

## Geometric Transformations

- **review of relevant math**
- **4x4 transformation matrices**

## Math Review

### matrix vector multiplication

- points as column vectors

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \quad P' = MP$$

*Handwritten notes: 'h=1' with arrows pointing to the h' and h elements in the vectors.*

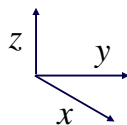
- points as row vectors

~~$$[x' \ y' \ z' \ h'] = [x \ y \ z \ h] \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}^T \quad P'^T = P^T M^T$$~~

## Math Review

### Coordinate Systems

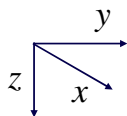
#### Right-handed Coordinate System



$$z = x \times y$$

using right-hand rule

#### Left-handed Coordinate System

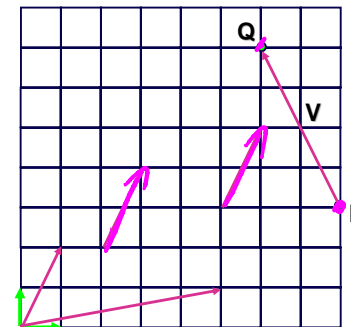


$$z = x \times y$$

using left-hand rule

## Math Review

### Points and Vectors



**vector space**  
vectors are invariant  
under translation

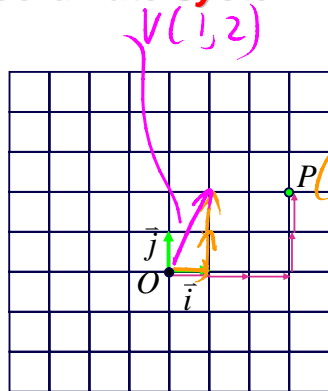
**affine space:** allows vector-to-point addition

$$P + V = Q$$

$$Q - P = V$$

# Math Review

## Coordinate System vs Frame



coordinate system: basis vectors  
frame: basis vectors + Origin

allows for points

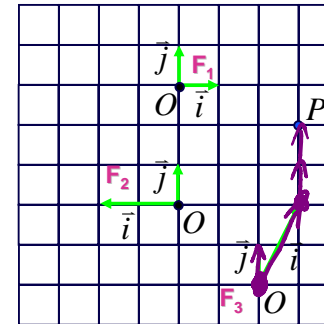
$$V = x\vec{i} + y\vec{j}$$

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

origin

# Math Review

## Working with Frames



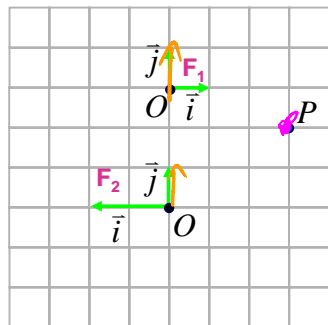
$$P = O + x\vec{i} + y\vec{j}$$

$F_1$   $P(3,-1)$   $0\vec{i}_1 + 3\vec{j}_1 - 1\vec{k}_1$   
 $F_2$   $P(-1.5, 2)$   
 $F_3$   $P(1, 2)$   $1\vec{i}_3 + 2\vec{j}_3$

# Transformations

## Transformations as a change of frame

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



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

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

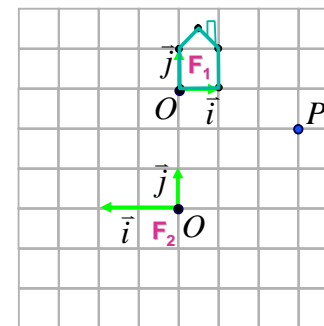
$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

$P_2 = MP_1$

check:  $P_1(3,-1)$  becomes  $P_2(-1.5, 2)$

# Transformations

## change of basis expressed using a matrix



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

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

# Usage of Transformations

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

$$\begin{bmatrix} x_{obj} \\ y_{obj} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{obj} + x_{obj} \begin{bmatrix} 1 \\ 0 \end{bmatrix}_{obj} + y_{obj} \begin{bmatrix} 0 \\ 1 \end{bmatrix}_{obj}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_w = \begin{bmatrix} 3 \\ 1 \end{bmatrix}_w + x_{obj} \begin{bmatrix} 2 \\ 0 \end{bmatrix}_w + y_{obj} \begin{bmatrix} 0 \\ 2 \end{bmatrix}_w$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

© Michiel van de Panne

# Using Transformations

$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

2D → 3D

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$

GLfloat T[16] = { 2,0,0,0, 0,2,0,0, 0,0,1,0, 3,0,0,1};

© Michiel van de Panne

# Usage of Transformations

```

GLfloat T[16] = { ... };
glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(T);

glBegin(GL_LINE_LOOP);
glVertex2f(0,0);
glVertex2f(2,0);
glVertex2f(2,2);
glVertex2f(1,3);
glVertex2f(0,2);
glEnd();
    
```

© Michiel van de Panne

# Usage of Transformations

**An easier way to do the same thing....**

```

glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

glTranslatef(3,1,0);
glScale(2,2,2);

glBegin(GL_LINE_LOOP);
glVertex2f(0,0);
glVertex2f(2,0);
glVertex2f(2,2);
glVertex2f(1,3);
glVertex2f(0,2);
glEnd();
    
```

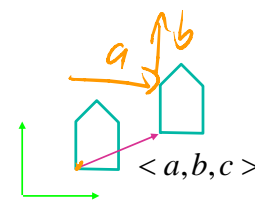
© Michiel van de Panne

# Transformations

## Translation

translate(a,b,c)

