Computer Graphics
2D Transformations

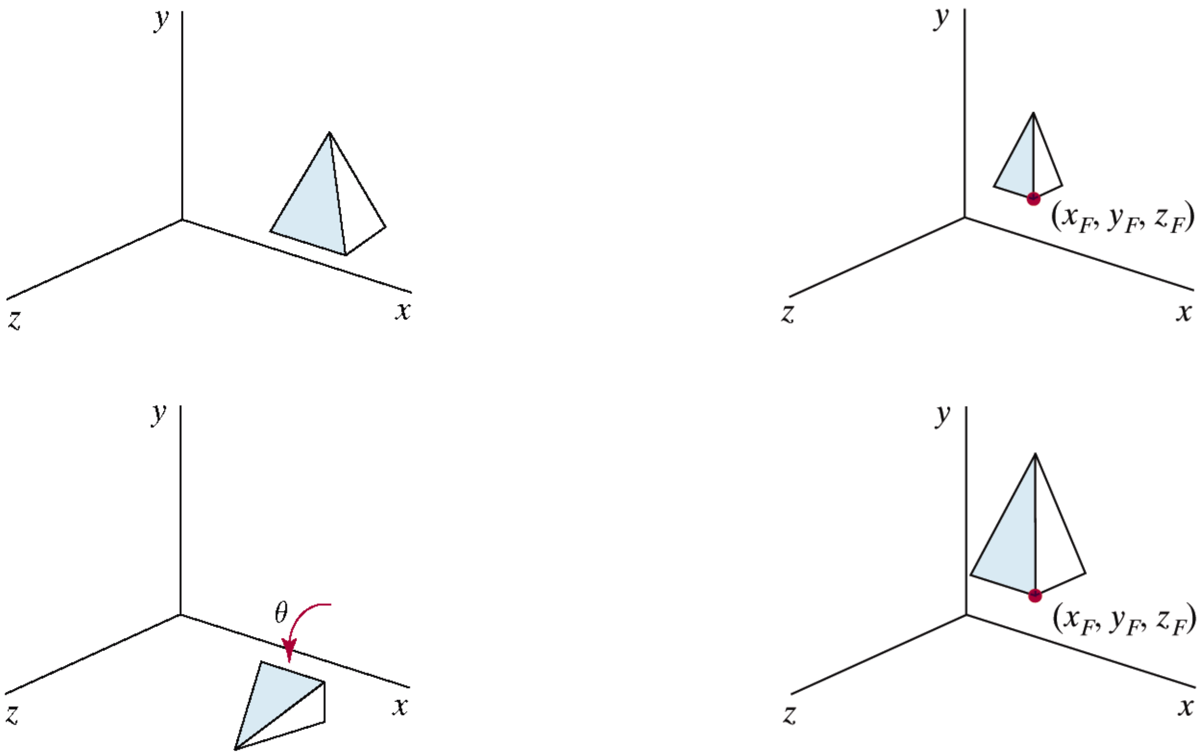
Contents

In today's lecture we'll cover the following:

- Why transformations
- Transformations
  - Translation
  - Scaling
  - Rotation
- Homogeneous coordinates
- Matrix multiplications
- Combining transformations

Why Transformations?

In graphics, once we have an object described, transformations are used to move that object, scale it and rotate it



Images taken from Hearn & Baker, "Computer Graphics with OpenGL" (2004)

Translation

- A **translation** is applied to an object by repositioning it along a straight-line path from one coordinate location to another.

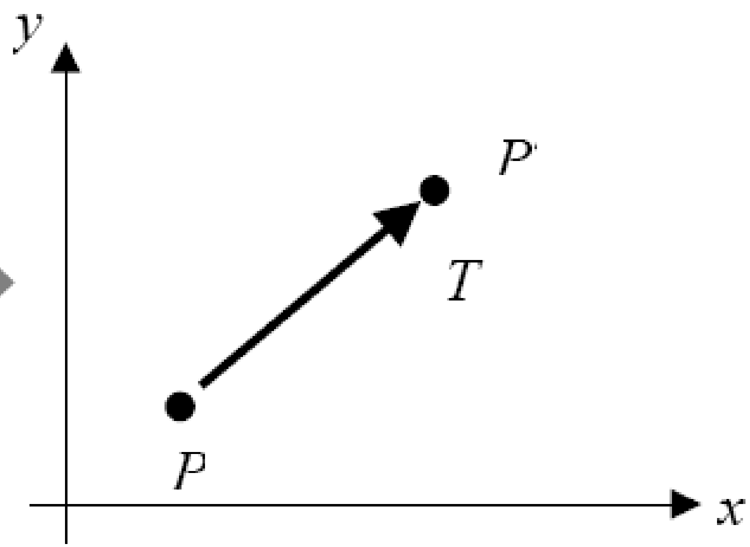
- We translate a two-dimensional point by adding **translation distances**, t_x and t_y , to the original coordinate position (x,y) to move the point to a new position (x', y')

$$x' = x + t_x \ , \ y' = y + t_y \qquad (1)$$

Translation

- The translation distance pair (t_x , t_y) is called a **translation vector** or **shift vector**

Fig. 1: *Translating a point from position P to position P′ with translation vector T.*



Translation

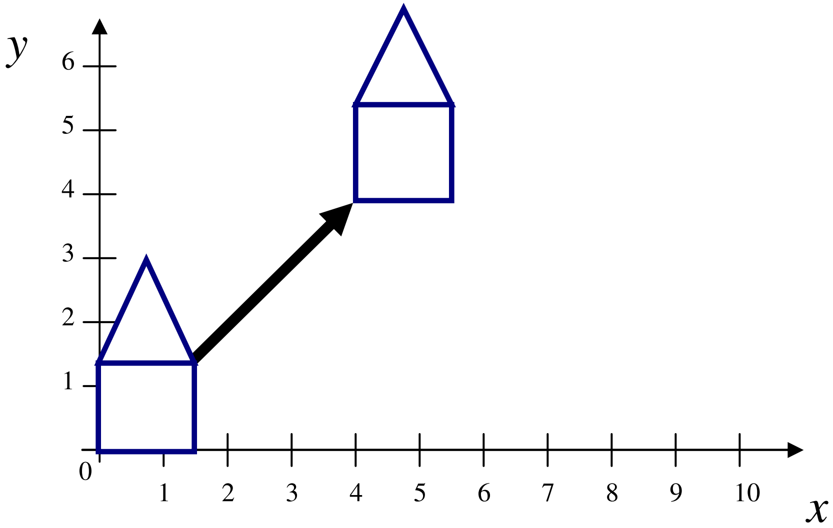
- This allows us to write the two-dimensional translation equations in the matrix form:

$$P' = P + T \qquad (2)$$

- where

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (3)$$

Example



Note: House shifts position relative to origin

Scaling

- A **scaling** transformation alters the size of an object.
- This operation can be carried out for polygons by multiplying the coordinate values (*x*,*y*) of each vertex by **scaling factors** s_x , s_y to produce the transformed coordinates (*x*′, *y*′):

$$x' = x \cdot s_x, \quad y' = y \cdot s_y. \qquad (10)$$

Scaling

•**Transformation Equations** (Matrix Form):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}. \qquad (11)$$

•Or

$$P' = S \,.\, P, \qquad (12)$$

where *S* is the 2 by 2 scaling matrix.

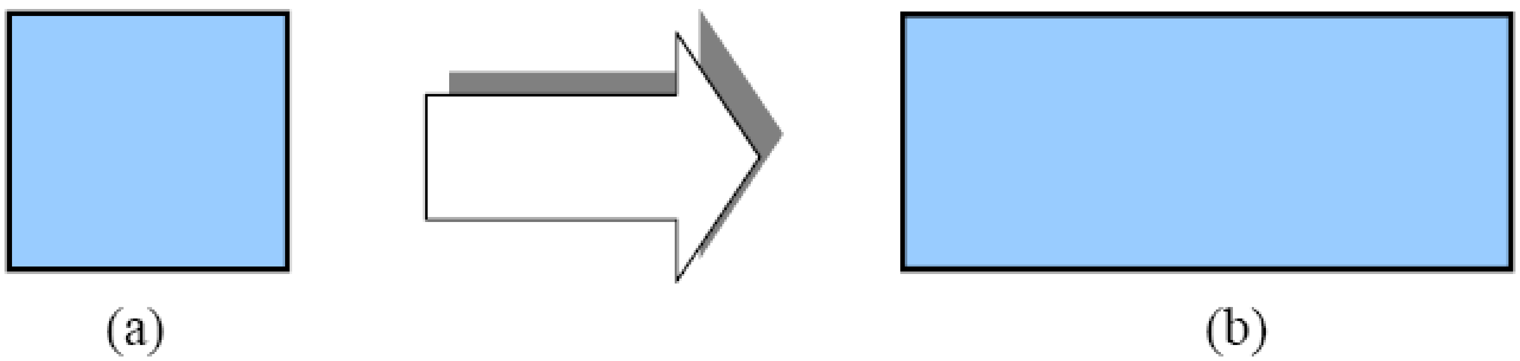
Scaling

Remarks:

1. When s_x and s_y are assigned the same value, a **uniform scaling** is produced that maintains relative object proportions.
2. Unequal values for s_x and s_y result in a **differential scaling** that is often used in design applications, where pictures are constructed from a few basic shapes that can be adjusted by scaling and positioning transformations.

Example

•Turning a square (a) into a rectangle (b) with scaling factors $s_x = 2$ and $s_y = 1$.



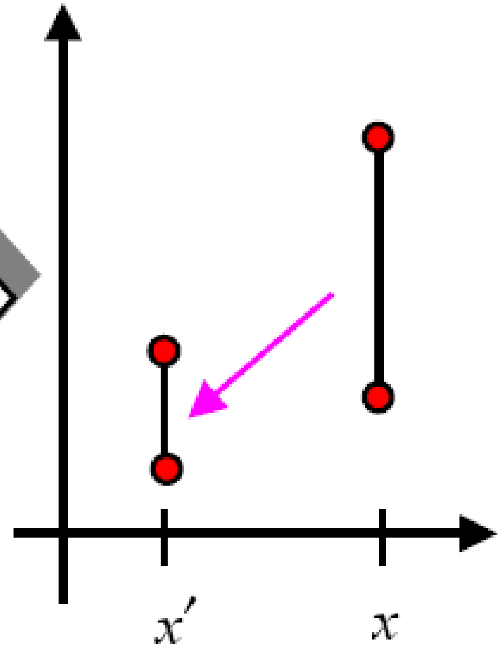
Scaling

• When the scaling factors s_x and s_y are assigned the values less than 1, the objects move closer to the coordinate origin.

• Similarly, the values of s_x and s_y greater than 1, move coordinate positions farther from the origin.

Example

Fig 7: *A line scaled with Eqn. (12) using* $s_x = s_y = 0.5$ *is reduced in size and moved closer to the coordinate origin*



Scaling

- When the scaling factors s_x and s_y are assigned the values less than 1, the objects reduce in size.

- Similarly, the values of s_x and s_y greater than 1, enlarge the object size (see Fig. 7).

- **Warning:** Negative values of s_x and s_y are not permissible.

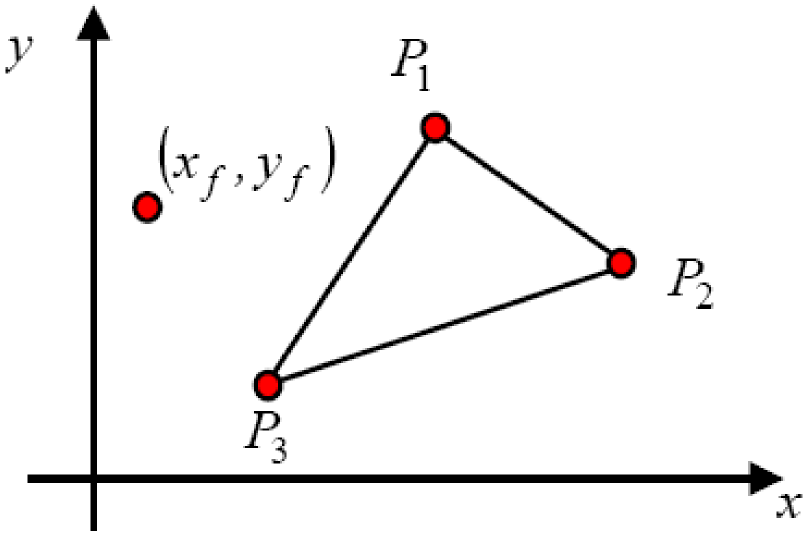
Scaling

•We can control the location of a scaled object by choosing a position, called the **fixed point**, that is to remain unchanged after the scaling transformation.

