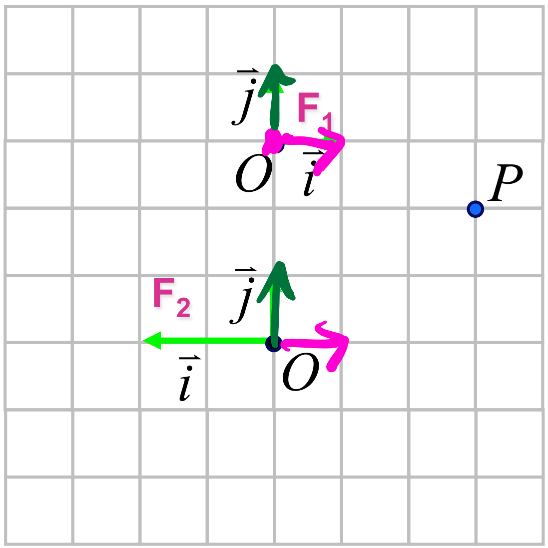
Transformations as a change of basis



$P_1 = (3,-1)$ $P_2 = (-1.5, 2)$ **Goal:** $P_2 = M_{1\to 2} P_1$

$$P = O + x\vec{i} + y\vec{j}$$

$$\begin{bmatrix}x_1\\y_1\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}_1 + x_1\begin{bmatrix}1\\0\end{bmatrix}_1 + y_1\begin{bmatrix}0\\1\end{bmatrix}_1$$

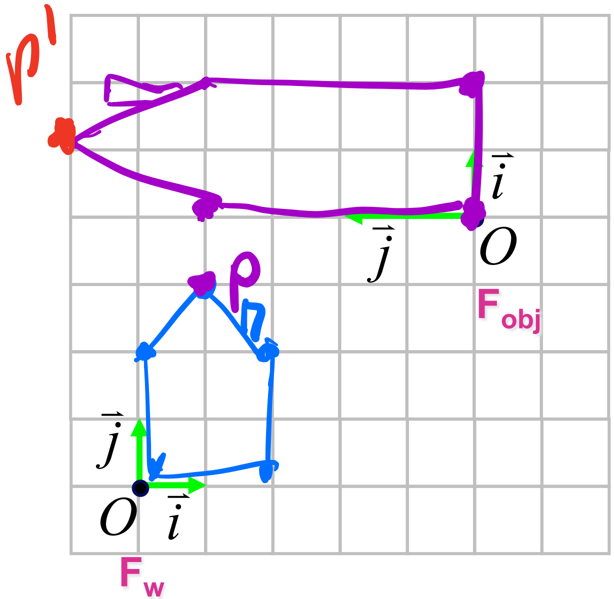
$$\begin{bmatrix}x_2\\y_2\end{bmatrix} = 1\begin{bmatrix}0\\3\end{bmatrix}_2 + x_1\begin{bmatrix}-0.5\\0\end{bmatrix}_2 + y_1\begin{bmatrix}0\\1\end{bmatrix}_2$$

$$\begin{bmatrix}x_2\\y_2\\1\end{bmatrix} = \begin{bmatrix}-0.5&0&0\\0&1&3\\0&0&1\end{bmatrix}\begin{bmatrix}x_1\\y_1\\1\end{bmatrix}$$

check:

$$\begin{bmatrix}-1.5\\2\\1\end{bmatrix}_2 = \begin{bmatrix}0.5&0&0\\0&1&3\\0&0&1\end{bmatrix}\begin{bmatrix}3\\-1\\1\end{bmatrix}_1$$

Transformations as a change of basis



Goal: $P_w = M_{obj \to w} \, P_{obj}$

$$\begin{bmatrix} -1 \\ 5 \\ 1 \end{bmatrix}_w = \begin{array}{c} \begin{array}{ccc} i_{obj} & j_{obj} & O_{obj} \end{array} \\ \begin{bmatrix} 0 & -2 & 5 \\ 1 & 0 & 4 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}_{obj}$$

tip of roof

3D Transformations

Affine transformations

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

$$\boxed{P' = M \cdot P + T}$$

rotations
scales
shears.

