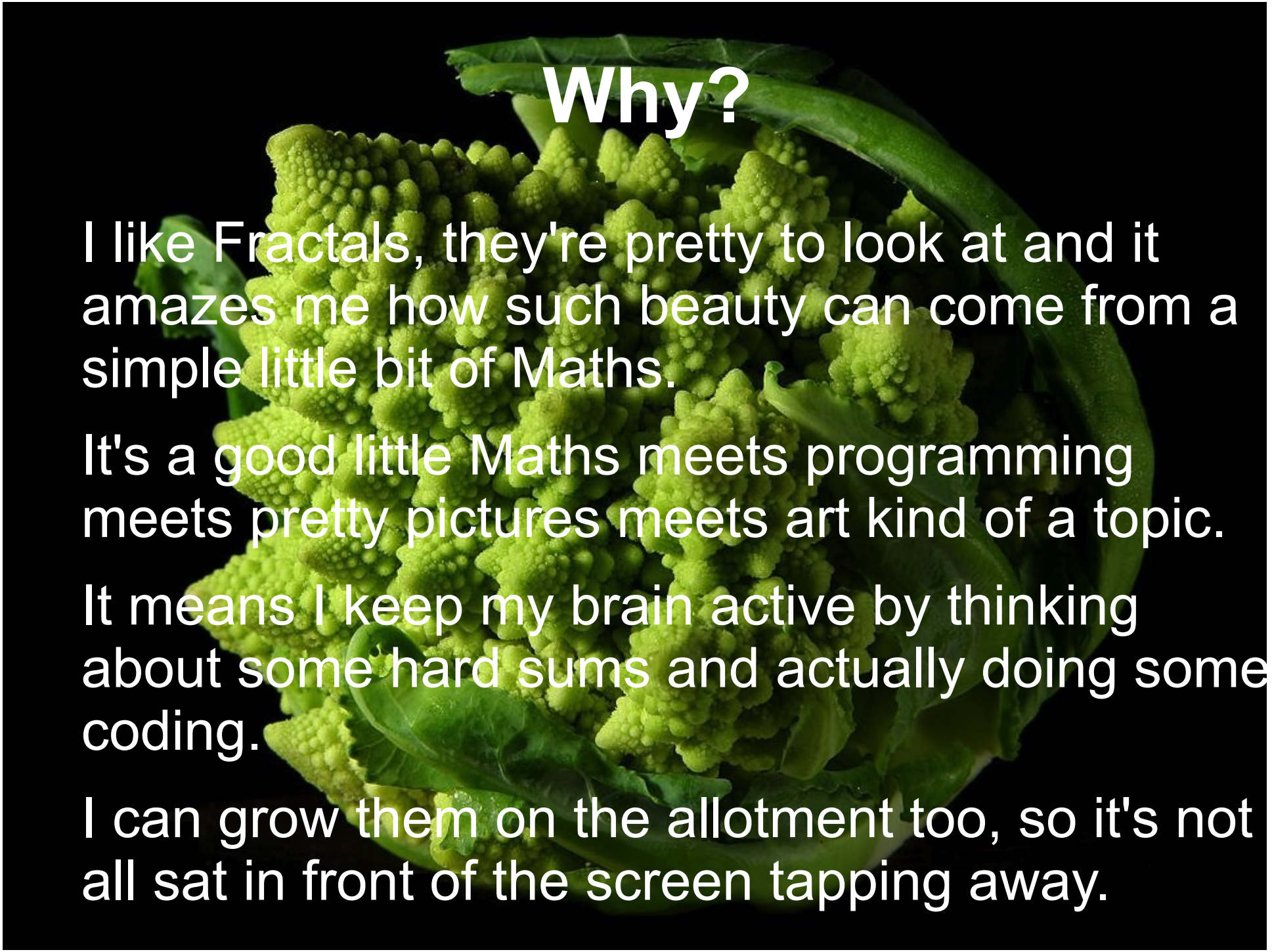

BarnCamp 2011
The HTML 5 <canvas> element as seen through a fractal eye: a brief introduction



Why?

I like Fractals, they're pretty to look at and it amazes me how such beauty can come from a simple little bit of Maths.

It's a good little Maths meets programming meets pretty pictures meets art kind of a topic.

It means I keep my brain active by thinking about some hard sums and actually doing some coding.

I can grow them on the allotment too, so it's not all sat in front of the screen tapping away.