*translations*



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

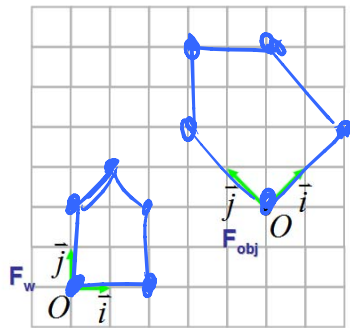
*Handwritten equations:*

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

glTranslatef(a,b,c);  
glTranslated(a,b,c);

*Handwritten equation:*

$$P = M P + T$$



*Handwritten list of points:*

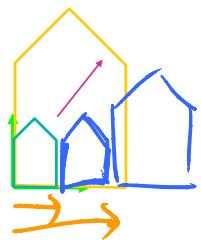
- P(0,0)
- P(2,0)
- P(2,2)
- P(1,3)
- P(0,2)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_w = \begin{bmatrix} 1 & 1 & 5 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

# Transformations

## Scaling

scale(a,b,c)



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

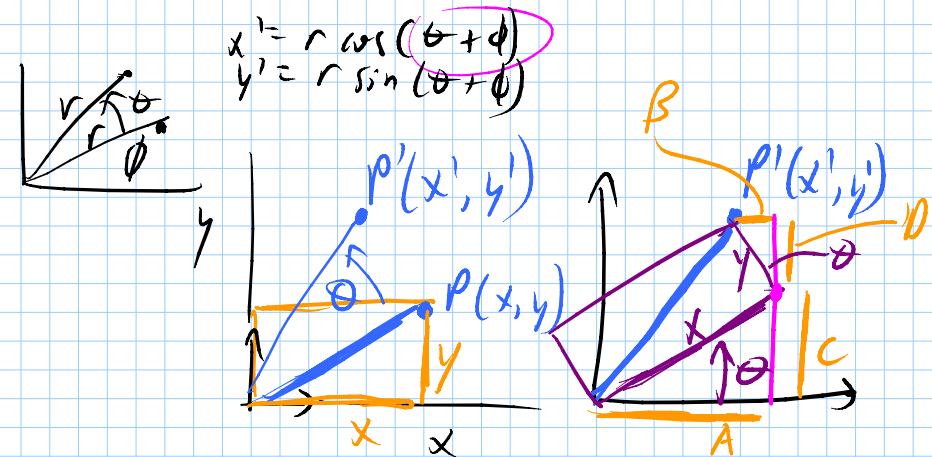
*Handwritten equations:*

$$\begin{aligned} x' &= ax \\ y' &= by \\ z' &= cz \end{aligned}$$

glScalef(a,b,c);  
glScaled(a,b,c);

*Handwritten note:*

$$a, a, a$$



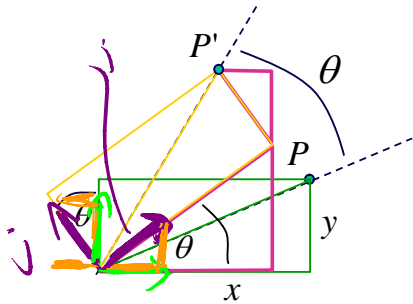
*Handwritten equations for rotation:*

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

$$\begin{cases} x' = A - B \\ y' = C + D \end{cases} \leftarrow \begin{cases} x' = x \cos\theta - y \sin\theta \\ y' = x \sin\theta + y \cos\theta \end{cases}$$

# Transformations

## Rotation



Rotate( $z, \theta$ )

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

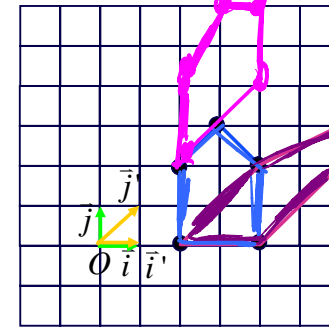
$$z' = z$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

glRotatef(angle,x,y,z);  
glRotated(angle,x,y,z); (30°, 0, 0, 1)

# Transformations

## Shear



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

shear

$$\begin{aligned} x' &= x + y \\ y' &= y \\ z' &= z \end{aligned}$$

$$\begin{aligned} x' &= x \\ y' &= y + x \\ z' &= z \end{aligned}$$

# Transformations

## Affine transformations

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

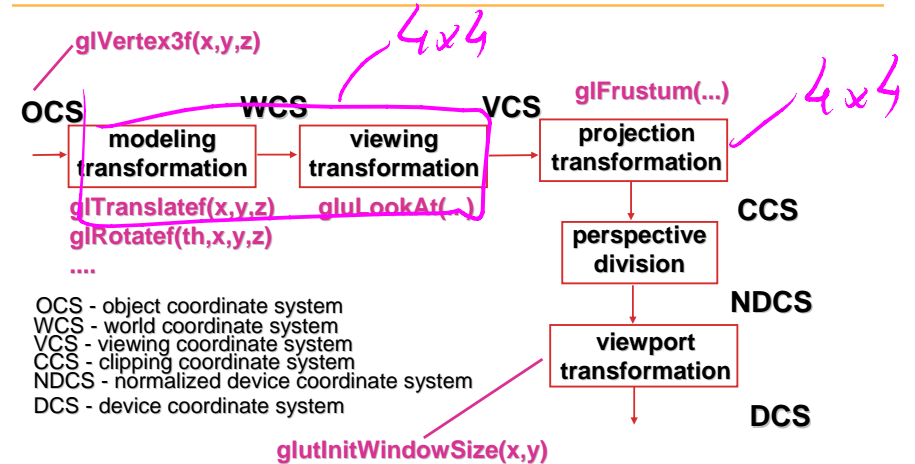
$$P' = M \cdot P + T$$

## 4x4 matrices

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

rotations  
scales  
shears  
translations

# Projective Rendering Pipeline



- 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

