

## **Fractals in Pascal deel 1**

**Henk van Wulpen,**

**MSX CLUB MAGAZINE 44**

*Scanned, ocr'ed and converted to PDF by HansO, 2001*

Henk van Wulpen schreef over fractals en perste en passant ook het uiterste uit zijn printer. Hij schakelde een jaartje geleden over van Basic op Pascal en een aantal programma's werd toen herschreven. Speciaal voor de Pascalliefhebber.

Eindejaar...Het betekent ook 2 weken vakantie. Eindelijk nog eens tijd om me met mijn laatste artikels bezig te houden. Vandaag enkele voorbeelden die je op weg moeten helpen bij het maken van eigen creaties.

Geloof het of niet, vandaag is het le Kerstdag... De programma's, die bij dit artikel en volgende horen, zijn al 4 maanden oud. Bij het verschijnen van dit artikel, mag je daar ongetwijfeld nog enkele maanden bij tellen. De redactie wordt immers nog steeds overstelpt met inzendingen, en dat is maar goed ook. Daar er echter sinds vorige maand (november dus...) geen MSX meer, maar een heuse PC op mijn 'desk' staat (daarom is het ook een desktop), zal het je niet verwonderen, dat deze artikelsree waarschijnlijk mijn laatste bijdrage aan dit tijdschrift zal zijn. Tot spijt van velen, tot opluchting van anderen... Het is nu Kerstmis, dus waarom deze keer geen abstracte kerstboom tekenen ?...

### **1. Triplex**

Dit was mijn eerste TMS-(Turbo Mega-Screen) probeersel. Ik dacht eens zelf een fractal uit te vinden, ha... Mijn redenering voor Triplex was als volgt. Deel een rechthoek (aanvankelijk de volledige grootte van het blad) op in 4 gelijke delen, opnieuw rechthoeken. Laat die rechtsonder wit en kleur de andere zwart. Herhaal deze bewerking op elk van deze 3 zwarte rechthoeken, en zo verder...ad infinitum. Dit moest duidelijk een fractal opleveren. In de vorm van een Tur-bo-Pascal-programma bekomt men TRIPLEX.PAS. Dankzij recur-sie in Pascal is het programma zo klaar als pompwater. Diepte bepaalt het aantal keren, dat de iteratie (het opdelen van de rechthoeken) herhaald moet worden. Merk op. dat het zwart kleuren van de rechthoek slechts gebeurt op het laagste niveau zoals hierboven voorgesteld. Dit verandert echter niks aan het resultaat. Het tekenen van de zwarte rechthoek gebeurt in de dubbele for-lus. Indien de gewenste diepte nog niet bereikt is, wordt de procedure recursief opgeroepen voor de drie deel-rechthoe-ken (met gemeenschappelijk hoekpunt  $x_3, y_3$ ) en niveau  $d+1$ . Je ziet het, een zeer eenvoudig programma, dat erg weinig toelichting behoeft. En dat ligt echt niet aan mij. Ik ben geen programmeerwonder. Iedereen die Pascal kent, is hiertoe in staat. Toen ik het programma startte bleek echter al vlug, dat ik toch geen nieuwe fractal geprogrammeerd had. Er verscheen namelijk niks anders dan de bekende fractal van Sierpinski (draai het blad om en met een beetje goede wil zie je de kerstboom). Nochtans

werd die volledig anders getekend, dan normaal het geval is (zie paragraaf 2). Het is echter enkel door de grote iteratie-diepte en de eindige resolutie, dat dit resultaat bekomen wordt. Verander de waarde Diepte eens in 5 en de Sierpinski-fractal is nog nauwelijks te herkennen. Als de ruimte in dit blad (en ook de redactie) het toelaat, staat deze fractal hier ergens afgedrukt.

## **2. Sierpinski**

De echte Sierpinski-fractal wordt namelijk met driehoeken geconstrueerd en niet met rechthoeken. Het enige verschil zit dan ook in de for-lus, waar de driehoek getekend wordt. De moet toegeven, dat de manier waarop deze getekend wordt, namelijk door een stel lijnen door hetzelfde punt  $x_1, y_2$ , wel niet de meest efficiënte maar wel de eenvoudigste is.

## **3. Vortex**

Een ander simpel voorbeeld. Op de zijde van de aanvangsrecht-hoek komt op afstand  $l/r$  van de hoeken een nieuwe rechthoek (gelogen, in feite is het een parallellogram) en dat tot in het oneindige. Maar omdat dat nogal lang is: tot een willekeurige toets ingedrukt wordt. Een afdruk maakt direct duidelijk wat het resultaat is.

## **4. Boom van Pythagoras**

Ik maak het mezelf gemakkelijk, door gewoon te verwijzen naar de vele artikels (ook in dit tijdschrift) over dit, toch wel bekende onderwerp. Laat dit programma een uurtje lopen, druk de figuur af en sta meestal versteld van het fraaie resultaat.

```

PROGRAM Triplex;

{           Triplex-Fractal
  -----

Eigen creatie : fractal der 3 rechthoeken

    Henk Van Wulpen
    Braambesstraat 8
    8210 Zedelgem
}

CONST Diepte = 10;           { 10 = 6 minuten }
      FileNaam = 'TRIPLEX';

{$I MEGASCRN.LIB}

PROCEDURE Recursie (x1,y1,x2,y2 : INTEGER;
                  d : INTEGER);
{$A-}

VAR x,y : INTEGER;
    x3,y3 : INTEGER;

BEGIN
  IF d = Diepte THEN
    FOR x := x1 TO x2 DO
      FOR y := y1 TO y2 DO
        MSet (x,y,B1)
      ELSE BEGIN
        x3 := x1 + (x2-x1) Div 2;
        y3 := y1 + (y2-y1) Div 2;
        Recursie (x1,y1,x3,y3,d+1);
        Recursie (x1,y3,x3,y2,d+1);
        Recursie (x3,y1,x2,y3,d+1)
      END
    END;

BEGIN
  MInit (Wh);
  Recursie (0,0,1023,1279,1);
  MSave (FileNaam);
  MExit;
  IF MError <> 0 THEN WriteLn ('IO-fout ',MError)
END.

