University of British Columbia

Composing Geometric Transformations

- composition of Transformations
- scene graphs

Transformations

University of British Columbia

Affine transformations

- linear transformation + translations
- can be expressed as a 3x3 matrix + 3 vector

$$P' = M \cdot P + T$$

4x4 matrices

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & T_x \\ m_{21} & m_{22} & m_{23} & T_y \\ m_{31} & m_{32} & m_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

h=1

Transformations

University of British Columbia

translate(a,b,c)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

scale(a,b,c)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$Rotate(z,\theta)$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Simple Compositions

University of British Columbia

$\cos\theta_1 \cdot \cos\theta_2 - \sin\theta_1 \cdot \sin\theta_2 = \cos(\theta_1 + \theta_2)$

translate(a,b,c) translate(d,e,f)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & d \\ & 1 & & e \\ & & 1 & f \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 & & & a+d \\ & 1 & & b+e \\ & & 1 & c+f \\ & & & 1 \end{bmatrix}$$

scale(a,b,c) scale(d,e,f)

$$\begin{bmatrix} a & & \\ & b & \\ & & c \end{bmatrix} \begin{bmatrix} d & & \\ & e & \\ & & f \end{bmatrix} = \begin{bmatrix} a \cdot d & & \\ & b \cdot e & \\ & & c \cdot f \end{bmatrix}$$

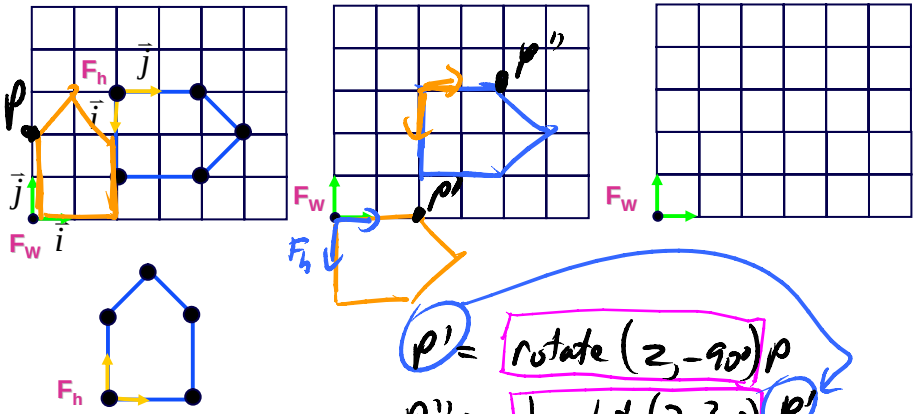
$Rotate(z,\theta_1)$ $Rotate(z,\theta_2)$

$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & & \\ \sin\theta_1 & \cos\theta_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & & \\ \sin\theta_2 & \cos\theta_2 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

Composing Transformations

University of British Columbia

suppose we want



Composing Transformations

University of British Columbia

$$P_w = Trans(2,3,0)\, Rot(z,-90)\, P_h$$

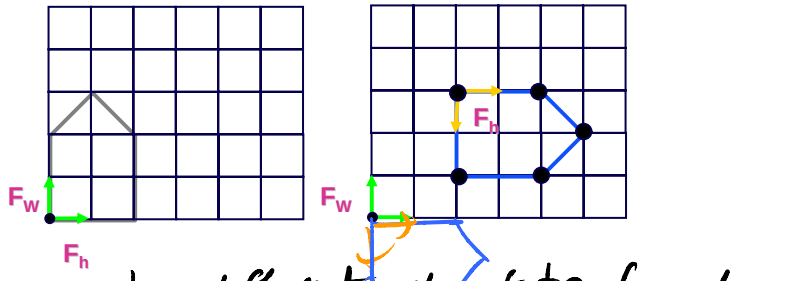
Translate (0,3,0)

- R-to-L: interpret operations wrt fixed coords
- L-to-R: interpret operations wrt local coords
- OpenGL (L-to-R, local coords)

glTranslate(2,3,0)
glRotate(-90°, 0,0,1)
glTranslate(0,3,0)

Composing Transformations

University of British Columbia



Same example, different order of transformations

glRotate(?)
glTranslate(?)

glRotate(z, -90°)
glTranslate(-3,2,0)

