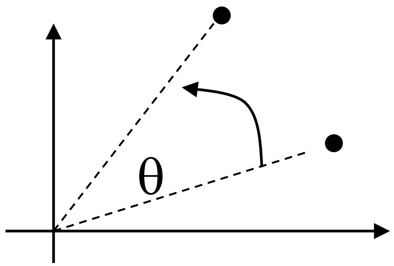
ΗΥ416
ΓΡΑΦΙΚΑ ΥΠΟΛΟΓΙΣΤΩΝ

Γεωμετρικοί Μετασχηματισμοί

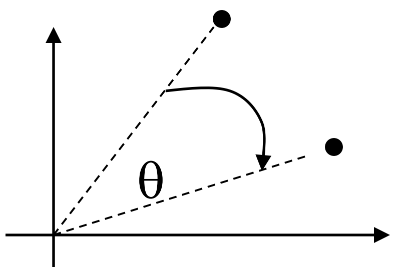
Π. ΤΣΟΜΠΑΝΟΠΟΥΛΟΥ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

2D Rotation

- Default rotation center: Origin (0,0)



θ> 0 : Rotate counter clockwise



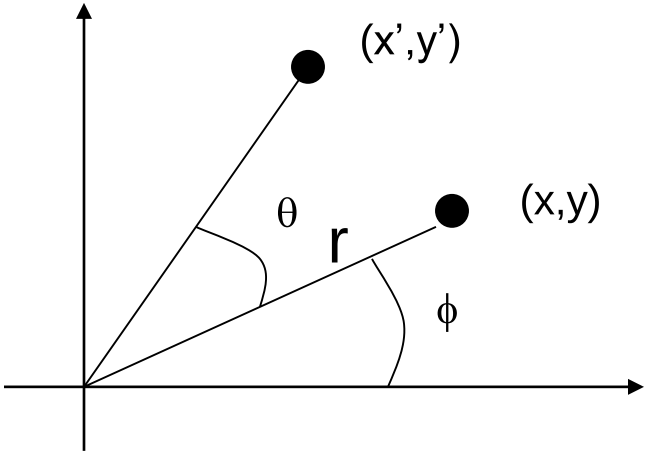
θ< 0 : Rotate clockwise

2D Rotation

(x,y) → *Rotate about the origin* by θ → (x’, y’)

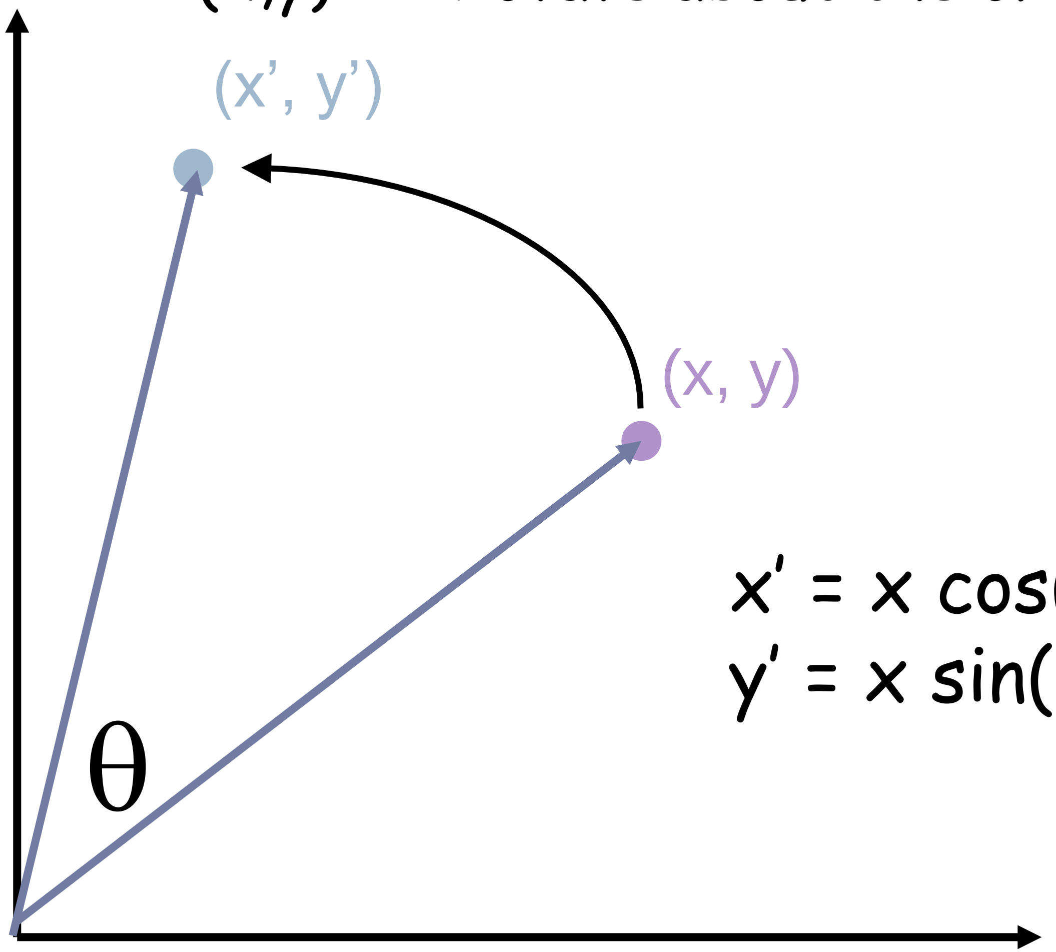
How to compute (x’, y’) ?

