Проект по Фрактали към Факултета по математика и информатика на СУ”Климент Охридски”

Изготвил:
Диляна Пламенова Тодорова
71128

Снежинка на Кокс

Рекурсивно изчертаване на фрактал, който се съдържа в себе си

Снежинката на Кокс е фрактал описан от шведския математик Нийлс Фабиан вон Кокс през 1904 година в неговия труд върху теорията на числата. Снежинката на Кокс е един от най-известните фрактали и е лесен за разбиране.

Крива на Кокс

Кривата на Кокс представлява в началото една права, която се разделя на три равни части. Средната част се използва за основа на равностранен триъгилник, след което основата на този триъгълник се премахва. Така се образуват четири нови прави. На следващата стъпка всяка от тези прави се разделя аналогично на още четири прави. Така кривата на Кокс може да расте безкрайно.

Снежинката на Кокс

Снежинката на Кокс представлява в началото един равностранен триъгълник, като всяка една от страните му представлява крива на Кокс.

Реализация на изчертаването на снежинката на Кокс

Технологии и системни изисквания

За реализацията на фрактала са използвани съвременни уеб технологии - HTML5, JavaScript и CSS3 за оформянето на външния вид. За изчертаването на фрактала е нужно да се стартира файлът snowflake.html. За да може да се използва програмата е нужно да се отвори с последно поколение браузър като firefox, chrome или safari.

Математически изчисления

Проекта използва рекурсивен алгоритъм за изчисляване на три точки и образуване на крива на Кокс. Алгоритъмът се прилага върху трите страни на равностранния триъгълник в началото на нулевата итерация, след което рекурсивно се прилага върху всяка една от новообразуваните страни. Така може да бъде прилаган безброй пъти, но поради ограничението на паметта и експоненциалното растене на функцията, съвременен браузър се справя добре с изчислението на 7 итерации.
Алгоритъма започва като изчертае равностранен триъгълник. Образуването на следващите итерации става чрез навигационните бутони следваща и предишна итерация, което подтиква ново изчертаване на фрактала с дълбочина на рекурсията съответната текуща итерация.
Нека да разгледаме изчисленията, които се извършват за една от страните на равностранния триъгълник (снежинката на Кокс в нулевата итерация). Изчисленията са напълно аналогични за останалите страни.
Нека да разгледаме страната AB. Тя се приема за права с начална точка A с координати (50, 150) и крайна точка B с координати (500,150). Първо се намират координатите на точката C, която разделя правата AB в съотношение 1:2. От аналитичната геометрия следва че координатите на точката C, която е част от правата AB са следните
$x = (2x_a + x_b) / 3$, $y = (2y_a + y_b) / 3$, където x_a и x_b са съответно абсцисите на точките A и B, y_a и y_b са ординатите съответно на точките A и B. Следва намирането на координатите на точката D, която разделя правата AB в отношение 2:1. Правата AB може да се приеме и за вектор, като точката D разделя вектора BA също в отношение 1:2, което означава че може да се приложат същите изчисления като за точката C, само че с обратния вектор.
Програмното реализиране е чрез използването на прототип в езика javaScript за задаването на обект представляваш точка с координати x и y и помошни функции за умножаване на координатите със скалар, събиране на координатите и делението и на двете координати на скалар.

```
var C = divide(add(multiply(A, 2), B), 3);

function multiply(v, num){
return { x: v.x * num, y: v.y * num};
};

function add(a, b){
return { x: a.x + b.x, y: a.y + b.y };
};
```

```
function divide(v, num){
return { x: v.x/num, y: v.y / num };
};
```

и съответно за точката D се извършва същото изчисление, но с обратния вектор BA

```
var D = divide(add(multiply(B, 2), A), 3);
```

Следва намирането на точката F, която служи за връх на равностранен триъгълник със основа CD.
Точката F лежи върху права, която е перпендикулярна на AB и минава през средата на AB. Тоест точката F лежи върху нормализирания вектор на AB. Ако M е средата на вектора AB, то дължината на MF може да се изрази като височина в равностранен триъгълник със страна 1/3 от AB, което е AB√3/6.

