

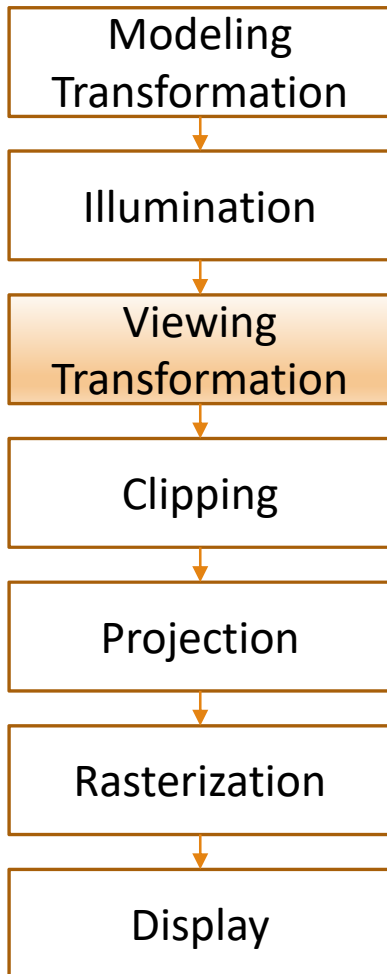
CT5510: Computer Graphics

Viewing Transformation

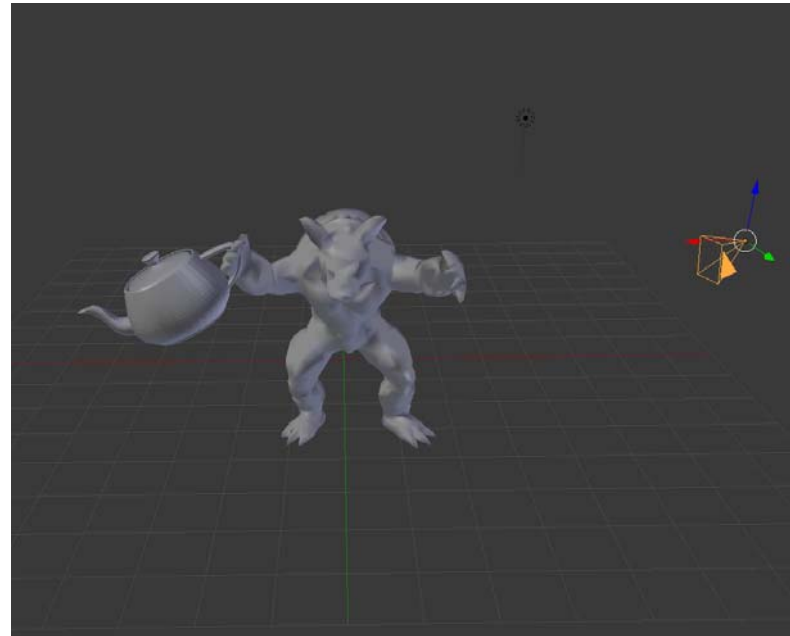
BOCHANG MOON



Viewing Transformation



- Transform all points from world space to *eye space*
 - Camera position transforms into the origin

