CPSC 436D
Video Game Programming

Transformations
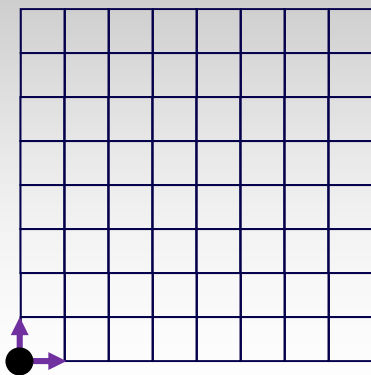

COORDINATE SYSTEMS
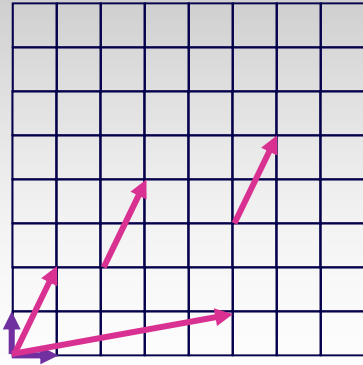
Coordinate system = Origin + Basis Vectors
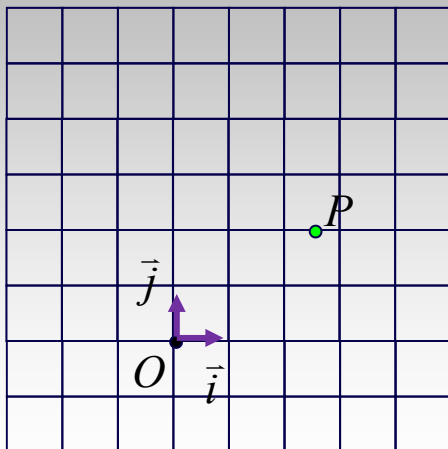
COORDINATE SYSTEMS

Coordinate system = Origin + Basis Vectors

© Alla Sheffer

COORDINATE SYSTEMS

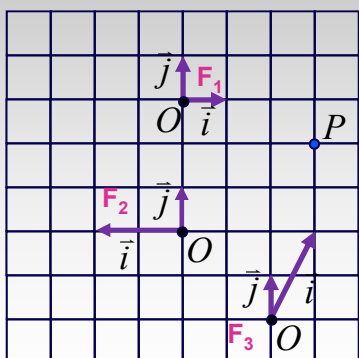$$P = O + x\vec{i} + y\vec{j}$$

equivalent: *P* = (*x, y*)

© Alla Sheffer

COORDINATE SYSTEMS

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

F₁

F₂

F₃
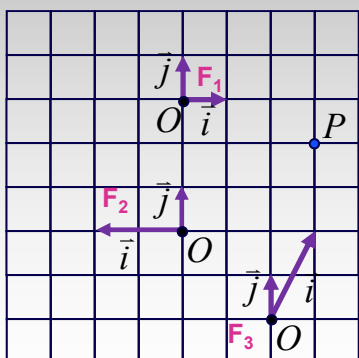
COORDINATE SYSTEMS

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

F₁ **P(3,-1)**

F₂

F₃

COORDINATE SYSTEMS

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

F₁   P(3,-1)

F₂   P(-1.5,2)

F₃
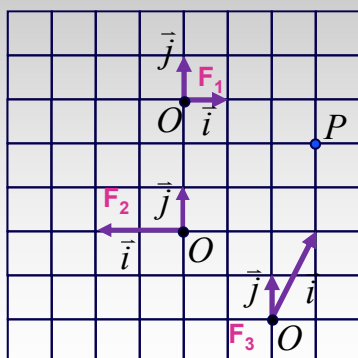
COORDINATE SYSTEMS
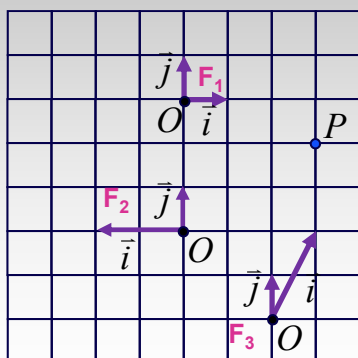
$$P = O + x\vec{i} + y\vec{j}$$

F₁   P(3,-1)

F₂   P(-1.5,2)

F₃   P(1,2)

