**Національний технічний університет України**

**“Київський політехнічний інститут”**

**Факультет прикладної математики**

**Кафедра спеціалізованих комп’ютерних систем**

**Лабораторна робота № 4**

з дисципліни «Комп’ютерна графіка»

Виконав: Місячний Ігор.

Студент групи КВ-51

Перевірив(ла)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**м. Київ 2017**

**Завдання**

Відтворити на екрані 3 групи фракталів:

1) Сніжинка «Коха»

2) Множина Мальдебронта

3) Трикутник Серпинського

**Текст програми**

$(document).ready(function () {

var ctx = document.getElementById('triangle').getContext('2d');

var top = new Point(500 / 2, 0),

left = new Point(0, 500),

right = new Point(500, 500);

drawTriangle(5, new Triangle(top, left, right), ctx);

});

class Point {

constructor(x, y) {

this.x = x;

this.y = y;

}

}

class Triangle {

constructor(top, left, right) {

this.top = top;

this.left = left;

this.right = right;

}

};

function draw(triangle, ctx) {

ctx.fillStyle = '#000000';

ctx.beginPath();

ctx.moveTo(triangle.top.x, triangle.top.y);

ctx.lineTo(triangle.left.x, triangle.left.y);

ctx.lineTo(triangle.right.x, triangle.right.y);

ctx.closePath();

ctx.fill();

}

function getMiddle(p1, p2) {

return new Point((p1.x + p2.x) / 2, (p1.y + p2.y) / 2);

}

function drawTriangle(level, triangle, ctx) {

if (level == 0) {

draw(triangle, ctx);

return;

}

var leftMid = getMiddle(triangle.top, triangle.left),

rightMid = getMiddle(triangle.top, triangle.right),

topMid = getMiddle(triangle.left, triangle.right);

drawTriangle(level - 1, new Triangle(triangle.top, leftMid, rightMid), ctx);

drawTriangle(level - 1, new Triangle(leftMid, triangle.left, topMid), ctx);

drawTriangle(level - 1, new Triangle(rightMid, topMid, triangle.right), ctx);

}

$(document).ready(function () {

var ctx = document.getElementById('carpet').getContext('2d');

carpet(5, new Rectangle(0, 0, 450, 450), ctx);

});

class Rectangle {

constructor(x, y, height, width) {

this.x = Math.round(x);

this.y = Math.round(y);

this.height = Math.round(height);

this.width = Math.round(width);

}

};

function draw(rect, ctx) {

ctx.fillStyle = '#000000';

ctx.fillRect(rect.x, rect.y, rect.height, rect.width);

ctx.stroke();

}

function carpet(level, rect, ctx) {

if (level == 0) {

draw(rect, ctx);

return;

}

var height = rect.height / 3,

width = rect.width / 3,

x1 = rect.x, x2 = x1 + width,

x3 = x1 + 2 \* width,

y1 = rect.y, y2 = y1 + height,

y3 = y1 + 2 \* height;

carpet(level - 1, new Rectangle(x1, y1, width, height), ctx);

carpet(level - 1, new Rectangle(x2, y1, width, height), ctx);

carpet(level - 1, new Rectangle(x3, y1, width, height), ctx);

carpet(level - 1, new Rectangle(x1, y2, width, height), ctx);

carpet(level - 1, new Rectangle(x3, y2, width, height), ctx);

carpet(level - 1, new Rectangle(x1, y3, width, height), ctx);

carpet(level - 1, new Rectangle(x2, y3, width, height), ctx);

carpet(level - 1, new Rectangle(x3, y3, width, height), ctx);

}

const WIDTH = 1000;

const HEIGHT = 1000;

const STEP\_X = 10;

const STEP\_Y = 10;

const SX = 0.005;

const SY = 0.005;

const DX = -250;

const DY = -135;

const COUNT\_ITER = 1000;

const BAIL\_OUT = 30;

var canvas1 = document.getElementById('cloud');

var context1 = canvas1.getContext('2d');

context1.beginPath();

context1.stroke();

context1.closePath();

drawClouds(context1, HEIGHT, WIDTH);

function draw(x, y, ctx) {

ctx.fillStyle = '#' + (Math.random() \* 0xFFFFFF << 0).toString(16);

ctx.fillRect(Math.round(x), Math.round(y), 1, 1);

ctx.stroke();

}

function drawClouds(ctx, height, width) {

for (var i = 0; i < width; i += STEP\_X) {

for (var j = 0; j < height; j += STEP\_Y) {

var c = SX \* (i + DX),

d = SY \* (j + DY),

x = c, y = d, t;

for (var k = 0; x \* x + y \* y < BAIL\_OUT && k < COUNT\_ITER; k++) {

t = x \* x - y \* y + c;

y = 2 \* x \* y + d;

x = t;

draw(400 + x / SX - DX, 100 + y / SY - DY, ctx);

}

}

}

}