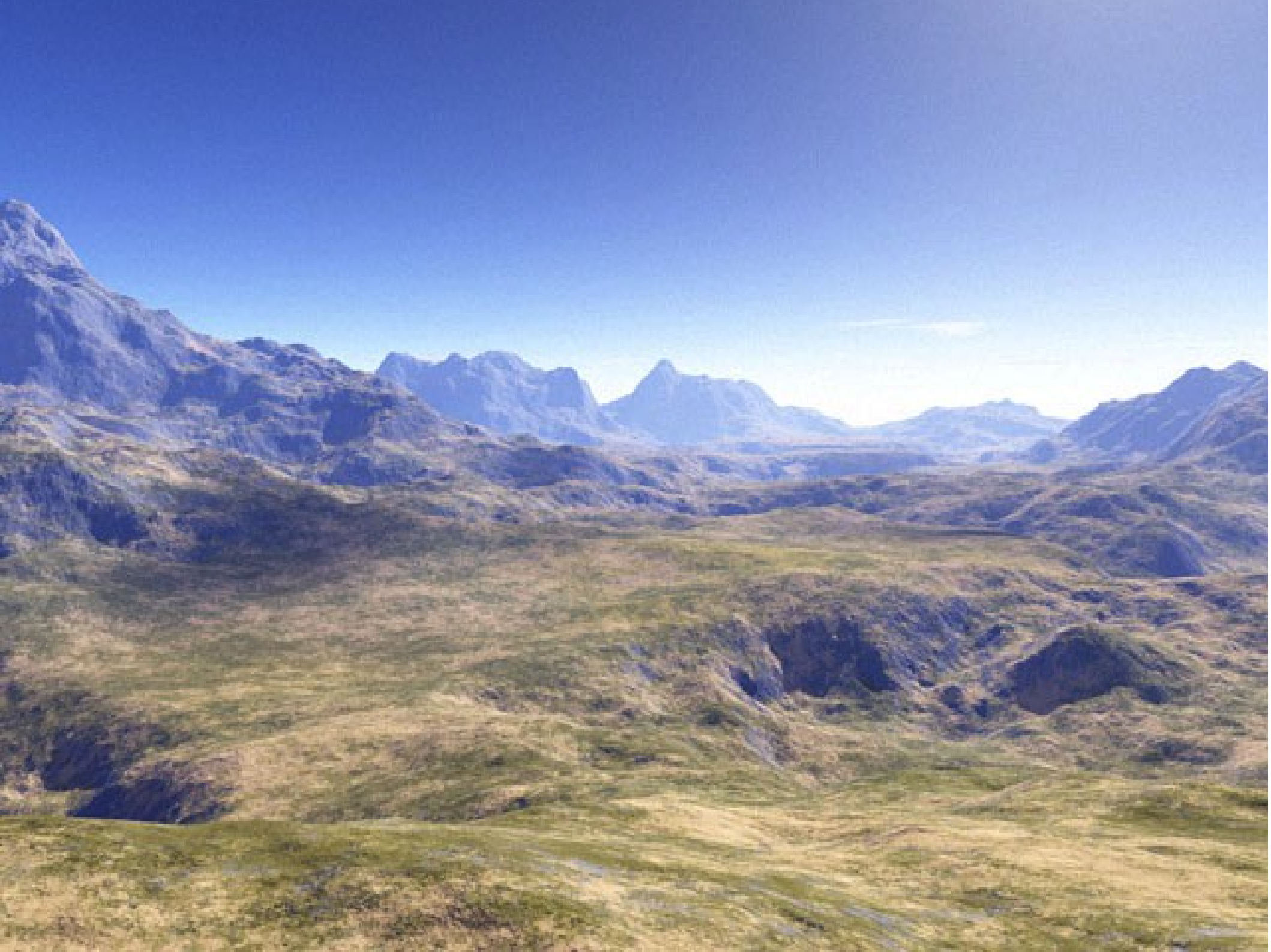
What is a fractal?

An object which has some of these properties:

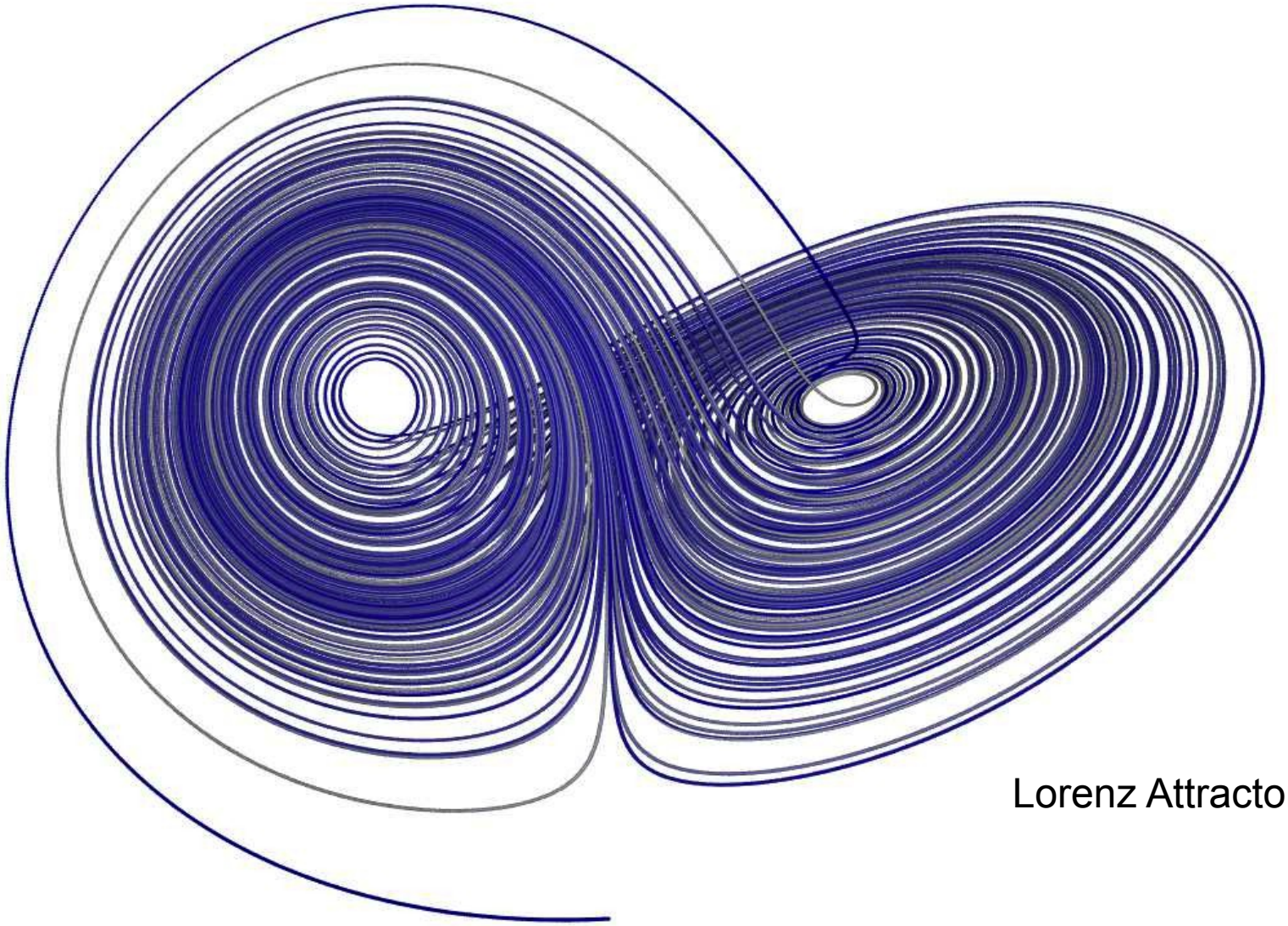
- Self-similar (the same patterns repeat themselves at different magnifications)
- Fine structure are arbitrary small scales.
- It can't be described by Euclidean geometry.
- It has a simple and recursive definition.
- It has a Hausdorff dimension which is greater than its topological dimension – I think this means that it's boundary is greater than the area that it bounds.

Types of fractals

- Iterated Function Systems (IFS) - Cantor set, Sierpinski carpet, Sierpinski gasket, Peano curve, Koch snowflake, Harter-Heighway dragon curve, T-Square, Menger sponge
- Strange attractors – Lorenz attractor
- Random fractals – such as those made from Brownian motion, fractal landscapes, etc.
- Escape-time fractals – Mandelbrot Set, Julia Set, Nova and Lyapunov fractals.

