University of British Columbia

Geometric Transformations

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

---

affine transformations
linear transformations

University of British Columbia

Math Review

matrix vector multiplication

- points as column vectors

point in 3D space

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

homogeneous coordinate $=1$

- points as row vectors

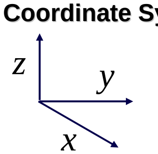
$$\begin{bmatrix} x' & y' & z' & h' \end{bmatrix} = \begin{bmatrix} x & y & z & h \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}^T \qquad P'^T = P^T M^T$$

University of British Columbia

Math Review

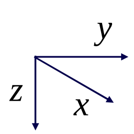
Coordinate Systems

Right-handed Coordinate System



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

Left-handed Coordinate System

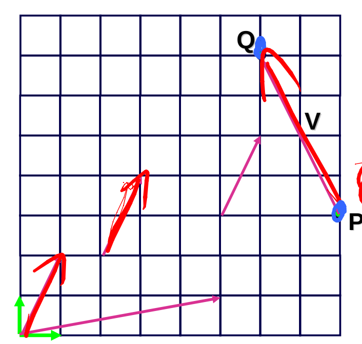


$z = x \times y$
using left-hand rule

University of British Columbia

Math Reviw

Points and Vectors



vector space
vectors are invariant under translation

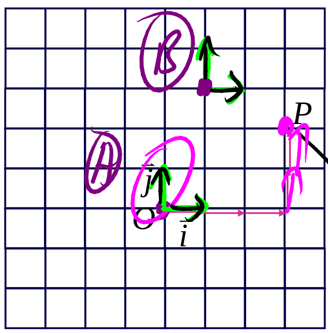
affine space:
allows vector-to-point addition

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

Math Reviw

University of British Columbia

Coordinate System vs Frame



coordinate system: basis vectors
frame: basis vectors + Origin
allows for points

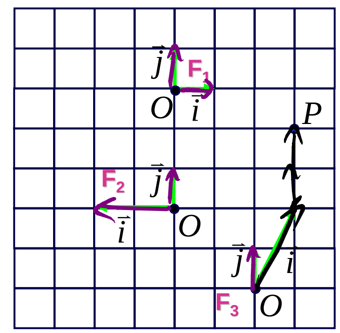
$P = O + x\vec{i} + y\vec{j}$

$P_A(3,2)$ $P_B(2,-1)$

Math Review

University of British Columbia

Working with Frames



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

$P(x_1, y_1)$

F1 P(3,-1)

F2 P(-1.5,2) $O_2 - 1.5\, i_2 + 2\, j_2$

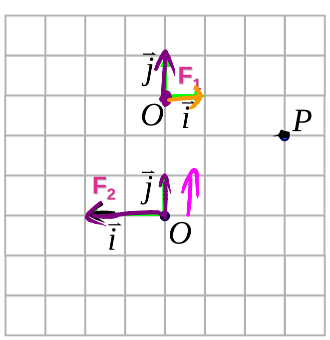
F3 P(1,2)

$P(x_3, y_3)$

Transformations

University of British Columbia

Transformations as a change of frame



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

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

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

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

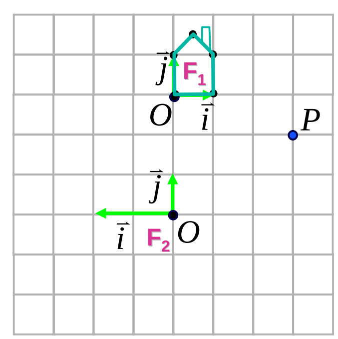
$P_2 = MP_1$

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

Transformations

University of British Columbia

change of basis expressed using a matrix

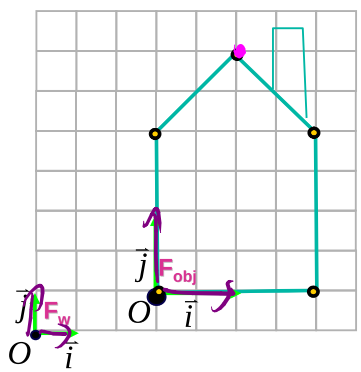


$$\begin{bmatrix}x\\y\end{bmatrix}_2 = \begin{bmatrix}-0.5&0&0\\0&1&3\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}_1$$

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

Usage of Transformations

University of British Columbia



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

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

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

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

(1) 0 0 1

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

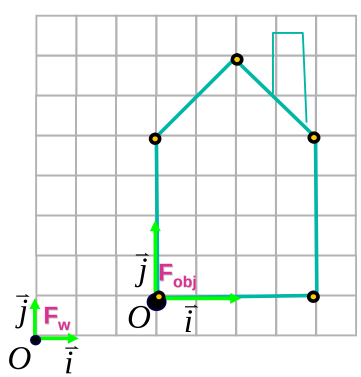
$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_w = \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} \\ \begin{bmatrix} 2 & 0 & 0 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$

