University of British Columbia

Geometric Transformations

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

© MIchiel van de Panne

University of British Columbia

Math Review

matrix vector multiplication

- points as column vectors

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

- points as row vectors

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

© Michiel van de Panne

University of British Columbia

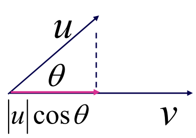
Math Review

dot product

- also called *inner product*

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \bullet \begin{bmatrix} a \\ b \\ c \end{bmatrix} = x*a + y*b + z*c \qquad P \bullet N$$

$$u \bullet v = |u||v|\cos\theta$$

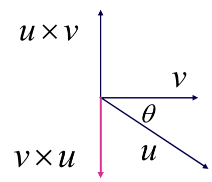


© Michiel van de Panne

University of British Columbia

Math Review

Cross Product



Right Handed Coordinate System

(curl fingers from u to v; thumb points to u x v)

$$|u \times v| = |u||v|\sin\theta$$

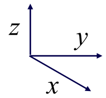
© Michiel van de Panne

University of British Columbia

Math Review

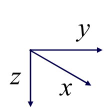
Coordinate Systems

Right-handed Coordinate System



$z = x \times y$
using right-hand rule

Left-handed Coordinate System



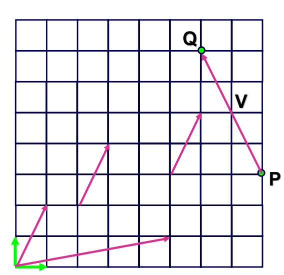
$z = x \times y$
using left-hand rule

© Michiel van de Panne

University of British Columbia

Math Reviw

Points and Vectors



vector space
vectors are invariant under translation

affine space:
allows vector-to-point addition

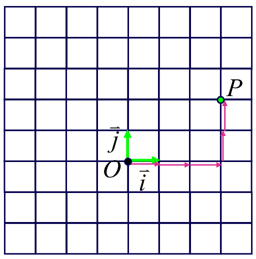
$$P + V = Q$$
$$Q - P = V$$

© Michiel van de Panne

Math Reviw

University of British Columbia

Coordinate System vs Frame



coordinate system: basis vectors
frame: basis vectors + Origin

allows for points

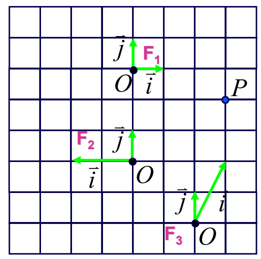
$$P = O + x\vec{i} + y\vec{j}$$

© Michiel van de Panne

Math Review

University of British Columbia

Working with Frames



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

F_1 **P(3,-1)**

F_2 **P(-1.5,2)**

F_3 **P(1,2)**

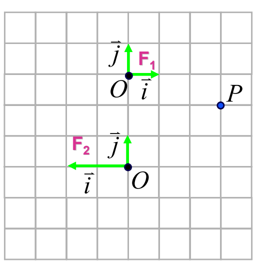
© Michiel van de Panne

Transformations

University of British Columbia

Transformations as a change of frame

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



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

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

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

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

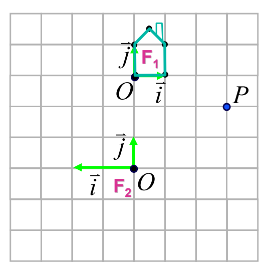
$$P_2 = MP_1$$

© Michiel van de Panne

Transformations

University of British Columbia

change of basis expressed using a matrix



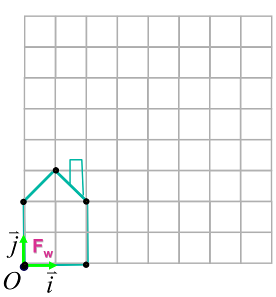
$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$