O_{obj}

4x4 matrices

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

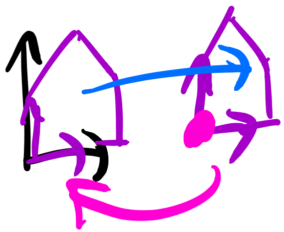
Translations

h=1

i_{obj} j_{obj} k_{obj}

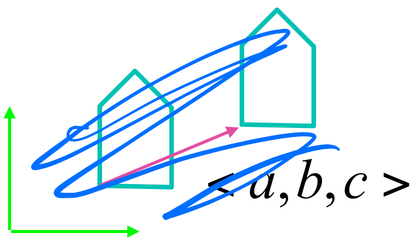
Transformations

Translation



Trans(5,0,0)

Translate(a,b,c)
Trans(a,b,c)



$$x' = x + a$$
$$y' = y + b$$
$$z' = z + c$$

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

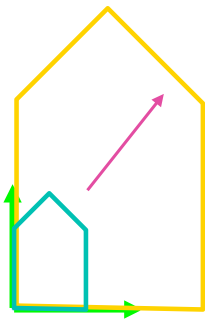
$$P' = MP$$

Trans(a,b,c)

Transformations

Scaling

Scale(a,b,c)



$$x' = a \cdot x$$
$$y' = by$$
$$z' = cz$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

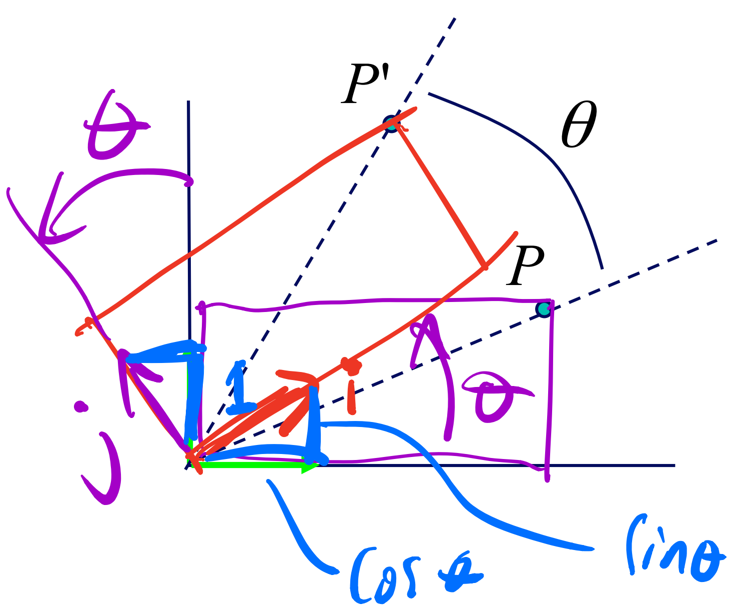
Scale(a,b,c)

Transformations

Rotation

$Rotate(z,\theta)$ CCW is +

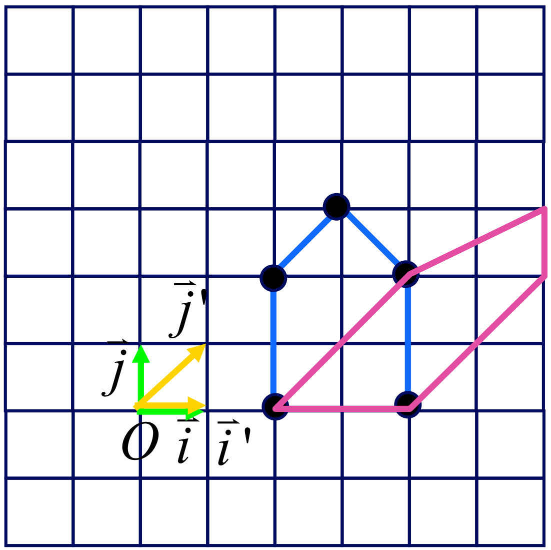
$Rot(z,\theta)$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}_W = \begin{array}{c} \begin{array}{cccc} \vec{i} & \vec{j} & & \end{array} \\ \begin{bmatrix} \cos\theta & -\sin\theta & & 0 \\ \sin\theta & \cos\theta & & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$