Transformations

Transformations as a change of frame

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

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_1 + x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix}_1 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_1$$
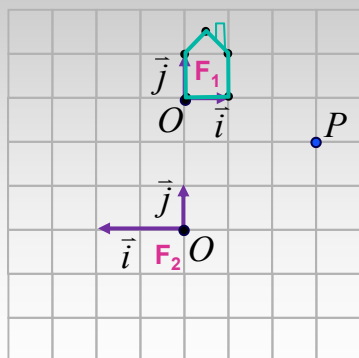
$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}_2 + x_1 \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}_2 + y_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_2$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1$$
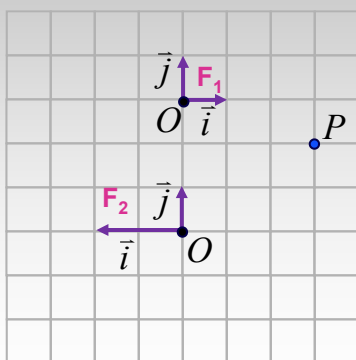
check: P_1(3,-1) becomes P_2(-1.5,2)
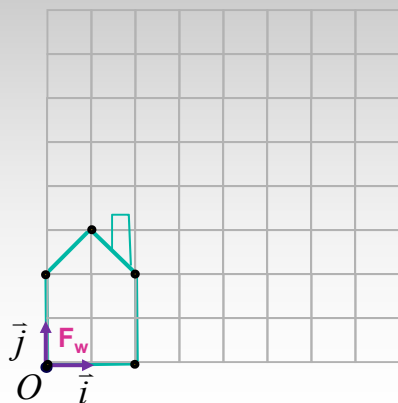
$$P_2 = MP_1$$

TRANSFORMATIONS

change of basis expressed using a matrix

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

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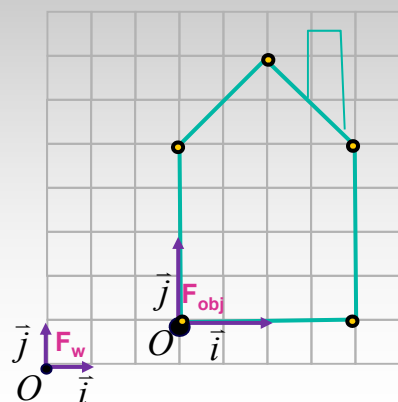
Usage of Transformations

set up the modeling matrix M

for each vertex v
  v' = Mv

© Alla Sheffer

Usage of Transformations

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

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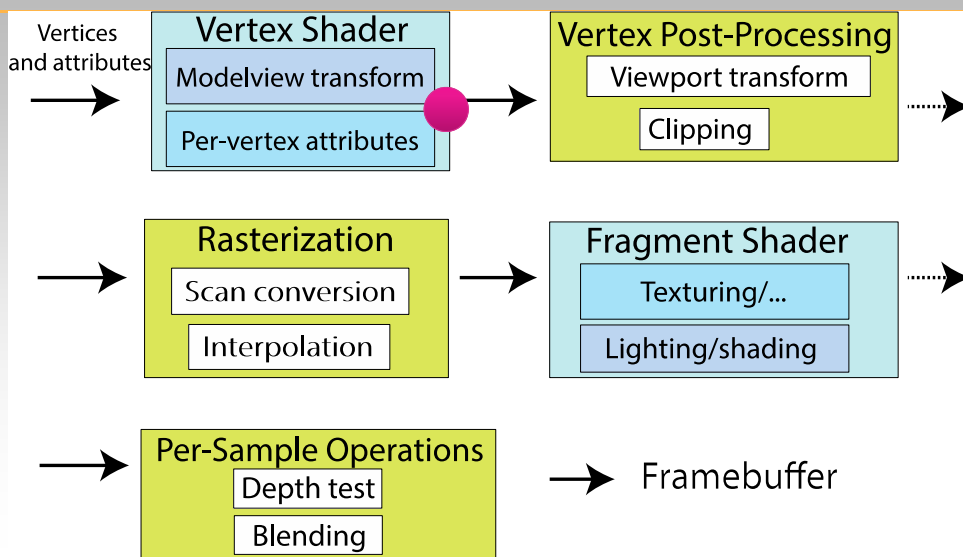
$$\begin{bmatrix} x \\ y \end{bmatrix}_{w} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}_{w} + x_{obj}\begin{bmatrix} 2 \\ 0 \end{bmatrix}_{w} + y_{obj}\begin{bmatrix} 0 \\ 2 \end{bmatrix}_{w}$$