•Coordinates for the fixed point (x_f , y_f)can be chosen as one of the vertices, the object centroid , or any other position see next Figure.

Example

Fig 8: *Scaling relative to a chosen fixed point (x_f, y_f). Distances from each polygon vertex to fixed point are scaled by transformation equations.*



Scaling

Transformation equations, with the fixed point (x_f , y_f), are calculated as:

$$x' = x_f + (x - x_f) s_x \ , \quad y' = y_f + (y - y_f) s_y \qquad (13)$$

We can rewrite these scaling transformations to separate the multiplicative and the additive terms as follows:

$$\begin{aligned} x' &= x \cdot s_x + x_f \ (1 - s_x) \\ y' &= y \cdot s_y + y_f \ (1 - s_y) \end{aligned} \qquad (14)$$

Scaling

•The additive terms x_f (1 - s_x) and y_f (1-s_y) are constant for all points in the object.

•Polygons are scaled by applying transformations (14) to each vertex and then regenerating the polygon using the transformed vertices.

•Other objects are scaled by applying transformations (14) to the parameters defining the objects.

•For example, an ellipse in the standard position is resized by scaling the semi-major and semi-minor axes and redrawing the ellipse about the designated center coordinates.

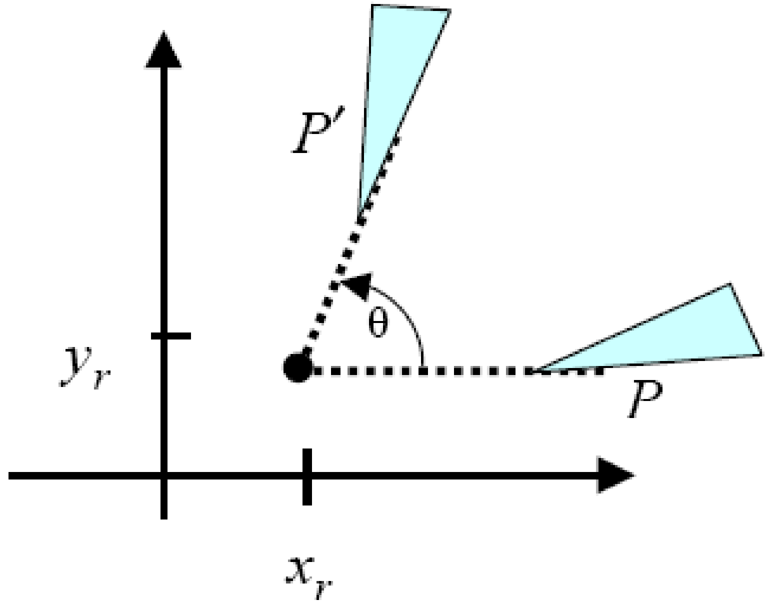
Rotation

- The two-dimensional **rotation** is applied to an object by repositioning it along a circular path in the *x-y* plane.

- To generate a rotation, we specify a **rotation angle** θ and the position (x_r, y_r) of the **rotation point** (or **pivot point**) about which object is rotated as shown in the Figure.

Rotation

Fig. 3: *Rotation of an object through angle* θ *about the pivot point* (x_r, y_r).

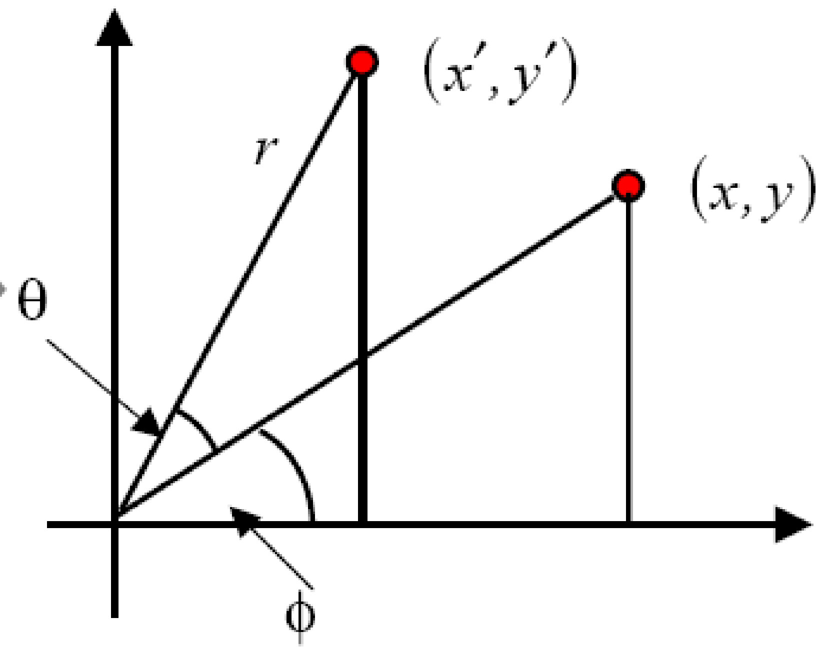


•This transformation can also be described as a rotation about a **rotation axis** that is perpendicular to the x-y plane and passes through the pivot point.