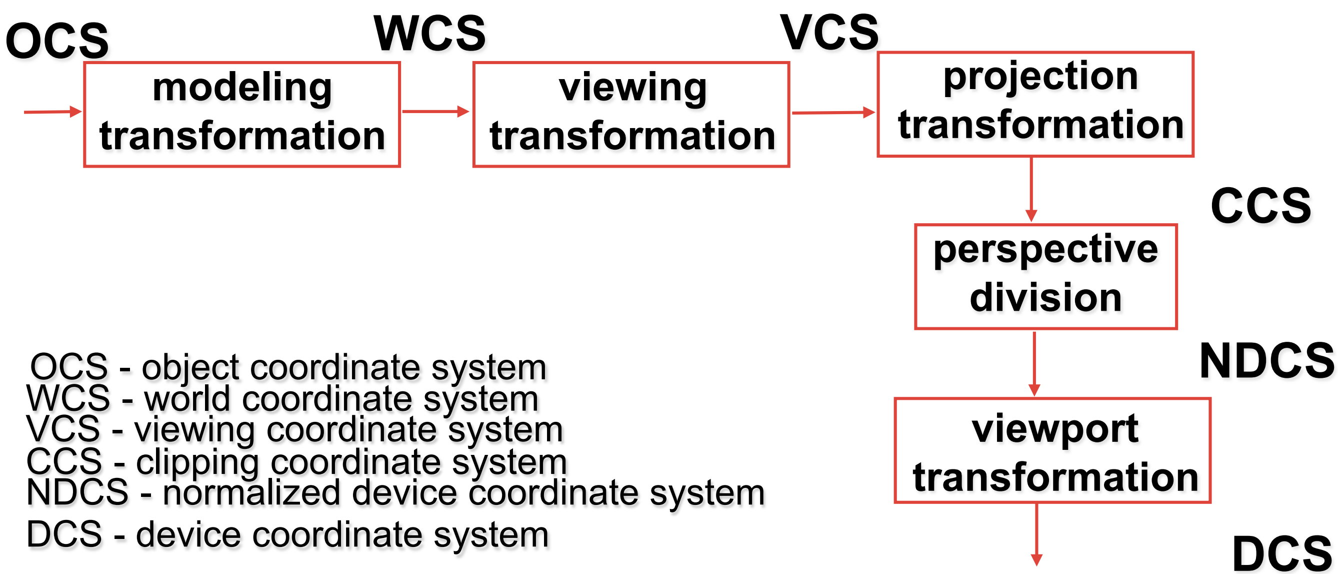
Transformations

Shear



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Vertex Transformations



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

Composition of Transformations

reminder:

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

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

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

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

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

$$\begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix}\begin{bmatrix} d & & & \\ & e & & \\ & & f & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} ad & & & \\ & be & & \\ & & cf & \\ & & & 1 \end{bmatrix} \equiv \text{Scale}(ad, be, cf)$$

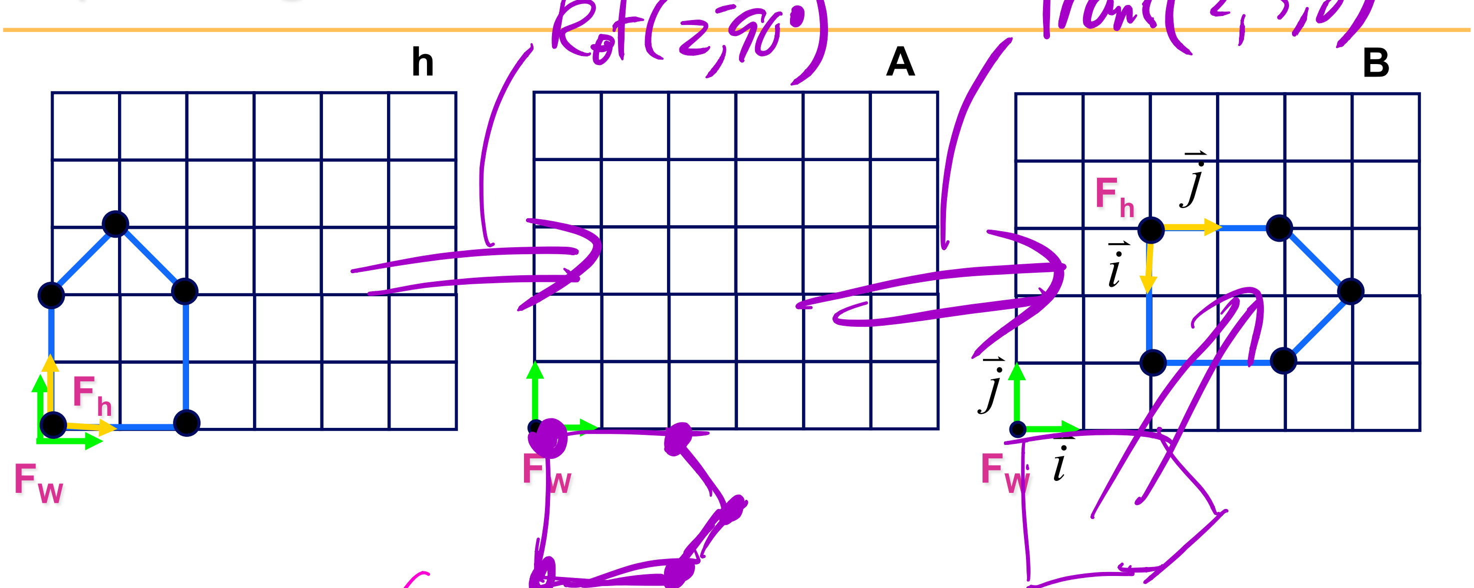
$$Rotate(z,\theta_2)\ Rotate(z,\theta_2) = \text{Rotate}(z, \theta_1+\theta_2)$$