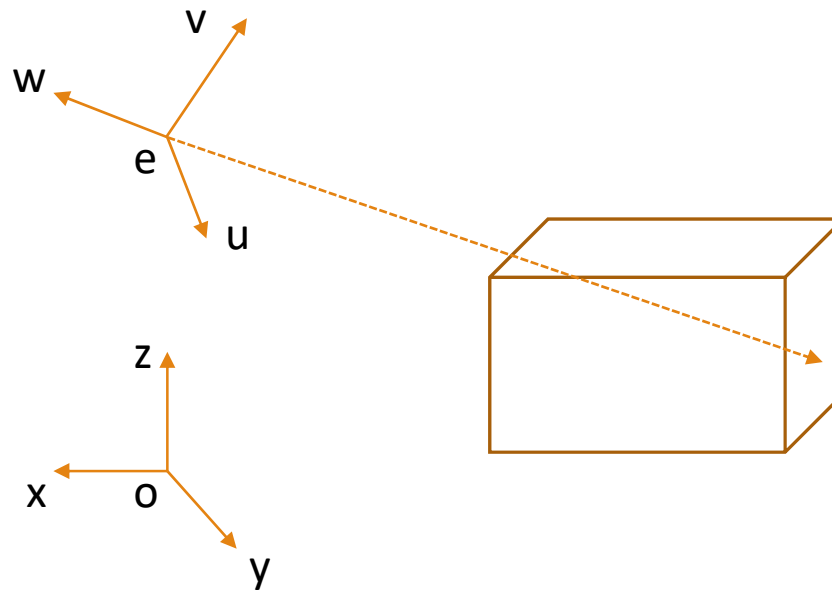


Viewing Transformation

- Define camera position and its orientation
- Specify the following:
 - Location of the camera, $\mathbf{e} = (x_e, y_e, z_e)$
 - Direction where the camera is aiming at, vector $\mathbf{g} = (x_g, y_g, z_g)$
 - Upward direction of the camera, vector $\mathbf{t} = (x_t, y_t, z_t)$
 - Roughly orthogonal to \mathbf{g} (not necessary)
- A user specifies these variables.
- These variables are defined in *world space*.

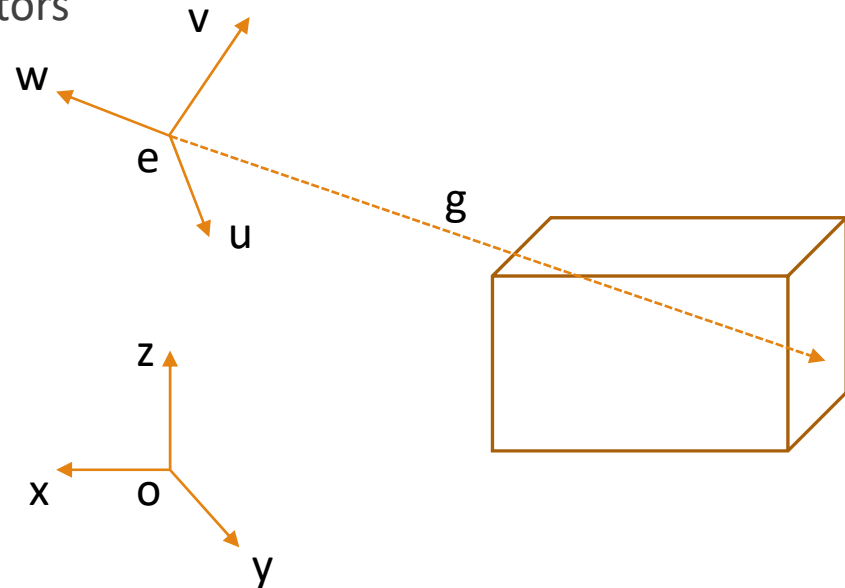
Viewing Transformation

- Our task: transform all points defined in world space into new points in eye space
- Need to build a coordinate system (eye space) from the specifications of the camera



Viewing Transformation

- Our task: transform all points defined in world space into new points in eye space
- Need to build a coordinate system (eye space) from the specifications of the camera
 - Construct basis vectors from two input vectors
 - $\mathbf{w} = -\frac{\mathbf{g}}{\|\mathbf{g}\|}$
 - $\mathbf{u} = \frac{\mathbf{t} \times \mathbf{w}}{\|\mathbf{t} \times \mathbf{w}\|}$
 - $\mathbf{v} = \mathbf{w} \times \mathbf{u}$



Viewing Transformation in OpenGL

- `void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,`
- `GLdouble centerX, GLdouble centerY, GLdouble centerZ,`
- `GLdouble upX, GLdouble upY, GLdouble upZ);`
- Parameters
 - `eyeX, eyeY, eyeZ`
 - Specifies the position of the camera
 - `centerX, centerY, centerZ`
 - Specifies the position of the reference point that your camera is looking at
 - `upX, upY, upZ`
 - Specifies the direction of the up vector
- Issue: `centerX, centerY, centerZ` is not the gaze vector \mathbf{g} . How can we compute this?



Viewing Transformation in OpenGL

- This can be considered as the following matrix transformations:
 - Step 1: translate the camera position \mathbf{e} to the origin in world space
 - Step 2: rotate $\mathbf{u}, \mathbf{v}, \mathbf{w}$ to be aligned to $\mathbf{x}, \mathbf{y}, \mathbf{z}$

- $$\mathbf{M}_{view} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$
$$= \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{why?})$$

