**Assignment No.: 06**

Name: Aditya Kangune

Roll No.: 23365

Class: SE11

Batch: G11

Subject: Computer Graphics Laboratory

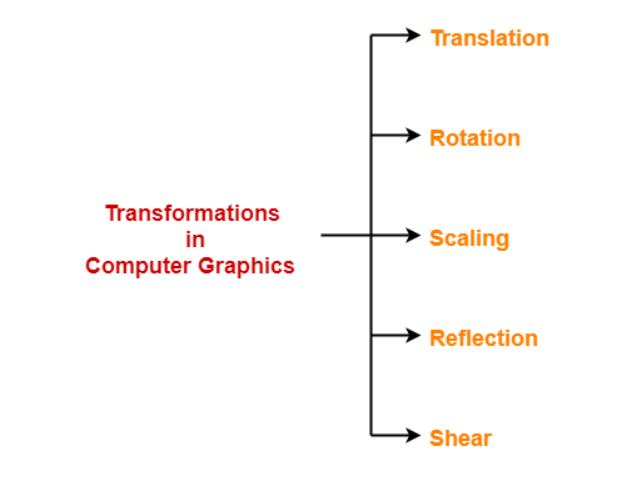
**Title:** 2D Transformation.

**Problem Statement:** Implement following 2D transformations on the object with respect to axis:

1. Scaling.
2. Rotation about arbitrary point.
3. Reflection.

**Theory:**

* Transformation means changing some graphics into something else by applying rules.
* We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation.
* Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.
* In computer graphics, various transformation techniques are shown the diagram below:



1. **Translation:**

* In Computer graphics, 2D Translation is a process of moving an object from one position to another in a two-dimensional plane.
* Consider a point object O has to be moved from one position to another in a 2D plane.
* Let

Initial coordinates of the object O = (Xold, Yold)

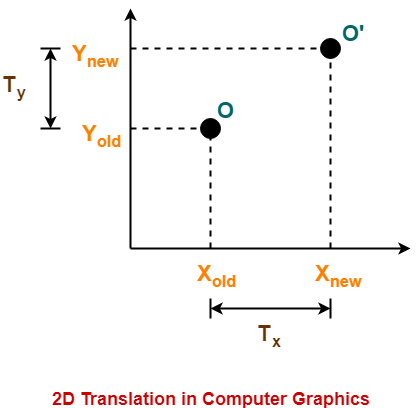
New coordinates of the object O after translation = (Xnew, Ynew)

Translation vector or Shift vector = (Tx, Ty)

* Given a Translation vector (Tx, Ty):

Tx defines the distance the Xold coordinate has to be moved.

Ty defines the distance the Yold coordinate has to be moved.

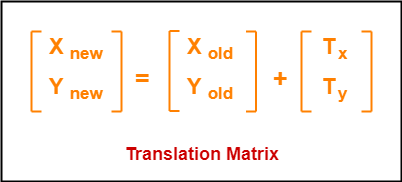


* This translation is achieved by adding the translation coordinates to the old coordinates of the object as:

Xnew = Xold + Tx (This denotes translation towards X axis)

Ynew = Yold + Ty (This denotes translation towards Y axis)

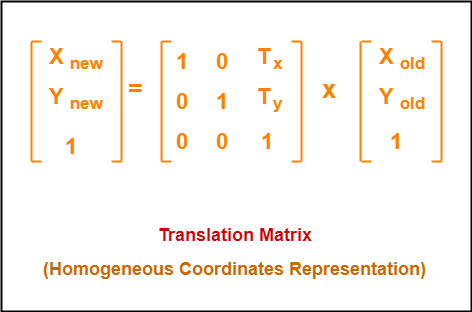
* In Matrix form, the above translation equations may be represented as:



* The homogeneous coordinates representation of (X, Y) is (X, Y, 1).

Through this representation, all the transformations can be performed using matrix / vector multiplications.

* The above translation matrix may be represented as a 3 x 3 matrix as:



1. **Rotation:**

* In Computer graphics, 2D Rotation is a process of rotating an object with respect to an angle in a two-dimensional plane.
* Consider a point object O has to be rotated from one angle to another in a 2D plane.