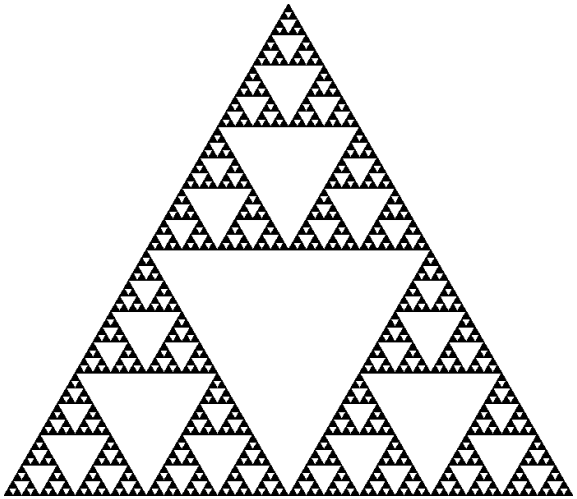


Strange Attractors

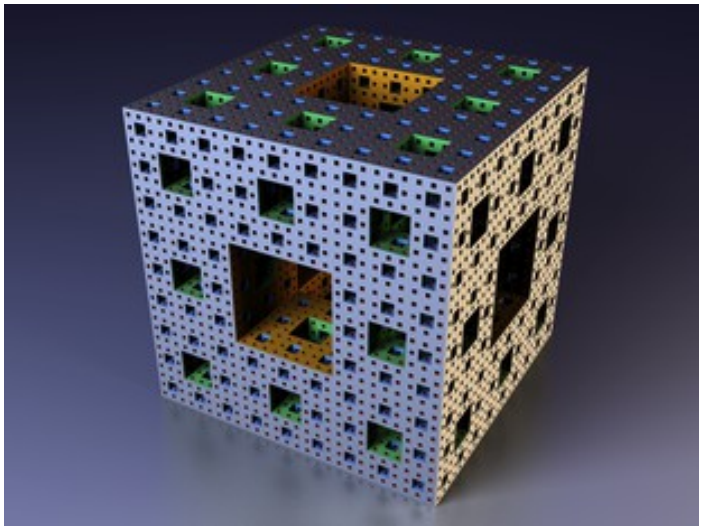


Lorenz Attractor

IFS

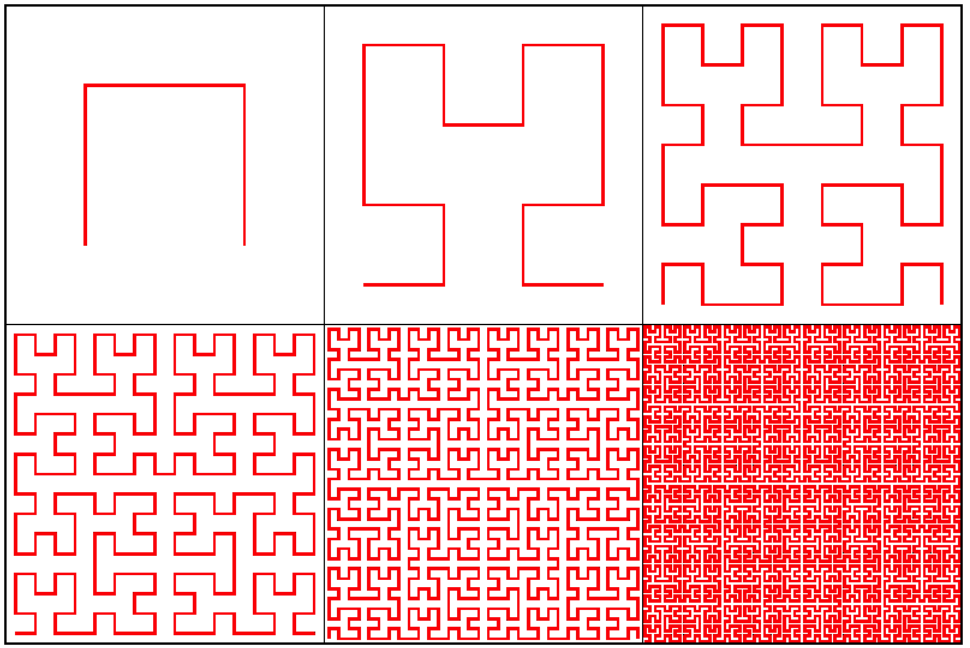


Sierpinski Triangle



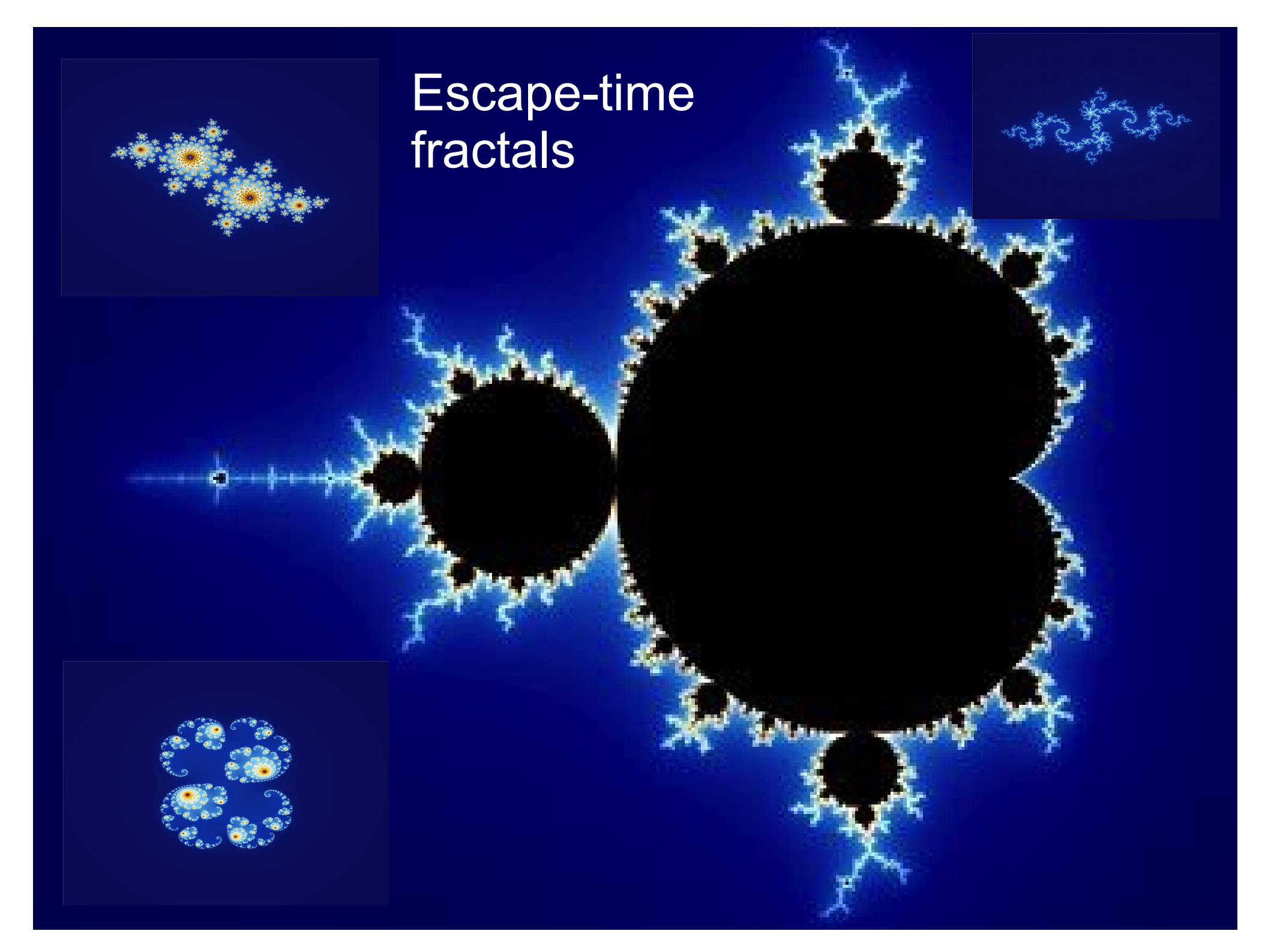
Menger Sponge