$x = r \cos(\varphi)$
$y = r \sin(\varphi)$
$x' = r \cos(\varphi + \theta)$
$y = r \sin(\varphi + \theta)$



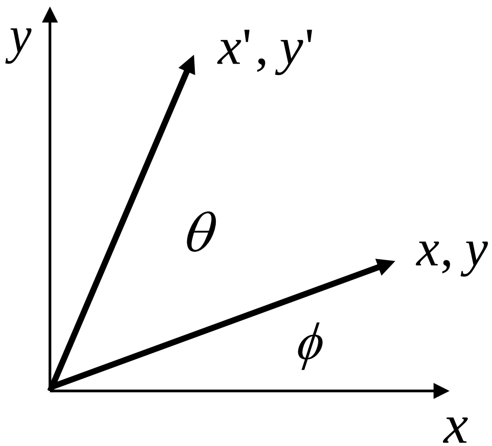
2-D Rotation

(x,y) → Rotate *about the origin* by θ → (x', y')



$$x' = x \cos(\theta) - y \sin(\theta)$$
$$y' = x \sin(\theta) + y \cos(\theta)$$

Rotations (2D)



How to compute (x', y') ?

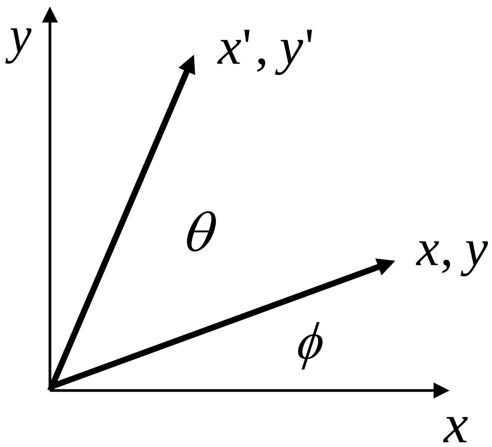
$$x = r\cos\phi$$

$$y = r\sin\phi$$

$$x' = r\cos(\phi + \theta)$$

$$y' = r\sin(\phi + \theta)$$

Rotations (2D)



How to compute (x', y') ?

$$x = r\cos\phi$$

$$y = r\sin\phi$$

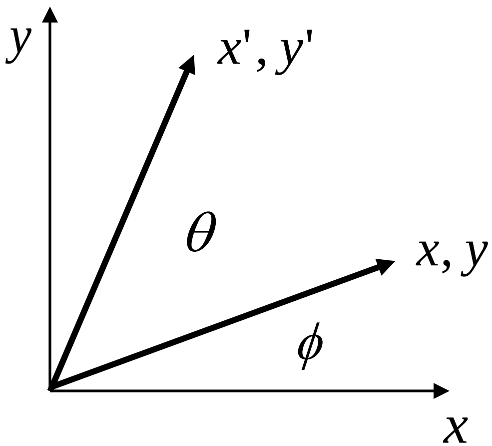
$$x' = r\cos(\phi + \theta)$$

$$y' = r\sin(\phi + \theta)$$

$$\cos(\phi + \theta) = \cos\phi\cos\theta - \sin\phi\sin\theta$$

$$\sin(\phi + \theta) = \cos\phi\sin\theta + \sin\phi\cos\theta$$

Rotations (2D)



How to compute (x′, y′) ?

$$x = r\cos\phi$$
$$y = r\sin\phi$$

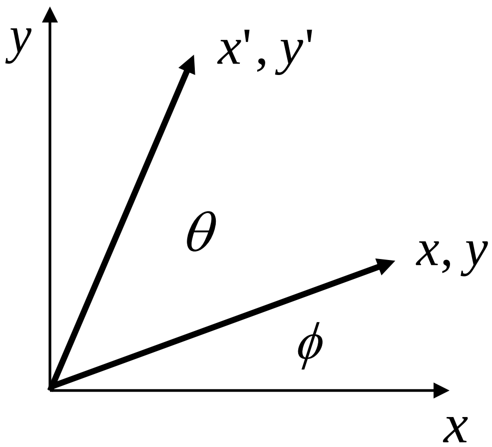
$$x' = r\cos(\phi + \theta)$$
$$y' = r\sin(\phi + \theta)$$

$$\cos(\phi + \theta) = \cos\phi\cos\theta - \sin\phi\sin\theta$$
$$\sin(\phi + \theta) = \cos\phi\sin\theta + \sin\phi\cos\theta$$

$$x' = (r\cos\phi)\cos\theta - (r\sin\phi)\sin\theta$$
$$y' = (r\cos\phi)\sin\theta + (r\sin\phi)\cos\theta$$

Rotations (2D)

How to compute (x', y') ?



$$x = r\cos\phi$$

$$y = r\sin\phi$$

$$x' = r\cos(\phi + \theta)$$

$$y' = r\sin(\phi + \theta)$$

$$\cos(\phi + \theta) = \cos\phi\cos\theta - \sin\phi\sin\theta$$

$$\sin(\phi + \theta) = \cos\phi\sin\theta + \sin\phi\cos\theta$$

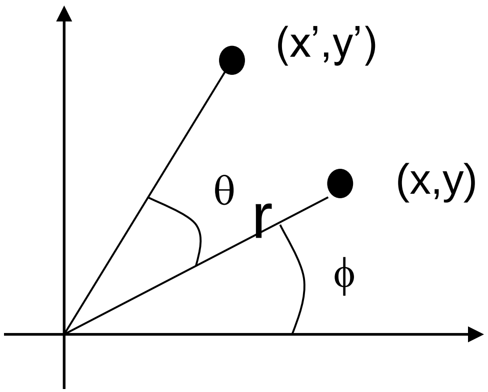
$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}}$$