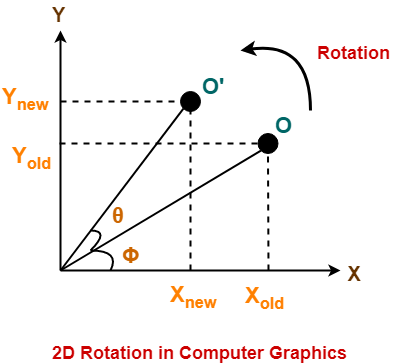
* Let

Initial coordinates of the object O = (Xold, Yold)

Initial angle of the object O with respect to origin = Φ

Rotation angle = θ

New coordinates of the object O after rotation = (Xnew, Ynew)

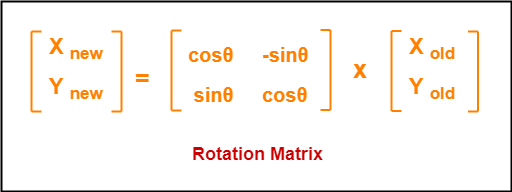


* This rotation is achieved by using the following rotation equations:

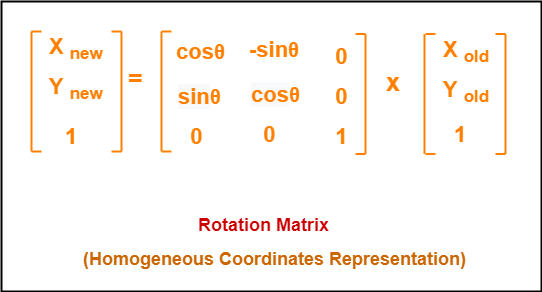
Xnew = Xold \* cosθ – Yold \* sinθ

Ynew = Xold \* sinθ + Yold \* cosθ

* In Matrix form, the above rotation equations may be represented as:



* For homogeneous coordinates, the above rotation matrix may be represented as a 3 x 3 matrix as:



1. **Scaling:**

* In computer graphics, scaling is a process of modifying or altering the size of objects.
* Scaling may be used to increase or reduce the size of object. Scaling subjects the coordinate points of the original object to change.
* Scaling factor determines whether the object size is to be increased or reduced.
  + If scaling factor > 1, then the object size is increased.
  + If scaling factor < 1, then the object size is reduced.
* Consider a point object O has to be scaled in a 2D plane.
* Let

Initial coordinates of the object O = (Xold, Yold)

Scaling factor for X-axis = Sx

Scaling factor for Y-axis = Sy

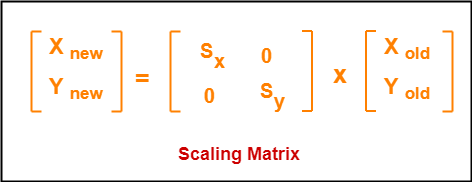
New coordinates of the object O after scaling = (Xnew, Ynew)

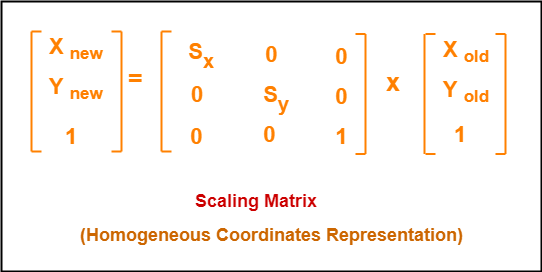
* This scaling is achieved by using the following scaling equations-

Xnew = Xold \* Sx

Ynew = Yold \* Sy

* In Matrix form, the above scaling equations may be represented as:



* For homogeneous coordinates, the above scaling matrix may be represented as a 3 x 3 matrix as:

1. **Reflection:**

* Reflection is a kind of rotation where the angle of rotation is 180 degrees.
* The reflected object is always formed on the other side of mirror. The size of reflected object is same as the size of original object.
* Consider a point object O has to be reflected in a 2D plane.
* Let

Initial coordinates of the object O = (Xold, Yold)

New coordinates of the reflected object O after reflection = (Xnew, Ynew)

* Reflection can be of three types as mentioned below:

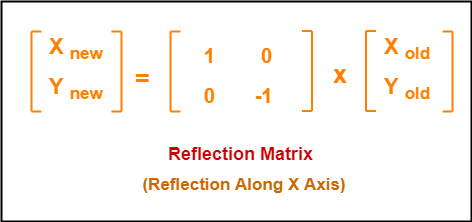
1. **Reflection on X-axis:**

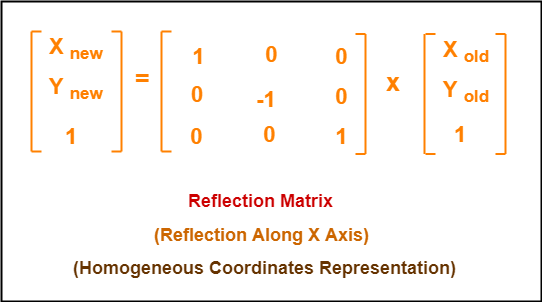
* This reflection is achieved by using the following reflection equations-

