|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |



Semest

er



S.E. Semester III



–



Computer Engineering



Subj



Computer Graphics



Subject Professor In



-



char



Prof. Sachin Deshpande



Assisting

Teachers



Prof. Sachin Deshpande



|  |  |
| --- | --- |
| Student Name | Trisha Shah |
| Roll Number | 20102A0004 |

**EXPERIMENT DETAILS:**

|  |  |
| --- | --- |
| Title | Write a program to implement 3D Transformations |
| Tools/Language Used | Dev C++ |

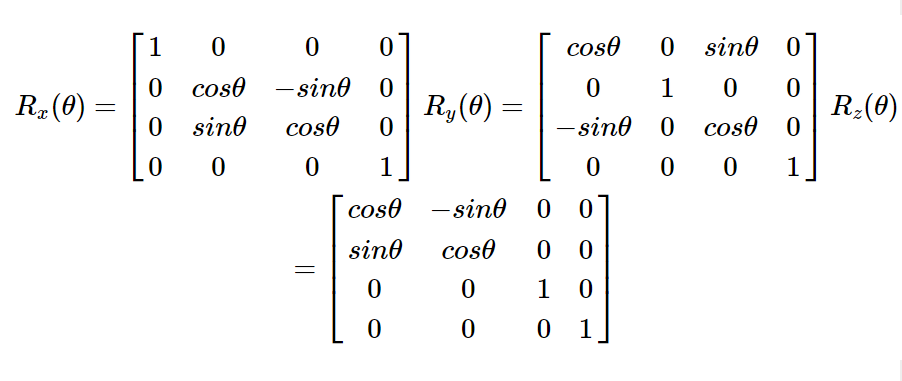
3D Transformations take place in a three dimensional plane.

3D Transformations are important and a bit more complex than 2D Transformations.

Transformations are helpful in changing the position, size, orientation, shape etc of the object.

**Rotation**

3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axes. They are represented in the matrix form as below –



Scaling

You can change the size of an object using scaling transformation. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result.

Shear

A transformation that slants the shape of an object is called the shear transformation. Like in 2D shear, we can shear an object along the X-axis, Y-axis, or Z-axis in 3D.

Transformation Matrices

Transformation matrix is a basic tool for transformation. A matrix with n x m dimensions is multiplied with the coordinate of objects. Usually 3 x 3 or 4 x 4 matrices are used for transformation. For example, consider the following matrix for various operation.

**Program:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

int maxx,maxy,midx,midy;

void axis()

{

getch();

cleardevice();

line(midx,0,midx,maxy);

line(0,midy,maxx,midy);

}

void main()

{

int gd,gm,x,y,z,ang,x1,x2,y1,y2;

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"C:/TC/BGI");

setfillstyle(3,25);

maxx=getmaxx();

maxy=getmaxy();

midx=maxx/2;

midy=maxy/2;

outtextxy(100,100,"ORIGINAL OBJECT");

line(midx,0,midx,maxy);

line(0,midy,maxx,midy);

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

axis();

outtextxy(100,20,"TRANSLATION");

printf("\n\n Enter the Translation vector: ");

scanf("%d%d",&x,&y);

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

bar3d(midx+(x+100),midy-(y+20),midx+(x+60),midy-(y+90),20,5);

axis();

outtextxy(100,20,"SCALING");

printf("\n Enter the Scaling Factor: ");

scanf("%d%d%d",&x,&y,&z);

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

bar3d(midx+(x\*100),midy-(y\*20),midx+(x\*60),midy-(y\*90),20\*z,5);

axis();

outtextxy(100,20,"ROTATION");

printf("\n Enter the Rotation angle: ");

scanf("%d",&ang);

x1=100\*cos(ang\*3.14/180)-20\*sin(ang\*3.14/180);

y1=100\*sin(ang\*3.14/180)+20\*sin(ang\*3.14/180);

x2=60\*cos(ang\*3.14/180)-90\*sin(ang\*3.14/180);

y2=60\*sin(ang\*3.14/180)+90\*sin(ang\*3.14/180);

axis();

printf("\n After rotating about z-axis\n");

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

bar3d(midx+x1,midy-y1,midx+x2,midy-y2,20,5);

axis();

printf("\n After rotating about x-axis\n");

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

bar3d(midx+100,midy-x1,midx+60,midy-x2,20,5);

axis();

printf("\n After rotating about y-axis\n");

bar3d(midx+100,midy-20,midx+60,midy-90,20,5);