$$x' = (r\cos\phi)\cos\theta - (r\sin\phi)\sin\theta$$

$$y' = (r\cos\phi)\sin\theta + (r\sin\phi)\cos\theta$$

3x3 2D Rotation Matrix

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix}$$

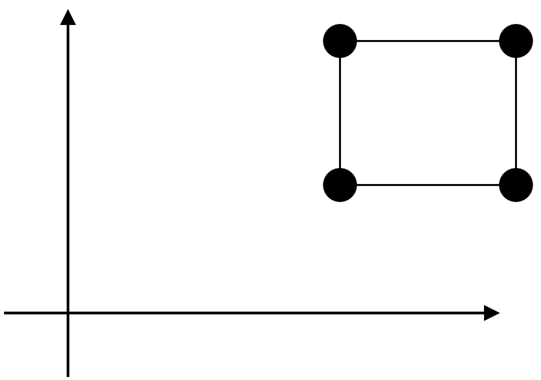


$$\Downarrow$$

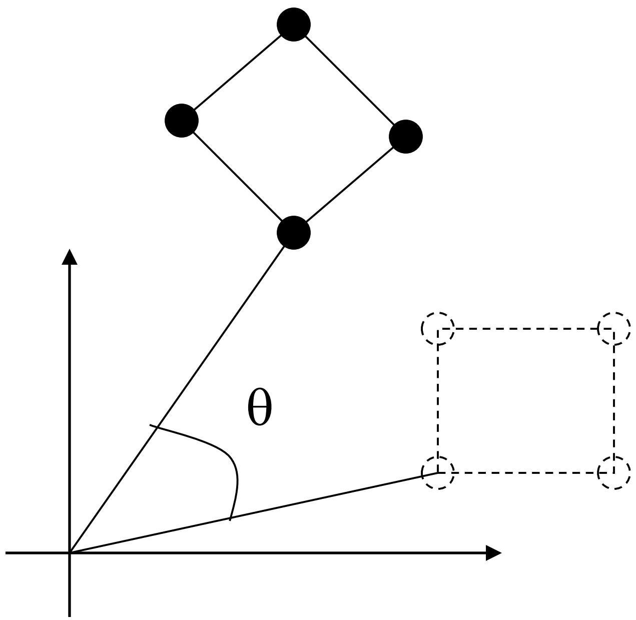
$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

2D Rotation of objects

- How to rotate an object with multiple vertices?



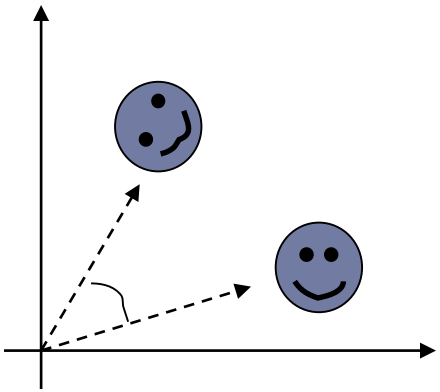
i) Rotate individual Vertices
ii) Redraw the object



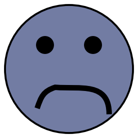
More on Rotation

- The standard rotation matrix is used to rotate about the origin (0,0)

$$\begin{matrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{matrix}$$

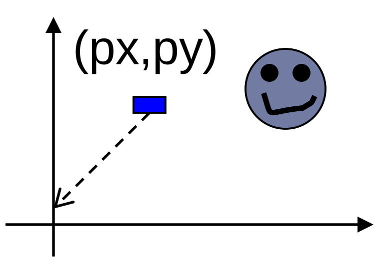


- What if I want to rotate about an arbitrary center?



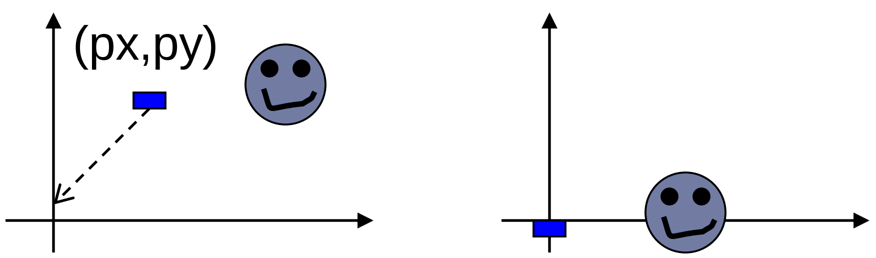
Arbitrary Rotation Center

- To rotate about an arbitrary point P (px,py) by θ:



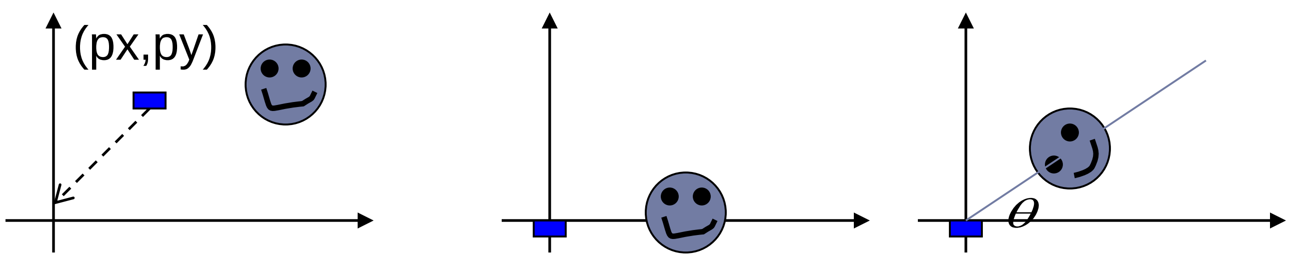
Arbitrary Rotation Center

- To rotate about an arbitrary point P (px,py) by θ:
  - Translate the object so that P will coincide with the origin: T(-px, -py)



