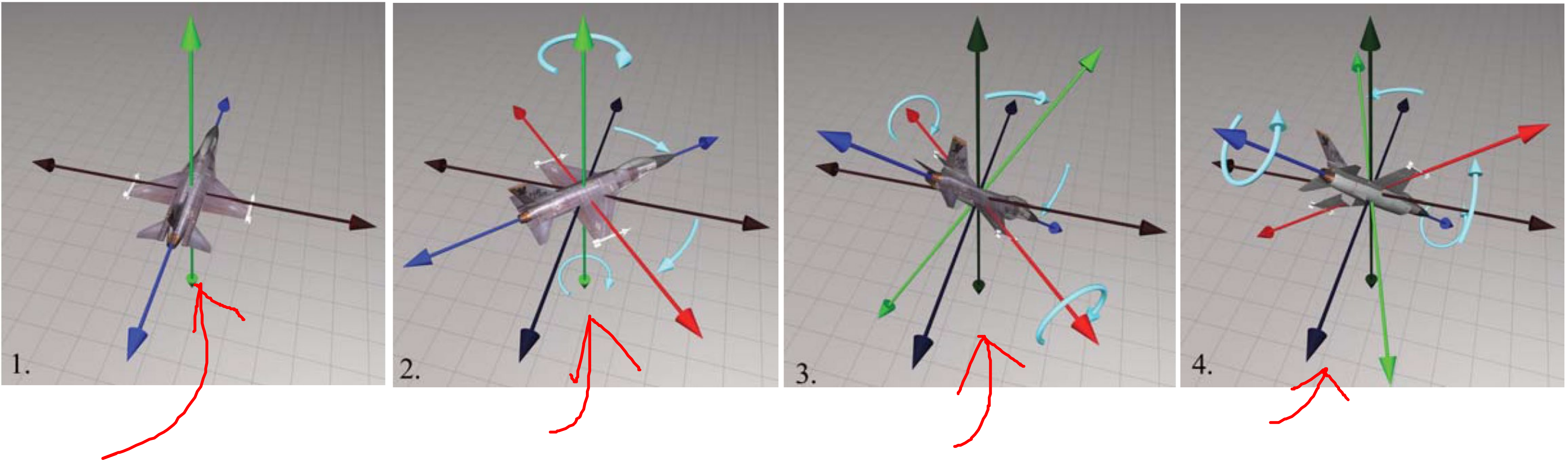
Assume we limit ourselves to 3 rotations

- no successive rotations about the same axis
- we could use any of the following 12 sequences to specify an orientation

XYZ	XZY	XYX	XZX
YXZ	YZX	YXY	YZY
ZXY	ZYX	ZXZ	ZYZ



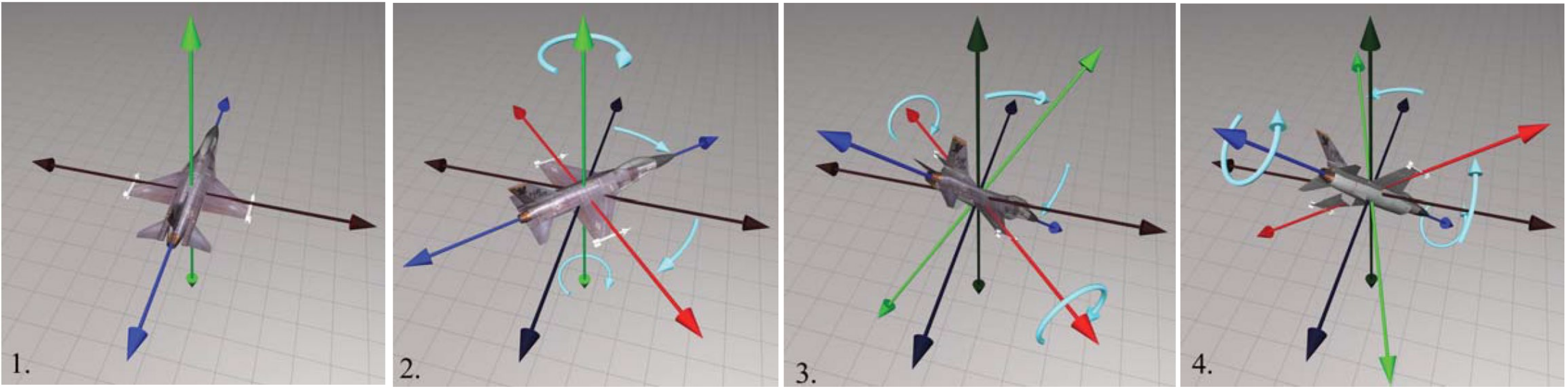
Applying an Euler Angle Sequence



Which axes are being rotated around in this sequence?

ILLINOIS

Applying an Euler Angle Sequence



Which axes are being rotated around in this sequence?

