**1.Breshnam line drawing**

#include <GL/glut.h> #include <stdio.h>

int x1, y1, x2, y2;

void myInit()

{ glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0, 500, 0, 500); }

void draw\_pixel(int x, int y)

{ glBegin(GL\_POINTS); glVertex2i(x, y); glEnd(); }

void draw\_line(int x1, int x2, int y1, int y2)

{ int dx, dy, i, e; int incx, incy, inc1, inc2;

int x,y; dx = x2-x1; dy = y2-y1;

if (dx < 0) dx = -dx;

if (dy < 0) dy = -dy;

incx = 1;

if (x2 < x1) incx = -1;

incy = 1;

if (y2 < y1) incy = -1;

x = x1; y = y1;

if (dx > dy)

{ draw\_pixel(x, y); e = 2 \* dy-dx;

inc1 = 2\*(dy-dx); inc2 = 2\*dy;

for (i=0; i<dx; i++)

{ if (e >= 0)

**7.3D SIERPINSKI GASKET**

#include <stdlib.h> #include <stdio.h> #include <GL/glut.h> typedef GLfloat point[3];

point v[]={{-1.0,-0.5,0.0},{1.0,-0.5,0.0},{0.0,1.0,0.0},

{0.0,0.0,1.0}};GLfloatcolors[4][3]={{1.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0},{1.0,1.0,0.0}}; int n;

void triangle(point a,point b,point c

{ glBegin(GL\_POLYGON);

glVertex3fv(a);glVertex3fv(b); glVertex3fv(c);glEnd(); } void tetra(point a,point b,point c,point d)

{glColor3fv(colors[0]);triangle(a,b,c); glColor3fv(colors[1]);triangle(a,c,d); glColor3fv(colors[2]);triangle(a,d,b); glColor3fv(colors[3]);triangle(b,d,c);

void divide\_tetra(point a,point b,point c,point d,int m)

{ point mid[6];int j; if(m>0) { for(j=0;j<3;j++) {

mid[0][j]=(a[j]+b[j])/2.0; mid[1][j]=(a[j]+c[j])/2.0;

mid[2][j]=(a[j]+d[j])/2.0; mid[3][j]=(b[j]+c[j])/2.0;

mid[4][j]=(c[j]+d[j])/2.0; mid[5][j]=(b[j]+d[j])/2.0;

} divide\_tetra(a,mid[0],mid[1],mid[2],m-1);

divide\_tetra(mid[0],b,mid[3],mid[5],m-1);

divide\_tetra(mid[1],mid[3],c,mid[4],m-1);

divide\_tetra(mid[2],mid[5],mid[4],d,m-1); }

else

tetra(a,b,c,d); } void display()

{glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT); glClearColor(1.0,1.0,1.0,1.0); divide\_tetra(v[0],v[1],v[2],v[3],n);glFlush(); } void myReshape(int w,int h)

{ glViewport(0,0,w,h); glMatrixMode(GL\_PROJECTION); glLoadIdentity();

if(w<=h)

glOrtho(-1.0,1.0,-1.0\*((GLfloat)h/(GLfloat)w),

1.0\*((GLfloat)h/(GLfloat)w),-1.0,1.0); else

glOrtho(1.0\*((GLfloat)w/(GLfloat)h),1.0\*((GLfloat)w/(GLflo at)h),-1.0,1.0,-1.0,1.0);

glMatrixMode(GL\_MODELVIEW); glutPostRedisplay(); } void main(int argc,char \*\* argv)

{ printf( "No of Division?: "); scanf("%d",&n); glutInit(&argc,argv);glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB|GLUT\_DEP TH);

glutInitWindowSize(500,500);

glutCreateWindow( "3D gasket" );glutDisplayFunc(display); glutReshapeFunc(myReshape); glEnable(GL\_DEPTH\_TEST); glutMainLoop(); }

{ y += incy;e += inc1; }

else

e += inc2; x += incx;

draw\_pixel(x, y); } }

else

{ draw\_pixel(x, y); e = 2\*dx-dy;

inc1 = 2\*(dx-dy); inc2 = 2\*dx;

for (i=0; i<dy; i++)

{ if (e >= 0)

{ x+= incx; e += inc1; }

else

e += inc2;y += incy;

draw\_pixel(x, y); } } }

void myDisplay() { glClear(GL\_COLOR\_BUFFER\_BIT);

draw\_line(x1, x2, y1, y2);glFlush();}

int main(int argc, char \*\*argv)

{printf( "Enter end points of the Line (x1, y1, x2, y2)\n");

scanf("%d %d %d %d", &x1, &y1, &x2, &y2);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500, 500);glutInitWindowPosition(0, 0);

glutCreateWindow("Bresenham's Line Drawing");

myInit();glutDisplayFunc(myDisplay);glutMainLoop();

return 0; }