$$\begin{bmatrix} c_1 & -s_1 & & \\ s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}\begin{bmatrix} c_2 & -s_2 & & \\ s_2 & c_2 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_1c_2 - s_1s_2 & -s_1c_2 - c_1s_2 & & \\ s_1c_2 + c_1s_2 & c_1c_2 - s_1s_2 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} c_{12} & -s_{12} & & \\ s_{12} & c_{12} & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

$c_1 = \cos\theta_1$ etc.

$c_{12} = \cos(\theta_1+\theta_2)$ etc.

Composing Transformations
(thinking in fixed coords)



① $P_A = Rot(z,-90°)\, P_h$

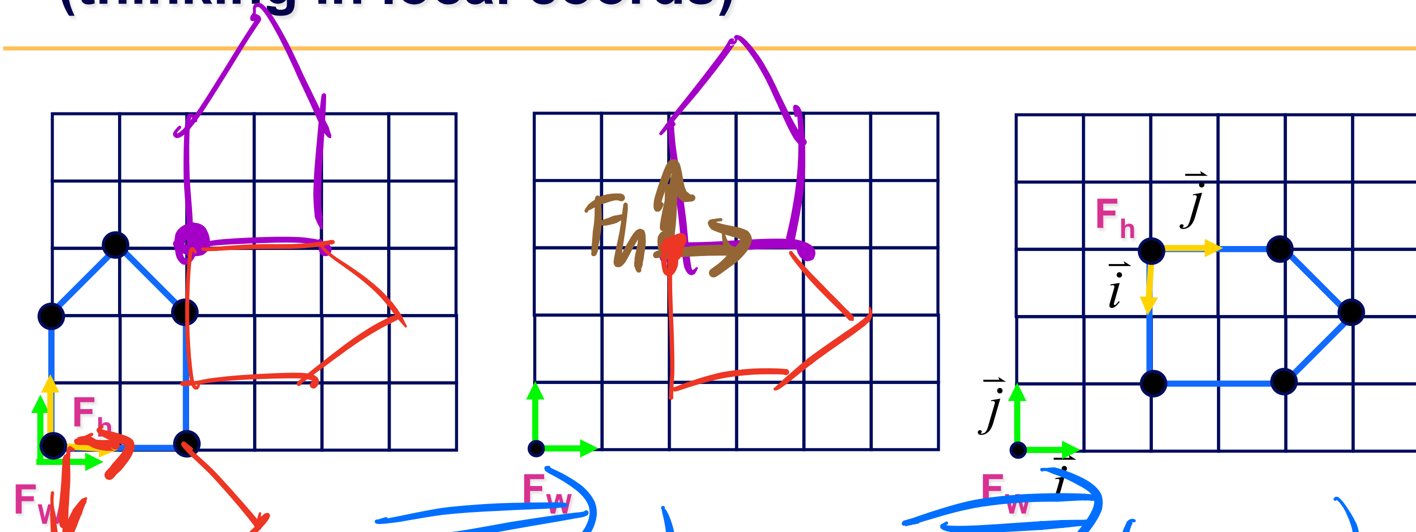
② $P_B = Trans(2,3,0)\, P_A$

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

R-to-L matrices

Composing Transformations

(thinking in local coords)



Rot(z,-90°)