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

$\underbrace{2,0,0,0}_{c_1}$, $\underbrace{0,2,0,0}_{c_2}$, $\underbrace{0,0,1,0}_{c_3}$ $\underbrace{3,1,0,1}_{c_4}$

Usage of Transformations

University of British Columbia



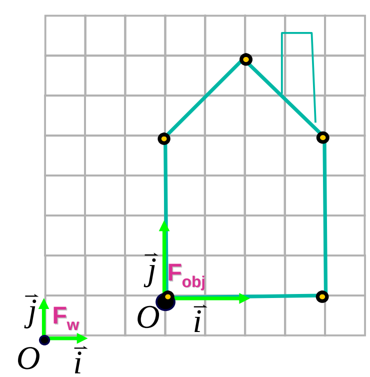

```
GLfloat  T[16] = { ... };
glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(T);

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


Usage of Transformations

University of British Columbia

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

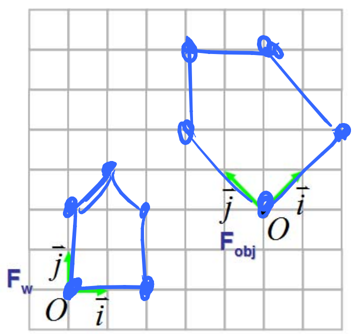



```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

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();
```

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



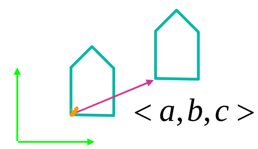
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_w = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

Transformations

University of British Columbia

Translation

translate(a,b,c)



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

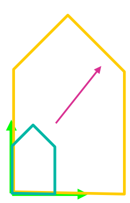
glTranslatef(a,b,c);
glTranslated(a,b,c);

Transformations

University of British Columbia

Scaling

scale(a,b,c)

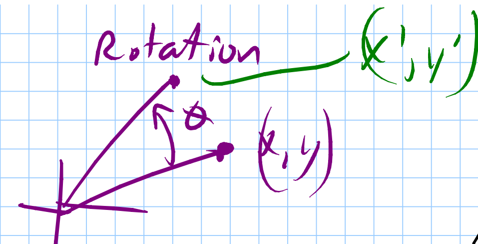


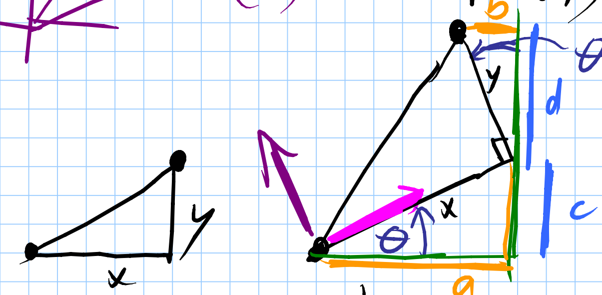
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$x' = ax$
$y' = by$
$z' = cz$

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

Rotation





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

$x' = a - b$

$\boxed{x' = x\cos\theta - y\sin\theta}$

$y' = c + d$

$\boxed{y' = x\sin\theta + y\cos\theta}$

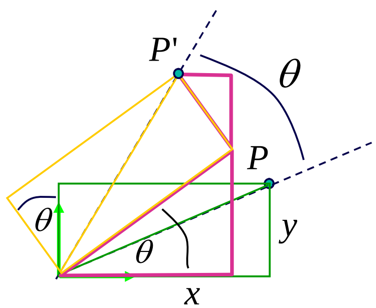
Transformations

University of British Columbia

Rotation

$Rotate(z,\theta)$

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



glRotatef(θ, 0, 0, 1)

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

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

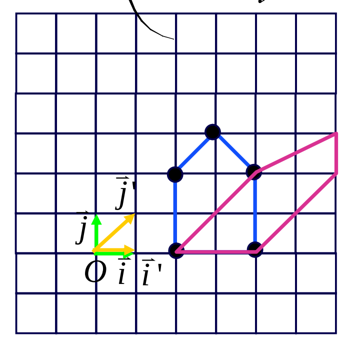
degrees

Transformations

University of British Columbia

Shear

(rarely use this!)



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

shear

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

Projective Rendering Pipeline

University of British Columbia

object CS

glVertex3f(x,y,z)

OCS → **modeling transformation** → **WCS** (world CS) → **viewing transformation** → **VCS** (viewing CS, camera coords) → **projection transformation** → **CCS** → **perspective division** → **NDCS** → **viewport transformation** → **DCS**

glTranslatef(x,y,z)
glRotatef(th,x,y,z)
....

gluLookAt(...)

glFrustum(...)

glutInitWindowSize(x,y)

device coords
screen coords

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