Composing Transformations

University of British Columbia

Equivalence

$$P_w = Trans(2,3,0)\, Rot(z,-90)\, P_h$$
$$P_w = Rot(z,-90)\, Trans(-3,2,0)\, P_h$$

Undoing Transformations

$(Trans(a,b,c))^{-1} = Trans(-a,-b,-c)$

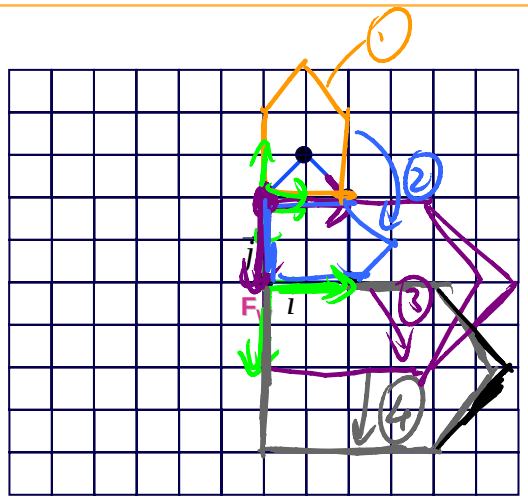
$Trans(a,b,c)\, Trans(-a,-b,-c) = I$

$Rot(z,\theta)^{-1} = Rot(z,-\theta) = Rot(-z,\theta)$

$Scale(a,b,c)^{-1} = Scale\left(\frac{1}{a},\frac{1}{b},\frac{1}{c}\right)$

Test yourself ...

University of British Columbia



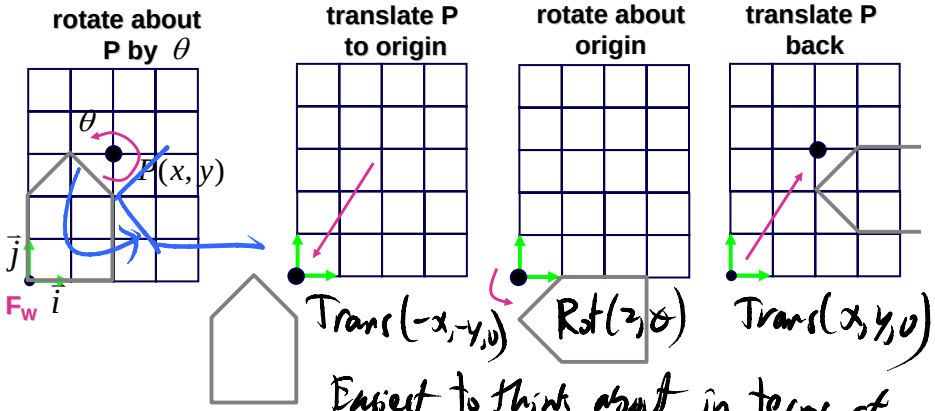
```
glTranslatef(0,2,0);
glRotatef(-90,0,0,1);
glScale(2,2,2);
glTranslate(1,0,0);
draw_house();
```

local coords L→R

$M = Trans(0,2,0)\,Rot(-90,z)\,Scale(2,2,2)\,Trans(1,0,0)$

Rotation about a point

University of British Columbia



Easiest to think about in terms of fixed coords. (R→L)

