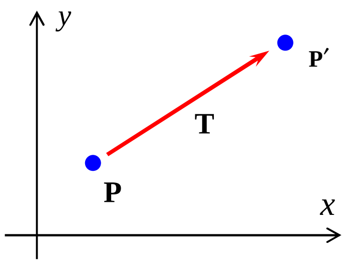
Chapter 5: (part 1)
2D Geometric Transformations

1

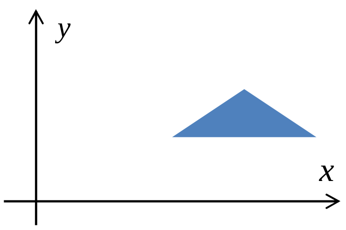
2D Translation

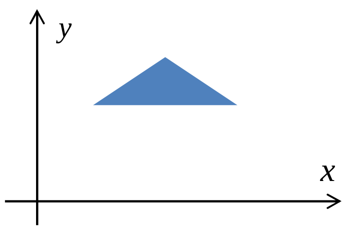


$$x' = x + t_x, \quad y' = y + t_y$$

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{T}$$





2

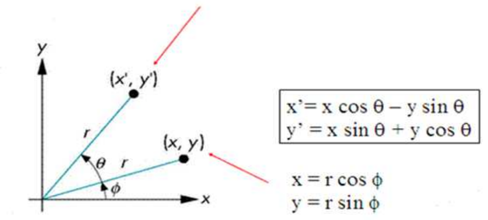
2D Translation

```
void TranslatePolygon (Point2D* points, const int size, float
tx, float ty)
{
        for (int i = 0; i < size; ++i) {
        points[i].x += tx;
        points[i].y += ty;
        }

        glBegin (GL_POLYGON);
                for (int i = 0; i < size; ++i)
                        glVertex2f (points[i].x, points[i].y);
        glEnd ();
}
```


2D Rotation

❑ Rotation around the origin



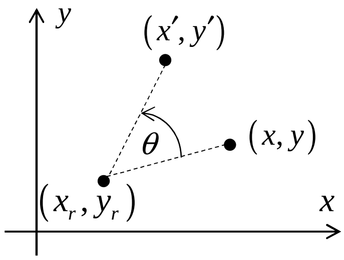
$$P' = R.P$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

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

2D Rotation

- Rotation of a point around an arbitrary pivot position



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

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

$$\mathbf{P}' = \mathbf{P}_r + \mathbf{R}\cdot(\mathbf{P} - \mathbf{P}_r)$$

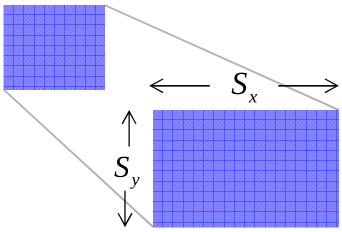
$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2D Rotation

```
void RotatePolygon (Point2D* points, const int size, Point2D
pntPivot, float theta)
{
        for (int i = 0; i < size; ++i) {
                points[i].x = pntPivot.x + (points[i].x - pntPivot.x)
                        * cos (theta) - (points[i].y - pntPivot.x) *
                        sin (theta);
                points[i].y = pntPivot.y + (points[i].x - pntPivot.x)
                        * sin (theta) + (points[i].y - pntPivot.y) *
                        cos (theta);
        }

        glBegin (GL_POLYGON);
                for (int i = 0; i < size; ++i)
                        glVertex2f (points[i].x, points[i].y);
        glEnd ();
}
```

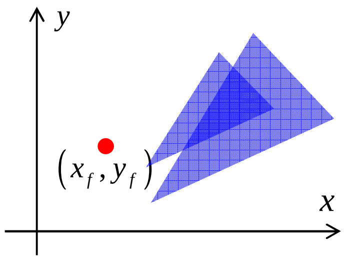
2D Scaling



$$x' = x \cdot s_x, \quad y' = y \cdot s_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$