Rotation

- **Rotation about the Origin:**

Fig. 4: *Rotation of a point from position (x,y) to position (x', y') through angle θ relative to the coordinate origin. The original angle of displacement of the point from the x axis is φ.*



Rotation

- **In the previous Figure:**
- r – is the constant distance of the point from the origin.
- ϕ is the original angular position of the point from th horizontal
- θ is the rotation angle.
- Using the standard trigonometric identities, we can express the transformed coordinates in terms of the two angles as

$$\left.\begin{aligned} x &= r\,cos(\phi+\theta) = r\,cos\,\phi\,cos\,\theta - r\,sin\,\phi\,sin\,\theta \\ y &= r\,sin(\phi+\theta) = r\,cos\,\phi\,sin\,\theta + r\,sin\,\phi\,cos\,\theta \end{aligned}\right\} \quad (4)$$

Rotation

The original coordinates of the point in polar coordinates are:

$$x = r \cos \varphi,$$
$$y = r \sin \varphi \qquad (5)$$

- Substituting expressions (5) into (4) , we obtain the transformation equations for rotating a point at position (x, y) through and an angle θ about the origin:

$$x' = x \cos \theta - y \sin \theta,$$
$$y' = x \sin \theta + y \cos \theta \qquad (6)$$

Rotation

- Rotation Equations in the Matrix Form:

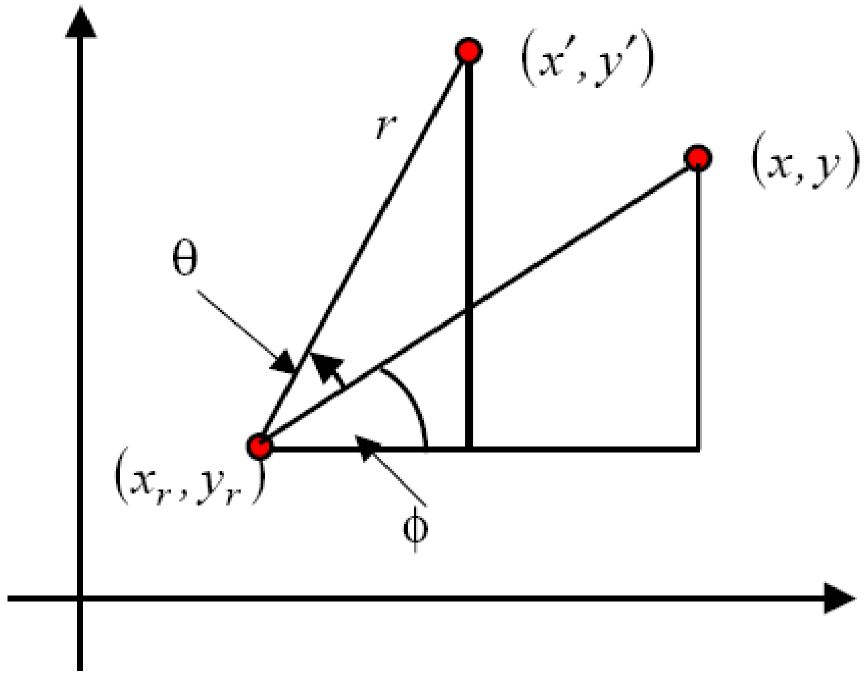
$$P' = R \,.\, P \tag{7}$$

- Where the rotation matrix is

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{8}$$

•Rotation of a point about an arbitrary pivot position is shown bellow:

Fig. 5: *Rotating a point from position (x,y) to position* (x', y') *through angle* θ *about rotation point* (x_r, y_r).



Rotation

•Using the trigonometric relationships indicated by the two right triangles in the previous Figue, we can generalize equation (6) to obtain the transformation equations for rotation of a point about any specified rotation position (x_r, y_r):

$$
\begin{aligned}
x' &= x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta, \\
y' &= y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta
\end{aligned}
\qquad (9)
$$

Rotation

•Rotation is a rigid-body transformations that move objects without deformation.

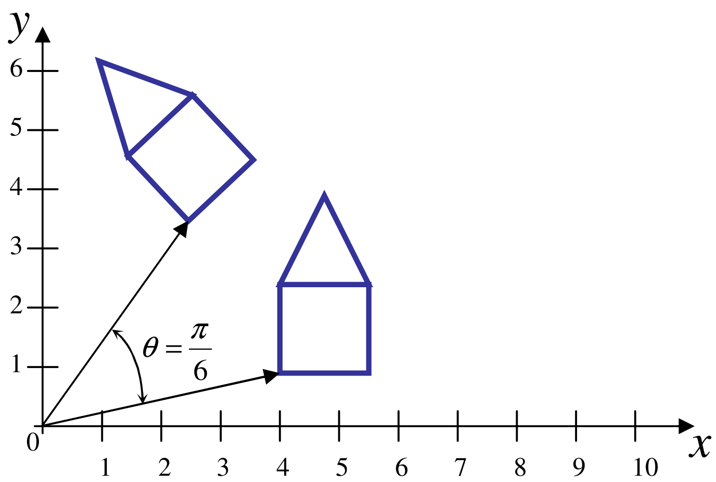
•Every point on an object is rotated through the same angle.

•A straight line segment is rotated by applying the rotation equation (9) to each of the two line endpoints and redrawing the line between the new endpoint positions.

Rotation

- A polygon is rotated by displaying each vertex using the specified rotation angle and then generating the polygon using the new vertices.
- A curve is rotated by repositioning the defining points for the curve and then redrawing it.
- An ellipse is rotated about its center coordinates simply by rotating the major and minor axes.

Example



Homogeneous Coordinates

A point (x, y) can be re-written in **homogeneous coordinates** as (x_h, y_h, h)

The **homogeneous parameter** h is a non-zero value such that:

$$x = \frac{x_h}{h} \qquad\qquad y = \frac{y_h}{h}$$

We can then write any point (x, y) as (hx, hy, h)

We can conveniently choose $h = 1$ so that (x, y) becomes $(x, y, 1)$

Why Homogeneous Coordinates?

Mathematicians commonly use homogeneous coordinates as they allow scaling factors to be removed from equations

We will see in a moment that all of the transformations we discussed previously can be represented as 3*3 matrices

Using homogeneous coordinates allows us use matrix multiplication to calculate transformations – extremely efficient!

