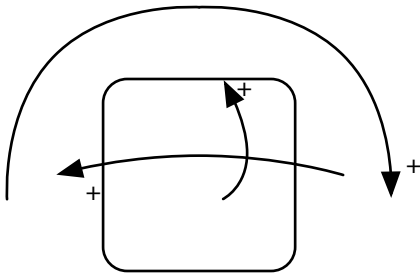


Unity



Transform Rotation:

X = Pitch (Top-to-front Positive // Nose down Positive)

Y = Yaw (Clockwise Positive)

Z = Roll (Top-to-left Positive)

euler.x = Yaw

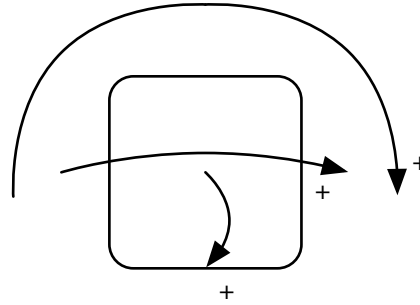
euler.y = Pitch

euler.z = Roll

currently: eulerAngles.y += 180;

SoundAlgorithm(eulerAngles.x, eulerAngles.y, eulerAngles.z)

Unreal Engine



Transform Rotation:

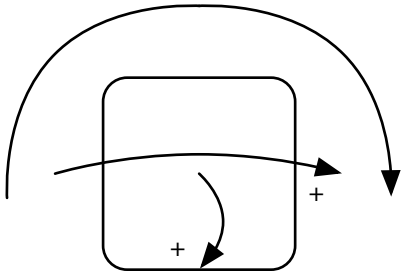
X = Roll (Top-to-right Positive)

Y = Pitch (Top-to-rear Positive // Nose up Positive)

Z = Yaw (Clockwise Positive)

SoundAlgorithm(quat.Euler().Y, (quat.Euler().Z < 0 ?
360 + quat.Euler().Z : quat.Euler().Z), quat.Euler().X)

OpenFrameworks



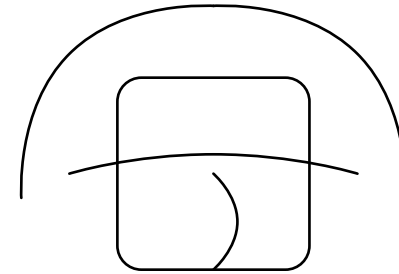
X = Pitch (Top-to-rear Positive // Nose up Positive)

Y = Yaw (Clockwise Positive)

Z = Roll (Top-to-right Positive)

currently: audioMixAlgorithm(angleY,
-angleX, angleZ);

Google Cardboard (Android)



matrix R = Rz(roll) * Rx(pitch) * Ry(yaw)

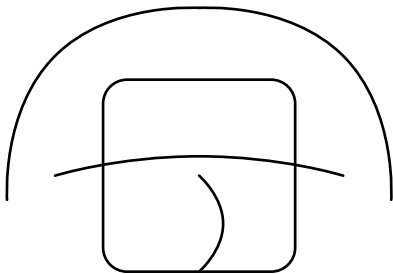
Pitch (X axis): $[-\pi/2, \pi/2]$

Yaw (Y axis): $[-\pi, \pi]$

Roll (Z axis): $[-\pi, \pi]$

X-Y-Z axes are the basis of a right-handed OpenGL-style coordinate system. During Gimbal lock this method enforces yaw to 0 and provides a valid roll angle.

CMDeviceMotion (iOS)



X = Pitch

Y = Roll

Z = Yaw

Note that there are 6 possible Euler decompositions and this y-x-z decomposition is different from the x-y-z decomposition used by some other systems.