Следователно за намирането на координатите на точката F имаме следната поредица от действия:

Намираме средата на AB точката M, която дели отсечката в отношение 1:1

```
var M = divide(add(A, B), 2)
```

Намираме единичния вектор на AB

```
var E = divide(minus(M, A), length(M, A));
```

където дължината се изчислява по формулата за разстояние между две точки

```
function length(a, b){
return Math.sqrt(Math.pow(a.x - b.x, 2) + Math.pow(a.y - b.y, 2));
};
```

След което намираме нормализирания вектор на единичния вектор E

```
var N = normala(E);
```

където нормализирания вектор се намира по формулата

```
function normala(v){
return { x: v.y, y: -v.x }
}
```

Точката F се намира върху вече намерения нормализиран вектор N, но както вече уточнихме трябва да е в точката M и да е с дължина AB√3/6, затова имаме следното изчисление:

```
var F = add(multiply(N, Math.sqrt(3)/6 * length(B, A)), M);
```

За красивата форма на фрактала остана само да се изчертаят четирите линии определяни от точките C, D и F и да се премахне основата на равностранния триъгълник правата CD. Изчертаването на четирите прави става рекурсивно, ако имаме още итерации всяка от тези прави ще бъде разделена на три равни части и ще се намери върха на равностранния триъгълник. Премахването на основата в конкретната реализация е осъществена чрез начертаването на права с цвят съвпадаш с фона.

За повече интерактивност е добена и анимация, по време на изчертаването на фрактала.
Цялото изчисление е осъществено вътре в клоужър, който връща функцията, която изчертава фрактала. Целта е да се предотврати извикването на помошните изчислителни функции извън файла с кода на фрактала.
Кода за изчертаването на снежинката на Кокс се намира във файла fractal.js

Свойства на Снежинката на Кокс

Кривите на Кокс, които изграждат фрактала на негово име са известни като безкрайни криви – на всяка итерация кривата се разделя на 4 прави всяка с дължина 1/3 от дължината на оригиналната кривата, следователно нополучената дължина се е увеличила с 1/3. Периметърът на Снежинката на Кокс за стъпка k е $(4/3)^k P_{k-1}$, където P_{k-1} е периметъра от предишната стъпка.
Фракталната размерност е log4/log3 = 1.26
В началото фрактала представлява равностранен триъгълник с лице $\frac{\sqrt{3}}{4}s^2$
където s е страната на този равностранен триъгълник
На следващата итерация към фрактала се добавят нови три равностранни триъгълника със страна s/3
Лицето на новополучената фигура е очевидно сбора на старото лице и новополучените три триъгълника

$$\frac{\sqrt{3}}{4}s^2+3.\frac{\sqrt{3}}{4}\left(\frac{s}{3}\right)^2$$

след преобразование получаваме $$\frac{\sqrt{3}}{4}s^2\left(1+\frac{3}{9}\right)$$

Ако направим абсолютно аналогичните разсъждения и изчисления ще намерим лицето на фигурата в следващата итерация

$$\frac{\sqrt{3}}{4}s^2\left(1+\frac{3}{9}+\frac{3\cdot 4}{9^2}\right)$$

Лесно е да забележим, че имаме геометрична прогресия относно лицата на малките равностранни триъгълничета.
За лицето в итерация n имаме

$$\frac{\sqrt{3}}{4}s^2\left(1+\sum_{k=1}^{n}\frac{3\cdot 4^{k-1}}{9^k}\right)$$

n клони към безкрайност, следователно получаваме

$$\lim_{n\to\infty}\frac{\sqrt{3}}{4}s^2\left(1+\sum_{k=1}^{n}\frac{3\cdot 4^{k-1}}{9^k}\right)$$