IFS – Iterative Function System means 'do something, then do it again, and again and again to the same thing ad infinitum'.



Hilbert Curve

Escape-time fractals



Who's behind them?

Edward N Lorenz
(1917-2008)

David Hilbert
(1862-1943)

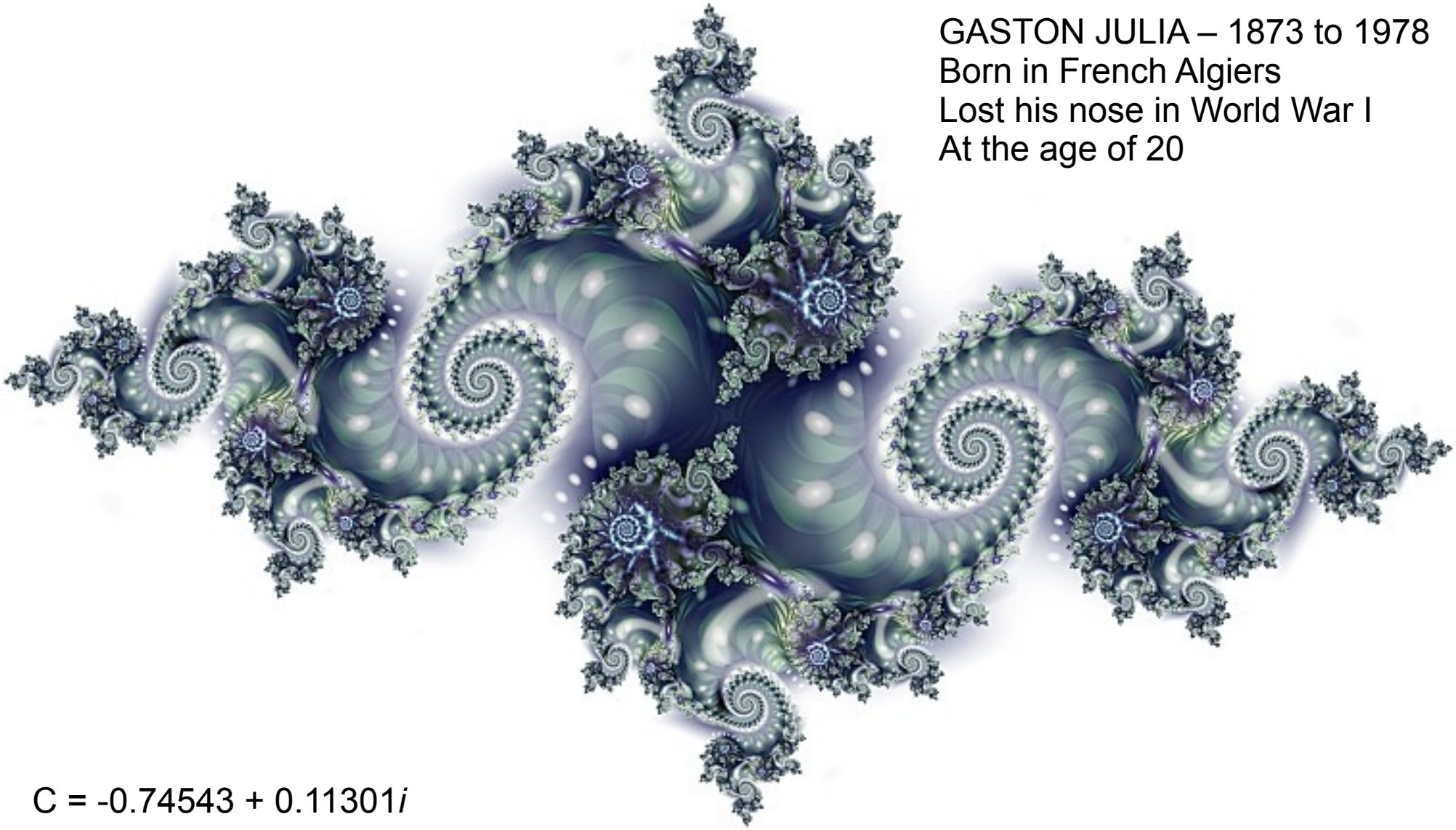
Wacław Sierpiński
(1882-1969)

