Assignment - 9

GoodLuck | Page No.
Date

Title :- Reflection transformation

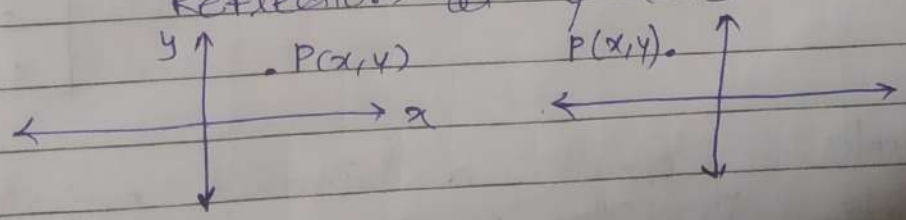
Problem statement :-
Write a c++ program to implement reflection of 2D object about x-axis, y-axis & about x=y line Also Rotate object about the arbitary point given by user

objective :-
To understand the working of transformation ie. reflection about different axes & rotation.

Outcome :-
We will be able to do reflection of partition line about x, y, x=y line

S/w requirement :-
Qt creator, 64 bit linux os.

Theory :-
Reflection at y-axis.

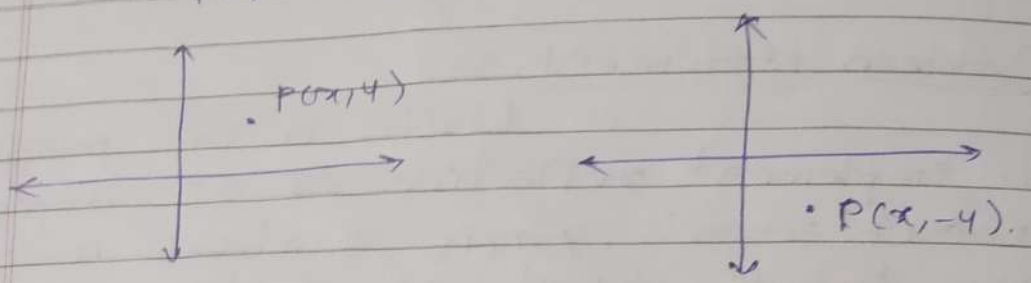


For this transformation, lets take 'T' Transformation matrix.

$P_1 \cdot T = P_2$

$[x, y] \cdot T = [-x, y]$ so

Reflection at x-axis



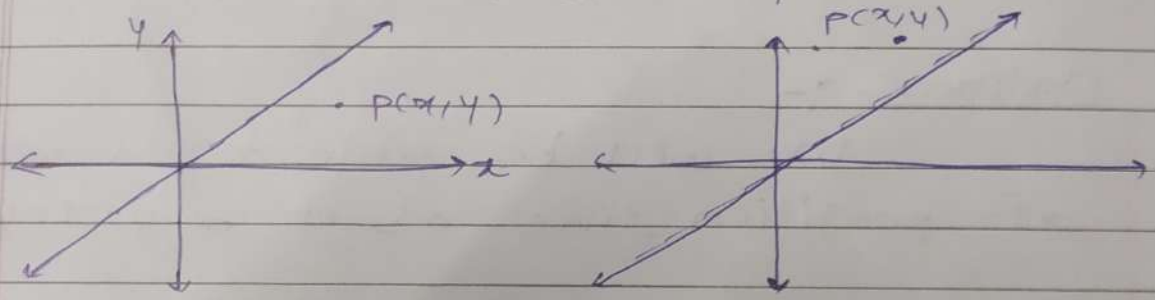
x coordinate is same but y gets changed.

$P_1 \cdot T = P_2$

$(x, y) \cdot T = (x, -y)$

$$T = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$$

Reflection about $x = y$ line



x coordinate becomes equal to y & y coordinate becomes equals x ie. they swap values.