Scaling about a fixed point (x_f, y_f)

$x' - x_f = (x - x_f)s_x, \; y' - y_f = (y - y_f)s_y$

$$x' = x \cdot s_x + x_f (1 - s_x)$$

$$y' = y \cdot s_y + y_f (1 - s_y)$$

$$\mathbf{P}' = \mathbf{P} \cdot \mathbf{S} + \mathbf{P}_f \cdot (\mathbf{1\text{-}S})$$

7

2D Scaling

```
void ScalePolygon (Point2D* points, const int size, Point2D
pntFixed, float sx, float sy)
{
	for (int i = 0; i < size; ++i) {
		points[i].x = points[i].x * sx + pntFixed.x *
			(1 - sx);
		points[i].y = points[i].y * sy + pntFixed.y *
			(1 - sy);
	}

	glBegin (GL_POLYGON);
		for (int i = 0; i < size; ++i)
			glVertex2f (points[i].x, points[i].y);
	glEnd ();
}
```

8

Homogeneous Coordinates

Rotate and then displace a point $\mathbf{P}$: $\mathbf{P}' = \mathbf{M}_1 \cdot \mathbf{P} + \mathbf{M}_2$

$\mathbf{M}_1$: 2×2 rotation matrix. $\mathbf{M}_2$: 2×1 displacement vector.

Displacement is unfortunately a non linear operation.

Make displacement linear with **Homoheneous Coordinates**.

$(x, y) \Rightarrow (x, y, 1)$. Transformations turn into 3×3 matrices.

Very big advantage. All transformations are concatenated by matrix multiplication.

9

2D Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P}$$

2D Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P}$$

2D Scaling

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{S}(S_x, S_y) \cdot \mathbf{P}$$

10

Inverse transformations:

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{S}^{-1} = \begin{bmatrix} 1/S_x & 0 & 0 \\ 0 & 1/S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composite transformations: $\mathbf{P}' = \mathbf{M}_2(\mathbf{M}_1 \cdot \mathbf{P}) = (\mathbf{M}_2 \cdot \mathbf{M}_1) \cdot \mathbf{P} = \mathbf{M} \cdot \mathbf{P}$

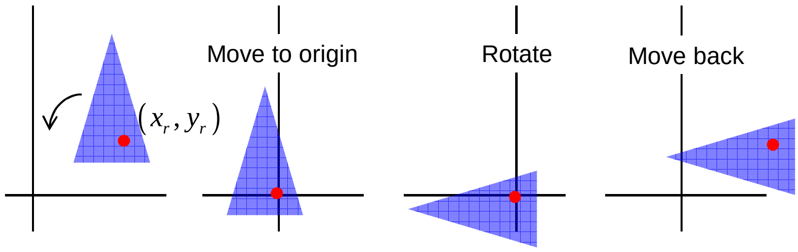
$$\mathbf{P}' = \mathbf{T}(t_{2x}, t_{2y}) \{ \mathbf{T}(t_{1x}, t_{1y}) \cdot \mathbf{P} \} = \{ \mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) \} \cdot \mathbf{P}$$

Composite translations:

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x} + t_{2x}, t_{1y} + t_{2y})$$

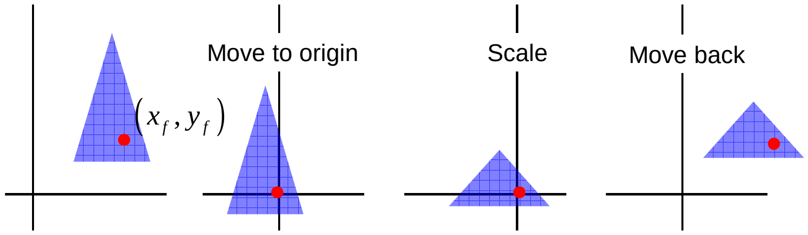
General 2D Rotation



$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r \sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r \sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

