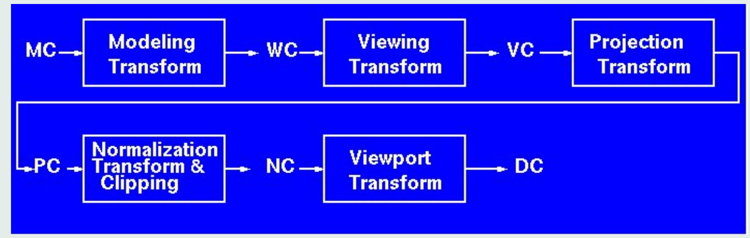
# Viewing in 3D

## Overview

- 3D Viewing Pipeline/Coordinate Systems
- Camera Specification and Transform
- View Volume

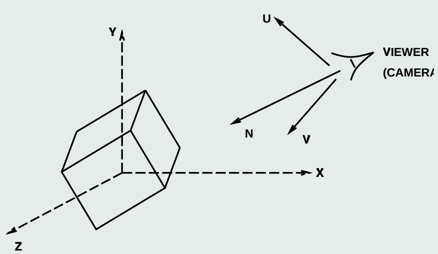


# 3D Viewing/Rendering Pipeline






# Synthetic Camera Model




- Pin-hole camera model (all objects in perfect focus)
- A convenient, flexible approach to specifying viewing parameters.
- Allows operations such as fly-by, swivel, zoom and head-tilt through 3D environments.
- All specification in a viewing (camera or eye) coordinate system.



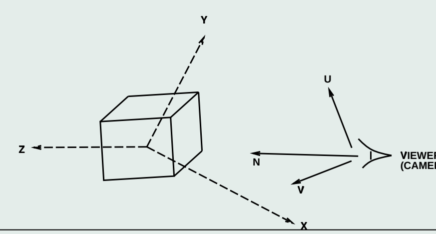
# Camera Specification (simplified)

## In World Coordinates

- View Point (viewer position or center of projection with view direction $\vec{N}$)
- View coordinate system defined with respect to view point.

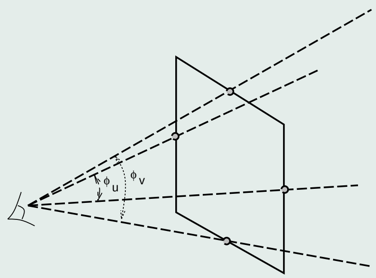
## In View Coordinates

- View plane onto which objects are projected to forma a 2D image.
- View frustum (volume) which defines the field of view.

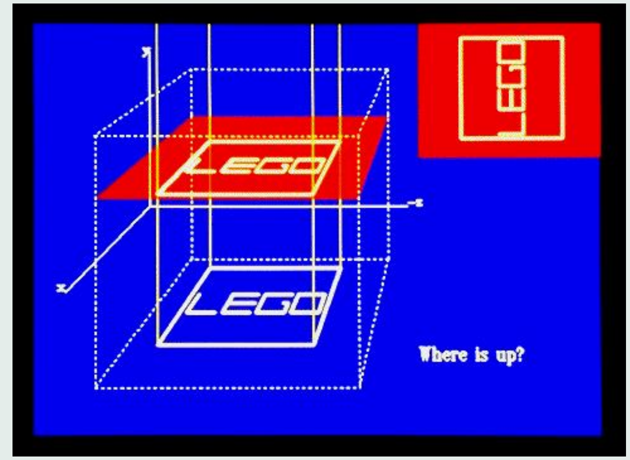


# Alternative Specification

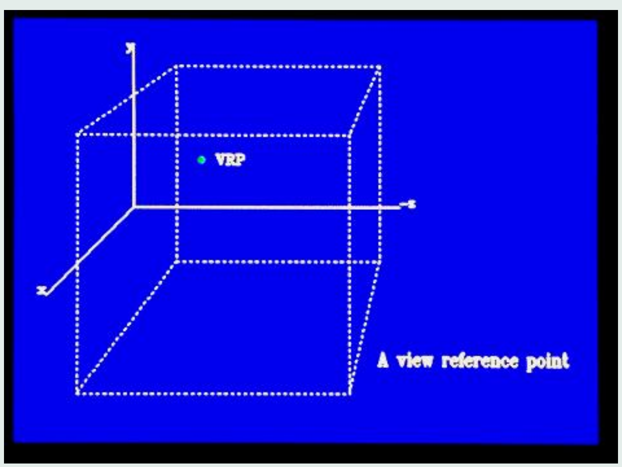
- *Window*, using field of view angles $(\theta_u, \theta_v)$ and $d$, the distance to the projection plane.