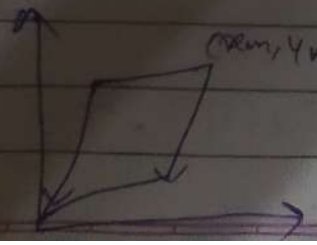
$P_1 \cdot T = P_2$

$[x, y] \cdot T = [y, x]$ $\therefore T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Rotate about arbitary point

Rotate $P_1 (x_m, y_m)$ we have to perform 3 steps. 1st we translate (x_m, y_m) to origin so translation matrix T_1 will become

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_m & -y_m & 1 \end{bmatrix}$$

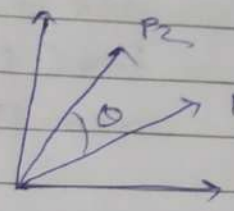


2020/4/20 20:03

GoodLuck | Page No.
Date

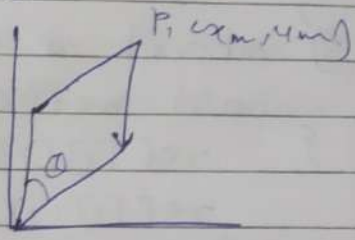
Secondly, we have to rotate pt Pn clockwise or Anticlockwise by angle θ, for that

$$R_1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Third one, to translate pt back to original position i.e. translation on matrix.

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_m & y_m & 1 \end{bmatrix}$$



Combined matrix is,

translation * Rotation * translation

$$= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_m\cos\theta \cdot y_m \sin\theta + x_m & -x_m\sin\theta \cdot -y_m\cos\theta + y_m & 1 \end{bmatrix}$$

Algorithm:-

1] About y axis

```
void reflect()
{   ref[1][2] · ref[2][1] = 0
    ref[1][1] = 1
    ref[2][2] = 1
for (i = 1 to edges) {
    resm[i][i] = (a[i][i] * ref[i][i]
                * (a[i][2] * b[2][i] * 32a);
    resm[1][2] = 2+0 [ (a[i][i] - ref[i][2]
                * (a[2][2] * resm)]
}
```

2020/4/20 20:03

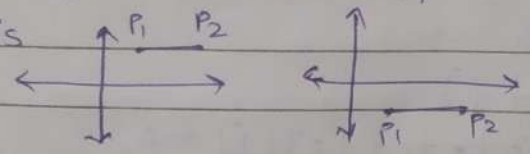
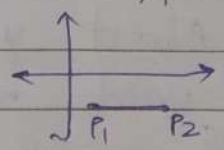
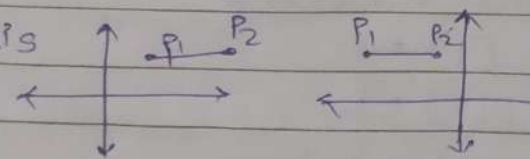
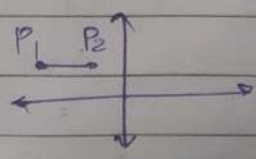
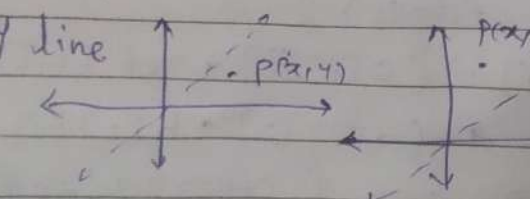
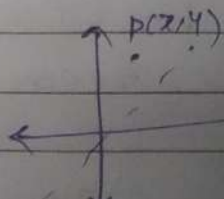
2] About x axis.

```
void reflect()
{
  ref[1][2] = ref[2][1] = 0;
  ref[1][1] = 1; ref[2][2] = -1;
  for(i=1 to edges){
  resm[i][1] = (a[i][1] * ref[1][1]) * (a[i][2] *
               ref[2][1] + 320;
  resm[i][2] = 240 - ((a[i][1] * ref[1][2]) *
               (a[i][2] * ref[2][2]))
  }
}
```

3] About x=y line.

```
void reflect()
{
  ref[1][2] = ref[2][1];
  ref[1][1] = 0; ref[2][2] = 0;
  for(i=1 to edges)
  {
    resm[i][1] = (a[i][1] * ref[i][1] +
                 (a[i][2] * ref[2][1] + 320;
    resm[2][2] = 240 - [(a[i][1] * ref[1][2])
                 + (a[i][2] * ref[2][2])];
  }
}
```

Testcases :-

	I/P	Expected O/P	Actual O/P	Result
1] X-axis				Success:
2] Y-axis				success
3] x=y line				Success

Conclusion :- We successfully implemented the reflection about axes & rotation about any point.

2020/4/20 20:03