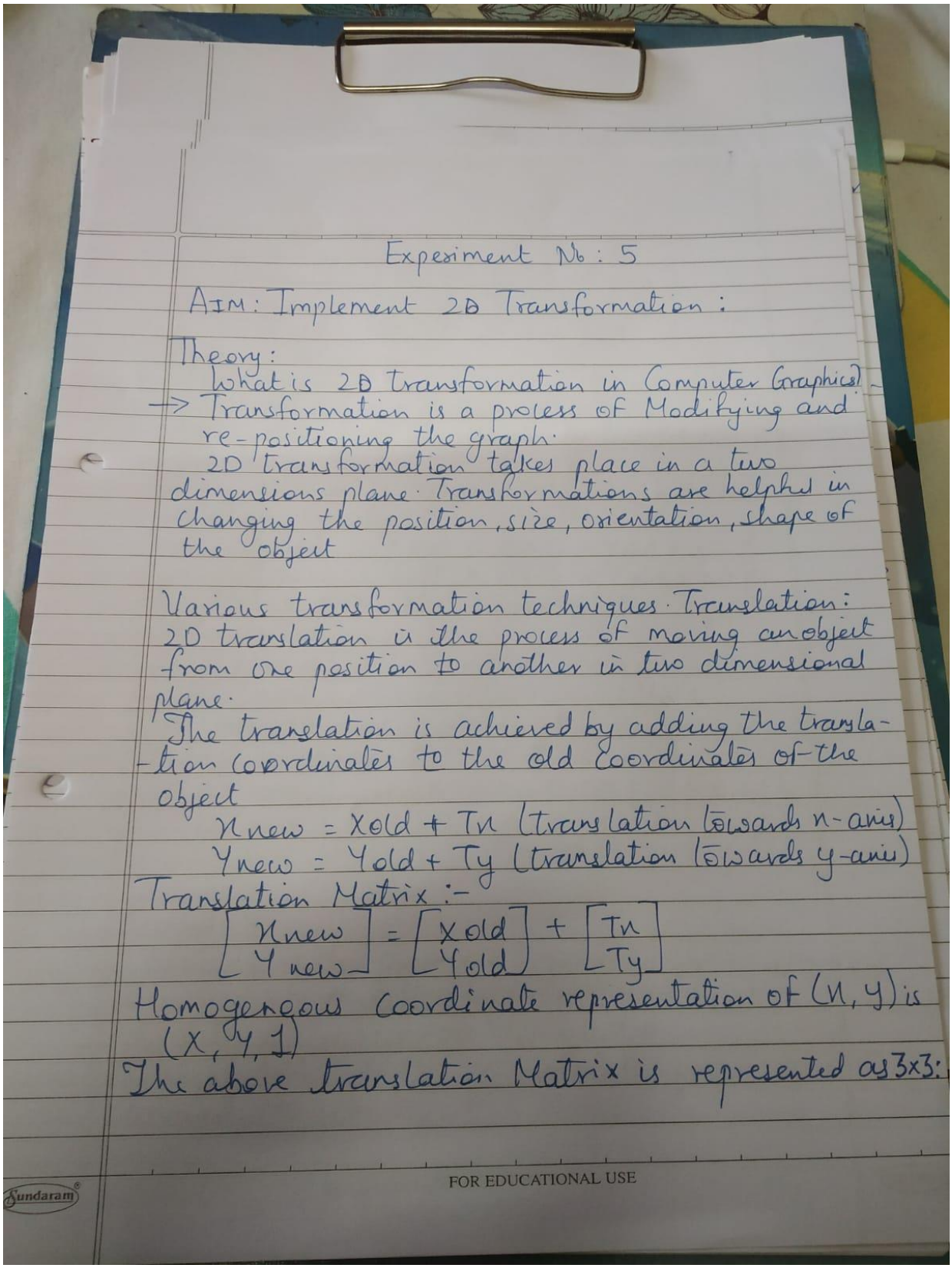
COMPUTER ENGINEERING

CG ODD SEM 2021-22/EXPERIMENT 5

NAME:- GAURAV AMARNANI (D7A, 67)



Experiment No: 5

AIM: Implement 2D Transformation:

Theory:

What is 2D transformation in Computer Graphics?