Gaston Julia
(1893-1978)

Benoit Mandelbrot
(1924-2010)

Pierre Fatou
(1878-1929)

Julia Sets



GASTON JULIA – 1873 to 1978
Born in French Algiers
Lost his nose in World War I
At the age of 20

C = -0.74543 + 0.11301*i*

Gaston Julia

Along with Pierre Fatou, Gaston Julia predicted that some pretty interesting shapes would come out from the iteration of the equation:

$$z \Rightarrow z^2 + c$$

For a fixed value of C, where Z and C are complex numbers.

A complex number is a number that has a *real* part and an *imaginary* part.

And is written in the format:

C, a complex number = a + b*i*,
where a is a real number and b*i* is the imaginary part.

$$i = \sqrt{-1}$$

Enter: Benoît Mandlebrot

Benoît is pronnounced 'ben-waa' and Mandlebrot is Yiddish for Almond Bread.

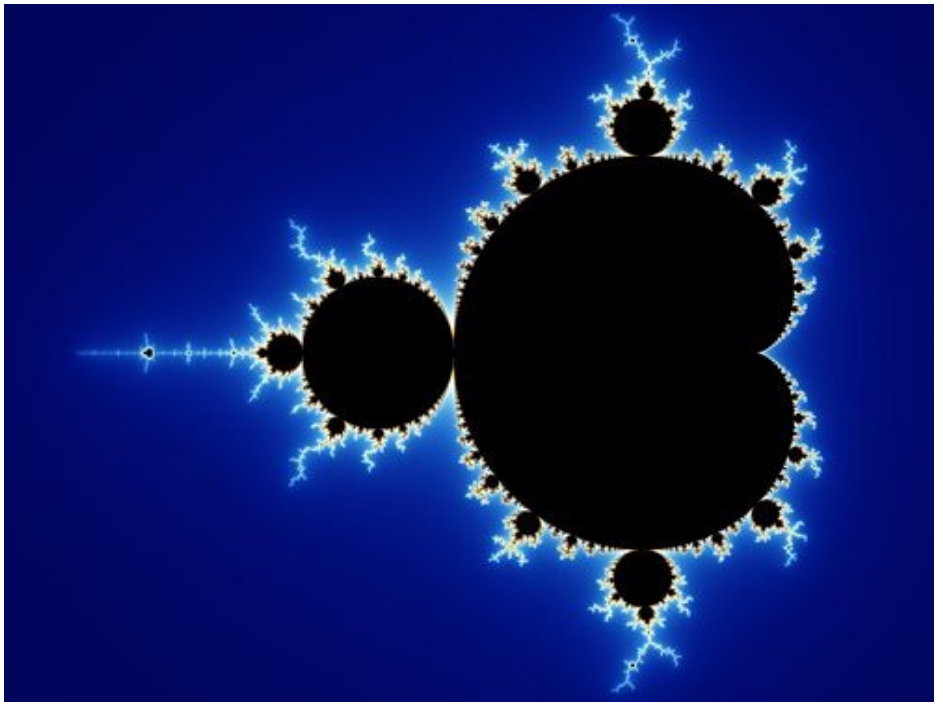
Born in Warsaw in 1924.

In 1979 he got hold of a big “supercomputer” at IBM and played about with writing computer programmes to plot Julia's work described many years before.

He came up with the idea of plotting a map of z => z^2 + c for every value of C in the complex plan and a miraculous pattern came out of the trusty old IBM plotter.

The Mandelbrot Set (Mandelbrot's Baby)

- Self-similar.
- Infinitely complex.
- A finite area enclosed by an infinite border.
- Plethora of beautiful images.
- Very simple equation.
- Easy to programme.



Fractals and I

- Was into them in the eighties because you could produce pretty pictures on your home computer. Bought of couple of hard maths books to try to understand them.
- Went to see Benoit Mandelbrot speak at the Arnolfini once, he signed my book!
- Wrote various programs with my mate Dan, including one for the Atari ST that was on an ST Format cover disk – Fractal Engine 3.
- Mandelbrot died last autumn and therefore I thought of doing this in tribute to him.

