←Back

# OpenGL Angles to Axes

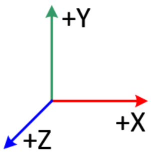


The rotation angles directly affects the first 3 columns of OpenGL **GL_MODELVIEW** matrix, precisely *left*, *up* and *forward* axis elements. For example, if a unit vector along X axis, (1, 0, 0) is multiplied by an arbitrary 3x3 rotation matrix, then the result of the vector after rotation is;

$$\begin{pmatrix} m_0 & m_3 & m_6 \\ m_1 & m_4 & m_7 \\ m_2 & m_5 & m_8 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} m_0 \\ m_1 \\ m_2 \end{pmatrix}$$

It means the first column ($m_0$, $m_1$, $m_2$) of the rotation matrix represents the coordinates of the *left* axis after rotated. Similarly, the second column is the *up* axis and the third column is the *forward* axis.

This article describes how to contruct GL_MODELVIEW matrix with rotation angles.



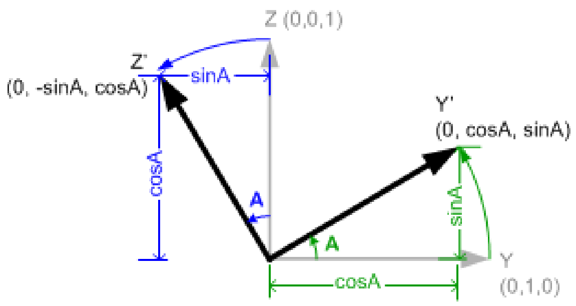

$$\begin{array}{cccc} \text{Left (X) Axis} & \text{Up (Y) Axis} & \text{Forward (Z) Axis} & \text{Translation} \end{array}$$
$$\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$$

4 Columns of GL_MODELVIEW matrix

## Axis Rotations

First, we look at a rotation around each axis; +X, +Y and +Z. We project three axes onto a plane in 3 different ways, so the axis that we want to rotate is facing toward you. The positive rotation direction becomes counter clockwise (right hand rule).

### Rotation about Left(X) Axis (Pitch)

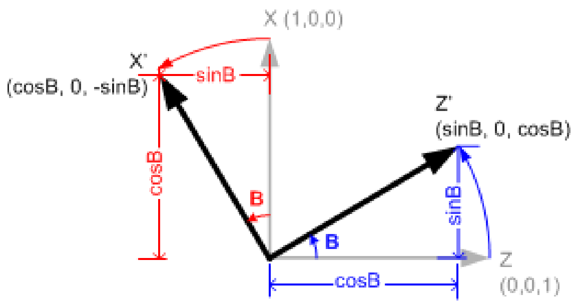



Rotation about Left(X) Axis

Initial value of up(Y) and forward(Z) axes are (0, 1, 0) and (0, 0, 1). If left(X) axis rotates A degree, then new up(Y') axis becomes (0, cosA, sinA) and forward(Z') becomes (0, -sinA, cosA). The new axes are inserted as column components of the 3x3 rotation matrix. Then, the rotation matrix becomes;

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

### Rotation about Up(Y) Axis (Yaw, Heading)

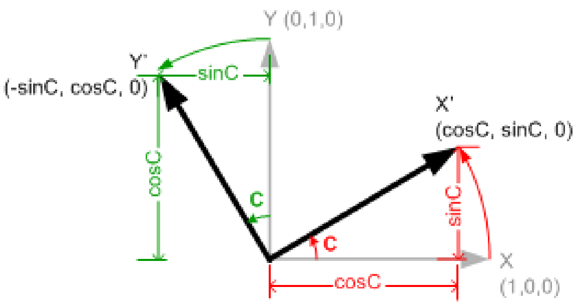



Rotation about Up(Y) Axis

Now, we rotate around up vector, which is facing toward you, with B angle . Left(X) axis is transformed from (1, 0, 0) to X' (cosB, 0, -sinB). And forward(Z) axis is from (0, 0, 1) to Z' (sinB, 0, cosB).

$$\begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

### Rotation about Forward(Z) Axis (Roll)




Rotation about Forward(Z) Axis

If we rotate forward(Z) axis with angle C degree, the original left(X) (1, 0, 0) axis becomes X' (cosC, sinC, 0), and up(Y) (0, 1, 0) axis becomes Y' (-sinC, cosC, 0).

$$\begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## Angles To Axes

We can combine these separate axis rotations into one matrix by multiplying above 3 matrices together. Note that multiplication of matrices is not commutative, so a different order of matrix multiplication results in a different outcome. There are 6 different combinations are possible; $R_xR_yR_z$, $R_xR_zR_y$, $R_yR_xR_z$, $R_yR_zR_x$, $R_zR_xR_y$ and $R_zR_yR_x$.

The left column of the combined rotation matrix is the *left* axis after rotated, the middle column is the *up* axis, and the right column is the *forward* axis.