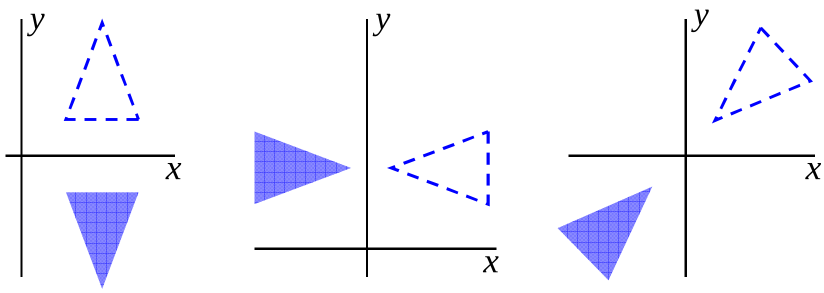
General 2D Scaling



$$\begin{bmatrix}1 & 0 & x_f\\ 0 & 1 & y_f\\ 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}S_x & 0 & 0\\ 0 & S_y & 0\\ 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 0 & -x_f\\ 0 & 1 & -y_f\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}S_x & 0 & x_f(1-S_x)\\ 0 & S_y & y_f(1-S_y)\\ 0 & 0 & 1\end{bmatrix}$$

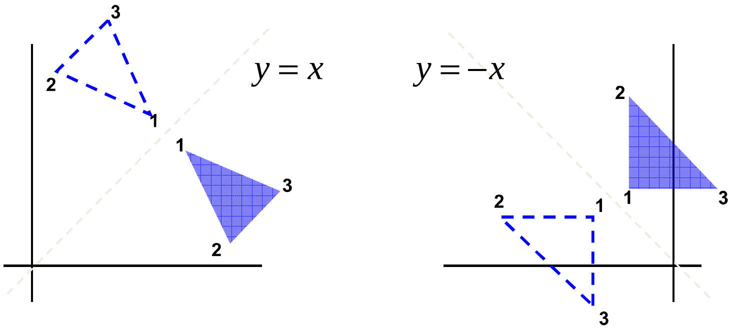
13

2D Reflections



$$\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}\bullet\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

14



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

2D Shearing

- Shear : A transformation that distorts the shape of an object

x-direction shear

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

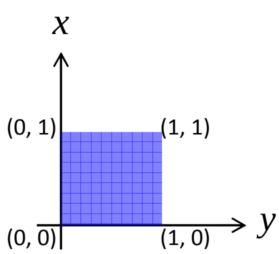
$x' = x + sh_x \cdot y, \; y' = y$

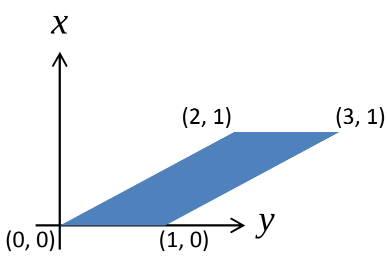
y-direction shear

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

$x' = x, \; y' = y + sh_y \cdot x$

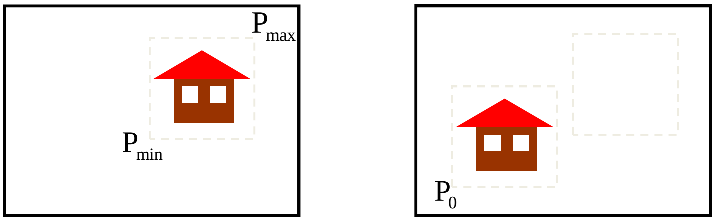
- E.g. $sh_x = 2$





Geometric Transformations by Rasterization

- The transformed shape needs to be filled.
  - A whole scan-line filling is usually in order.
- However, simple transformations can save new filling by manipulating blocks in the frame buffer.



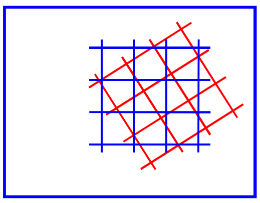
Translation:

Move block of pixels of frame buffer into new destination.