След изчисление на израза намираме, че лицето на снежинката на Кокс е

$$\frac{\sqrt{3}}{4}s^2\left(1+\frac{3/9}{1-4/9}\right)$$

което означава след преобразование на израза, че лицето е

$$\frac{2\sqrt{3}}{5}s^2$$

Извода е, че снежинката на кокс има безкраен периметър, но крайно лице

Кода за изчертаването на фрактала Снежинка на Кокс

```
var iteration = 0;

function next_iteration(step)
{
var iterationCounter = document.getElementById("iterationCounter");
iteration = iteration + step < 0 ? 0 : iteration + step ;
iterationCounter.innerHTML = iteration;
init(iteration);
}

var init = (function(){

var W = 550;
var H = 510;
var timeout = 100;
var strokeColor = "#ce534d";
var bgColor = "#ded4b9";
var canvas;

return function(iteration){
        canvas = document.getElementById("imageView");

    context = canvas.getContext('2d');
    context.beginPath();
    context.stroke();
    context.closePath();

        context.clearRect(0, 0, W, H);

    kochCurve({x: 50,  y: 150}, {x: 500, y: 150}, iteration);
    kochCurve({x: 500, y: 150}, {x: 270, y: 490}, iteration);
    kochCurve({x: 270, y: 490}, {x: 50,  y: 150}, iteration);

};

function kochCurve(A, B, iteration){
```

```
    if (iteration < 0){
        return null;
    }

    var C = divide(add(multiply(A, 2), B), 3);
    var D = divide(add(multiply(B, 2), A), 3);
    var M = divide(add(A, B), 2);

    var E = divide(minus(M, A), length(M, A));
    var N = normala(E);

    var F = add(multiply(N, Math.sqrt(3)/6 * length(B, A)), M);


    setTimeout(function(){
        line(A, B, strokeColor);

        if (iteration !=0){
            setTimeout(function(){
                for (var i = 0; i < 7; i++)
                    line(C, D, bgColor);
            }, timeout);
        };

        kochCurve(A, C, iteration-1);
        kochCurve(C, F, iteration-1);
        kochCurve(F, D, iteration-1);
        kochCurve(D, B, iteration-1);

    },timeout);


};

function normala(v){
    return { x: v.y, y: -v.x }
}

function multiply(v, num){
    return { x: v.x * num, y: v.y * num};
};

function divide(v, num){
    return { x: v.x/num, y: v.y / num };
};

function add(a, b){
    return { x: a.x + b.x, y: a.y + b.y };
};
```

```
function minus(a, b){
    return {
        x: a.x - b.x, y: a.y - b.y };
};

function length(a, b){
    return Math.sqrt(Math.pow(a.x - b.x, 2) +
                  Math.pow(a.y - b.y, 2));
};

function line(a, b, c){
    context.beginPath();
    context.strokeStyle = c;
    context.moveTo(a.x, a.y);
    context.lineTo(b.x, b.y);
    context.stroke();
    context.closePath();
};

})();
```

За визуализацията на фрактала и подреждането на визуалните компоненти се грижи файлът snowflake.html, който има следното съдържание:

```
<html>
  <head>
    <title>Koch Snowflake Fractal</title>
    <link rel="stylesheet" title="style" href="style.css">
    <script type="text/javascript" src="fractal.js"></script>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>

  <body onload="init(iteration);">

  <div id="wrapper">
    <div id="header">
        <h1>Снежинка на Кокс</h1>
    </div>
    <div id="container">
    <p class="center biggerfont">Текуща итерация <span id="iterationCounter">0<span></p>
          <canvas id="imageView" width="550" height="510">
          За съжаление използвания от вас браузър не поддържа новите стандарти.
          Моля използвайте по-нова версия
          </canvas>

           <div id="controls">
```

```html
        <ul>
            <li><a class="large button red" onclick="next_iteration(1);">Следваща
итерация</a></li>
            <li><a class="large button red" onclick="next_iteration(-1);">Предишна
итерация</a></li>
        <ul>
    </div>
        </div>
    <div id="kox">
        <div id="kox_information">
        <h1>Нийлс Кокс</h1>
        <p class="indent">Нийлс Кокс е шведски математик, живял през 20ти век.
        През 1904 година Кокс публикува труд, в който описва специален вид - наречена също
на негово име и използвана за образуването на фрактала снежинка.</p>
        <h2>Крива на Кокс</h2>
        <p class="indent">Кокс описва права крива, която се разделя на три равни части.
Средната част на кривата образува равностранен триъгълник, след което средната третина
от отсечката се премахва. Така кривата е вече начупена. Тава представлява първата
итерация. Кривата може да се начупва безкраен брой пъти.</p>
        <h2>Снежинка на Кокс</h2>
        <p class="indent">Снежинката започва като равностранен триъгълник. Всяка една
страна представлява права на Кокс, която се начупва експоненциално, образувайки
снежинка.</p>
        </div>
    </div>

    <div id="footer"><p class="center">Диляна Тодорова, 71128</p>
    <p class="center">ФМИ, 2012</p>
    </div>

    </div>
  </body>
</html>
```