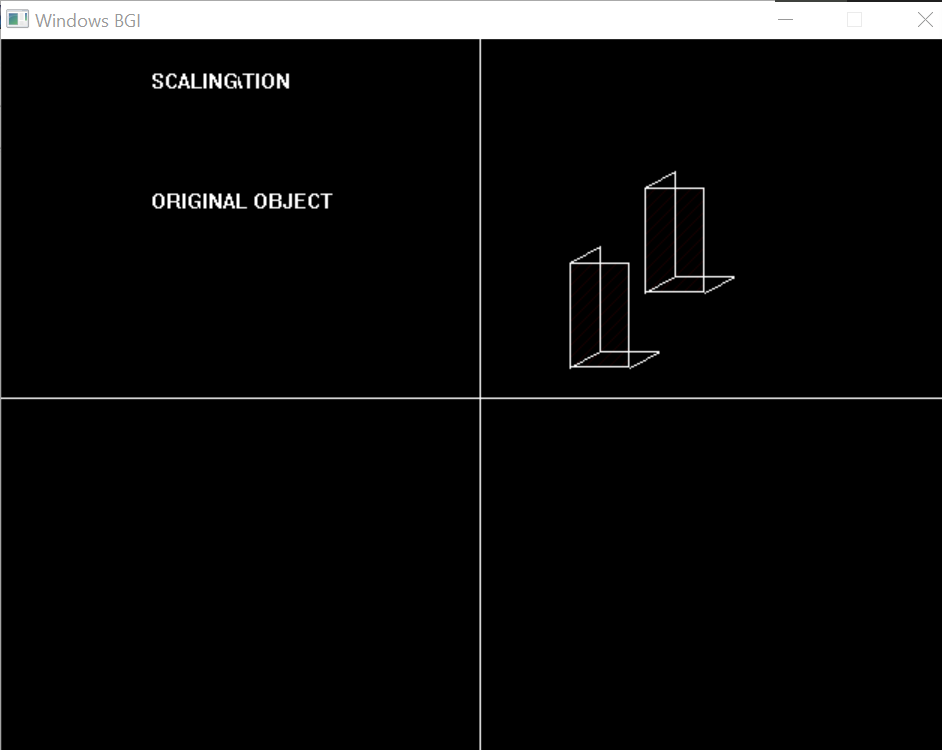
bar3d(midx+x1,midy-20,midx+x2,midy-90,20,5);

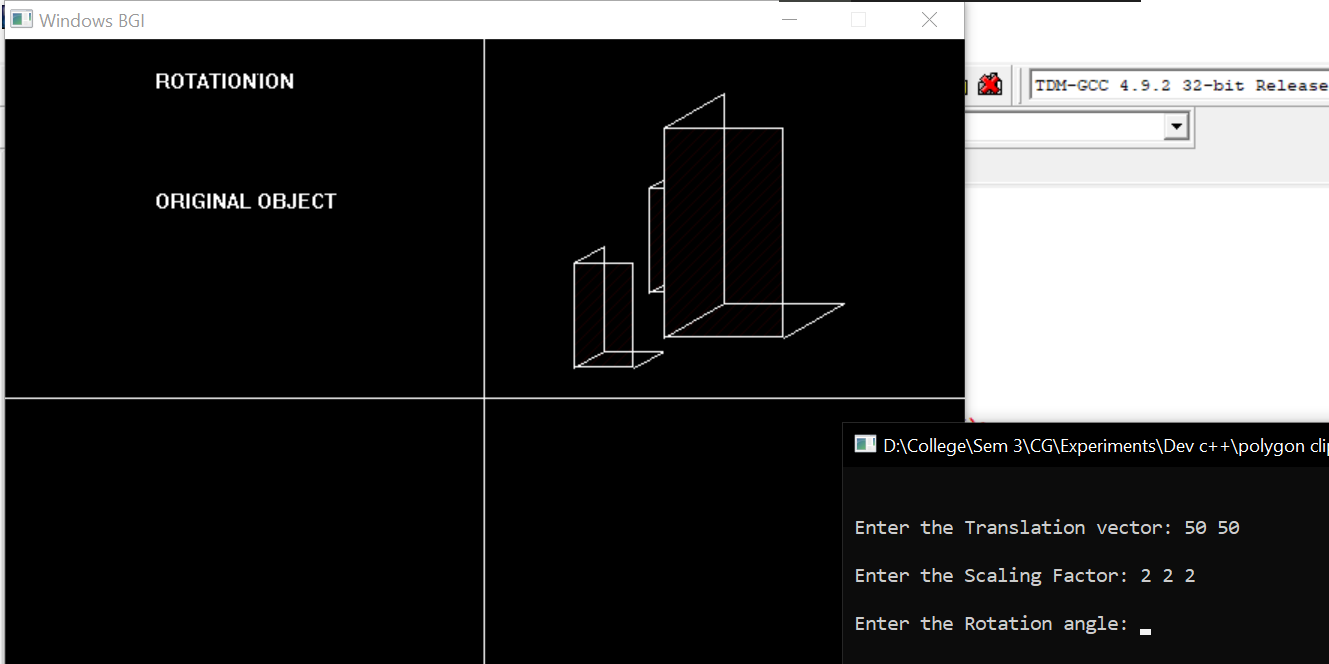
axis();

closegraph();

}

**Output:**

****

****