Xnew = Xold

Ynew = -Yold

* In Matrix form, the above reflection equations may be represented as:



* For homogeneous coordinates, the above reflection matrix may be represented as a 3 x 3 matrix as:

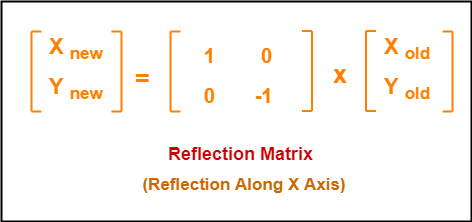
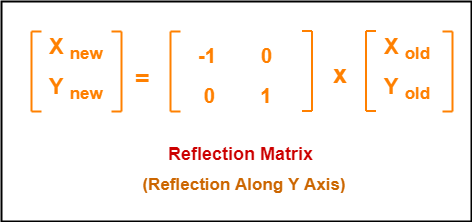
1. **Reflection on Y-axis:**

* This reflection is achieved by using the following reflection equations-

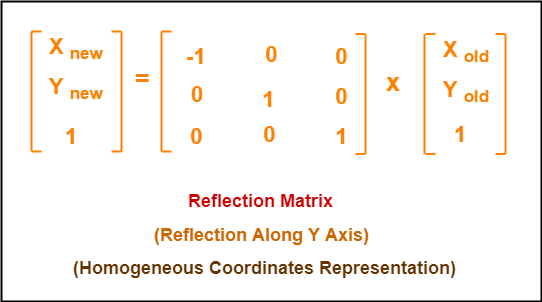
Xnew = -Xold

Ynew = Yold

* In Matrix form, the above reflection equations may be represented as:



* For homogeneous coordinates, the above reflection matrix may be represented as a 3 x 3 matrix as:



1. **Reflection about Origin:**

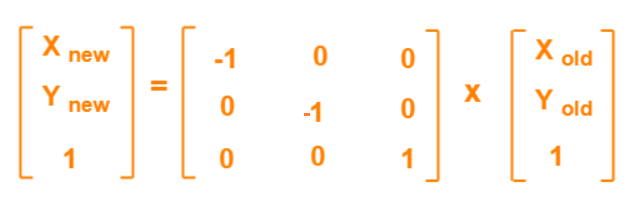
* This reflection is achieved by using the following reflection equations-

Xnew = -Xold

Ynew = -Yold

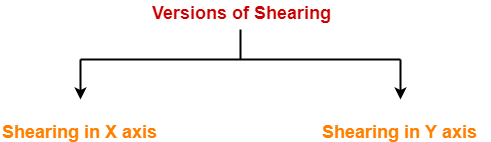
* In Matrix form, the above reflection equations may be represented as:



* For homogeneous coordinates, the above reflection matrix may be represented as a 3 x 3 matrix as:

1. **Shear:**

* In Computer graphics, 2D Shearing is an ideal technique to change the shape of an existing object in a two-dimensional plane.
* In a two-dimensional plane, the object size can be changed along X direction as well as Y direction. So, there are two versions of shearing:



* Consider a point object O has to be sheared in a 2D plane.
* Let

Initial coordinates of the object O = (Xold, Yold)

Shearing parameter towards X direction = Shx

Shearing parameter towards Y direction = Shy

New coordinates of the object O after shearing = (Xnew, Ynew)

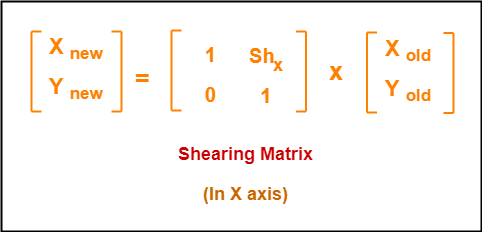
1. **Shearing in X-axis:**

* Shearing in X axis is achieved by using the following shearing equations:

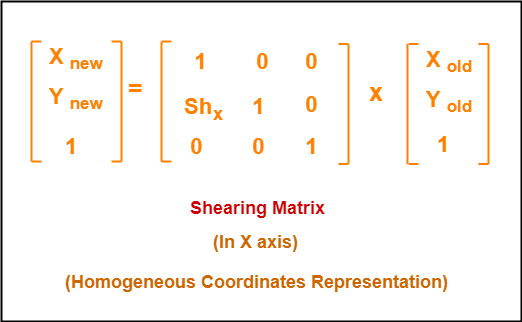
Xnew = Xold + Shx \* Yold

Ynew = Yold

* In Matrix form, the above shearing equations may be represented as:



* For homogeneous coordinates, the above shearing matrix may be represented as a 3 x 3 matrix as-

****

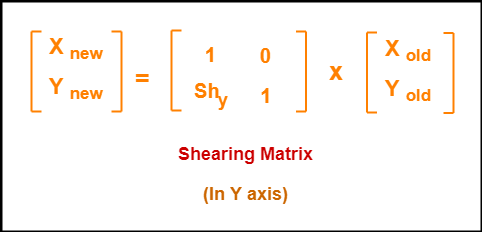
1. **Shearing in Y-axis:**

* Shearing in X axis is achieved by using the following shearing equations:

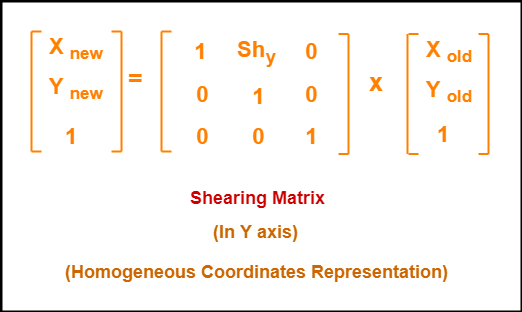
Xnew = Xold

Ynew = Yold + Shy \* Xold

* In Matrix form, the above shearing equations may be represented as:



* For homogeneous coordinates, the above shearing matrix may be represented as a 3 x 3 matrix as:



* **Composite Transformation:**
* A composite transformation (or composition of transformations) is two or more transformations performed one after the other. Sometimes, a composition of transformations is equivalent to a single transformation. The following is an example of a translation followed by a reflection.
* If a transformation of the plane T1 is followed by a second plane transformation T2, then the result itself may be represented by a single transformation T which is the composition of T1 and T2 taken in that order. This is written as T = T1∙T2.
* Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.
* A combined matrix :

[T][X] = [X] [T1] [T2] [T3] [T4] …. [Tn]

where [Ti] is any combination of

* Translation
* Scaling
* Shearing
* Rotation
* Reflection
* The change in the order of transformation would lead to different results, as in general matrix multiplication is not cumulative, that is [A] . [B] ≠ [B] . [A] and the order of multiplication.
* The basic purpose of composing transformations is to gain efficiency by applying a single composed transformation to a point, rather than applying a series of transformation, one after another.
* For example, to rotate an object about an arbitrary point (Xp, Yp), we have to carry out three steps:
* Translate point (Xp, Yp) to the origin.
* Rotate it about the origin.
* Finally, translate the center of rotation back where it belonged.

**Algorithm:**

1. Start
2. Initialize the graphics mode.
3. Construct a 2D object  (use Drawpoly()) e.g. (x,y)
4. **Translation:**
   * 1. Get the translation value Tx, Ty
     2. Move the 2d object with tx, ty (x’=x+Tx, y’=y+Ty)
     3. Plot (x’,y’)
5. **Rotation:**
   * 1. Get the Rotation angle
     2. Rotate the object by the angle ф
        + 1. x’=x cos ф - y sin ф
          2. y’=x sin ф - y cosф
     3. Plot (x’,y’)
6. **Scaling:**
   * 1. Get the scaling value Sx,Sy
     2. Resize the object with Sx,Sy  (x’=x\*Sx, y’=y\*Sy)
     3. Plot (x’,y’)
7. **Reflection**
   * 1. About X-axis
        + 1. x’= x , y’ = -y
          2. Plot (x’,y’)
     2. About Y-axis
        + 1. x’= -x , y’=y
          2. Plot (x’,y’)
8. **Shearing:**
   * 1. X-shear
        + 1. Get the shearing value Shx
          2. x’=x + Shx \* y, y’=y
          3. Plot (x’,y’)
     2. Y-shear
        + 1. Get the shearing value Shy
          2. x’=x , y’=y+ Shy \* x
          3. Plot (x’,y’)

**Input:**

**Output:**

**Conclusion:**

2D Transforms techniques were implemented successfully on the object with respect to axis.

Following techniques were implemented:

* + 1. Scaling
    2. Rotation about arbitrary point
    3. Reflection