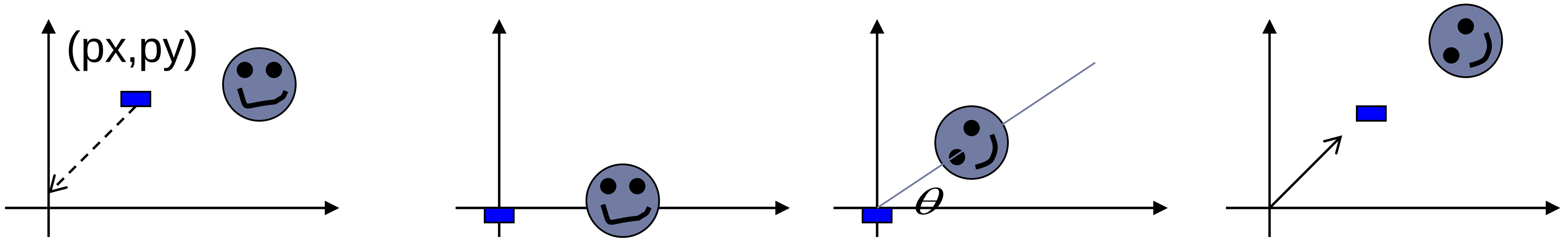
Arbitrary Rotation Center

- To rotate about an arbitrary point P (px,py) by θ:
  - Translate the object so that P will coincide with the origin: T(-px, -py)
  - Rotate the object: R(θ)



Arbitrary Rotation Center

- To rotate about an arbitrary point P (px,py) by θ:
  - Translate the object so that P will coincide with the origin: T(-px, -py)
  - Rotate the object: $R(\theta)$
  - Translate the object back: T(px,py)



Arbitrary Rotation Center

- Translate the object so that P will coincide with the origin: T(-px, -py)
- Rotate the object: R(q)
- Translate the object back: T(px,py)

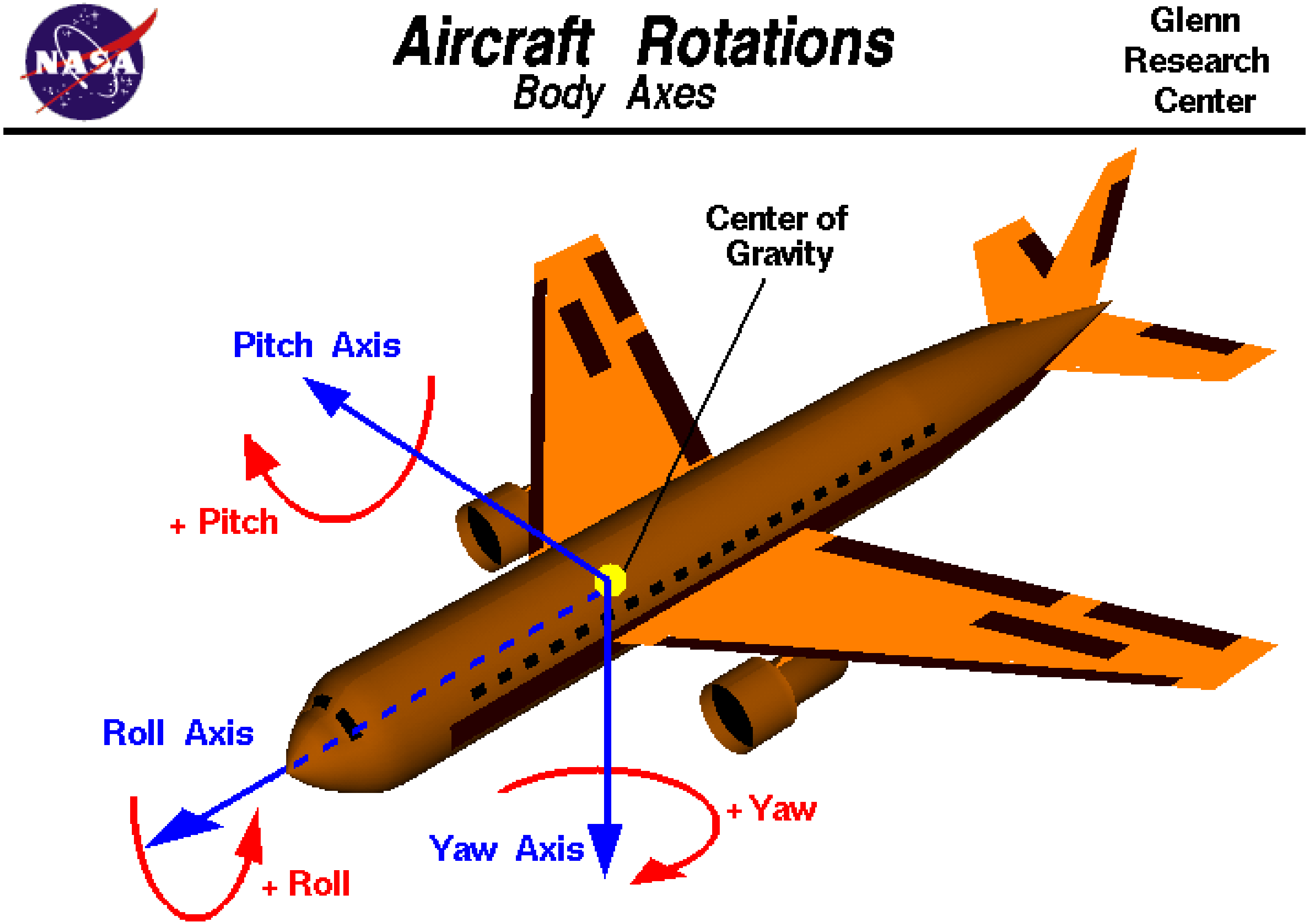
- Put in matrix form: T(px,py) R(q) T(-px, -py) * P