Trans(2,3,0)

Rot(Z,-90°)

$$P_w = Trans(2,3,0)\; Rot(Z,-90°) \cdot P_h$$

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

fixed

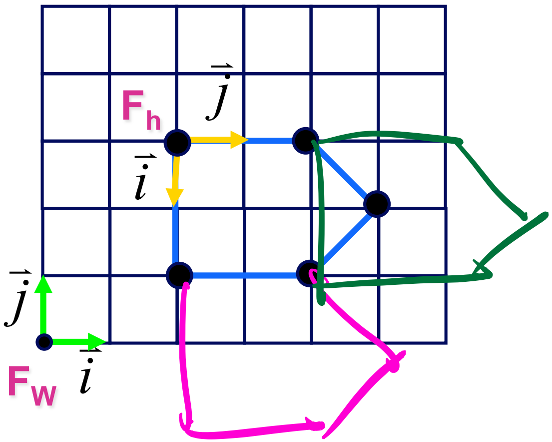
local

Composing Transformations

- left multiply: R-to-L
  - *interpret operations wrt fixed coords*
- right multiply: L-to-R   (default for **code**)
  - *interpret operations wrt local coords*

Summary Example

$T = \text{Translate}(2,0,0)$



$$P'_w = \overset{T}{\downarrow} M \overset{T}{\downarrow} P_h$$

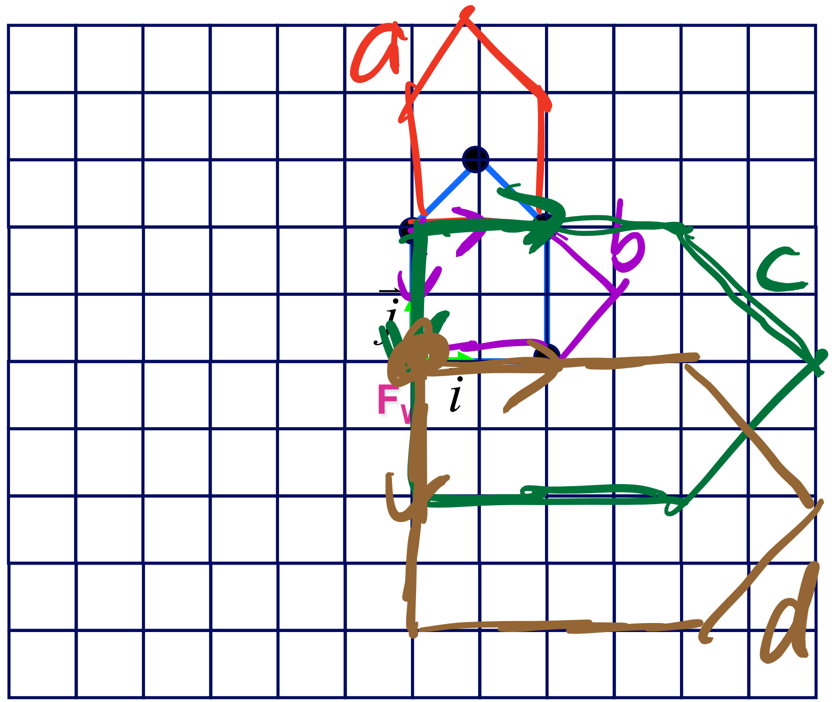
left multiply

translate in F_w

right multiply

translate in F_h

Test yourself ...



$$P_w = \underbrace{M_a M_b M_c M_d}\, P_h$$

a Translate(0,2,0);
b Rotate(z,-90);
c Scale(2,2,2);
d Translate(1,0,0);
DrawHouse();

Test yourself

$P_w = M\, P_h$

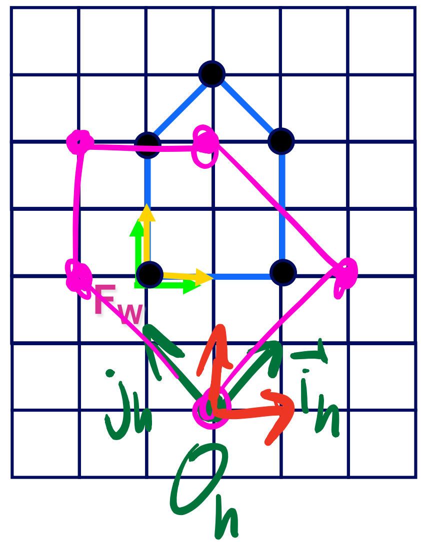
$$M = \begin{array}{c} \begin{array}{cccc} i & j & k & \text{Origin} \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