За подредения и издържан вид на проекта се грижи файлът style.css, където чрез css3 са зададени стиловите параметри на всеки един компонент.

```css
body,
html {
        margin:0;
        padding:0;
        color:#46433a;
        background:#ded4b9;
}
#wrapper   { width:850px; margin:0 auto; }
#header    { border-bottom: 3px solid #46433a; margin-bottom: 15px;}

h1 {
        padding-top: 25px;
```

```
	font-size: 45px;
	font-weight:bold;
	text-align: center;
}

h2			{ text-align: center;	}
p			{ padding : 0;	margin : 0; }
#imageView	{ padding : 0; margin : 0; }

#controls ul{
	list-style: none;
	margin: 0;
	padding: 0;
}
#controls	li{ float: right; padding: 45px; }

#container {
	background:#ded4b9;
	width: 550px;
	float: left;
	padding: 0;
}
#imageView		{ padding: 0; margin: 0; }
p.center		{ text-align: center; }
p.biggerfont	{ font-size: 25px; }
p.indent		{ text-indent: 15px; }
#kox{
	float: right;
	background: #64B6B1;
	box-shadow: 2px 1px 10px 2px #46433a;
	width:200px;
	padding: 15px;
}
#kox_information { border: 1px dashed #46433a; padding: 5px; }
#footer { clear: both; }

.button {
	display: inline-block;
	outline: none;
	cursor: pointer;
	text-align: center;
	text-decoration: none;
	font: 14px/100% Arial, Helvetica, sans-serif;
	padding: .5em 2em .55em;
	text-shadow: 0 1px 1px rgba(0,0,0,.3);
	-webkit-border-radius: .5em;
	-moz-border-radius: .5em;
	border-radius: .5em;
	-webkit-box-shadow: 0 1px 2px rgba(0,0,0,.2);
```

```
        -moz-box-shadow: 0 1px 2px rgba(0,0,0,.2);
        box-shadow: 0 1px 2px rgba(0,0,0,.2);
}
.button:hover               { text-decoration: none; }
.button:active              {  position: relative; top: 1px; }

.button, .button:visited {
        background: #222 url(overlay.png) repeat-x;
        display: inline-block;
        padding: 5px 10px 6px;
        color: #fff;
        text-decoration: none;
        -moz-border-radius: 6px;
        -webkit-border-radius: 6px;
        -moz-box-shadow: 0 1px 3px rgba(0,0,0,0.6);
        -webkit-box-shadow: 0 1px 3px rgba(0,0,0,0.6);
        text-shadow: 0 -1px 1px rgba(0,0,0,0.25);
        border-bottom: 1px solid rgba(0,0,0,0.25);
        position: relative;
        cursor: pointer
}
.button:hover                                      { background-color: #111; color: #fff; }
.button:active                                     { top: 1px; }
.large.button, .large.button:visited  { font-size: 14px; padding: 8px 14px 9px; }
.red.button, .red.button:visited             { background-color: #e62727; }
.red.button:hover                                { background-color: #cf2525; }
```