→ Transformation is a process of Modifying and re-positioning the graph.

2D transformation takes place in a two dimensions plane. Transformations are helpful in changing the position, size, orientation, shape of the object

Various transformation techniques. Translation: 2D translation is the process of moving an object from one position to another in two dimensional plane.

The translation is achieved by adding the translation coordinates to the old coordinates of the object

$X_{new} = X_{old} + T_x$ (translation towards x-axis)

$Y_{new} = Y_{old} + T_y$ (translation towards y-axis)

Translation Matrix :-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

Homogeneous coordinate representation of (X, Y) is (X, Y, 1)

The above translation Matrix is represented as 3x3:

Sundaram

FOR EDUCATIONAL USE

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

2. Rotation:
2D Rotation is a process of rotating an object with respect to an angle in two dimensional plane.
This rotation is achieved by using following rotation equation:-

$$X_{new} = X_{old} \times \cos\theta - Y_{old} \times \sin\theta$$
$$Y_{new} = Y_{old} \times \sin\theta + Y_{old} \times \cos\theta$$

In Matrix form, the above rotation equation is represented as:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} \cos & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

for homogeneous co-ordinates, the above rotation Matrix is represented as 3×3:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

Scaling:-
Scaling is the process of modifying or altering the size of objects. It is usually used to increase or reduce the size of the object.
Scaling subjects the coordinates points of the original to change.

→ If scaling factor > 1, then object size is increased
→ If scaling factor < 1, then object size is reduced

Scaling Equation :-

$$X_{new} = X_{old} \times S_x$$
$$Y_{new} = Y_{old} \times S_y$$

In matrix form, the above scaling equation is :-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

for homogenous coordinates,

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

4 Reflection :-

Reflection is a kind of rotation where the angle of rotation is 180 degree. 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

→ Reflection Equation :- (Reflection on X-axis)

$$X_{new} = X_{old}$$
$$Y_{new} = Y_{old}$$

Matrix form :-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Homogenous coordinates for 3x3:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

→ Reflection Equation:- (Reflection on Y-axis)

$X_{new} = -X_{old}$

$Y_{new} = Y_{old}$

Matrix:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Homogenous Coordinate as 3x3:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

5 Shear:-

2D Shearing is an ideal technique to change the shape of an existing object in two dimensional plane.

In two dimensional plane, the object size can be changed along x direction as used as Y direction

Two versions of Shearing:-

a Shearing in X-direction:

Shearing in X-axis is achieved by the following eq^n

$X_{new} = X_{old} + Sh_x \times Y_{old}$

$Y_{new} = Y_{old}$

Matrix form:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & Sh_x \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Homogeneous coordinate:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

b. Shearing in Y-direction:-

Shearing in Y axis is achieved by:-

$X_{new} = X_{old}$

$Y_{new} = Y_{old} + Sh_y \times X_{old}$

Matrix formed:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Sh_y & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Homogeneous Coordinate:-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ 1 \end{bmatrix}$$

Conclusion: Hence, we have successfully performed the lab practical based on 2D transformation and executed the program using switch case.

