

Composing Geometric Transformations

- composition of Transformations
- scene graphs

Transformations

Affine transformations

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

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

4x4 matrices

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

Transformations

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, θ)

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

$$\begin{aligned} & \cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2 \\ & = \cos(\theta_1 + \theta_2) \end{aligned}$$

Simple Compositions

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

$$\begin{bmatrix} 1 & & a \\ & 1 & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & d \\ & 1 & e \\ & & 1 & f \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & a+d \\ & 1 & b+e \\ & & 1 & c+f \\ & & & 1 \end{bmatrix}$$

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

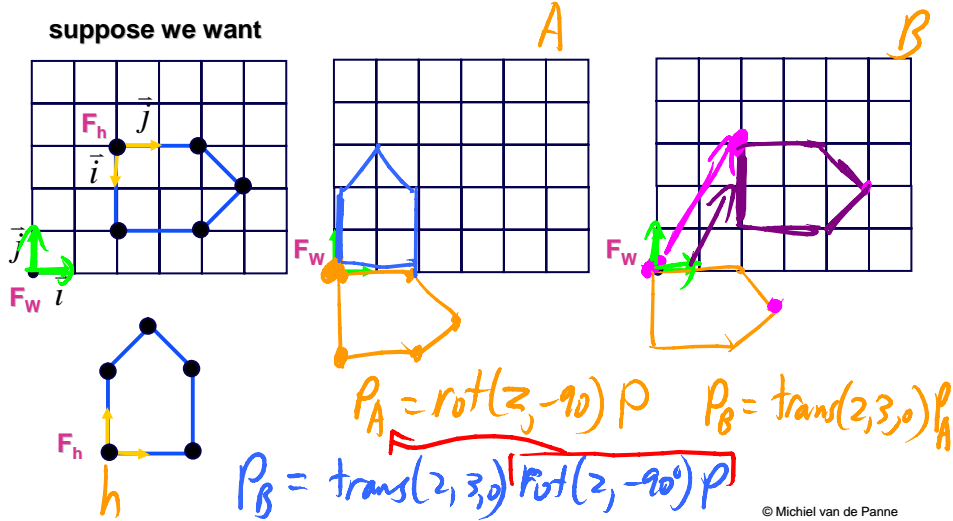
$$\begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix} \begin{bmatrix} d & & & \\ & e & & \\ & & f & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} a \cdot d & & & \\ & b \cdot e & & \\ & & c \cdot f & \\ & & & 1 \end{bmatrix}$$

Rotate(z, θ₁) Rotate(z, θ₂)

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

Composing Transformations

suppose we want



Composing Transformations

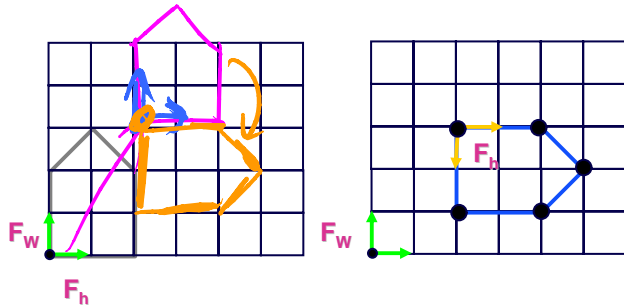
$$P_w = \text{Trans}(2, 3, 0) \text{Rot}(z, -90) P_h$$

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

OpenGL

↓ gl Translatef(2, 3, 0)
 ↓ gl Rotatef(-90, 0, 0, 1)
 ↓ draw House()

Composing Transformations



In Local Coords: $\text{Trans}(2, 3, 0)$ ✓
 Local Coords: $\text{Rot}(z, -90)$

Composing Transformations

Equivalence

- ① $P_w = \text{Trans}(2, 3, 0) \text{Rot}(z, -90) P_h$
- ② $P_w = \text{Rot}(z, -90) \text{Trans}(-3, 2, 0) P_h$