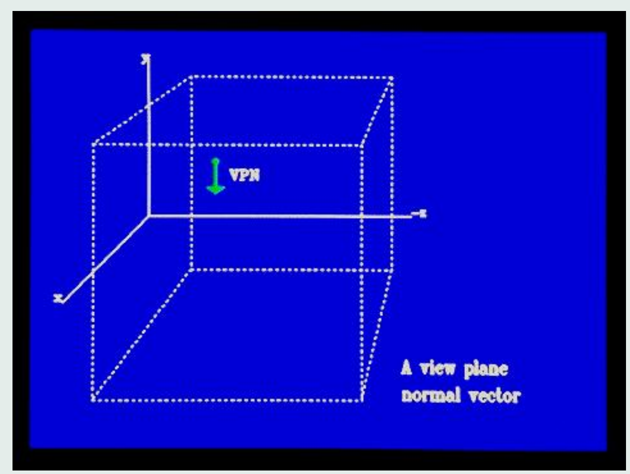


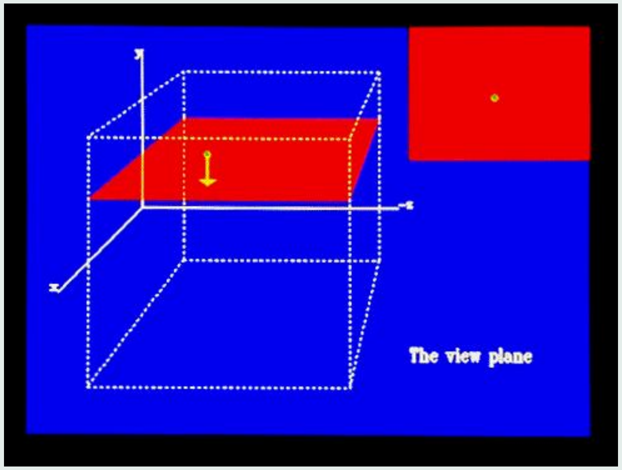





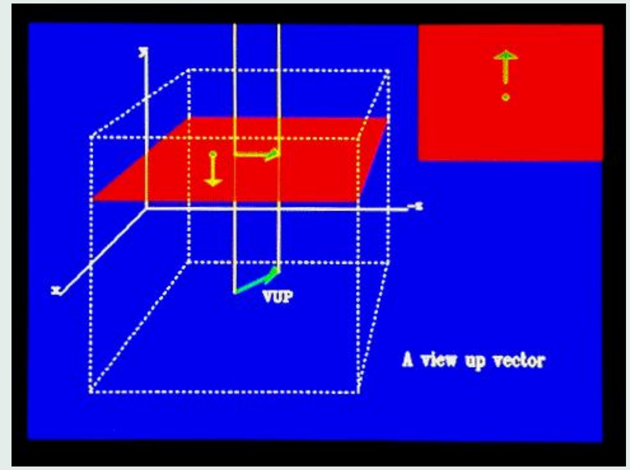





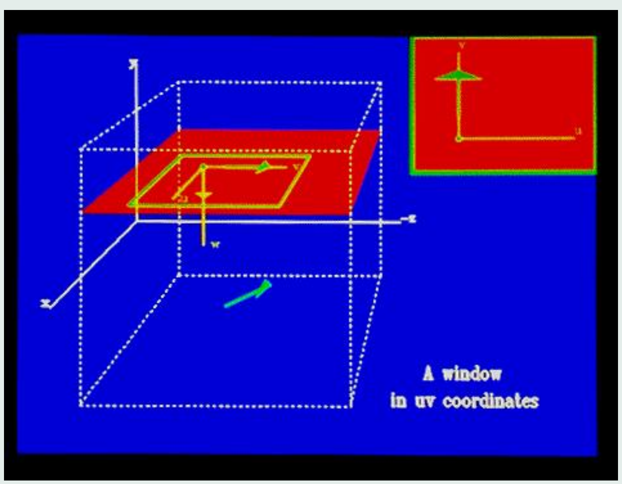




The view plane

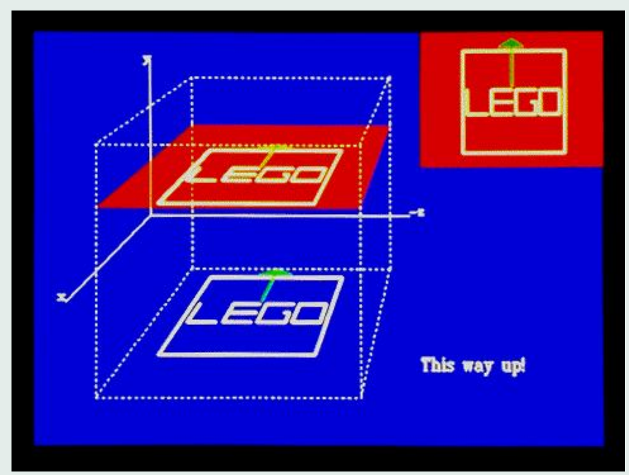


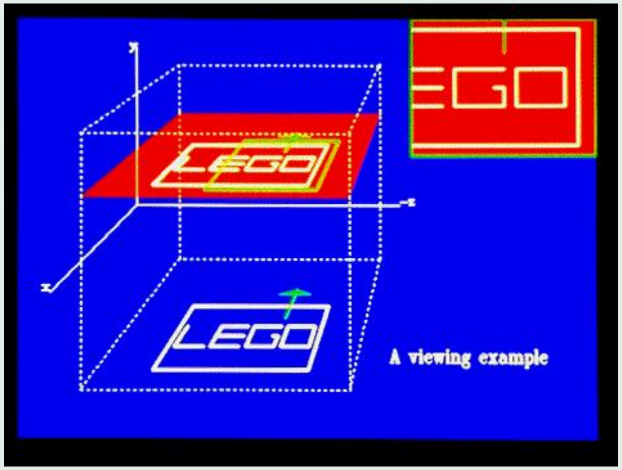

VUP
A view up vector



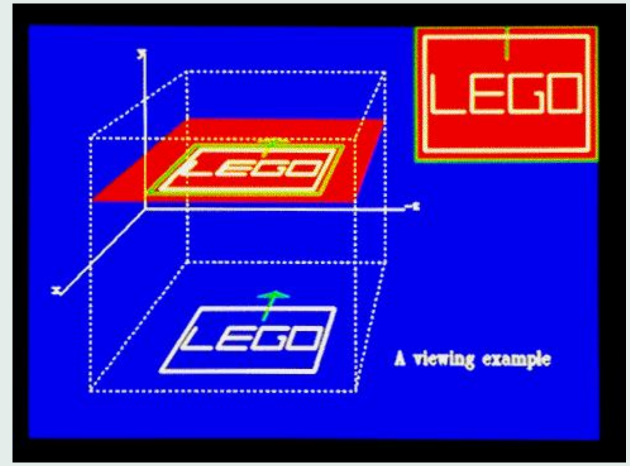

A window
in uv coordinates



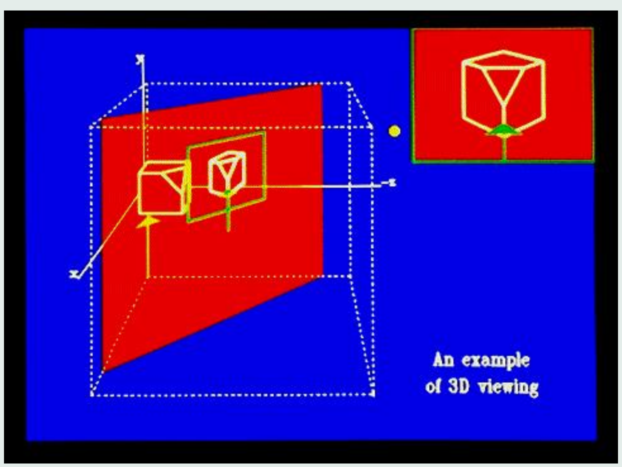

LEGO
LEGO
LEGO
This way up!










# The Viewing Transformation




$M_{wv}$ maps

$$\begin{aligned}(0,0,0) &\Longrightarrow (r_x, r_y, r_z)\\ (\vec{X},\vec{Y},\vec{Z}) &\Longrightarrow (\vec{U},\vec{V},\vec{N})\end{aligned}$$

Hence

$$\begin{aligned}\vec{U}^T &= \vec{M}\vec{X}^T = \text{left column of } \vec{M}\\ \vec{V}^T &= \vec{M}\vec{Y}^T = \text{middle column of } \vec{M}\\ \vec{N}^T &= \vec{M}\vec{Z}^T = \text{right column of } \vec{M}\end{aligned}$$

## The Viewing Transformation

$$M = \begin{bmatrix}\vec{U}^T & \vec{V}^T & \vec{N}^T\end{bmatrix} = \begin{bmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{bmatrix}$$

$M$ is the **rotation matrix** that transforms $(\vec{X}, \vec{Y}, \vec{Z})$ to $(\vec{U}, \vec{V}, \vec{N})$.