Homogeneous Translation

The translation of a point by *(dx, dy)* can be written in matrix form as:

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

Representing the point as a homogeneous column vector we perform the calculation as:

$$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1*x+0*y+dx*1 \\ 0*x+1*y+dy*1 \\ 0*x+0*y+1*1 \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \\ 1 \end{bmatrix}$$

Remember Matrix Multiplication

Recall how matrix multiplication takes place:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a*x+b*y+c*z \\ d*x+e*y+f*z \\ g*x+h*y+i*z \end{bmatrix}$$

Homogenous Coordinates

To make operations easier, 2-D points are written as homogenous coordinate column vectors

Translation: $$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \\ 1 \end{bmatrix} : v' = T(dx, dy)v$$

Scaling: $$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \times x \\ s_y \times y \\ 1 \end{bmatrix} : v' = S(s_x, s_y)v$$

Homogenous Coordinates (cont…)

Rotation:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta \times x - \sin\theta \times y \\ \sin\theta \times x + \cos\theta \times y \\ 1 \end{bmatrix} : v' = R(\theta) v$$

Inverse Transformations

Transformations can easily be reversed using inverse transformations

$$T^{-1} = \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

$$S^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Combining Transformations

A number of transformations can be combined into one matrix to make things easy

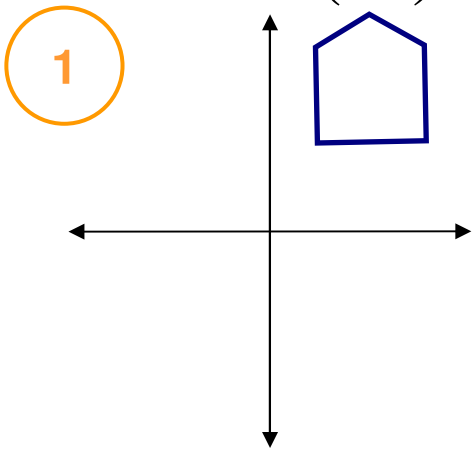
- Allowed by the fact that we use homogenous coordinates

Imagine rotating a polygon around a point other than the origin

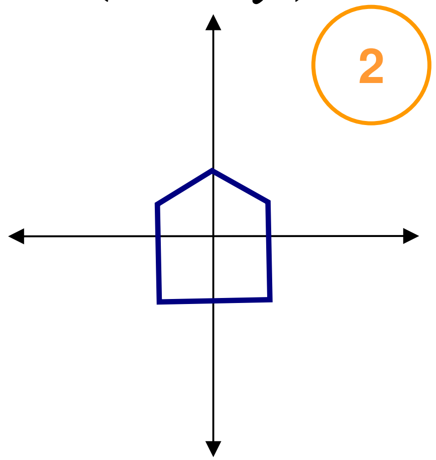
- Transform to centre point to origin
- Rotate around origin
- Transform back to centre point

Combining Transformations (cont…)

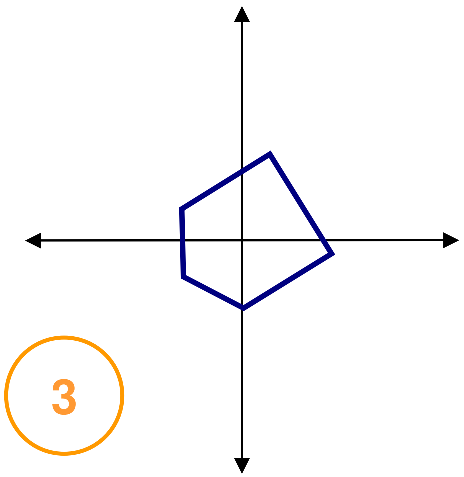
$House\ (H)$



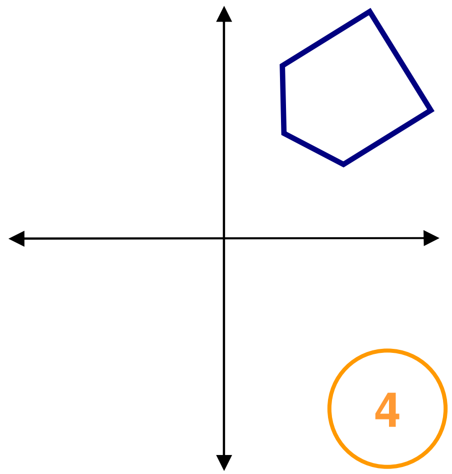
$T(dx, dy)H$



$R(\theta)T(dx, dy)H$



$T(-dx, -dy)R(\theta)T(dx, dy)H$



Combining Transformations (cont…)

The three transformation matrices are combined as follows

$$\begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$v' = T(-dx, -dy) R(\theta) T(dx, dy) v$$

REMEMBER: Matrix multiplication is not commutative so order matters

Summary

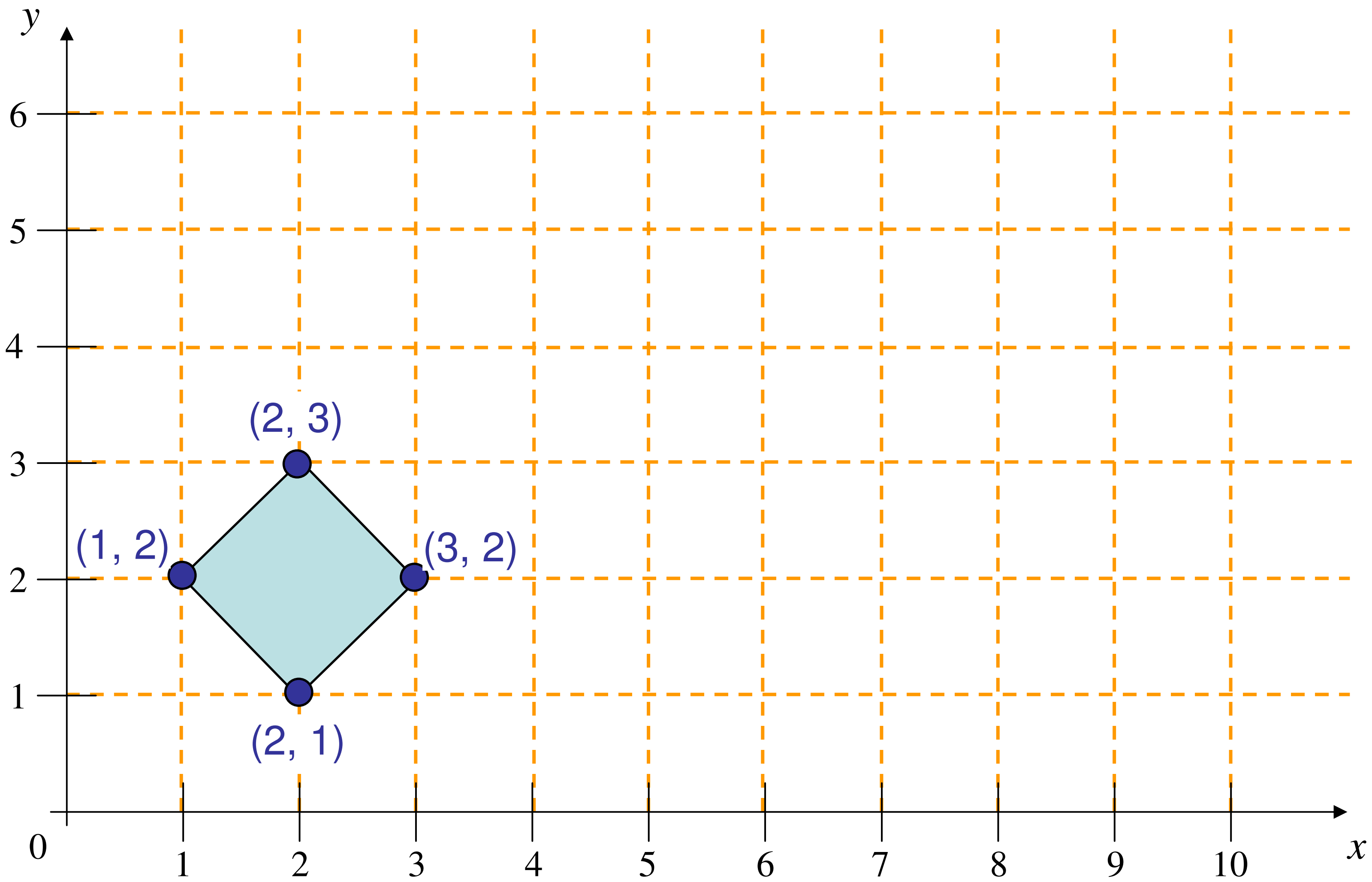
In this lecture we have taken a look at:

- 2D Transformations
  - Translation
  - Scaling
  - Rotation
- Homogeneous coordinates
- Matrix multiplications
- Combining transformations

Next time we'll start to look at how we take these abstract shapes etc and get them on-screen

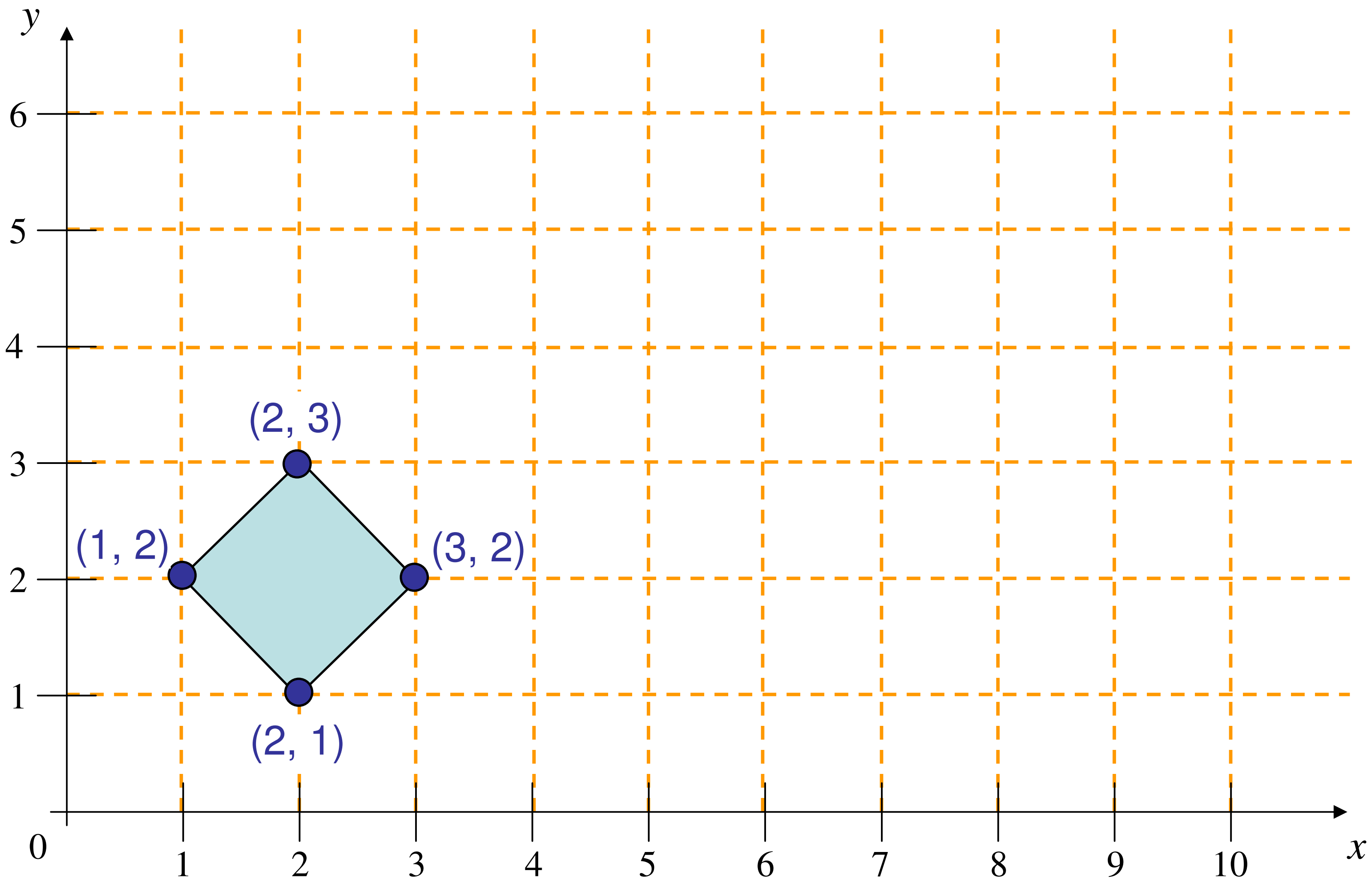
Exercises 1

Translate the shape below by (7, 2)