$M = Trans(x,y,0)\,Rot(z,\theta)\,Trans(-x,-y,0)$

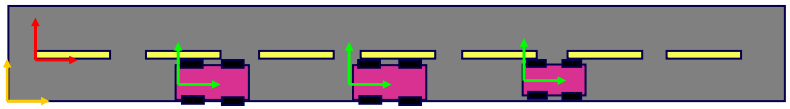
Rotation about an arbitrary axis

University of British Columbia

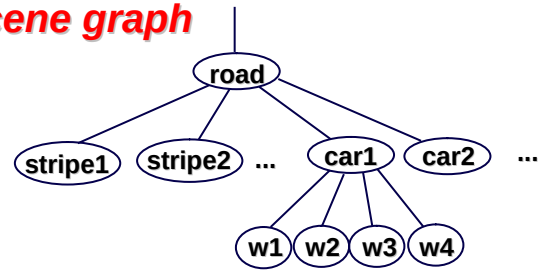
```
glRotatef( angle, x, y, z);
```


Transformation Hierarchies

University of British Columbia

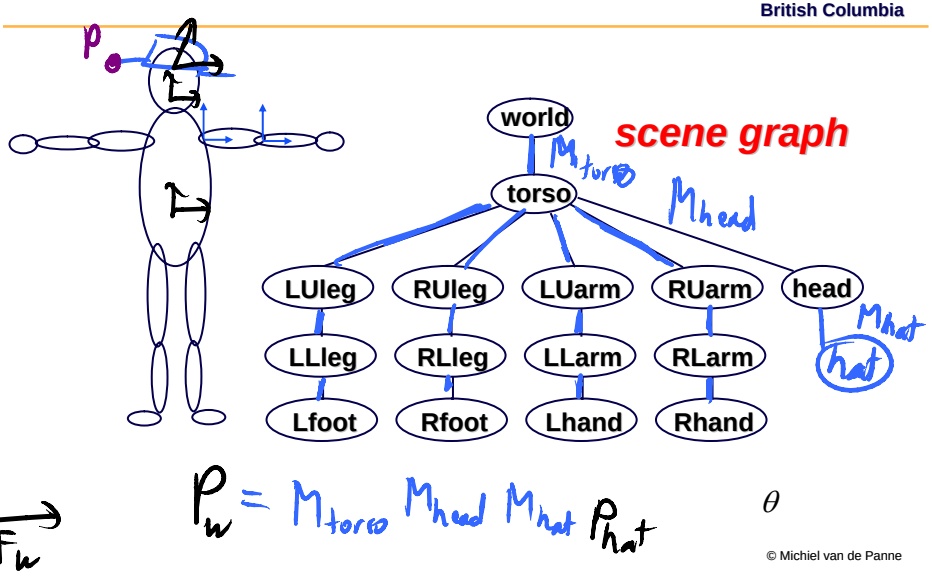


scene graph



Transformation Hierarchies

University of British Columbia

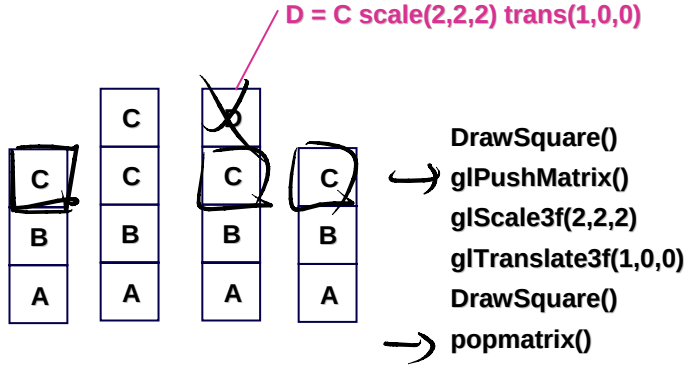


$P_w = M_{torso} M_{head} M_{hat} P_{hat}$

Transformation Hierarchies

University of British Columbia

Matrix Stack



glVertex(p)
$p' = Cp$

Hierarchical Modeling with Matrix Stacks

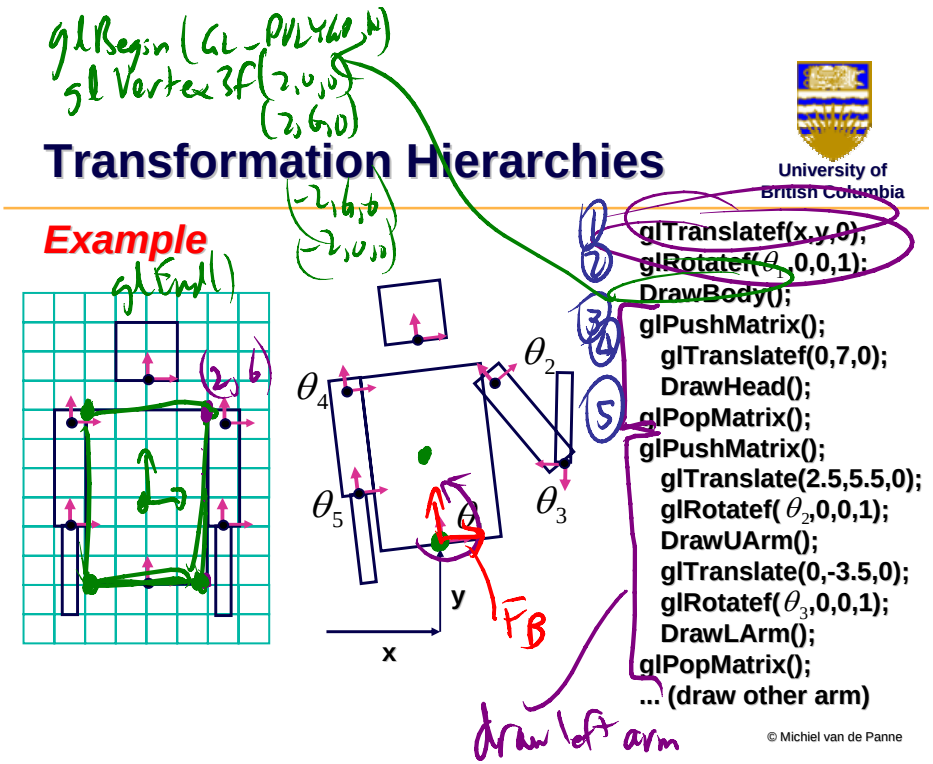
University of British Columbia

