Transformations, Viewing and Projection

HIMANSHU RAJ, Indraprastha Institute of Information Technology Delhi, India

1 TRANSFORMATIONS

To rotate anti-clockwise about an arbitrary vector a (1, 2, 2) by 30 degrees:

a) Form an orthonormal uvw basis with w = a

$$w = \frac{a}{||a||} = \frac{1}{3}i + \frac{2}{3}j + \frac{2}{3}k$$

$$t = 1i + \frac{2}{3}j + \frac{2}{3}k$$

$$u = \frac{t \times w}{||t \times w||} = 0i + \frac{-1}{\sqrt{2}}j + \frac{1}{\sqrt{2}}k$$

$$v = w \times u = \frac{2\sqrt{2}}{3}i + \frac{-1}{3\sqrt{2}}j + \frac{-1}{3\sqrt{2}}k$$

b) Rotate uvw basis to canonical basis xyz

$$R = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{2\sqrt{2}}{3} & \frac{-1}{3\sqrt{2}} & \frac{-1}{3\sqrt{2}} \\ \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \end{bmatrix}$$

c) Rotate about the z-axis by ϕ

$$R_z = \begin{bmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{-1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

d) Rotate canonical basis xyz back to uvw basis

$$R^T = \begin{bmatrix} u & v & w \end{bmatrix} = \begin{bmatrix} 0 & \frac{2\sqrt{2}}{3} & \frac{1}{3} \\ \frac{-1}{\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{2}{3} \\ \frac{1}{\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{2}{3} \end{bmatrix}$$

Author's address: Himanshu Raj, himanshu22216@iiitd.ac.in, Indraprastha Institute of Information Technology Delhi, India.

The final matrix product looks like

$$R_a(\phi) = R^T R_z R = \begin{bmatrix} 0 & \frac{2\sqrt{2}}{3} & \frac{1}{3} \\ \frac{-1}{\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{2}{3} \\ \frac{1}{\sqrt{2}} & \frac{-1}{3\sqrt{2}} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{-1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{2\sqrt{2}}{3} & \frac{-1}{3\sqrt{2}} & \frac{-1}{3\sqrt{2}} \\ \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \end{bmatrix}$$

The anti-clockwise 30 degrees rotation about (1,2,2) looks like

(a) Before transformation

(b) After transformation

2 VIEWING

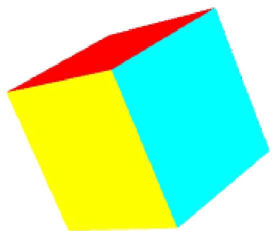
The routine customLookAt() is the scratch implementation of the GLM lookAt() function.
The custom routine takes the eye vector, centre vector and up vector as parameters, the same as the GLM lookAt() function.
We calculate (e,g,t) triplet to construct the viewing matrix. e is the same as the eye vector, and t is the up vector from parameters, and we find g (gaze vector) by (centre - eye).
Then, we find the uvw basis

$$w = \frac{-g}{||g||}, \quad u = \frac{t \times w}{||t \times w||}, \quad v = w \times u$$

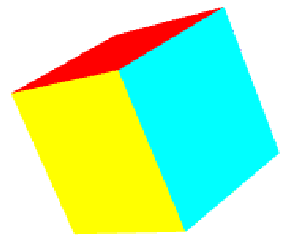
The final viewing matrix is given by

$$\begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rotated cube with eye vector (50, 100, 20), centre vector (0, 0, 0) and up vector (0, 0, 1) looks like



(c) with customLookAt()



(d) with glm::lookAt()

We can see that both functions produce the same output. This indicates that the scratch implementation of customLookAt() is correct and is working as expected.

3 PROJECTION

We generate different perspectives using appropriate viewing transformations on the unoriented cube.

1 point perspective
eye vector- (80, 0, 20), centre vector- (0, 0, 0), up vector- (0, 0, 1)

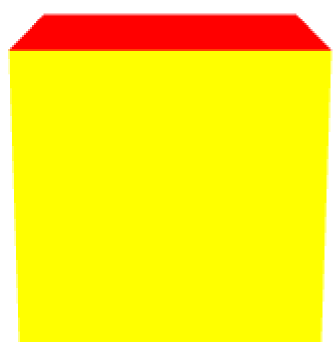


Fig. 1. 1 point perspective

2 point perspective
eye vector- (80, 80, 20), centre vector- (0, 0, 0), up vector- (0, 0, 1)

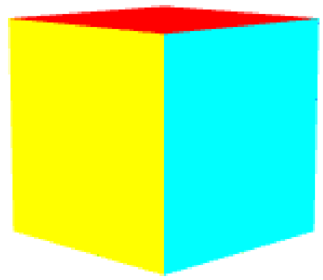
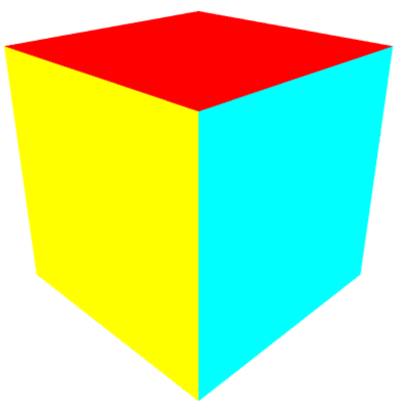
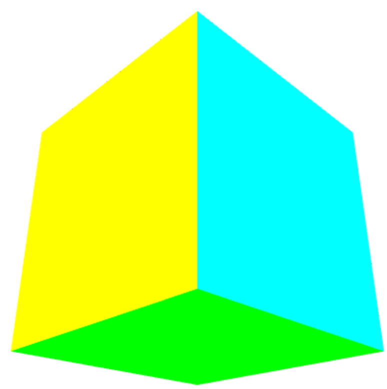


Fig. 2. 2 point perspective

3 point perspective
bird's eye view: eye vector- (30, 30, 20), centre vector- (0, 0, 0), up vector- (0, 0, 1)
rat's eye view: eye vector- (30, 30, -20), centre vector- (0, 0, 0), up vector- (0, 0, 1)



(a) bird's eye view



(b) rat's eye view

Fig. 3. 3 point perspective