But first...

A rather daft previous Fractal experiment....

The <td> Mandelbrot Generator

HTML 5 (= HTMLXR3*i*)

- New HTML standard developed by the World Wide Web Consortium (W3C)
- A hybrid of HTML 4.01 & XHTML 1.1 and a lot more.
- <font>HTML 4.01 was a mess
- But XHTML 1.1 was too idealistic for folk to do interesting stuff.

- HTML 5 is a compromise and features the exciting <video>, <audio> and <canvas> tags and features like drag-and-drop and cross-document messaging

HTML 5 & the <canvas> element

- The canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly.
- Great!
- Latest published version of HTML5 (May 2011) is available at http://www.w3.org/TR/html5/

Defining and using <canvas>

In HTML:

```
<canvas id="mset_canvas" width="320" height="240"
style="float:left"></canvas>
```

In JavaScript:

```
var canvas =
      document.getElementById("mset_canvas");

var ctx = canvas.getContext("2d");

ctx.fillStyle = "#000000";

ctx.fillRect(ix,iy,1,1);
```

Mandelbrot/Julia Algorithm: Maths

$$\mathbf{Z \rightarrow Z^2 + C}$$

Where Z and C are complex numbers.

- Thus; Z = *x* + *iy* and C = *a* + *ib*
- So; (x + iy) → (x+iy) (x+iy) + (a+ib).
- (x + iy) → (x*x + iy*iy + 2*x*iy) + (a + ib)
- (x + iy) → x2 - y2 +2iy + a + ib

- A *complex number* is an *imaginary number* + a *real number* (it is written as a+ib)
- A *real number* (a) is an ordinary number.
- An *imaginary number* (ib) is a real number multiplied by i, the square root of 1.

Mandelbrot/Julia Algorithm: Maths

$$Z \longrightarrow Z^2 + C$$

Where Z and C are complex numbers.

As $i = \sqrt{1}$, so $i^2 = 1$

(x + iy) →

x2 - y2 +2iy + a + ib

Therefore:

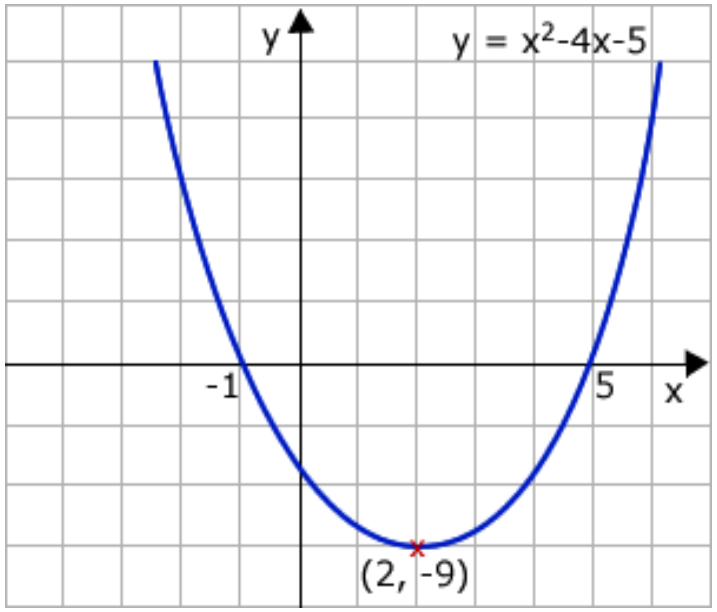
x → x2 - y2 + a

y → ib + 2iy

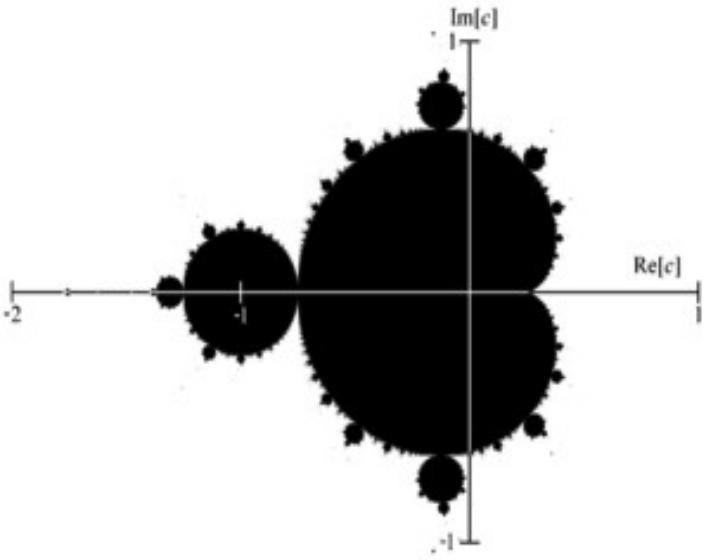
- A computer cannot handle complex numbers directly*, so we split up the real and imaginary parts to form two equations and calculate them separately.
- * note that C++ does have a type **complex**

Why's this equation interesting?