$M = R_z R_x R_y$

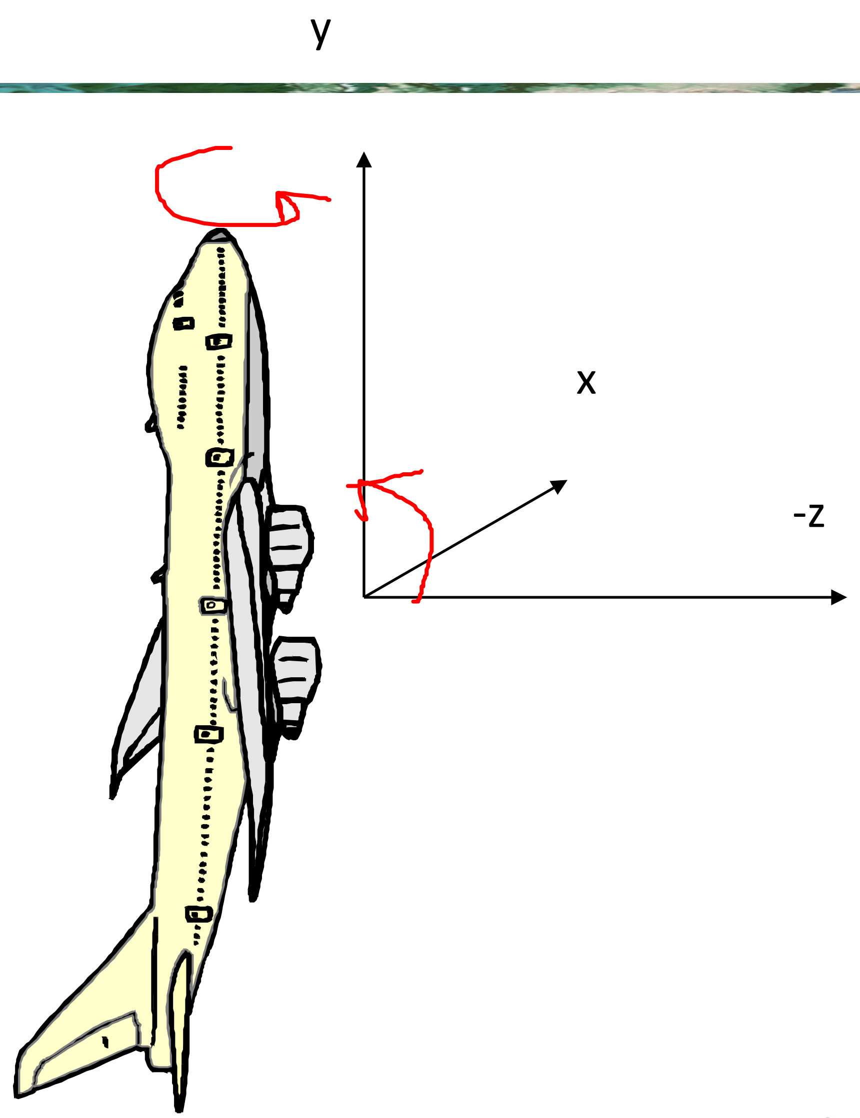
This is a very commonly used order…

Gimbal Lock

- Airplane orientation
  - Rz(roll) Rx(pitch) Ry(yaw)
- Two axes have collapsed onto each other
- Think about this from a user-interface perspective

Imagine you have 3 dials…one for each angle

What action caused the orientation you see?
What happens when z dial is moved now?



What problem could this cause for someone playing a game with this interface?

ILLINOIS

Thinking about Gimbal Lock

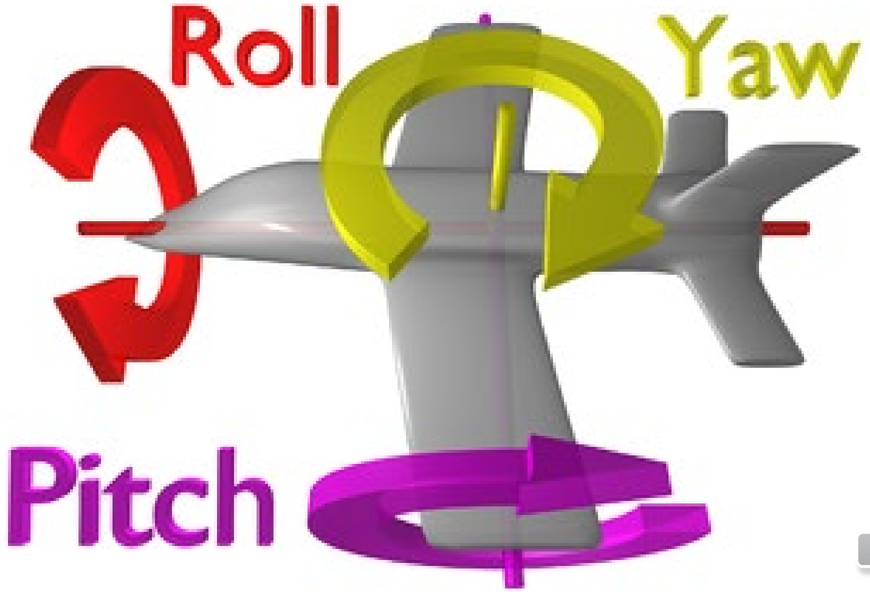
$M=R_zR_xR_y$

+- 90

In what order are the transformations applied to the points?