Undoing Transformations

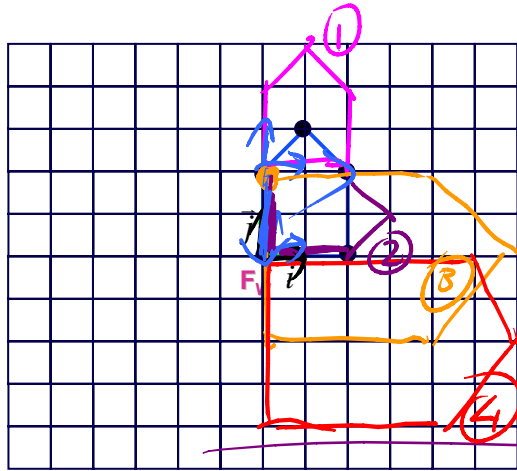
$$I = \text{Trans}(x, y, z) \text{Trans}(-x, -y, -z)$$

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

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

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

Test yourself ...



```

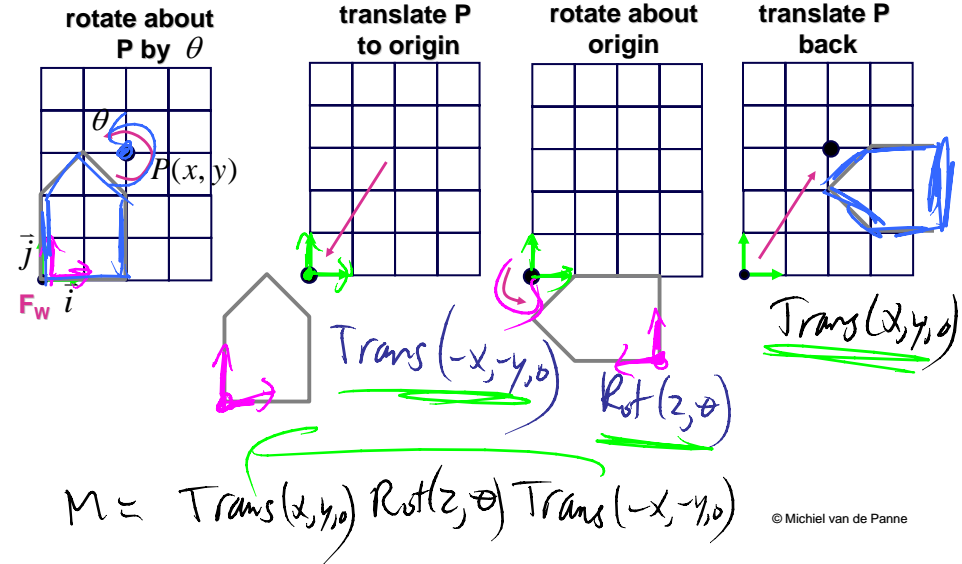
1 glTranslatef(0,2,0);
2 glRotatef(-90,0,0,1);
3 glScale(2,2,2);
4 glTranslate(1,0,0);
draw_house();

