CGPractical7b

#include <GL/glut.h>

#include <iostream>

#include <cstdlib>

#include <cmath>

using namespace std;

GLfloat oldx = 500, oldy = 800;

GLint i = 3;

void koch(GLdouble dir, GLfloat len, GLint itr)

{

GLfloat newx, newy;

GLdouble rdir = 3.14159265358979323846 / 180.0 \* dir;

newx = oldx + len \* cos(rdir);

newy = oldy + len \* sin(rdir);

if (itr == 0)

{

glVertex2f(oldx, oldy);

glVertex2f(newx, newy);

oldx = newx;

oldy = newy;

}

else

{

itr--;

koch(dir, len / 3, itr);

dir += 60;

koch(dir, len / 3, itr);

dir -= 120;

koch(dir, len / 3, itr);

dir += 60;

koch(dir, len / 3, itr);

}

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1, 0, 0);

glBegin(GL\_LINES);

// Triangle base coordinates (starting Koch Snowflake)

oldx = 300; oldy = 400;

koch(0, 400, i); // Bottom side

oldx = 700; oldy = 400;

koch(120, 400, i); // Right side

oldx = 500; oldy = 400 + 400 \* sqrt(3) / 2;

koch(-120, 400, i); // Left side

glEnd();

glFlush();

}

void mouse(int button, int state, int x, int y)

{

if (state == GLUT\_DOWN)

{

if (button == GLUT\_RIGHT\_BUTTON)

{

i++;

glutPostRedisplay(); // Refresh after increasing iteration

}

else if (button == GLUT\_LEFT\_BUTTON)

{

oldx = x;

oldy = fabs(720 - y);

display();

}

else if (button == GLUT\_MIDDLE\_BUTTON)

{

i = 3;

glClear(GL\_COLOR\_BUFFER\_BIT);

glFlush();

}

}

}

void init()

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0, 1280, 0, 720);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGBA);

glutInitWindowSize(1280, 720);

glutCreateWindow("Koch Snowflake - Practical 7");

init();

cout << "Left mouse button to draw Koch's Snowflake.\n";

cout << "Right Mouse button to increase the iterations.\n";

cout << "Middle mouse button to clear the screen.\n";

glutDisplayFunc(display);

glutMouseFunc(mouse);

glutMainLoop();

return 0;

}