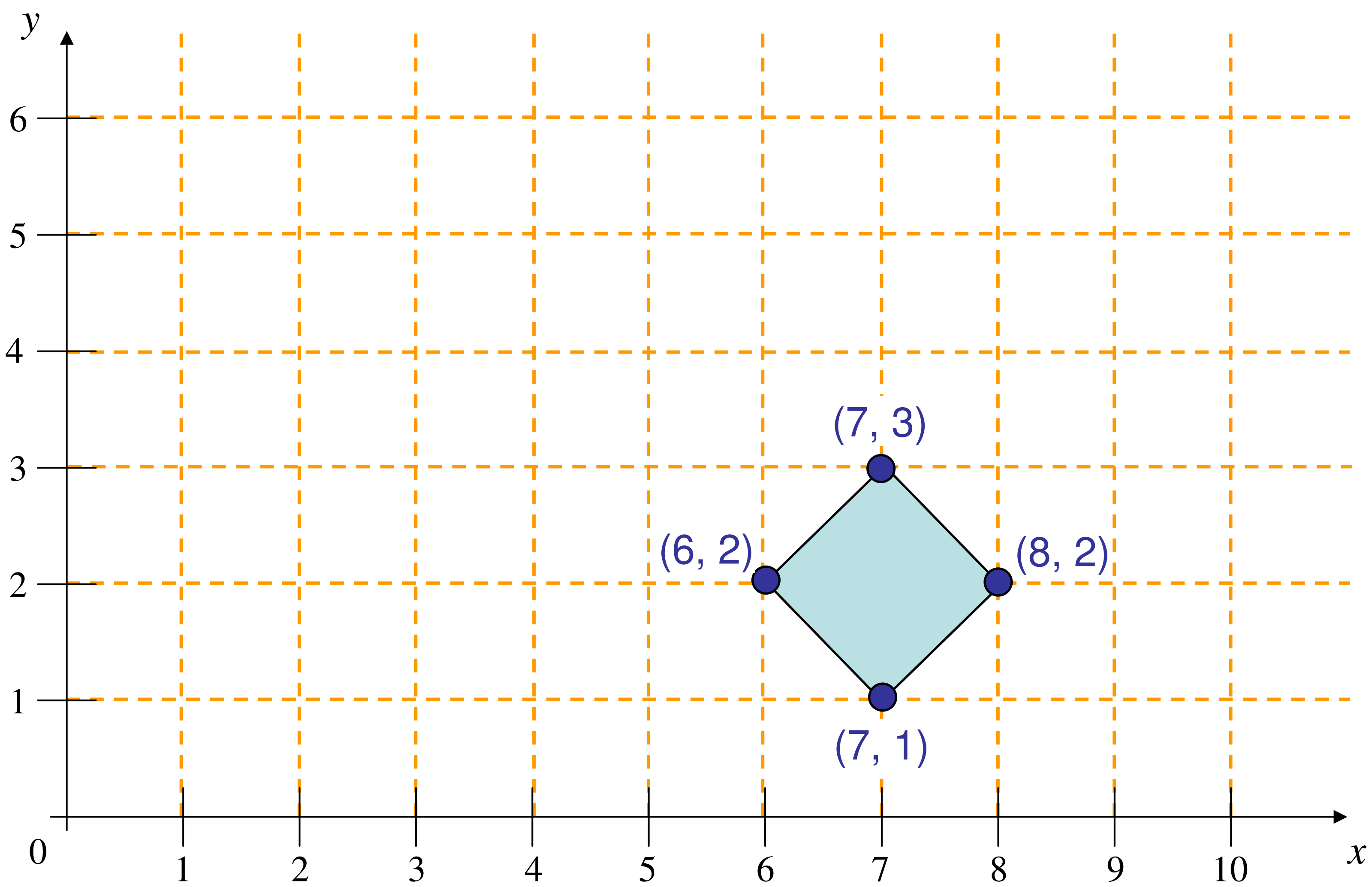
Exercises 2

Scale the shape below by 3 in x and 2 in y



Exercises 3

Rotate the shape below by 30° about the origin



Exercise 4

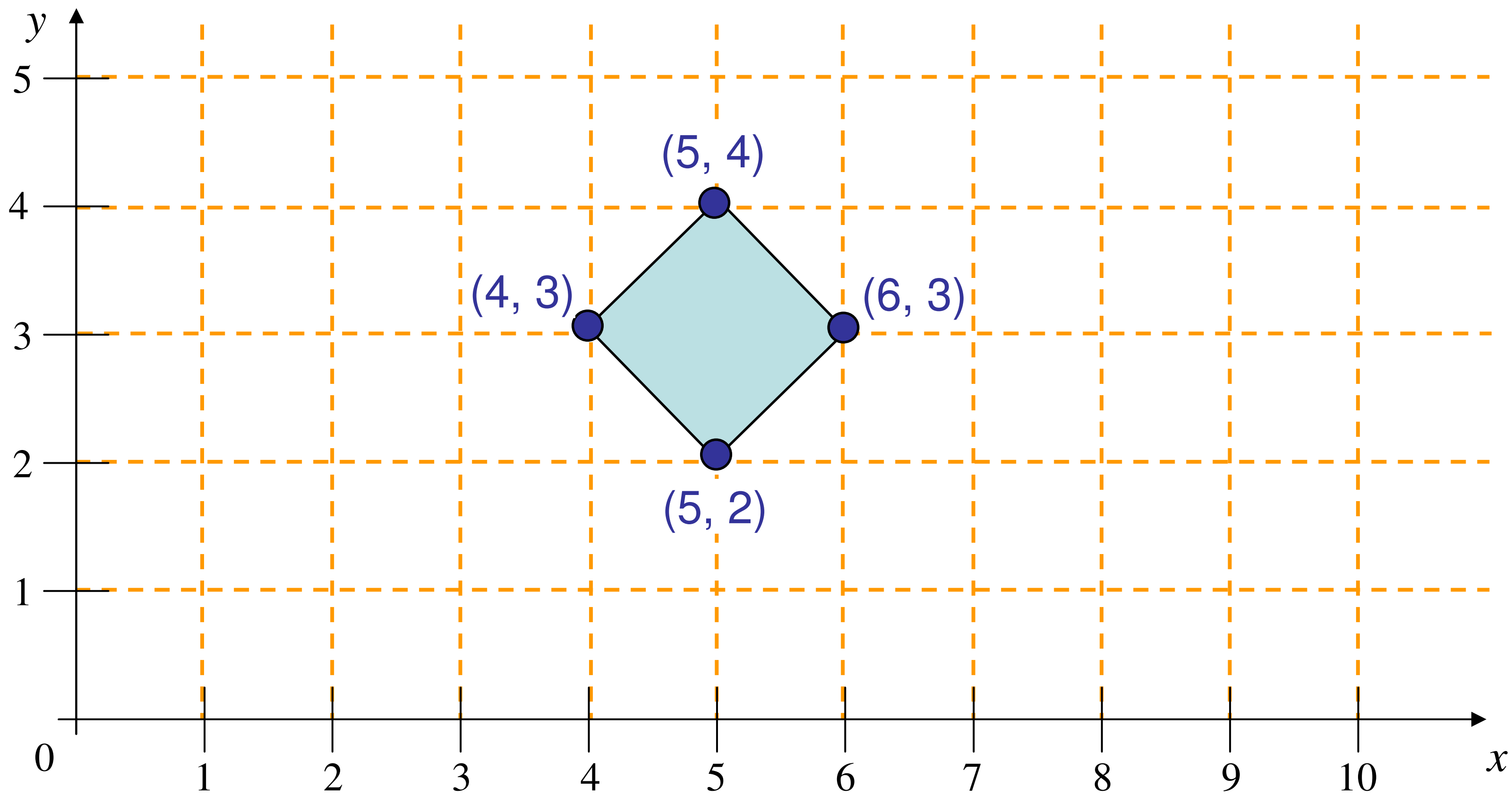
Write out the homogeneous matrices for the previous three transformations

