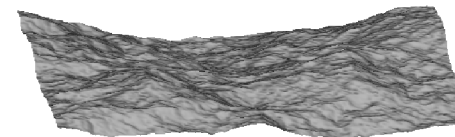
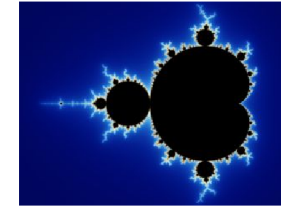


And now ...

- Go back to modelling for a while.

Fractals

- Infinite detail.
- Go on forever.



Fractals

- Problem: World is full of infinite detail.
- Hand modeling some things is too difficult.
- Can we make realistic-looking natural things?

Fractal properties

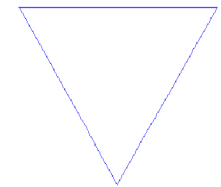
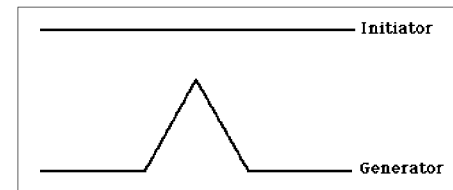
- Most important property: self similarity.
- Shape is the same at the microscopic level as it is at the large scale level.
- Two types of self-similarity:
 - Exact self-similarity: different levels are exactly the same shape.
 - Stochastic self-similarity: different levels "look similar" to one another.

Types of fractals

- Line fractals aka L-systems
- Iterated function systems.
- Mandelbrot sets, Julia sets, etc.
- Terrain generation by midpoint displacement

L-systems

- Fractals generated by replacement rules.
- E.g. Koch curve



L-systems

- Can actually model this as a repeated sequence of string substitutions. E.g. F for forward, + for turn 60 and - for turn -60. Then Koch can be represented as:
F -> F+F--F+F
- Many things can be represented this way.
- Can also be used for 3D.

Implementing in OpenGL

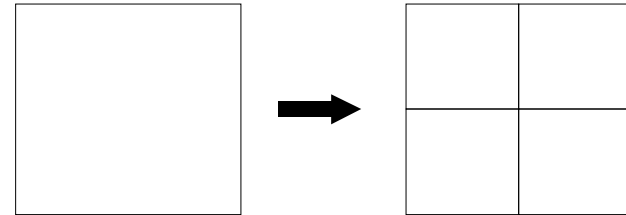
- Implementing fractals in OpenGL is really easy.
- Use recursion + matrix stack.
- Problem: Matrix stack has limited depth.
- Can have special conditions for flowers etc.
- Demo: FractalTree.

Randomness

- Can add randomness also rather than having fixed angles to make it look more realistic and natural.

Terrain formation by subdivision

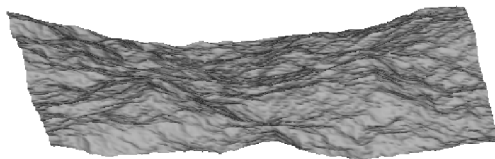
- Can do a similar thing for a surface



- Not very interesting. But what if we perturb z value?

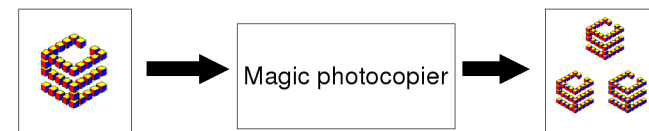
Terrain

- Can change elevation of middle point by random value.
- Then recursively subdivide and repeat:
- Note! the range of random values for subdivided points must also be halved.



Iterated function systems

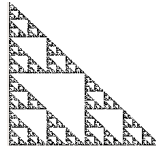
- Imagine a magic photocopier:



- What happens when we keep photocopying?

The Chaos game

- Previous slide would lead to Sierpinski triangle.
- But amazingly can also be achieved by:
P = random point
while(!bored){
 Choose random vertex
 $P = \frac{1}{2} P + \frac{1}{2} \text{random vertex}$
 Dot at p
}
- How does it work?



More on IFS

- Can it be used backwards?
 - What if, given an image, we try to find the magic photocopier that will produce it?
 - Called fractal image compression.
 - Hasn't really worked.
 - Best known algorithm: "Graduate student algorithm"



Mandelbrot and friends

- These things live on the complex plane.
- Basic equation: $f(z) = z^2 + c$
- Assume $z=0$ to start, what happens when we apply this equation repeatedly for a given c ?
- Two possibilities:
 - Can quickly head off to infinity (not in set)
 - Can bounce around (in Mandelbrot set)

Mandelbrot

- Evaluate it for every point and you get the Mandelbrot set.
- Julia set: Fix c , change starting s . What happens for different values of s ?

