East Delta University

Experiment Name: Implementation of Sierpinsky gasket

Name: Pranesh Chowdhury

ID: 202003112

Section: 02

Instructor Name: Tasnimatul Jannah

Course Name: CSE 322 Computer Graphics Lab

Title : Implementation of Sierpinsky Gasket

Introduction :

Sierpinsky gasket algorithm is exercise to understand recursion and plotting shapes. It is a fractal and attractive fixed set with the overall shape of an equilateral triangle. It sub divides recursively into smaller triangle.

Description :

First I have taken three points for the vertices of the triangle taken the order n and used the function to draw a triangle. This fractal triangle composed of smaller and smaller triangles and divided into four smaller triangle and divided into four smaller equilateral triangles using the mid points of the three sides of the original triangles triangle as the new vertices. Remove the interior of the middle triangle and to get S(1). Then we will get the sienpinski gasket congruent equilateral triangle.

Conclusion:

In this experiment, Sierpinsky gasket algorithm I learned how a pattern can repeat again at a different scales and how this complex shape can be formed by the simple repetition. I didn't encounter any difficulties during the implementation.

Code:

```cpp
#include<windows.h>
#include<GL/glut.h>
#include<stdlib.h>
#include<stdio.h>
#include<bits/stdc++.h>
#include<math.h>

float x1,x2,x3,z1,y2,y3;
int n;

void triangle(float x1, float z1, float x2, float y2, float x3, float y3)
{
    glBegin(GL_TRIANGLES);
    glVertex2f(x1,z1);
    glVertex2f(x2,y2);
    glVertex2f(x3,y3);
    glEnd();
}
void SG(float x1, float z1, float x2, float y2, float x3, float y3, int n)
{
    float x12,y12,x13,y13,x23,y23;
    if(n>0)
    {
        x12=(x1+x2)/2;
        y12=(z1+y2)/2;
        x13=(x1+x3)/2;
        y13=(z1+y3)/2;
        x23=(x2+x3)/2;
        y23=(y2+y3)/2;
        SG(x1,z1,x12,y12,x13,y13,n-1);
        SG(x12,y12,x2,y2,x23,y23,n-1);
        SG(x13,y13,x23,y23,x3,y3,n-1);
    }
    else
    {
        triangle(x1,z1,x2,y2,x3,y3);
    }
```

```cpp
        SG(x13,y13,x23,y23,x3,y3,n-1);
    }
    else
    {
        triangle(x1,z1,x2,y2,x3,y3);
    }
}
void display(void)
{
    SG(x1,z1,x2,y2,x3,y3,n);
    glFlush();
}
void init(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(0,0,0,0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-100,100,-100,100);
}
int main(int argc, char** argv)
{
    printf("Enter the three points: \n");
    scanf("%f %f %f %f %f %f", &x1, &z1, &x2, &y2, &x3, &y3);
    scanf("%d", &n);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Gasket");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```