Which axis and angle(s) can cause Gimbal Lock?

Y? X? Z?



Experiment

http://danceswithcode.net/engineeringnotes/rotations_in_3d/demo3D/rotations_in_3d_tool.html

Euler Angle Visualization Tool

Euler Angles (Deg)

Yaw 0

Pitch 90

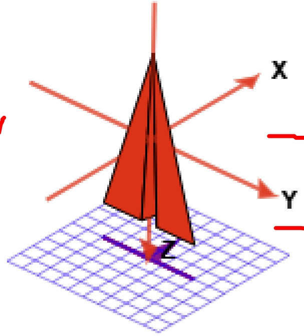
Roll 0

Convert

Rotation Matrix

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

Z



This tool converts Tait-Bryan Euler angles to a rotation matrix, and then rotates the airplane graphic accordingly. The Euler angles are implemented according to the following convention (see the main paper for a detailed explanation):

- Rotation order is yaw, pitch, roll, around the z, y and x axes respectively
- Intrinsic, active rotations
- Right-handed coordinate system with right-handed rotations

Gimbal lock occurs when the pitch angle is +90° or -90°. Under these conditions, the yaw and roll axis become aligned and have the same effect.

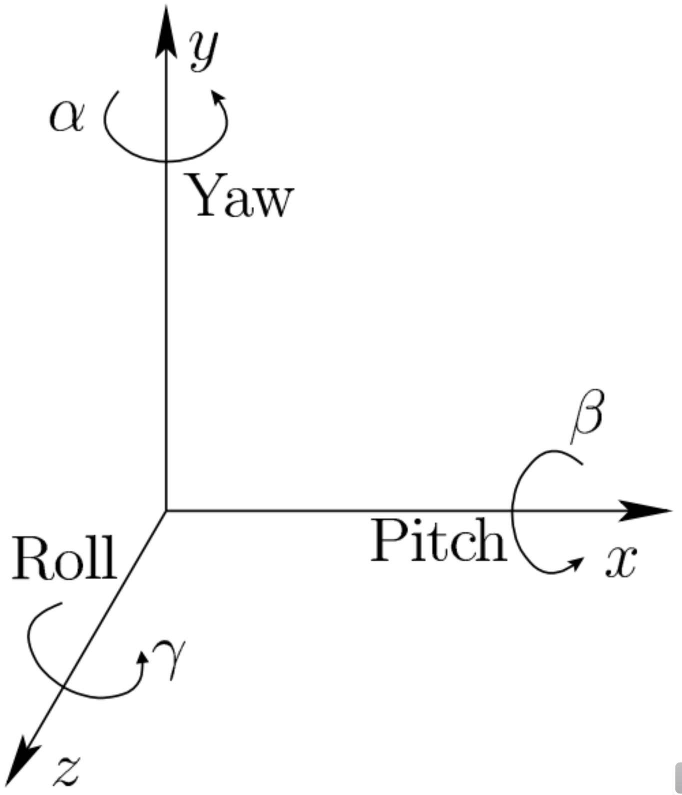
Is Gimbal Lock that Important?

"In the video game industry there has been some back-and-forth battles about whether this problem is crucial. In an FPS game, the avatar is usually not allowed to pitch his head all the way to $\pm\pi/2$, thereby avoiding this problem. In VR, it happens all the time that a user could pitch her head straight up or down. The kinematic singularity often causes the viewpoint to spin uncontrollably…"

-- *Virtual Reality* by Lavalle Section 3.3

Euler Angles

- We will define
  - Roll
    - rotation about z
  - Pitch
    - rotation about x
  - Yaw
    - rotation about y
- Orientation
  - Rz(roll) Rx(pitch) Ry(yaw)



Euler Angles to Matrix Conversion

To build a matrix from a set of Euler angles...
...just multiply a sequence of rotation matrices together:

$$\mathbf{R}_x \cdot \mathbf{R}_y \cdot \mathbf{R}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \cdot \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \cdot \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ s_x s_y c_z + c_x s_z & -s_x s_y s_z + c_x c_z & -s_x c_y \\ -c_x s_y c_z + s_x s_z & c_x s_y s_z + s_x c_z & c_x c_y \end{bmatrix}$$

Why would we care about being able to convert to a matrix ?

Euler Angles…Good and Bad

- Euler angles can generate any possible orientation in 3D (Good!)
- Euler angles are used in a lot of applications…they are believed to be ~~intuitive~~ (Good?)
- They are compact…requiring only 3 numbers (Good!)
- Ambiguous: different triples can be same orientation (Bad?)
- They do not interpolate in a obvious way (Bad!)
- They can suffer from Gimbal lock (Bad! But not as bad as it sounds)
- Conversion to/from a matrix requires several trig operations (Bad!)

ILLINOIS