Translation Scaling Rotation

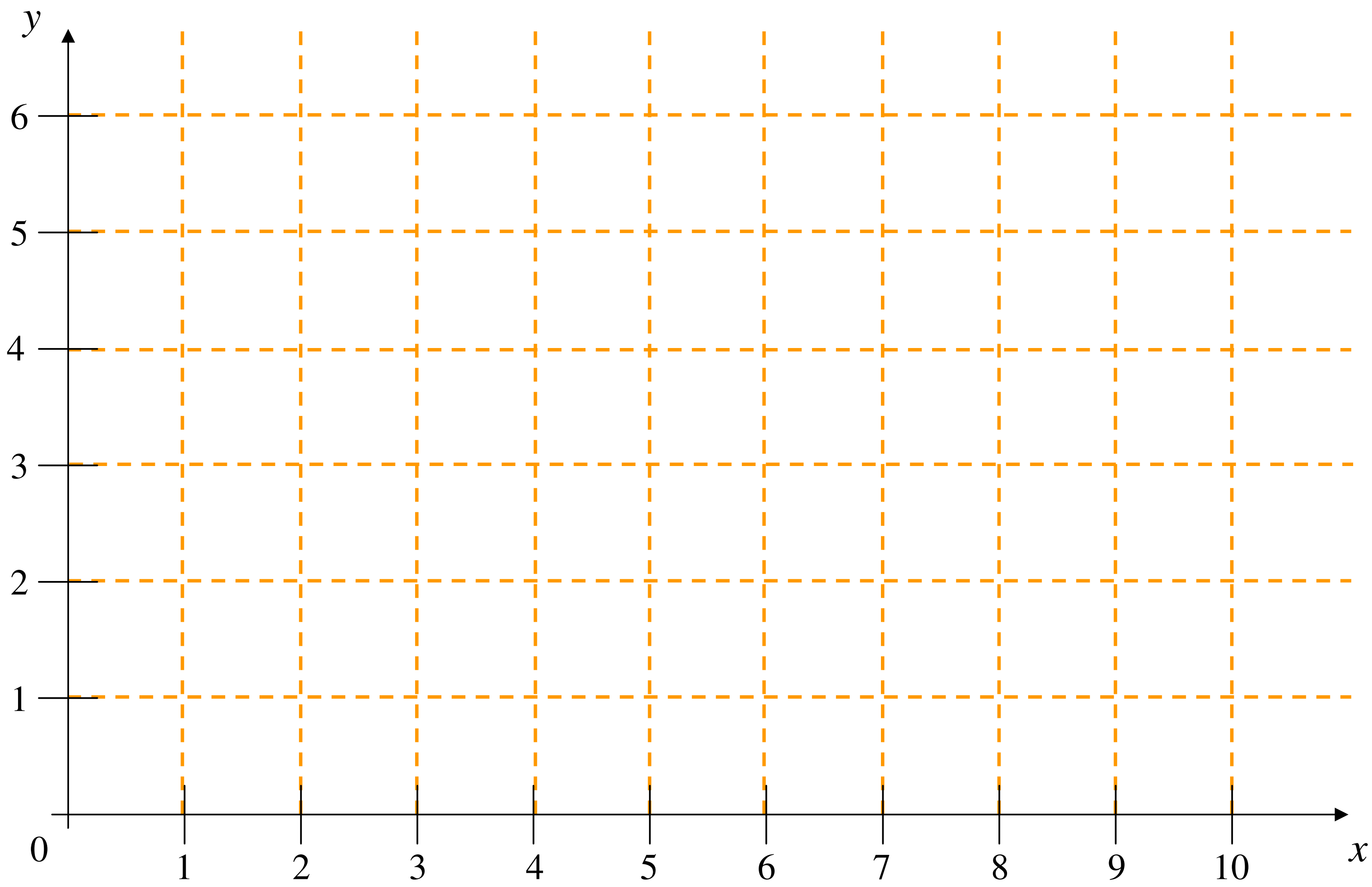
$$\begin{bmatrix} \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \end{bmatrix} \begin{bmatrix} \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \end{bmatrix} \begin{bmatrix} \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \\ \_\_ & \_\_ & \_\_ \end{bmatrix}$$

Exercises 5

Using matrix multiplication calculate the rotation of the shape below by 45° about its centre (5, 3)



Scratch



Equations

Translation:

$$x_{new} = x_{old} + dx \qquad y_{new} = y_{old} + dy$$

Scaling:

$$x_{new} = Sx \times x_{old} \qquad y_{new} = Sy \times y_{old}$$

Rotation

$$x_{new} = x_{old} \times \cos\theta - y_{old} \times \sin\theta$$

$$y_{new} = x_{old} \times \sin\theta + y_{old} \times \cos\theta$$