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

© Alla Sheffer

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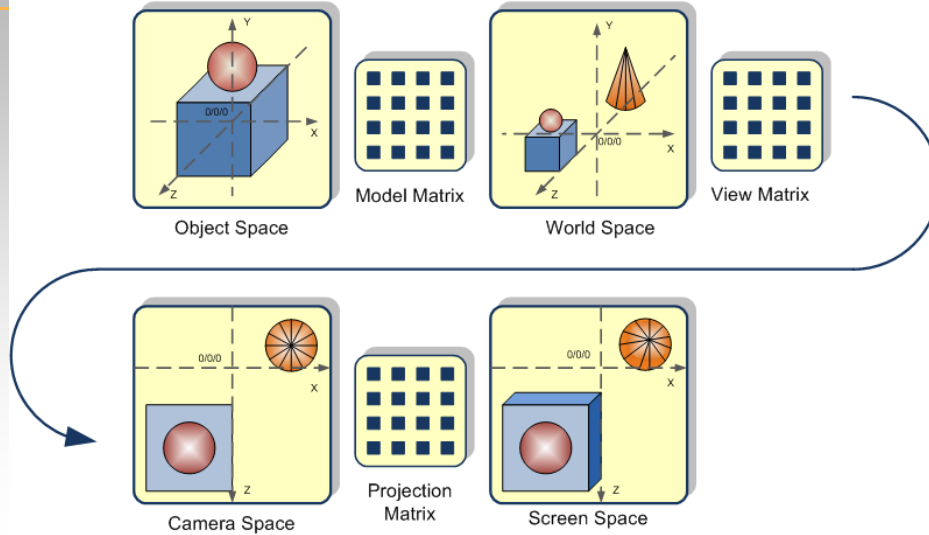
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_w = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{obj}$$

3D →

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

PIPELINE

Vertices and attributes →

Vertex Shader
- Modelview transform
- Per-vertex attributes

→ **Vertex Post-Processing**
- Viewport transform
- Clipping

→ **Rasterization**
- Scan conversion
- Interpolation

→ **Fragment Shader**
- Texturing/…
- Lighting/shading

→ **Per-Sample Operations**
- Depth test
- Blending

→ Framebuffer

coordinate systems

Object Space — Model Matrix — World Space — View Matrix

Camera Space — Projection Matrix — Screen Space

HOW TO TRANSFORM COORDINATES

- Between coordinate frames
- Or animate objects

linear transformations

Translation
Rotation
Scaling
Reflection
Shear

© Alla Sheffer

Matrix representations

Scale:

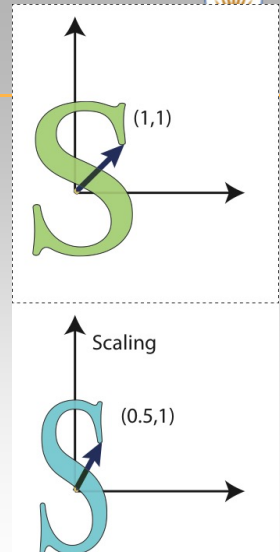(1,1)

Scaling

(0.5,1)
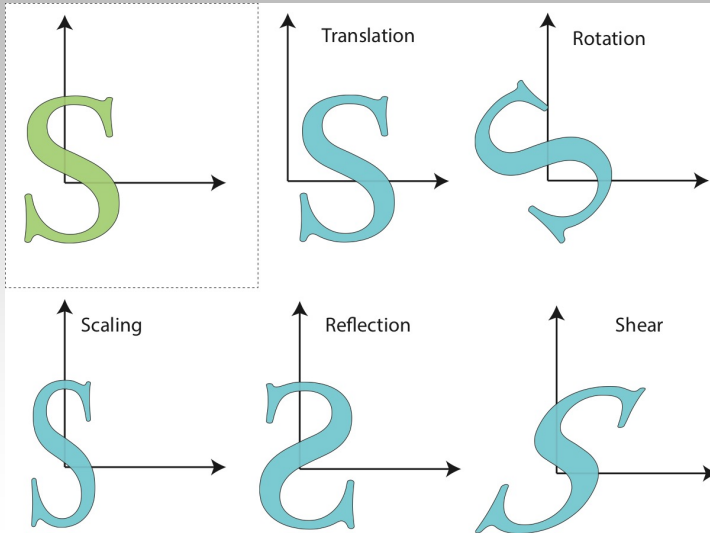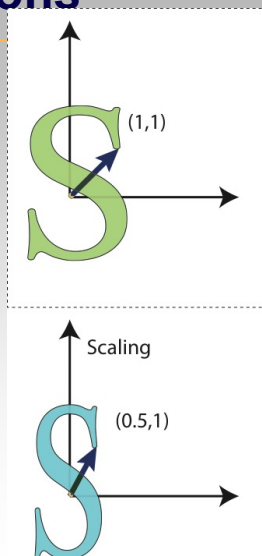
© Alla Sheffer