(a) Sketch the origin and basis vectors of the transformed house
(b) Draw the transformed house
(c) Give a sequence of translate(), rotate(), and scale() that implements this

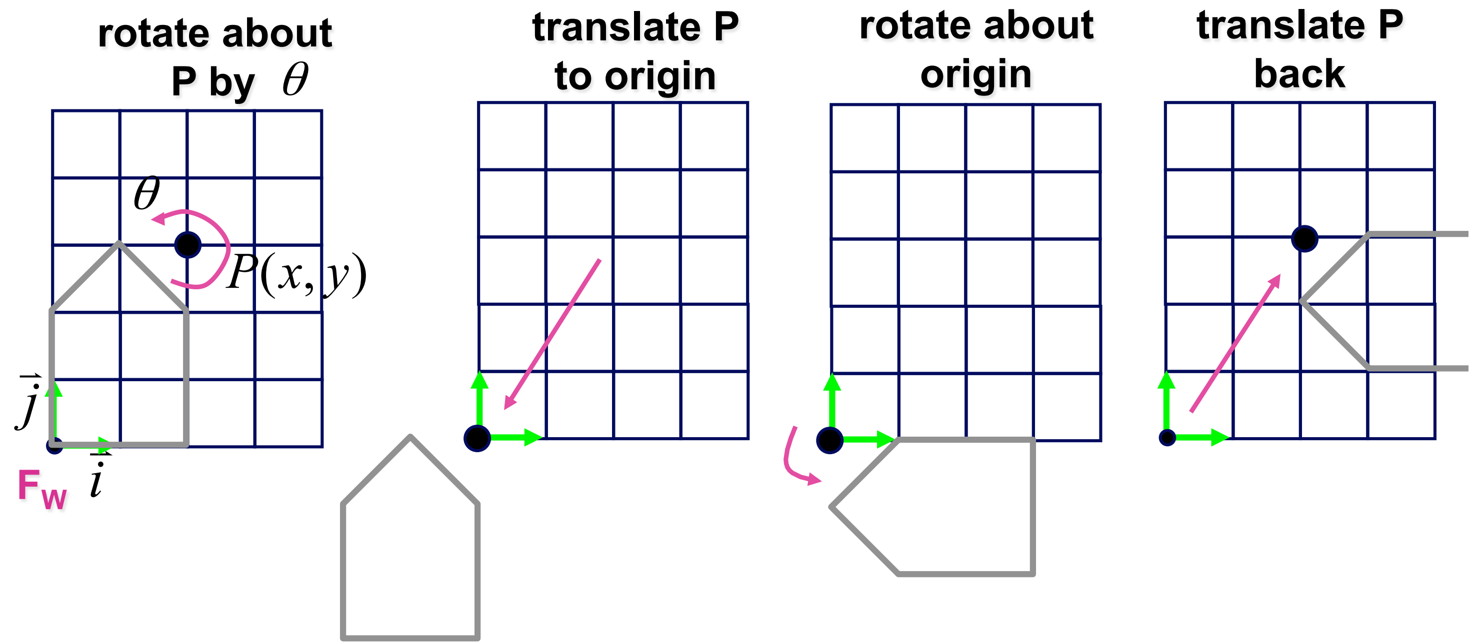
local coord, right multiplying

translate$(1, -2, 0) \equiv T$
rotate$(z, +45^\circ) \equiv R$
scale$(\sqrt{2}, \sqrt{2}, \sqrt{2}) \equiv S$