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & px \\ 0 & 1 & py \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & -px \\ 0 & 1 & -py \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

3-D Rotation

- 3-D is more complicated
  - Need to specify an *axis of rotation*
  - Simple cases: rotation about X, Y, Z axes

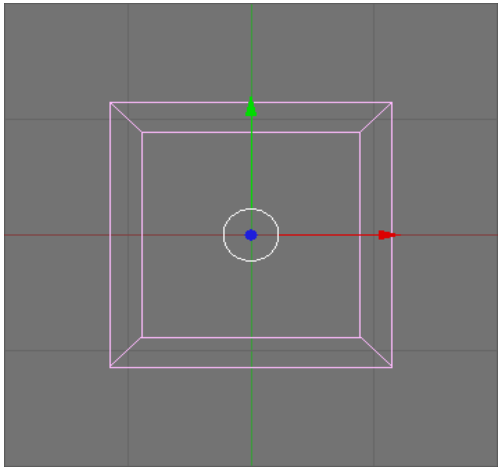
Rotation example: airplane



3-D Rotation

- *What does the 3-D rotation matrix look like for a rotation about the Z-axis?*
  - Build it coordinate-by-coordinate

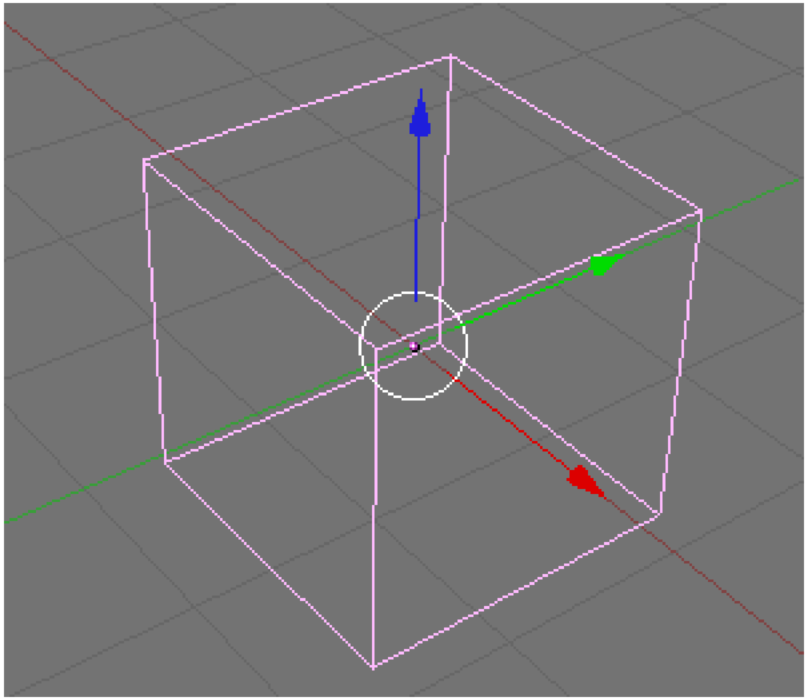
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

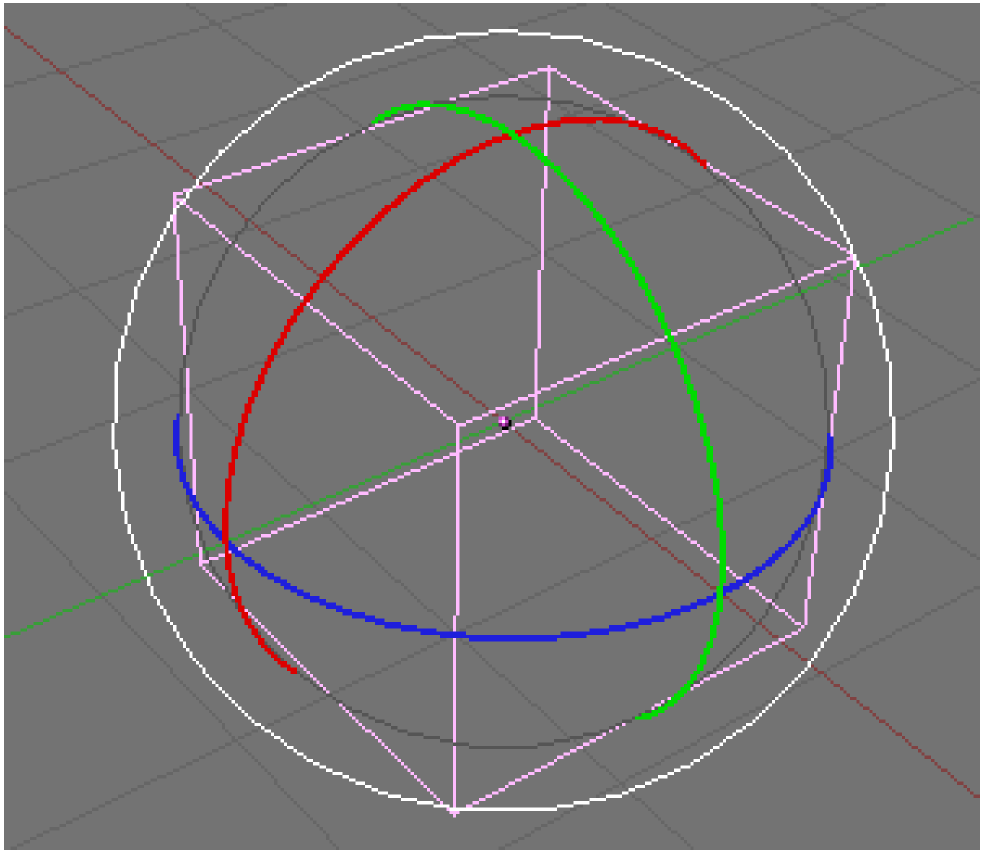


- 2-D rotation from last slide:

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

Rotations about the axes





3-D Rotation

- *What does the 3-D rotation matrix look like for a rotation about the* Y*-axis?*
  - Build it coordinate-by-coordinate

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

3-D Rotation

- *What does the 3-D rotation matrix look like for a rotation about the X-axis?*
  - Build it coordinate-by-coordinate

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

Rotations (3D)

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties of Rotations

$$R_a(0) = I$$

$$R_a(\theta)R_a(\phi) = R_a(\phi + \theta)$$

$$R_a(\theta)R_a(\phi) = R_a(\phi)R_a(\theta)$$

$$R_a^{-1}(\theta) = R_a(-\theta) = R_a^{T}(\theta)$$

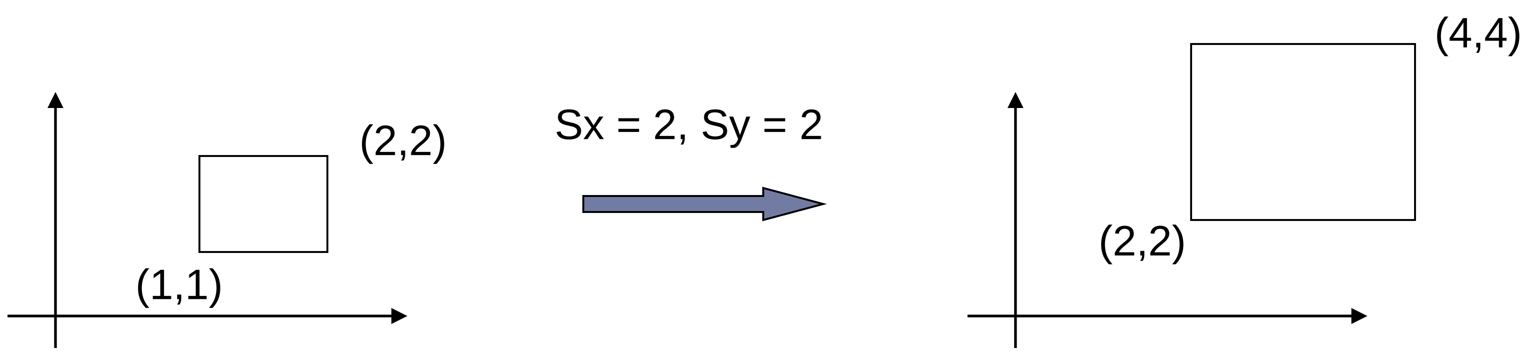
$$R_a(\theta)R_b(\phi) \neq R_b(\phi)R_a(\theta)$$ order matters!

2D Scaling

Scale: Alter the size of an object by a scaling factor (Sx, Sy), i.e.

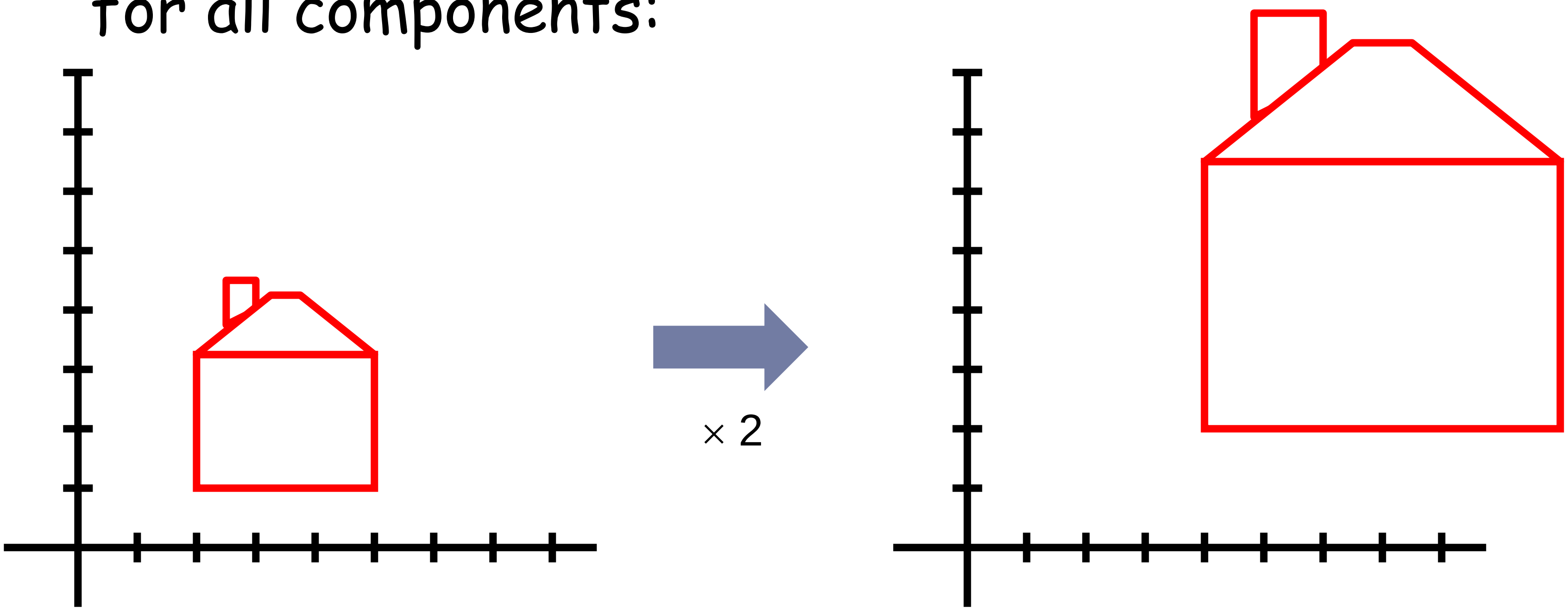
x' = x . Sx
y' = y . Sy

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} Sx & 0 \\ 0 & Sy \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix}$$