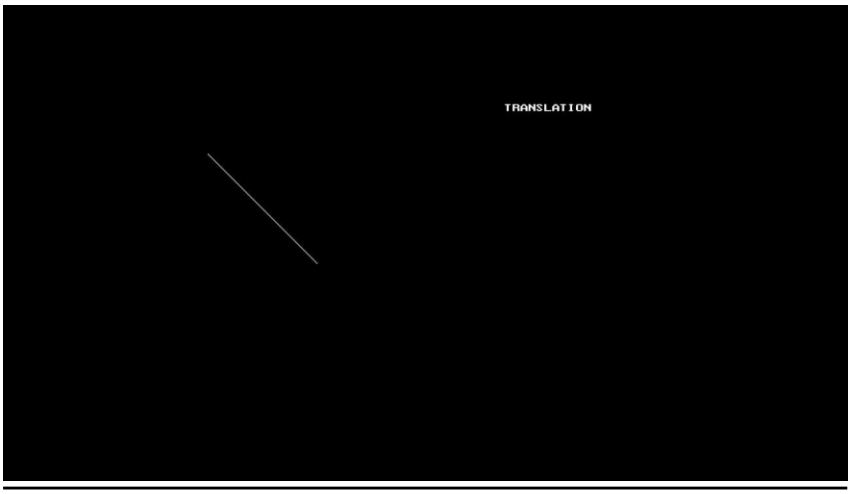
Program:

```
#include<stdio.h>
#include<graphics.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>

int x1,y1,x2,y2;

void translation() {
int  tx,ty,xn1,xn2,yn1,yn2;
printf("\n Enter the translation\n");
scanf("%d%d",&tx,&ty);
cleardevice();
outtextxy(400,100,"TRANSLATION");
xn1=x1+tx;
yn1=y1+ty;
xn2=x2+tx;
yn2=y2+ty;
line(x1,y1,x2,y2);
line(xn1,yn1,xn2,yn2);
getch();
}

void scaling() {
int xn1,xn2,yn1,yn2;
float sx,sy;
printf("Enter the scaling factor");
scanf("%f%f",&sx,&sy);
cleardevice();
outtextxy(300,200,"SCALING");
xn1=x1*sx;
yn1=y1*sy;
xn2=x2*sx;
yn2=y2*sy;
line(x1,y1,x2,y2);
line(xn1,yn1,xn2,yn2);
getch();
}


void rotation() {
int r;
float rx,xn1,xn2,yn1,yn2;
printf("\n enter the angle for rotation");
scanf("%d",&r); cleardevice();
outtextxy(500,200,"ROTATION");
rx=(r*3.14)/180;
xn1=x1*cos(rx)-y1*sin(rx);
yn1=y1*cos(rx)+x1*sin(rx);
xn2=x2*cos(rx)-y2*sin(rx);
yn2=y2*cos(rx)+x2*sin(rx);
line(x1,y1,x2,y2);
line(xn1,yn1,xn2,yn2);
getch();
}


void shearing() {
float sh;
float xn1,xn2,yn1,yn2;
printf("\n Enter the value for shearing");
scanf("%f",&sh);
```

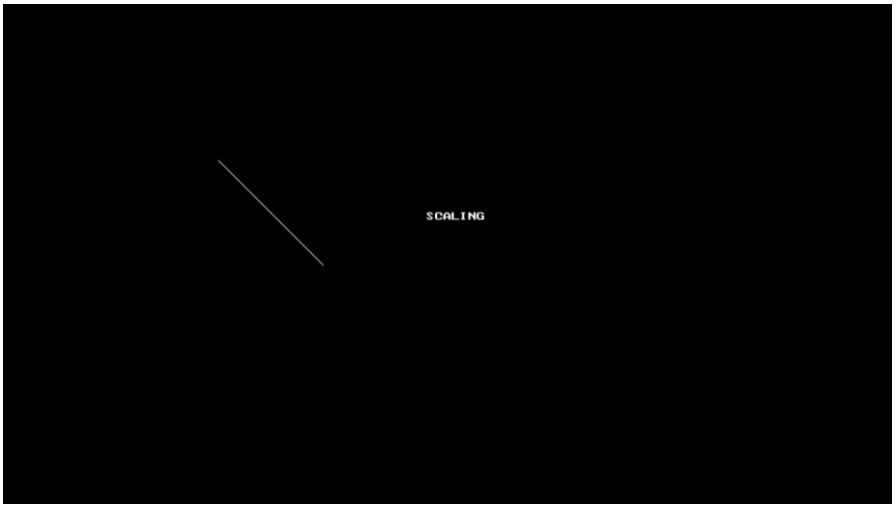
```
cleardevice();
outtextxy(500,100,"SHEARING");
xn1=x1+sh*y1;
yn1=y1;
xn2=x2+sh*y2;
yn2=y2;
line(x1,y1,x2,y2);
line(xn1,yn1,xn2,yn2);
getch();
}


void reflection() {
int xn1,xn2,yn1,yn2;
cleardevice();
outtextxy(300,100,"REFLECTION");
if((x1<y1)^(x2<y2)) {
xn1=x1+50;
xn2=x2+50;
yn1=y1;
yn2=y2;
} else {
xn1=x1;
xn2=x2;
yn1=y1+50;
yn2=y2+50;
}
line(x1,y1,x2,y2);
line(xn1,yn1,xn2,yn2);
getch();
}


void get() {
printf("\n Enter the coordinates x1,y1,x2,y2");
scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
outtextxy(200,100,"ORIGINAL OBJECT");
line(x1,y1,x2,y2); getch();
}


void main() {
int ch,gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
get();
do {
cleardevice();
outtextxy(10,10,"1)TRANSLATION");
outtextxy(10,20,"2)SCALING");
outtextxy(10,30,"3)ROTATION");
outtextxy(10,40,"4)SHEARING");
outtextxy(10,50,"5)REFLECTION");
outtextxy(10,60,"6)EXIT");
outtextxy(10,70,"ENTER UR CHOICE:");
scanf("%d",&ch);
switch(ch) {
case 1: translation(); break;
case 2: scaling(); break;
case 3: rotation(); break;
case 4: shearing(); break;
case 5: reflection(); break;
case 6: exit(0);
}
} while(ch<6);
}
```