## The Viewing Transformation




$$\begin{aligned}
\vec{P} &= \vec{M}\vec{Q} + \vec{r} \\
\vec{Q} = (a, b, c) &= \vec{M}^{-1}(\vec{P} - \vec{r}) \\
&= \vec{M}^T(\vec{P} - \vec{r}) \\
&= \vec{M}^T\vec{P} - \vec{M}^T\vec{r}
\end{aligned}$$

If $\vec{P} = (0, 0, 0)$ then

$$\vec{M}_{wv} = \vec{R} \bullet \vec{T}$$

$$\vec{M}_{wv} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -r_x \\ 0 & 1 & 0 & -r_y \\ 0 & 0 & 1 & -r_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



## Determining View Parameters

- Specify view point and view direction $\vec{N'}$ in world coordinates (eg., point camera at origin or center of scene)
- Normalize $\vec{N}$ to unit length

$$\vec{N} = \frac{\vec{N'}}{|\vec{N'}|}$$

- Specify "$\vec{UP}$" vector. Normalize to unit length

$$\vec{V'} = \frac{\vec{UP}}{|\vec{UP}|}$$

- Calculate $\vec{U}$ and $\vec{V}$

$$\vec{U} = \vec{N} \times \vec{V'}, \vec{V} = \vec{U} \times \vec{N}$$



## View Volume

- The volume of 3D space that is projected and displayed.
- Volume defined by a 2D window on the projection plane and **front** and **back** clipping planes (hither/yon, near/far).
- View volume is specified in camera coordinates.
- View volume is a **truncated frustum**.