Matrix representations

Scale:

$$M = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$$
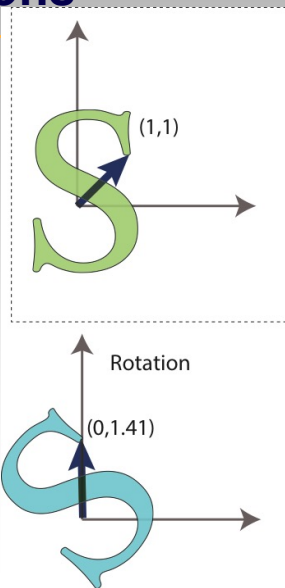
Example:

$$\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \alpha \\ 2\beta \end{pmatrix}$$

(1,1)

Scaling

(0.5,1)

© Alla Sheffer

Matrix representations

Rotation

(1,1)
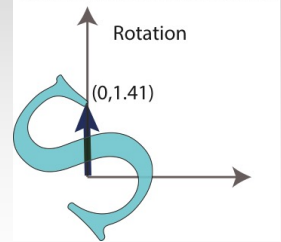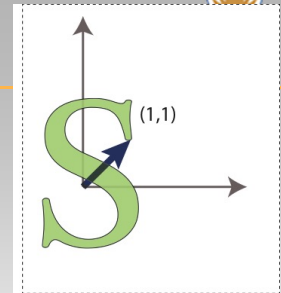
Rotation

(0,1.41)

© Alla Sheffer

Matrix representations

Rotation

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

Example:

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

(1,1)

Rotation

(0,1.41)

© Alla Sheffer

What does this 2D transformation do?

A. Rotates by 90 deg
B. Scales by a factor of 2
C. Rotates by -90 deg
D. Nothing

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

What does this 2D transformation do?

A. Rotates by 90 deg
B. Reflects the object
C. Rotates by -90 deg
D. Scales the object

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

© Alla Sheffer

TRANSLATION

There's a minor glitch.

- Translation: can't be represented as 2x2 matrix multiplication

general transformations

*We need to represent all the
linear transformations + translation.*

Ideas?

$$T(v) = Mv + b$$

AUGMENTED MATRIX

$$M_{2x2} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{bmatrix} M_{2x2} & & 0 \\ & & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Haven't changed much, have we?

AUGMENTED MATRIX

$$\begin{bmatrix} M_{2x2} & & b_x \\ & & b_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' + b_x \\ y' + b_y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} + \boldsymbol{b}$$

Translation

Affine transformations

- Linear (rotation, scaling, shear, reflections) + TRANSLATION

Affine transformations

- Linear (rotation, scaling, shear, reflections) + TRANSLATION

- How to convert a linear transformation matrix into affine matrix?
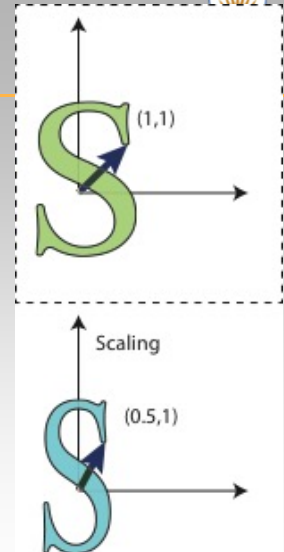
AFFINE Transformations

Scale:

$$M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

$$M = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example:

$$\begin{pmatrix} a \cdot 1 \\ b \cdot 2 \\ 1 \end{pmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

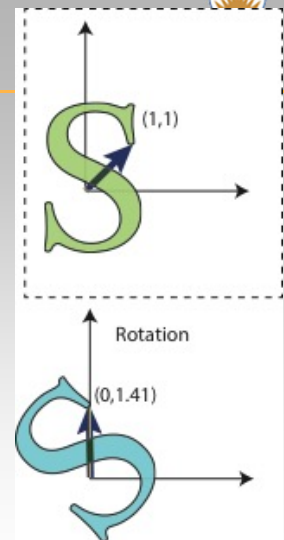© Alla Sheffer

AFFINE Transformations

Rotation

$$M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

$$M = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Example:

$$\begin{pmatrix} a \cdot 1 \\ b \cdot 2 \\ 1 \end{pmatrix} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

© Alla Sheffer

AFFINE Transformations

Translation

$$M = \begin{bmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

Example:

$$\begin{bmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + C_x \\ y + C_y \\ 1 \end{pmatrix}$$