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

© Michiel van de Panne

Usage of Transformations

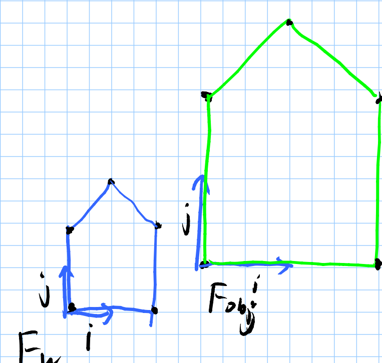
University of British Columbia




```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

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

$$P' = T_{MV} P$$

© Michiel van de Panne



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

$$\cancel{\begin{bmatrix} x_w \\ y_w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_w + x_w \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y_w \begin{bmatrix} 0 \\ 1 \end{bmatrix}}$$

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

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

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

Usage of Transformations

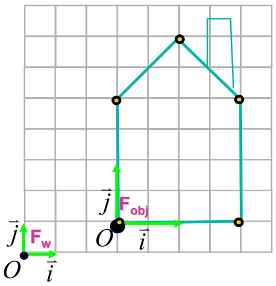
University of British Columbia

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

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

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

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



© Michiel van de Panne

Using Transformations

University of British Columbia

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

2D

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

3D

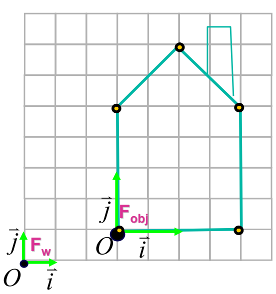
$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$

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

© Michiel van de Panne

Usage of Transformations

University of British Columbia



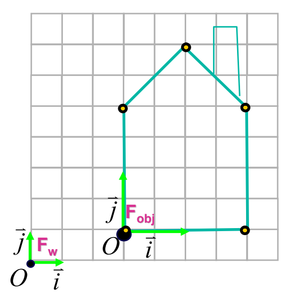

```
GLfloat  T[16] = { ... };
glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(T);

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

University of British Columbia

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();
```

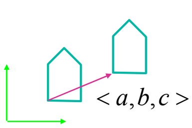
© Michiel van de Panne

Transformations

University of British Columbia

Translation

translate(a,b,c)



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

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

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

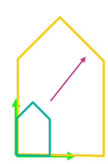
© Michiel van de Panne

Transformations

University of British Columbia

Scaling

scale(a,b,c)

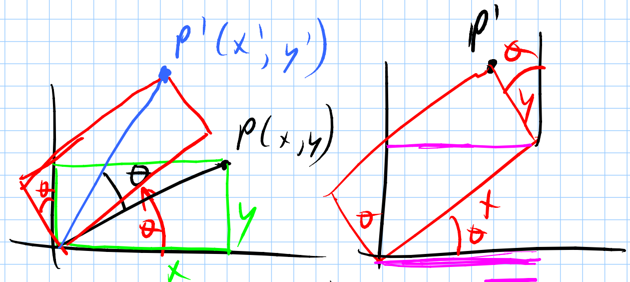


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$x' = x \cdot a$
$y' = y \cdot b$
$z' = z \cdot c$

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

© Michiel van de Panne



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

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

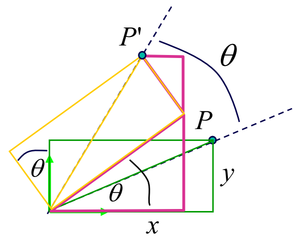
$y' = x\sin\theta + y\cos\theta$

$z' = z$

Transformations

University of British Columbia

Rotation



$Rotate(z,\theta)$

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

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

$z' = z$

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

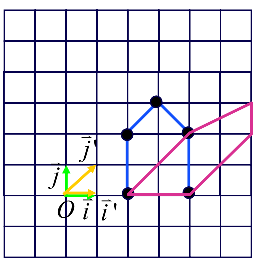
glRotatef(angle,x,y,z);
glRotated(angle,x,y,z);

© Michiel van de Panne

Transformations

University of British Columbia

Shear



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

shear

$x' = x + y$

$y' = y$

$z' = z$

© Michiel van de Panne

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

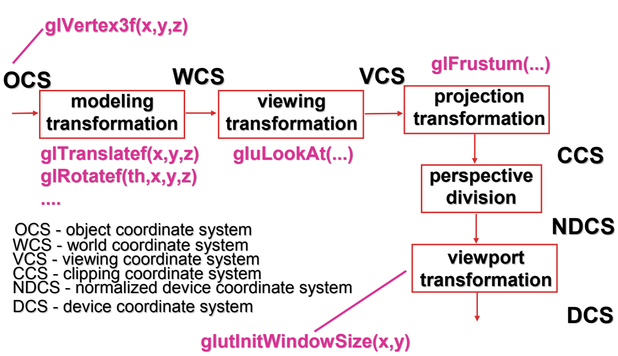
rotations
scales
shear
translate

always (0 0 0 1)

© Michiel van de Panne

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

© Michiel van de Panne