```

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

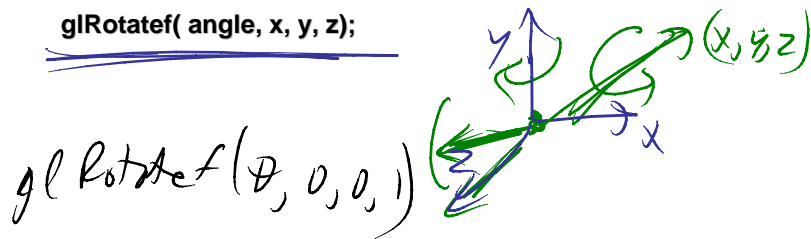
© Michiel van de Panne

Rotation about a point

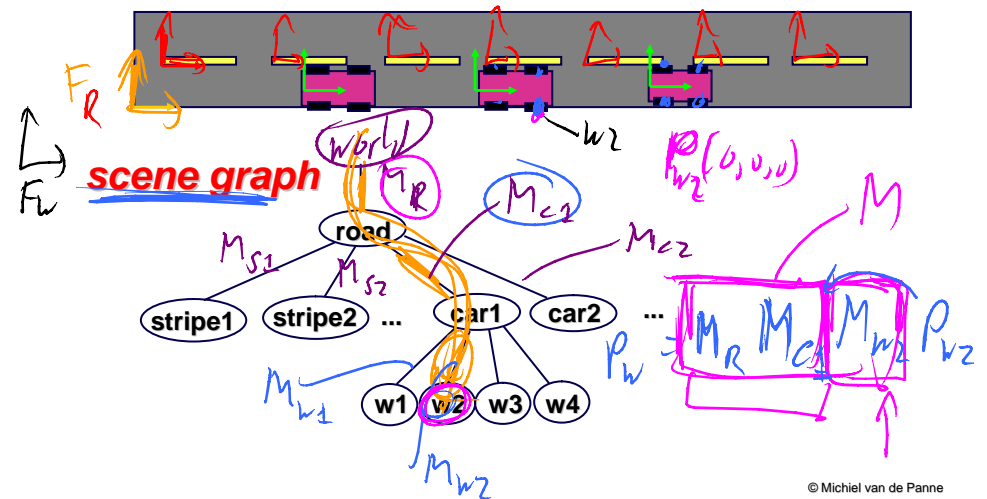


Rotation about an arbitrary axis

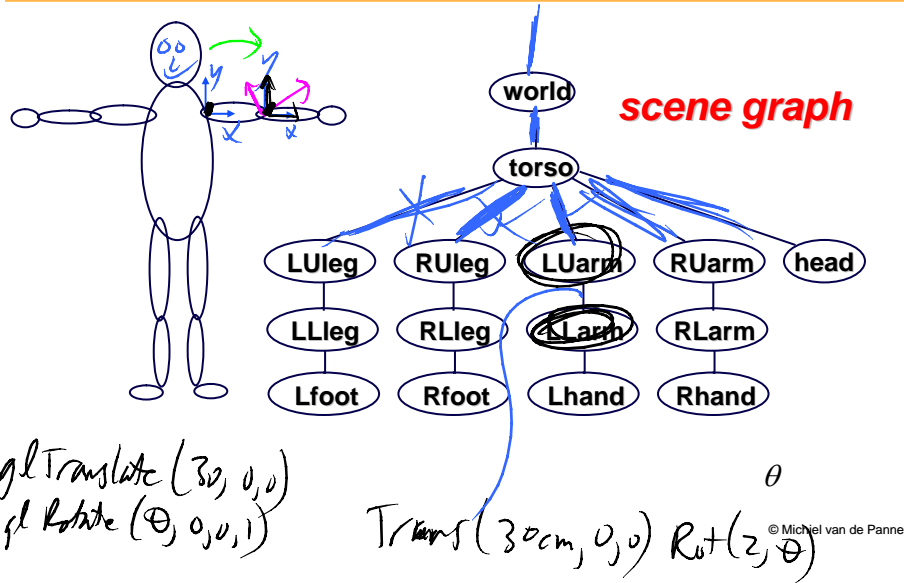
```
glRotatef( angle, x, y, z);
```



Transformation Hierarchies

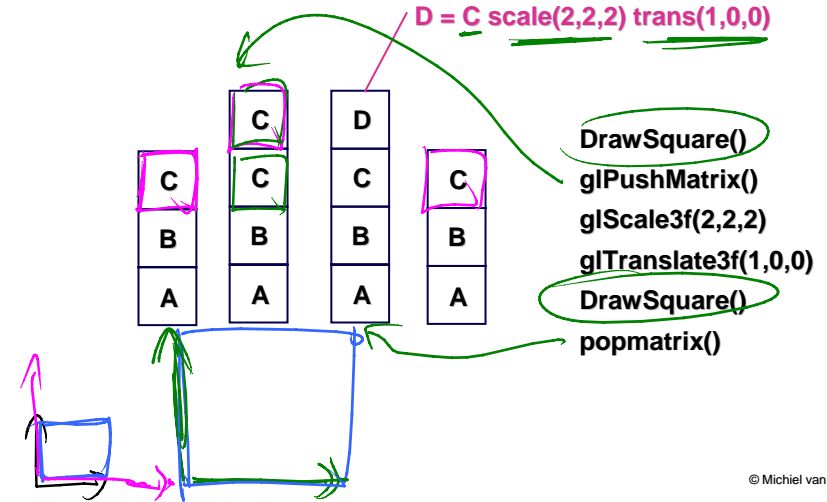


Transformation Hierarchies



Transformation Hierarchies

Matrix Stack

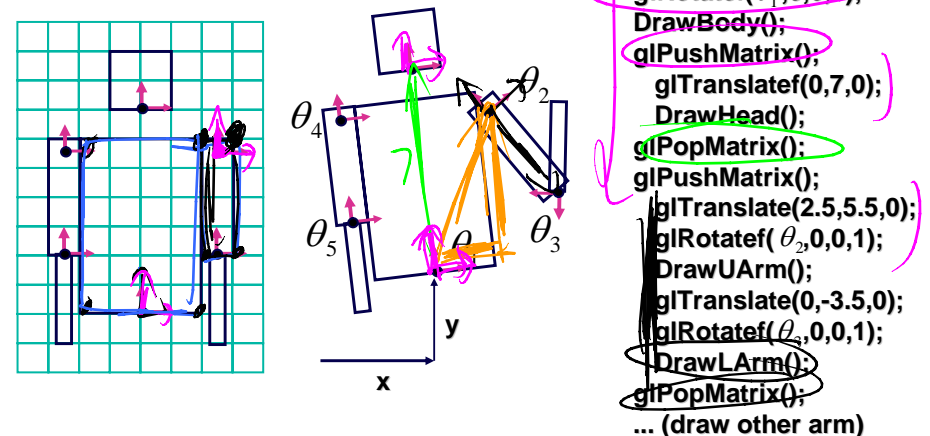