```

```

PROGRAM Sierpinski;

{      Fractal van Sierpinski
-----

De bekende fractal van de driehoekjes

    Henk Van Wulpen
    Braambesstraat 8
    8210 Zedelgem
}

CONST Diepte = 10;           { 10 = 6 minuten }
      FileNaam = 'SIERPINS';

{$I MEGASCRN.LIB}

{$I MEGASCR1.LIB}

PROCEDURE Recursie (x1,y1,x2,y2 : INTEGER;
                  d : INTEGER);
{$A-}

VAR x : INTEGER;
    x3,y3 : INTEGER;

BEGIN
  IF d = Diepte THEN
    FOR x := x1 TO x2 DO
      MLine (x,y1,x1,y2,B1)
    ELSE BEGIN
      x3 := x1 + (x2-x1) Div 2;
      y3 := y1 + (y2-y1) Div 2;
      Recursie (x1,y1,x3,y3,d+1);
      Recursie (x1,y3,x3,y2,d+1);
      Recursie (x3,y1,x2,y3,d+1)
    END
  END;

BEGIN
  MInit (Wh);
  Recursie (0,0,1023,1279,1);
  MSave (FileNaam);
  MExit;
  IF MError <> 0 THEN WriteLn ('IO-fout ',MError)
END.

```

```

PROGRAM Vortex;

{
    Vortex
    -----

Dit programma tekent een kolkende spiraaltrap

    Henk Van Wulpen
    Braambesstraat 8
    8210 Zedelgem
}

CONST r = 8;
      FileNaam = 'VORTEX';

VAR x1,y1,x2,y2,x3,y3,x4,y4 : INTEGER;
    xh,yh : INTEGER;

{$I MEGASCRN.LIB}

{$I MEGASCR1.LIB}

BEGIN
  MInit (Wh);
  x1 := 0;
  y1 := 0;
  x2 := 1023;
  y2 := 0;
  x3 := 1023;
  y3 := 1279;
  x4 := 0;
  y4 := 1279;
  REPEAT
    MLine (x1,y1,x2,y2,B1);
    MLine (x2,y2,x3,y3,B1);
    MLine (x3,y3,x4,y4,B1);
    MLine (x4,y4,x1,y1,B1);
    xh := x1 + (x2-x1) Div r;
    yh := y1 + (y2-y1) Div r;
    x2 := x2 + (x3-x2) Div r;
    y2 := y2 + (y3-y2) Div r;
    x3 := x3 + (x4-x3) Div r;
    y3 := y3 + (y4-y3) Div r;
    x4 := x4 + (x1-x4) Div r;
    y4 := y4 + (y1-y4) Div r;
    x1 := xh;
    y1 := yh
  UNTIL KeyPressed;
  MFrame;
  MSave (FileNaam);
  MExit;
  IF MError <> 0 THEN WriteLn ('IO-fout ',MError)
END.

```

```

PROGRAM PythagorasBoom;

{           De Boom van Pythagoras
  -----

Dit programma tekent een willekeurige Pythagoras-boom

    Henk Van Wulpen
    Braambesstraat 8
    8210 Zedelgem
}

CONST Diepte = 16;           { 16 = 1 uur }
      FileNaam = 'PYTHBOOM';

{$I MEGASCRN.LIB}

{$I MEGASCR1.LIB}

PROCEDURE Recursie (x1,y1,x2,y2 : INTEGER;
                  d : INTEGER);
{$A-}

VAR dx,dy : INTEGER;
    x3,y3,x4,y4,x5,y5 : INTEGER;
    a : REAL;
    CosA,SinA : REAL;

BEGIN
  IF MOK (x1,y1) And MOK (x2,y2) THEN BEGIN
    MLine (x1,y1,x2,y2,B1);
    dx := x2 - x1;
    dy := y2 - y1;
    x3 := x2 - dy;
    y3 := y2 + dx;
    x4 := x1 - dy;
    y4 := y1 + dx;
    IF MOK (x3,y3) And MOK (x4,y4) THEN BEGIN
      MLine (x2,y2,x3,y3,B1);
      MLine (x3,y3,x4,y4,B1);
      MLine (x4,y4,x1,y1,B1);
      IF d < Diepte THEN BEGIN
        dx := x4 - x3;
        dy := y3 - y4;
        a := Random + 0.3;
        CosA := Cos (a);
        SinA := Sin (a);
        x5 := x3 + Round ( CosA * (CosA*dx - SinA*dy) );
        y5 := y3 - Round ( CosA * (SinA*dx + CosA*dy) );
        Recursie (x5,y5,x3,y3,d+1);
        Recursie (x4,y4,x5,y5,d+1)
      END
    END
  END
END;

```

```
BEGIN
  MInit (Wh);
  Randomize;
  Recursie (1023,575,1023,705,1);
  MSave (FileNaam);
  MExit;
  IF MError <> 0 THEN WriteLn ('IO-fout ',MError)
END.
```