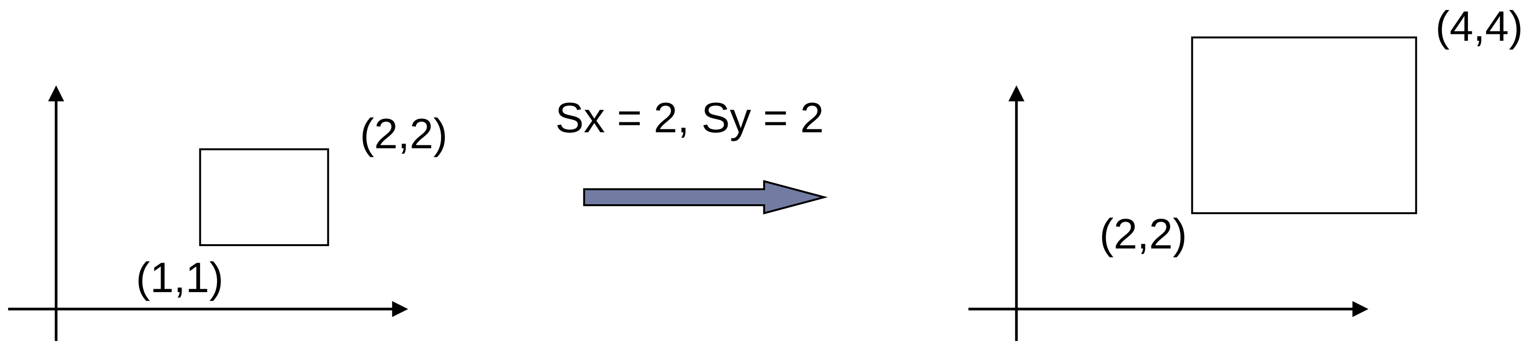


Scaling

- *Scaling* a coordinate means multiplying each of its components by a scalar
- *Uniform scaling* means this scalar is the same for all components:



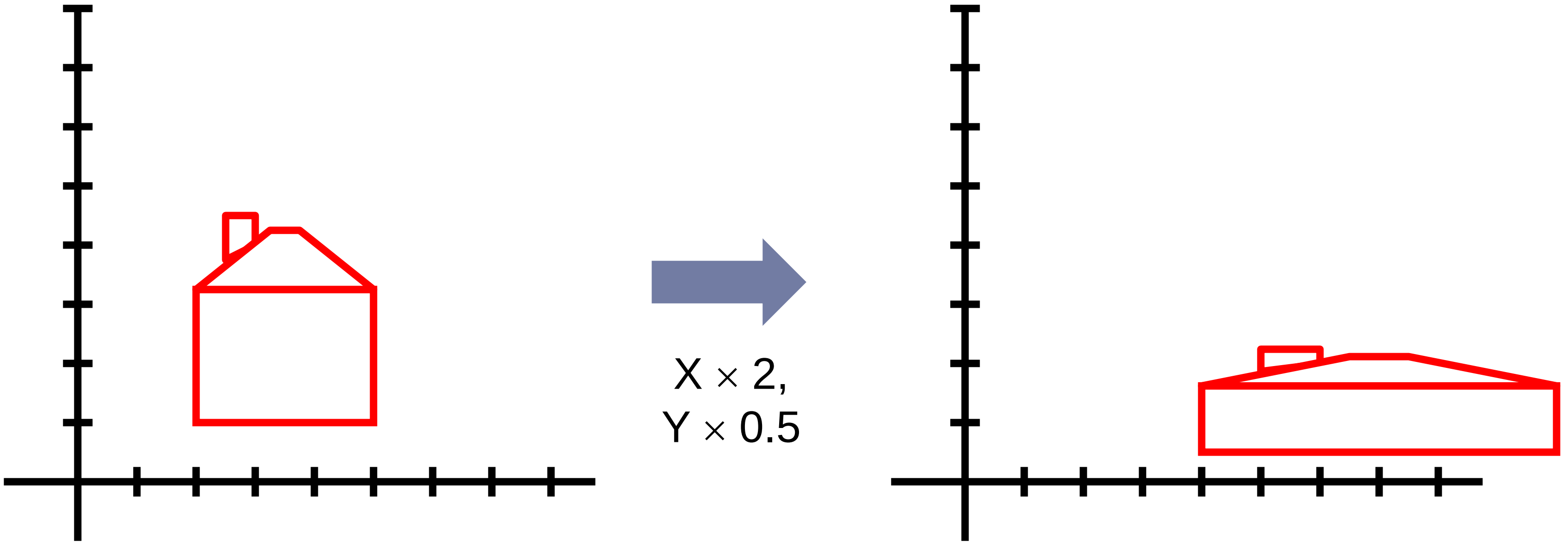
2D Scaling of objects



- Not only the object size is changed, it also moved!!
- Usually this is an undesirable effect
- We will discuss later (soon) how to fix it

Scaling

- *Non-uniform scaling*: different scalars per component:



- *How can we represent this in matrix form?*

3x3 2D Scaling Matrix

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} Sx & 0 \\ 0 & Sy \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix}$$

