EXPERIMENT NO.12

**Aim** : To Project a 3D object on a Projection plane Theory :

-Explain parallel 3D projections with an example

* Parallel 3D projection are also known as orthographic projection.
* These are the type of projection where objects in a 3D scene are projected onto a 2D plane in such a way that lines that are parallel in 3D space remain parallel in the 2D projection
* This type of projection is commonly used for technical drawings and CAD applications.

Example:

Consider a simple object : A rectangular Box

* Front view :In the front view we would see the front face of the box.
* Side view : In the side view ,we would see the side face of the box.

-Explain perspective 3D projections with an example

* Perspective 3D projections are a type of projection that simulates the way objects appear in the real world ,taking into account the eﬀects of perspective and depth.
* Unlike parallel 3d projection ,where parallel lines remain parallel in the projection ,perspective projection create the illusion of depth and convergence of parallel lines towards a vanishing point .
* This eﬀect make objects in the foreground appear larger and those in the background appear smaller ,just as they do in reality.

Example

Consider the drawing of a simple box in two point perspective

* You draw vanishing points on the horizon line
* The front face of the box , facing the viewer ,is drawn in a standard manner with straight ,parallel lines.
* The sides of the box which are not facing the viewer directly, are drawn with lines that converge towards the perspective vanishing point on the horizon line.
* The top and bottom faces of the box are drawn as horizontal lines

-Pseudo code for perspective projection of 3D object.

* Step1: Define the 3D object.
* Step2: Define the viewers position.
* Step3: Set projection parameters.
* Step4: Iterate through vertices.
* For each vertex in the 3D object, translate the vertex to the viewers co- ordinate system.
* Apply the perspective transformation.
* Map the projected point on the 2D screen space
* Step5: Clipping
* Step6: Draw the vertex on the 2D screen at the calculated screen coordinates
* Step7: Repeat steps 4-6 for all vertices of 3D object.

Write a Program to draw basic graphics construction like line, circle, arc, ellipse and rectangle.

Program:

#include<graphics.h> #include<conio.h> void main()

{

intgd=DETECT,gm;

initgraph (&gd,&gm,"c:\\tc\\bgi"); setbkcolor(GREEN); printf("\t\t\t\n\nLINE"); line(50,40,190,40);

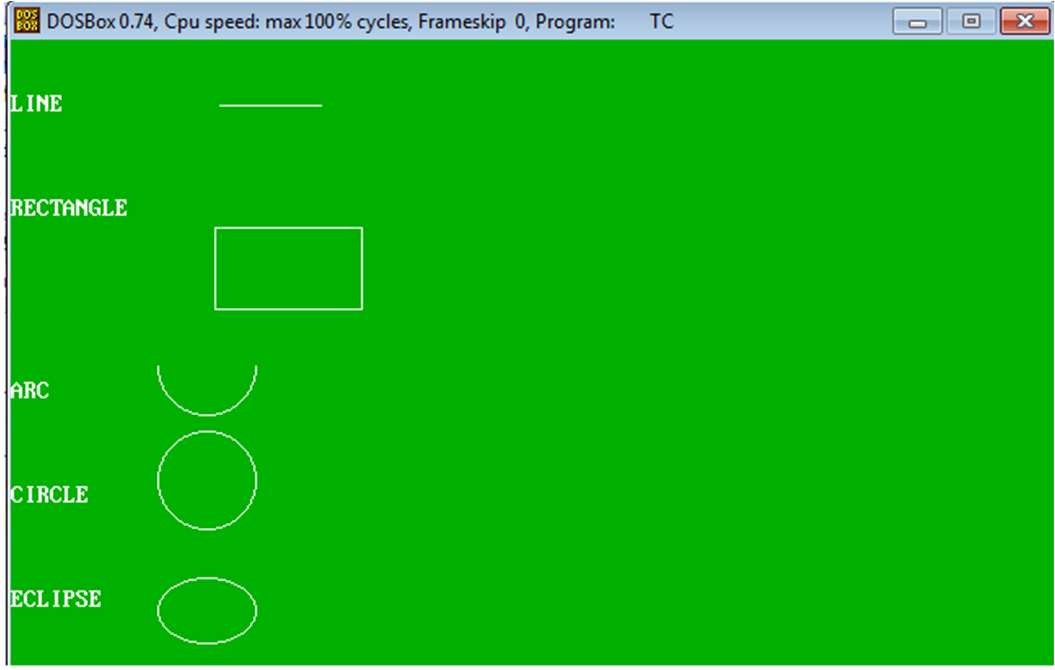
printf("\t\t\n\n\n\nRECTANGLE"); rectangle(125,115,215,165); printf("\t\t\t\n\n\n\n\n\n\nARC"); arc(120,200,180,0,30);

printf("\t\n\n\n\nCIRCLE"); circle(120,270,30); printf("\t\n\n\n\nECLIPSE"); ellipse(120,350,0,360,30,20);

getch();

}

Output:



Conclusion: I have understood how to implement fractals in CG.