90° counterclockwise rotation

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 3 & 6 & 9 & 12 \\ 2 & 5 & 8 & 11 \\ 1 & 4 & 7 & 10 \end{bmatrix} \qquad \begin{bmatrix} 12 & 11 & 10 \\ 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

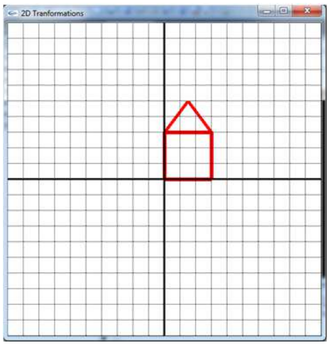
180° rotation



Rotated pixel block

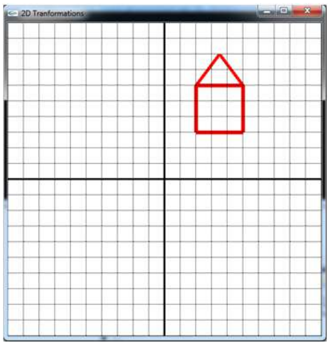
Destination pixel array

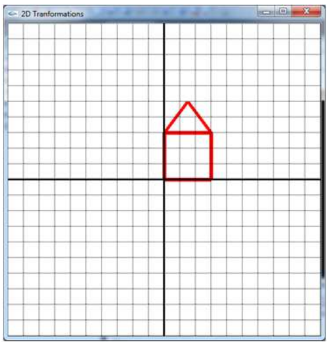
RGB of destination pixel can be determined by averaging rotated ones (as antialiasing)



$$P' = T(2,3).P$$

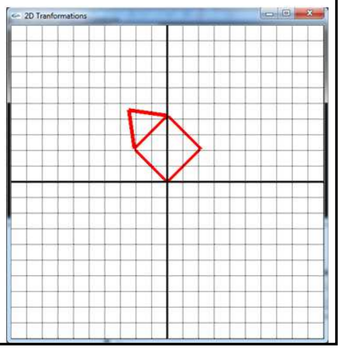
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

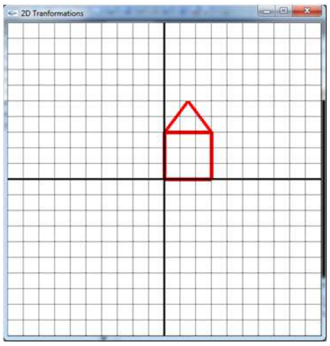




$$P' = R(45).P$$

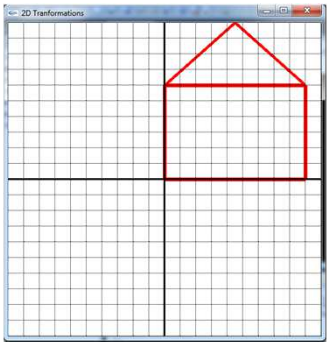
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos45 & -sin45 & 0 \\ sin45 & cos45 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

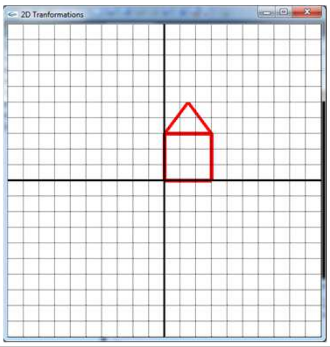




$$P' = S(3,2).P$$

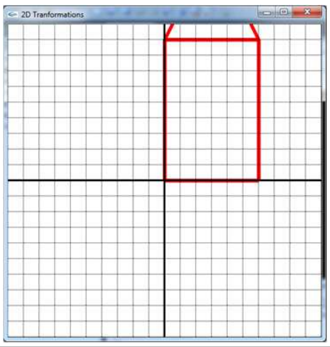
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

