Java Script - Płatek Kocha

© Copyright by 3bird Projects 2021, http://edukacja.3bird.pl

Opis

Płatek Kocha jest fraktalem budowanym za pomocą **funkcji rekurencyjnej** (funkcja wywołuje samą siebie, aż do spełnienia pewnego warunku). Płatek rysowany jest za pomocą *JavaScript* wewnątrz znacznika *<canvas>* (wprowadzony w HTML5). Funkcja najpierw rysuje rekurencyjnie jeden bok z odnogami, a następnie kopiuje ten bok w całości dwa razy i składa w „trójkąt”.

Kod skryptu

```
<body>
<canvas width="500" height="550" id="miejsceNaPlatek" style="border-style: solid; border-color: #f8f8ff;"></canvas>

<script>
// Rysowanie 2-wymiarowe (2D). Inna opcja (3D) to "webgl":
var rysowanyObiekt = miejsceNaPlatek.getContext("2d");

// Położenie obiektu na płótnie (canvas), współrzędne XY (domyślnie lewy górny punkt to 0,0).
// Innymi słowy: zdefiniuj na nowo punkt (0,0) w dowolnym miejscu mapy/współrzędnych:
rysowanyObiekt.translate(20,140); // Ten punkt na osi (X,Y) jest od tego momentu punktem 0,0 dla rysowanego obiektu.

function narysujPlatek(dlugoscBokuNadrzednegoTrojkata) {
    rysowanyObiekt.save(); // Zapis każdego przebiegu rekurencyjnego
    // Rysuje pojedynczą linię:
    rysowanyObiekt.beginPath(); // Będziemy zaraz definiować linię, to jest jej początek (potrzebne na wypadek potrzeby zamknięcia ścieżki, patrz: closePath())
    rysowanyObiekt.moveTo(0,0); // Przenieś się (bez rysowania!) do następującego punktu na osi współrzędnych (X,Y), czyli ile pikseli od lewej strony, ile pikseli od góry.
    rysowanyObiekt.lineTo(dlugoscBokuNadrzednegoTrojkata,0); // Współrzędne końca linii (X,Y), czyli ile pikseli od lewej strony, ile od góry. Polecenie tworzy linię w pamięci, ale jeszcze bez jej rysowania.
    rysowanyObiekt.strokeStyle="#cd1585"; // Kolor linii
    rysowanyObiekt.lineWidth="1"; // grubość linii w pikselach
    // rysowanyObiekt.shadowColor = "black"; // Kolor cienia
    // rysowanyObiekt.shadowBlur = 20; // Stopień rozproszenia tego cienia (im większa wartość, tym łagodniejszy cień)
    // rysowanyObiekt.shadowOffsetX = 20; // Przesunięcie cienia od obiektu
    // rysowanyObiekt.shadowOffsetY = 20;       // Przesunięcie cienia od obiektu
    rysowanyObiekt.stroke(); // Narysuj wcześniej zdefiniowaną pojedynczą poziomą linię!

    // Jeśli długość boku trójkąta (pierwszej linii) podana przez użytkownika jest większa niż 5px,
    // to rysuje ten bok za pomocą 4 łamanych linii (z wybrzuszeniem 60 stopni).
    // Linia będzie więc łamana na coraz mniejsze części, aż osiągnie 5px (szczegółowość płatka).
    if (dlugoscBokuNadrzednegoTrojkata>5) {
```

```
            // Dzieli podaną długość boku na trzy równe odcinki:
            var dlugoscBokuMniejszegoTrojkata=dlugoscBokuNadrzednegoTrojkata/3;

            // Rekurencja. Funkcja wywołuje samą siebie, ale z nową mniejszą wartością:
            narysujPlatek(dlugoscBokuMniejszegoTrojkata);
            rysowanyObiekt.translate(dlugoscBokuMniejszegoTrojkata,0);
            rysowanyObiekt.rotate(-60*Math.PI/180);

            // Rekurencja. Funkcja wywołuje samą siebie, ale z nową mniejszą wartością:
            narysujPlatek(dlugoscBokuMniejszegoTrojkata);
            rysowanyObiekt.translate(dlugoscBokuMniejszegoTrojkata,0);
            rysowanyObiekt.rotate(120*Math.PI/180);

            // Rekurencja. Funkcja wywołuje samą siebie, ale z nową mniejszą wartością:
            narysujPlatek(dlugoscBokuMniejszegoTrojkata);
            rysowanyObiekt.translate(dlugoscBokuMniejszegoTrojkata,0);
            rysowanyObiekt.rotate(-60*Math.PI/180);

            // Rekurencja. Funkcja wywołuje samą siebie, ale z nową mniejszą wartością:
            narysujPlatek(dlugoscBokuMniejszegoTrojkata);
        }

        rysowanyObiekt.restore();
}

// Uruchamiamy funkcję odpowiedzialną za narysowanie jednego boku trójkąta,
// i określamy tutaj jego długość w pikselach:
wstepnaDlugoscBokuTrojkata=460;
narysujPlatek(wstepnaDlugoscBokuTrojkata); // Wywołanie funkcji


// Kopiuje utworzony przez funkcję bok dwukrotnie, każdy z nich obraca o 120 stopni i łączy wszystko w trójkąt:
for (var i=0;i<3;i++) {
  narysujPlatek(wstepnaDlugoscBokuTrojkata);
  rysowanyObiekt.translate(wstepnaDlugoscBokuTrojkata,0);
  rysowanyObiekt.rotate(120*Math.PI/180);
}

</script>
</body>
</html>
```

Ostatnia aktualizacja: 16 listopada 2021.