Output:

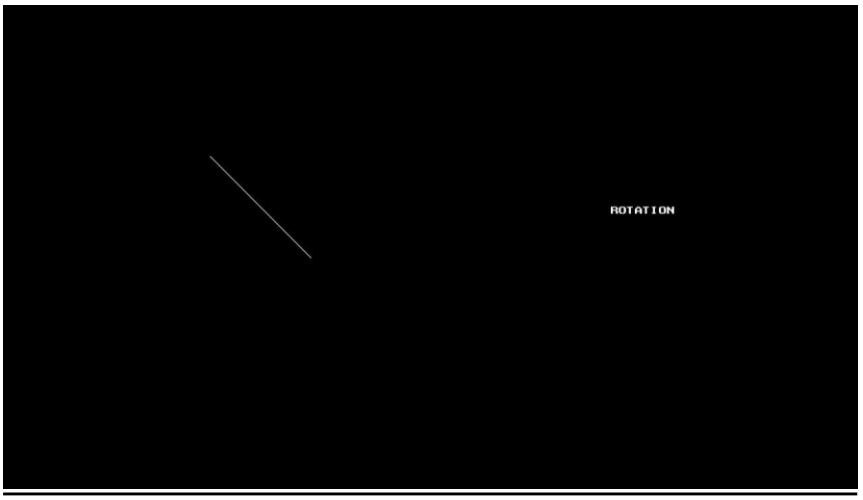
1. TRANSLATION:-



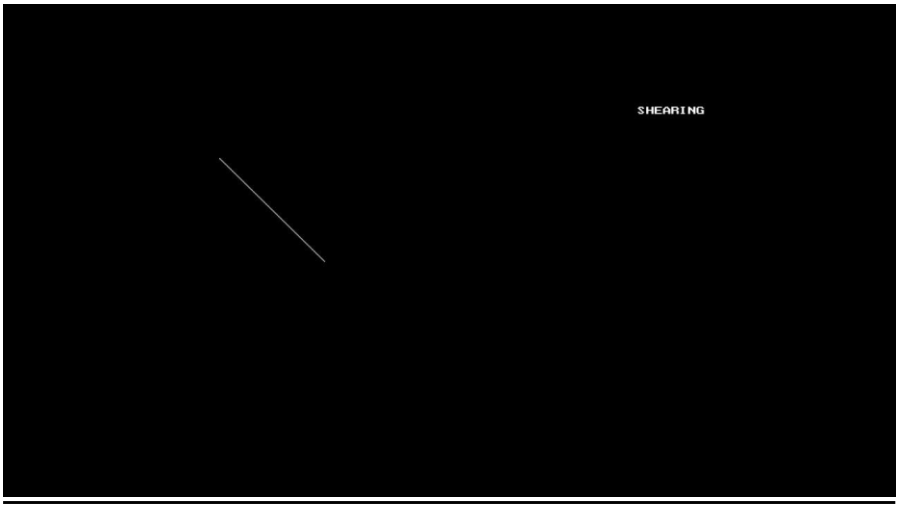
2. **SCALING:-**



3. **ROTATION:-**



4. **SHEARING:-**



5