$M = T \cdot R \cdot S$



Rotation about a point

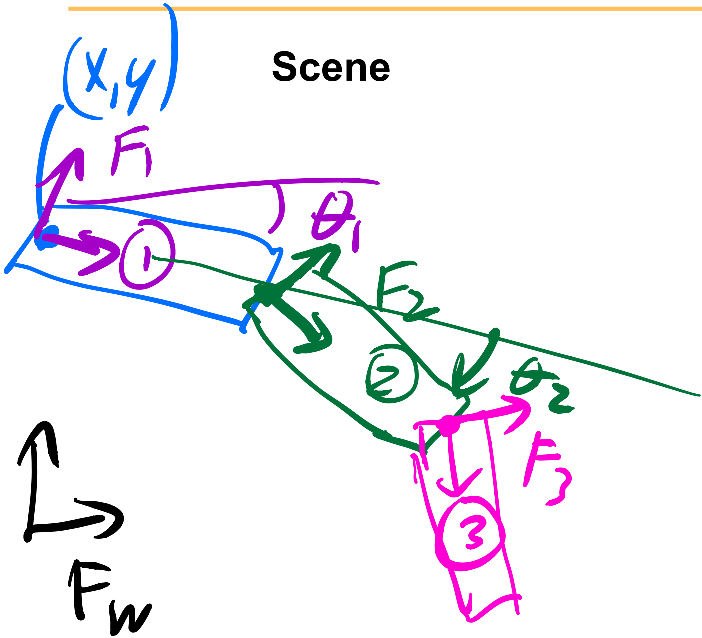


Rotation about an arbitrary axis

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

Transformations in Scene Graphs (1)

Scene

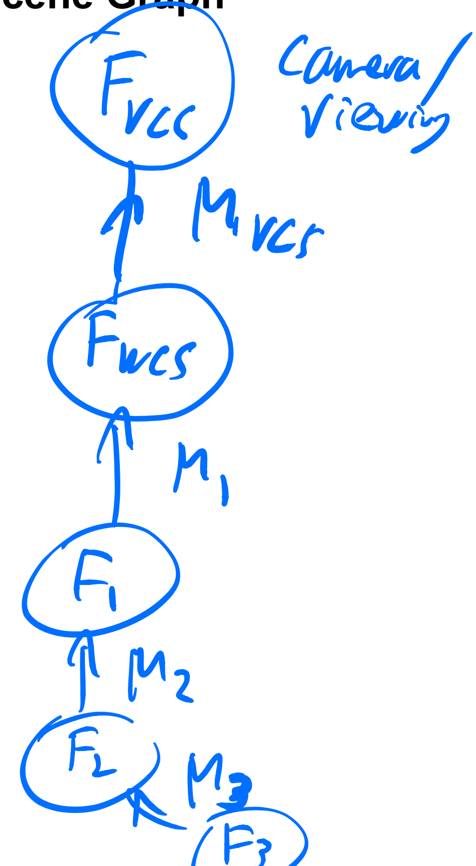


links: 4 x 1

degrees of freedom

$x, y, \theta_1, \theta_2, \theta_3$

Scene Graph



Transformations in Scene Graphs (2)

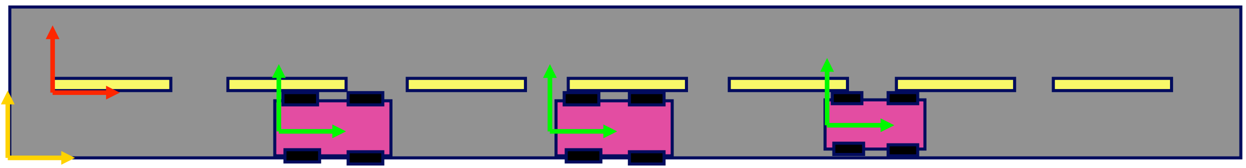
Transforming Vertices

Math

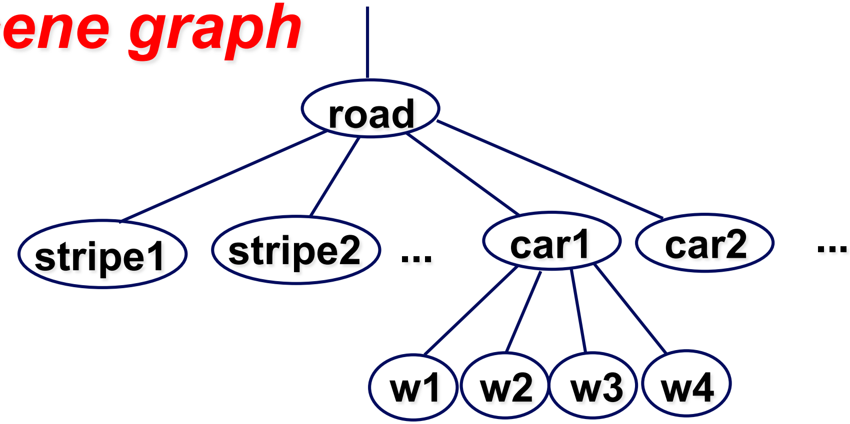
Code

how we'll usually draw it:

Transformation Hierarchy

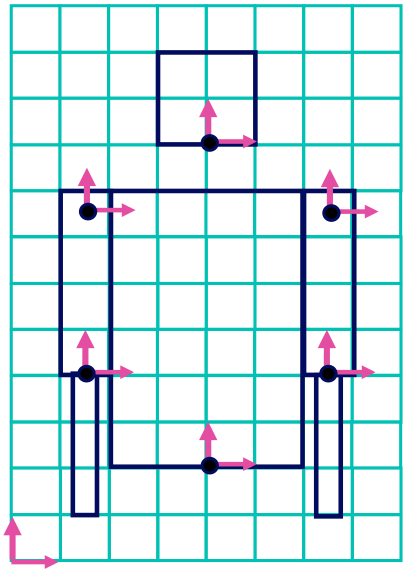


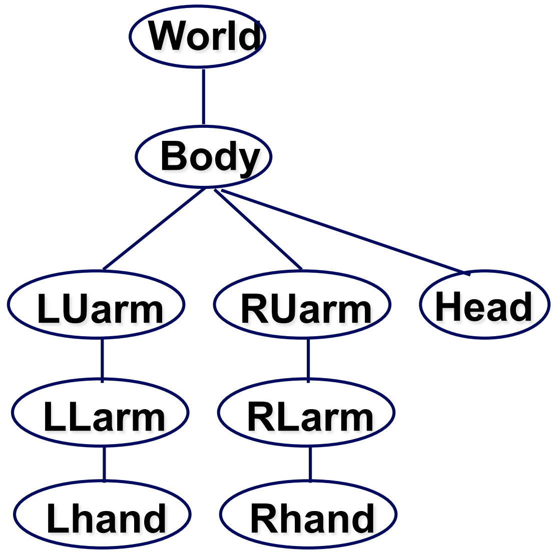
scene graph



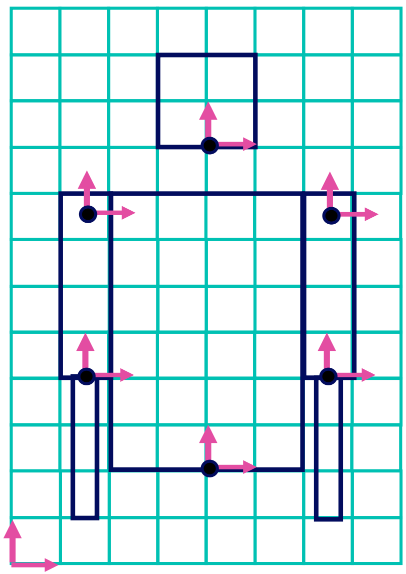
Transformation Hierarchy

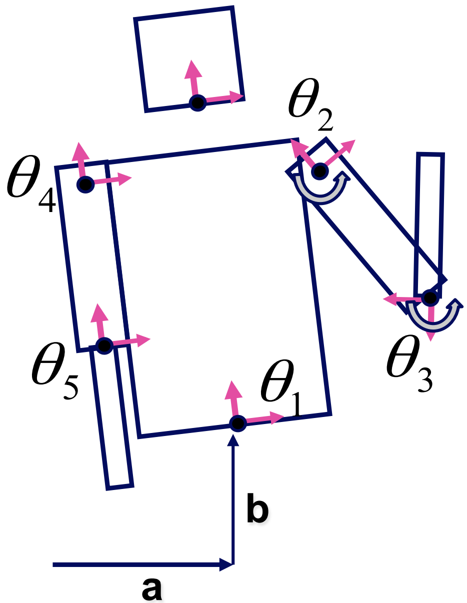
A matrix stack allows for convenient return to a previous coordinate frame.





Code to Draw using a Matrix Stack





looking at character from behind

```
M.Translate(a,b,0);
M.Rotatef(θ1,0,0,1);
DrawBody();
PushMatrix(M);
  M.Translate(0,7,0);
  M.DrawHead();
M=PopMatrix();
PushMatrix(M);
  M.Translate(2.5,5.5,0);
  M.Rotate(θ2,0,0,1);
  DrawRUarm();
  M.Translate(0,-3.5,0);
  M.Rotate(θ3,0,0,1);
  DrawRLarm();
M=PopMatrix();
... (draw left arm)
```