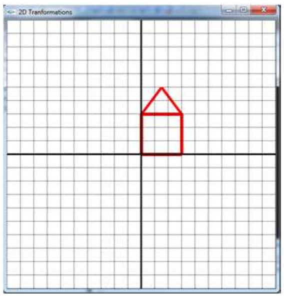




$$P' = S(2,3).P$$

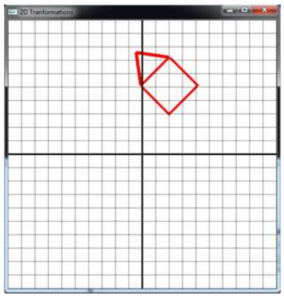
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

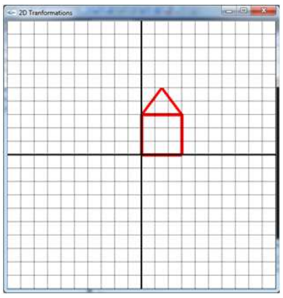




$$P' = T(2,3).R(45).P$$

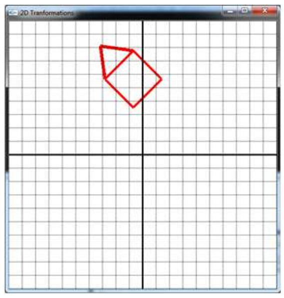
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos45 & -sin45 & 0 \\ sin45 & cos45 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$





$$P' = R(45).T(2,3).P$$

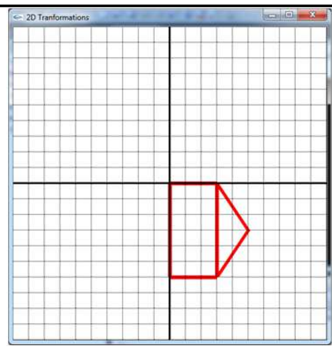
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos45 & -sin45 & 0 \\ sin45 & cos45 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



21

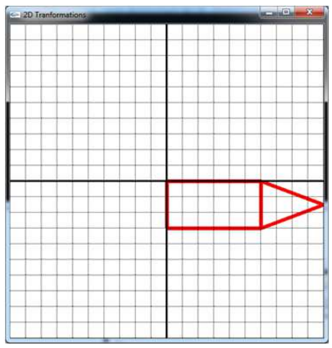
$$P' = S(1,2).R(270).P$$

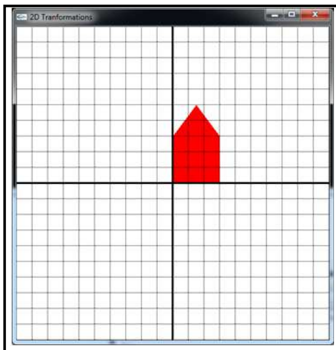
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos270 & -sin270 & 0 \\ sin270 & cos270 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



$$P' = R(270).S(1,2).P$$

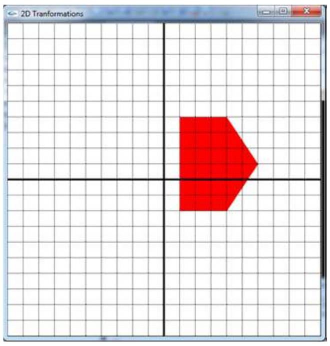
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos270 & -sin270 & 0 \\ sin270 & cos270 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

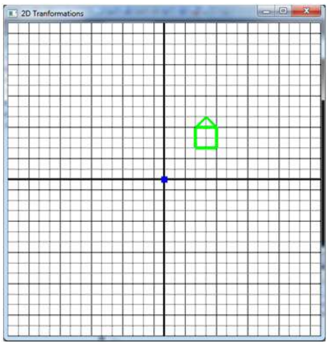




$$P' = S(1,2).T(1,2).R(270).P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos270 & -sin270 & 0 \\ sin270 & cos270 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