- Normal quadratic equation in real plane
  $y = x^2 - 4x - 5$



- Iterative equation in the complex plan
  $z \rightarrow z^2 + c$



The Mandelbrot/Julia Algorithm

We call the following algorithm every point on our “slice” of the complex plane. Our range is from -2.5 <= x <= 0.8 and -1.25 <= iy <= 1.25.

ALGORITHM: MsetLevel (cx, cy, maxiter)

```
BEGIN
      x:=y:=x2:=y2:=0.0
      iter:=0
      WHILE (iter<maxiter) AND (x2 + y2 < 10000.0) DO
            temp:=x2 – y2 + cx
            y:= 2 * x * y + cy
            x:=temp
            x2:=x * x
            y2:=y * y
            iter:=iter+1
      END WHILE
      RETURN (iter)
END
```

Rendered in JavaScript

```
function compute_point(x,y,cx,cy,maxiter,thresh)
{
  var x2 = x * x; var y2 = y * y;
  var iter = 0;

  while ( (iter < maxiter) &&
          ( (x2 + y2) < thresh) ) {

    var temp = x2 - y2 + cx;
    y = 2 * x * y + cy;
    x = temp;
    x2 = x * x;
    y2 = y * y;

    iter++;

   }
   return iter;
}
```

Plotting our image

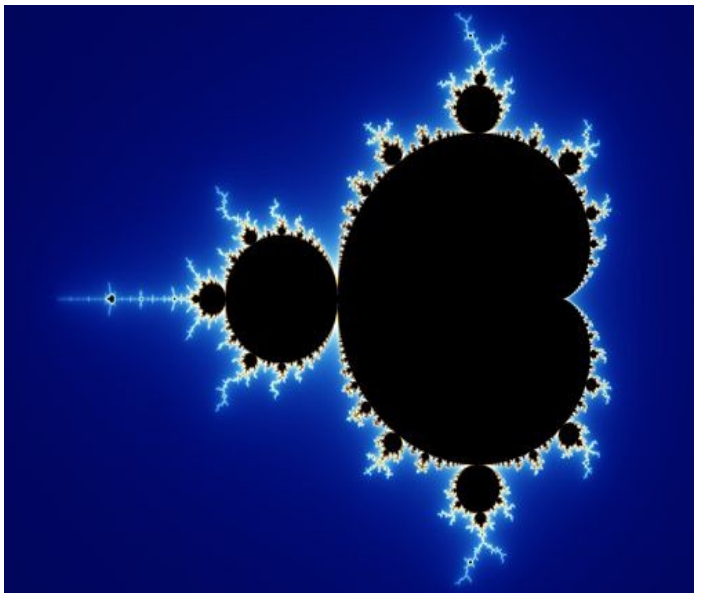
… WHILE (iter<maxiter) AND (x2 + y2 < 10000.0) ....

If the sum of the squares is greater than 10000 (the threshold) then we assume the value of Z is spiralling to infinity, we break out and plot a colour depending on the number of iterations of the algorithm that it took.

If we hit maxiter (the maximum number of iterations) then plot black.

So, the black set is the orderly part that is not tending to infinity, that part that is on the inside.

The colourful part is the chaotic part, tending to infinity.



The Mandelbrot Set plotting code

```
function mandelbrot() {

  var canvas = document.getElementById("mset_canvas");
  var ctx = canvas.getContext("2d");

  for (var iy = 0; iy < res_y; iy++) {

    var cy = y_min + iy * y_prop;

    for (var ix = 0; ix < res_x; ix++) {

      var cx = x_min + ix * x_prop;
      var x = 0.0; var y = 0.0;
      var iter = compute_point(x,y,cx,cy,maxiter,thresh);

      if(iter == maxiter) {

        // if we didn't get to infinity by the time we
        // used up all the iterations, then we're in the set
        // colour it black
        ctx.fillStyle = "#000000";

      } else {

        // otherwise colour it according to the number
        // of iterations it took to get to infinity (thresh)
        ctx.fillStyle = pallette[iter % num_colours];
      }

      ctx.fillRect(ix,iy,1,1);

   }
  }
}
```

The end result

http://mbharris.co.uk/fe3/jsetexplorer.html

In summary

- Fractals are pretty and still cool.
- XHTML 1.0/1 was nice and clean, but anally so and people forgot to live a little
- HTML 5 looks promising and the new <canvas> as well as <video> and <audio> tags look well exciting.
- Reconstructive surgery has improved greatly since Gaston Julia's time.
- Maths can be fun!

Thanks

http://mbharris.co.uk/fe3/
http://hatari.berlios.de/

http://mbharris.co.uk/fe3/jsetexplorer.html
http://www.w3.org/TR/html5/

http://gfabasic32.googlepages.com/
http://en.wikipedia.org/wiki/Romanesco_broccoli