---

$$R_X R_Y R_Z$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos B & 0 & \sin B \\ \sin A \sin B & \cos A & -\sin A \cos B \\ -\cos A \sin B & \sin A & \cos A \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos B \cos C & -\cos B \sin C & \sin B \\ \sin A \sin B \cos C + \cos A \sin C & -\sin A \sin B \sin C + \cos A \cos C & -\sin A \cos B \\ -\cos A \sin B \cos C + \sin A \sin C & \cos A \sin B \sin C + \sin A \cos C & \cos A \cos B \end{pmatrix}$$

---

$$R_X R_Z R_Y$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

$$= \begin{pmatrix} \cos C & -\sin C & 0 \\ \cos A \sin C & \cos A \cos C & -\sin A \\ \sin A \sin C & \sin A \cos C & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

$$= \begin{pmatrix} \cos C \cos B & -\sin C & \cos C \sin B \\ \cos A \sin C \cos B + \sin A \sin B & \cos A \cos C & \cos A \sin C \sin B - \sin A \cos B \\ \sin A \sin C \cos B - \cos A \sin B & \sin A \cos C & \sin A \sin C \sin B + \cos A \cos B \end{pmatrix}$$

---

$$R_Y R_X R_Z$$

$$= \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos B & \sin B \sin A & \sin B \cos A \\ 0 & \cos A & -\sin A \\ -\sin B & \cos B \sin A & \cos B \cos A \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos B \cos C + \sin B \sin A \sin C & -\cos B \sin C + \sin B \sin A \cos C & \sin B \cos A \\ \cos A \sin C & \cos A \cos C & -\sin A \\ -\sin B \cos C + \cos B \sin A \sin C & \sin B \sin C + \cos B \sin A \cos C & \cos B \cos A \end{pmatrix}$$

---

$$R_Y R_Z R_X$$

$$= \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

$$= \begin{pmatrix} \cos B\cos C & -\cos B\sin C & \sin B \\ \sin C & \cos C & 0 \\ -\sin B\cos C & \sin B\sin C & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

$$= \begin{pmatrix} \cos B\cos C & -\cos B\sin C\cos A + \sin B\sin A & \cos B\sin C\sin A + \sin B\cos A \\ \sin C & \cos C\cos A & -\cos C\sin A \\ -\sin B\cos C & \sin B\sin C\cos A + \cos B\sin A & -\sin B\sin C\sin A + \cos B\cos A \end{pmatrix}$$

---

$$R_Z R_X R_Y$$

$$= \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

$$= \begin{pmatrix} \cos C & -\sin C\cos A & \sin C\sin A \\ \sin C & \cos C\cos A & -\cos C\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix}$$

$$= \begin{pmatrix} \cos C\cos B - \sin C\sin A\sin B & -\sin C\cos A & \cos C\sin B + \sin C\sin A\cos B \\ \sin C\cos B + \cos C\sin A\sin B & \cos C\cos A & \sin C\sin B - \cos C\sin A\cos B \\ -\cos A\sin B & \sin A & \cos A\cos B \end{pmatrix}$$

---

$$R_Z R_Y R_X$$

$$= \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

$$= \begin{pmatrix} \cos C\cos B & -\sin C & \cos C\sin B \\ \sin C\cos B & \cos C & \sin C\sin B \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix}$$

$$= \begin{pmatrix} \cos C\cos B & -\sin C\cos A + \cos C\sin B\sin A & \sin C\sin A + \cos C\sin B\cos A \\ \sin C\cos B & \cos C\cos A + \sin C\sin B\sin A & -\cos C\sin A + \sin C\sin B\cos A \\ -\sin B & \cos B\sin A & \cos B\cos A \end{pmatrix}$$

Here is a C++ example code for $R_xR_yR_z$ combination. It performs 3 rotations in order of $R_z$ (roll), $R_y$ (yaw) then $R_x$ (pitch). The results of *left*, *up* and *forward* axis can be used to construct GL_MODELVIEW matrix.

```
struct Vector3
{
    float x;
    float y;
    float z;
    Vector3() : x(0), y(0), z(0) {}; // initialze when created
};


///////////////////////////////////////////////////////////////////////////////
// convert Euler angles(x,y,z) to axes(left, up, forward)
// Each column of the rotation matrix represents left, up and forward axis.
// The order of rotation is Roll->Yaw->Pitch (Rx*Ry*Rz)
// Rx: rotation about X-axis, pitch
// Ry: rotation about Y-axis, yaw(heading)
// Rz: rotation about Z-axis, roll
```

```
//     Rx            Ry           Rz
// |1  0   0| | Cy  0 Sy| |Cz -Sz 0|   | CyCz        -CySz         Sy  |
// |0 Cx -Sx|*|  0  1  0|*|Sz  Cz 0| = | SxSyCz+CxSz -SxSySz+CxCz -SxCy|
// |0 Sx  Cx| |-Sy  0 Cy| | 0   0 1|   |-CxSyCz+SxSz  CxSySz+SxCz  CxCy|
///////////////////////////////////////////////////////////////////////////////
void anglesToAxes(const Vector3 angles, Vector3& left, Vector3& up, Vector3& forward)
{
    const float DEG2RAD = 3.141593f / 180;
    float sx, sy, sz, cx, cy, cz, theta;

    // rotation angle about X-axis (pitch)
    theta = angles.x * DEG2RAD;
    sx = sinf(theta);
    cx = cosf(theta);

    // rotation angle about Y-axis (yaw)
    theta = angles.y * DEG2RAD;
    sy = sinf(theta);
    cy = cosf(theta);

    // rotation angle about Z-axis (roll)
    theta = angles.z * DEG2RAD;
    sz = sinf(theta);
    cz = cosf(theta);

    // determine left axis
    left.x = cy*cz;
    left.y = sx*sy*cz + cx*sz;
    left.z = -cx*sy*cz + sx*sz;

    // determine up axis
    up.x = -cy*sz;
    up.y = -sx*sy*sz + cx*cz;
    up.z = cx*sy*sz + sx*cz;

    // determine forward axis
    forward.x = sy;
    forward.y = -sx*cy;
    forward.z = cx*cy;
}
```



←Back