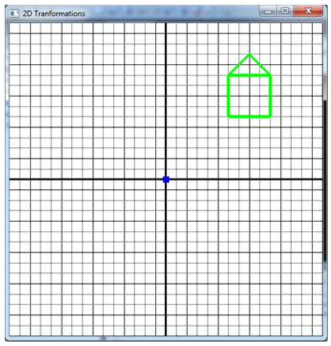


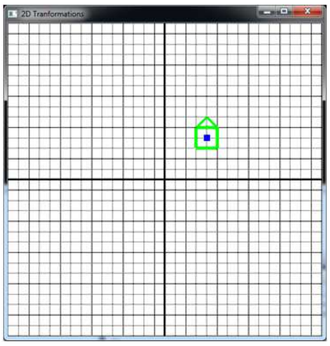


Scaling about origin

$$P' = S(2,2).P$$

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

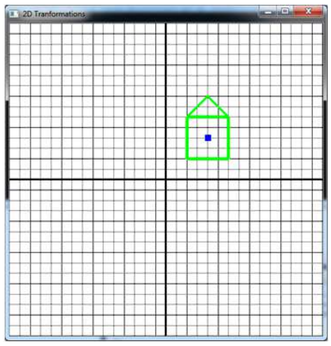


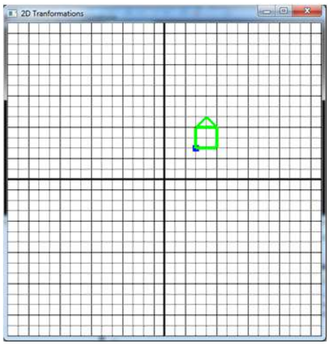


Scaling about (4, 4)

$$P' = T(4,4).S(2,2).T(-4,-4).P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

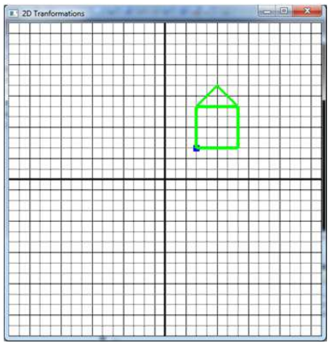


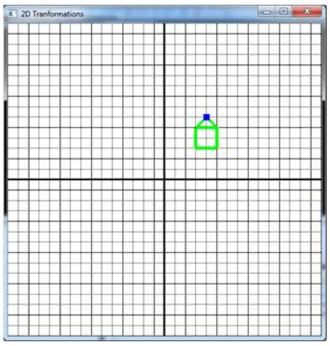


Scaling about (3, 3)

$$P' = T(3,3).S(2,2).T(-3,-3).P$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

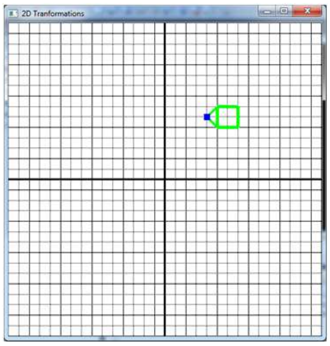


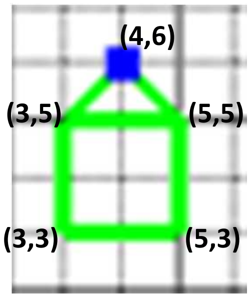


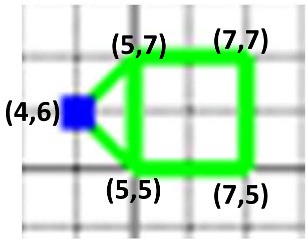
Rotation around (4, 6)

$$P' = T(4,6).R(90).T(-4,-6).P$$

$$\begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos90 & -sin90 & 0 \\ sin90 & cos90 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix}$$







$$M = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & -1 & 4 \\ 1 & 0 & 6 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 7 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 7 \\ 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 5 \\ 7 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 4 \\ 6 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 10 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

26