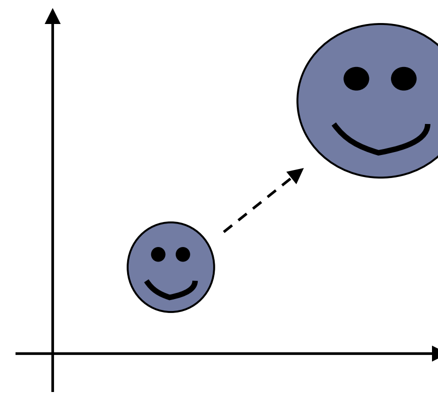
$$\Downarrow$$

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

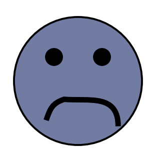
More on Scaling

- The standard scaling matrix will only anchor at (0,0)

$$\begin{matrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{matrix}$$

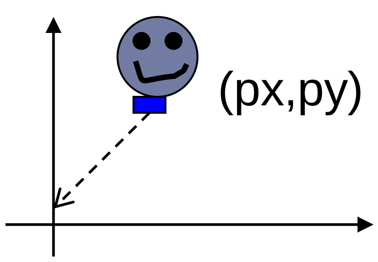


- What if I want to scale about an arbitrary pivot point?



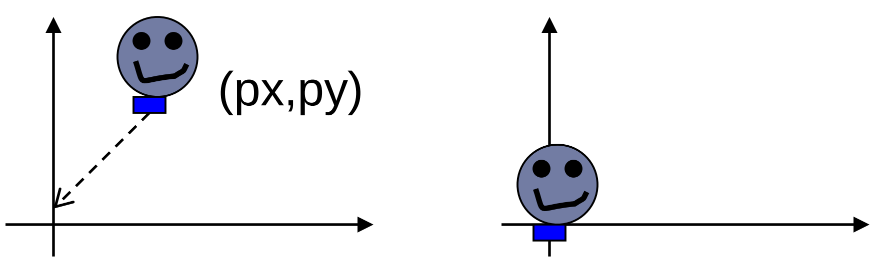
Arbitrary Scaling Pivot

- To scale about an arbitrary fixed point P (px,py):



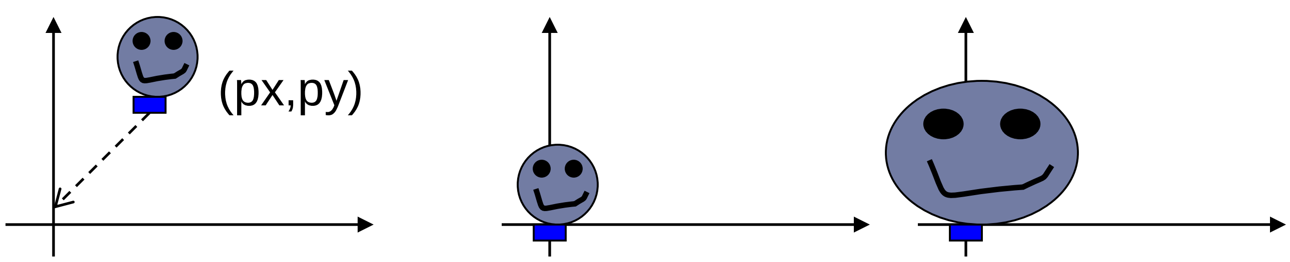
Arbitrary Scaling Pivot

- To scale about an arbitrary fixed point P (px,py):
  - Translate the object so that P will coincide with the origin: T(-px, -py)



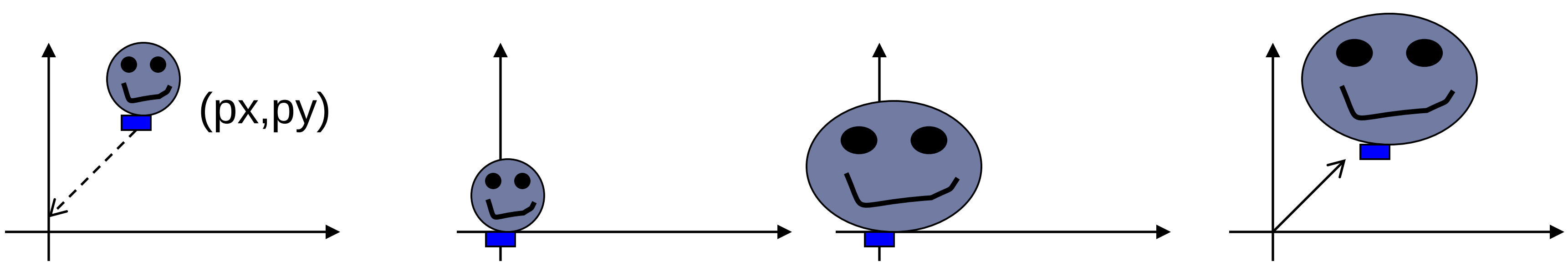
Arbitrary Scaling Pivot

- To scale about an arbitrary fixed point P (px,py):
  - Translate the object so that P will coincide with the origin: T(-px, -py)
  - Scale the object: S(sx, sy)



Arbitrary Scaling Pivot

- To scale about an arbitrary fixed point P (px,py):
  - Translate the object so that P will coincide with the origin: T(-px, -py)
  - Scale the object: S(sx, sy)
  - Translate the object back: T(px,py)



3-D Scaling

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \end{bmatrix}$$

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

Uniform scaling *iff* $s_x = s_y = s_z$

Scaling

- Scaling operation:

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

- Or, in matrix form:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}}_{\text{scaling matrix}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Put it all together in 2D cases

- Translation:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} t_x \\ t_y \end{vmatrix}$$

- Rotation:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix} \cdot \begin{vmatrix} t_x \\ t_y \end{vmatrix}$$

- Scaling:

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} s_x & 0 \\ 0 & s_y \end{vmatrix} \cdot \begin{vmatrix} t_x \\ t_y \end{vmatrix}$$

Or, 3x3 Matrix Representations

- Translation:

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- Rotation:

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- Scaling:

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$