**Practical No.11**

|  |
| --- |
| Write a C program for 3D Translation and Scaling . |

***Title:-***

***Course outcome*** :- Perform and demonstrate basic and composite graphical transformations on given object.

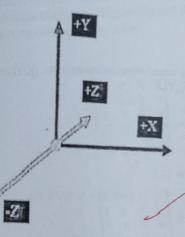
***Resources Required (Hardware & Software):-***

1. A Desktop PC/ Laptop
2. ANSI C/ Turbo C/ (Any distribution) installed

***Theory:-***

In the 2D system, we use only two coordinates X and Y but in 3D, an extra coordinate Z is added. 3D graphics techniques and their application are fundamental to the entertainment, games, and computerized design in industries. It is a continuing area of research in scientific visualization.

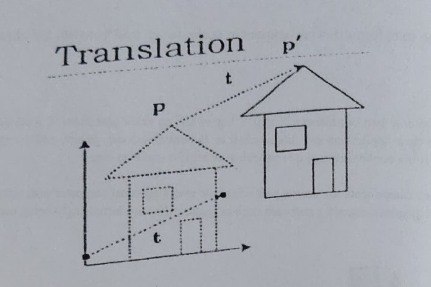
Furthermore, 3D graphics components are now a part of almost every personal computer and, although additionally intended for graphics-intensive software such as games, they are increasingly being used by other applications.



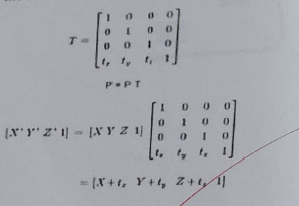
***Translation:***

In 3D translation, we transfer the Z coordinate along with the X and Y coordinates. translation in 3D is similar to 2D translation. At translation moves an object into a The process for different position On the screen.

The following figure shows the effect of translation -

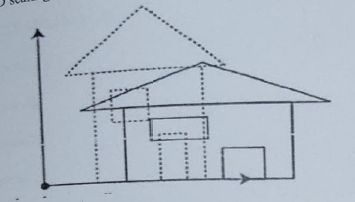


A point can be translated in 3D by adding translation coordinate (tx, ty,tz) to the original Coordinate X,Y,Z to get the new coordinate X,Y2.

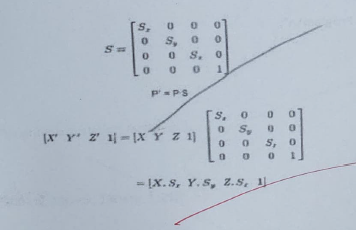


***Scaling:***

You can change size of an object using scaling transformation. In the scaling process, you can either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with scaling factor to get desired result. The following figure show the effect of 3D scaling.



In 3D scaling operation, three coordinates are used. Let us assume that the original coordinates are YYZ, Scaling factors are (Sx,Sy, Sz) respectively, and the pro duced coordinates are X,Y,Z'. This can be mathematically represented as shown below



**Program for 3D Translation & Scaling**

#include<stdio.h>

#include<dos.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

#include<math.h>

#include<iostream.h>

int main()

int gd=DETECT,gm;

int ans=0;

while(ans!l=4)

initgraph(&gd, &gm,"c\\turboc3\bgi");

cleardevice();

cout<<"\n-- -Welcome to The Program\n";

cout<<"\n1. Translation";

cout<<"\n2. Scaling";

cout<<"\n3. Exit";

cout<<"\n\nEnter your choice: ";

cin>>ans;

sleep(2);

cleardevice ();

xmax=getmaxx():

ymax=getmaxy():

setcolor(RED);

bar3d(100, 200, 190, 1 30, 1 5, 1):

xmax=getmaxx();

ymax=getmaxy();

int xmid=xmax/2;

int ymid=ymax/2;

setcolor(GREEN);

line(0,ymid,xmax,ymid);

line(xmid,0,xmid,ymax);

if(ans==1)

int tx,ty;

outtextxy (0,0,"\n enter translation factor \n");

cin>>tx>ty;

bar3d(100+tx, 200+ty, 190+tx, 130+ty, 15,1);

sleep(5);

else if (ans==2)

int sx,sy;

outtextxy(o,0,"\n enter scalling factor \n");

cin>>sx>>sy;

if(sx > 0 && sy>0)

bar3d(100\*sx), (200\*sy), (190\*sx), (130\*sy), 20,5);

else if(sx<0 && sy<0)

float neg\_Sx =1.0/ (-sx);

float neg\_sy =1.0 -sy);

bar3d((100\* neg\_sx), (200\*neg\_sy),(190\*neg\_sx),(130\*neg\_sy), 20, 5);

else

outtextxy(0,0,"Both scaling fcator should have the same sign for uniform scaling"):

}

sleep(2);

sleep(5);

else if(ans=-3)

exit(1);

else

outtextxy(50,50,"\n\nEnter a valid choice: ");

getch();

return 0;}

**Output : - ( Paste your own Output )**

***Conclusion:-*** Thus, we have written a C program for 3D Translation and Scaling .