Hierarchical Modeling with Matrix Stacks

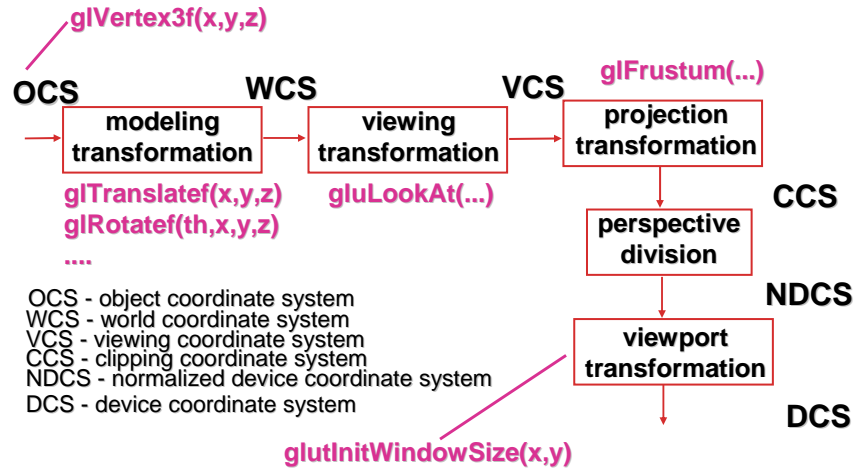
- provides a means of returning to a previously-used coordinate system
- graphical scenes and characters have a natural hierarchical structure
- depth of matrix stacks is limited in hardware
 - typically: 16 for ModelView, 4 for Projection

Transformation Hierarchies

Example



Projective Rendering Pipeline



Coming Up...

- projection transformations
- viewing transformations