- provides a means of returning to a previously-used coordinate system
- graphical scenes and characters have a natural hierarchical structure
- depth of matrix stacks is limited in hardware
  - *typically: 16 for ModelView, 4 for Projection*

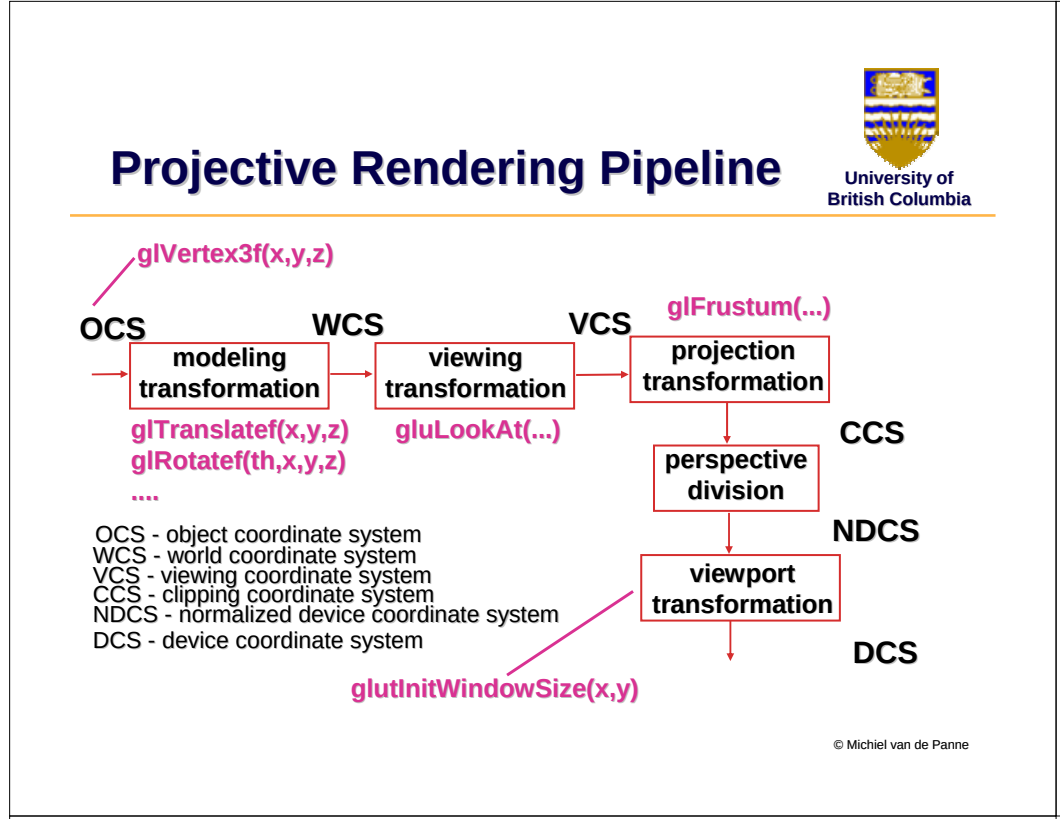
Transformation Hierarchies

University of British Columbia

Example



Projective Rendering Pipeline



University of British Columbia

OCS - object coordinate system
WCS - world coordinate system
VCS - viewing coordinate system
CCS - clipping coordinate system
NDCS - normalized device coordinate system
DCS - device coordinate system

Coming Up…

University of British Columbia

- projection transformations
- viewing transformations