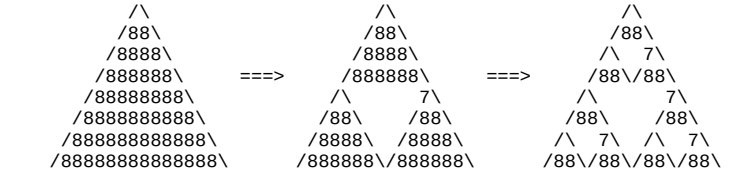
CS 4: Lecture 20
Wednesday, April 5, 2006

FRACTALS AND RECURSION
======================
Fractals are shapes that are _self_similar_, meaning that they appear similar or identical at all scales of magnification.

A simple example is called the _Sierpinski_triangle_. Here's the idea. Start with an equilateral triangle, filled in with black. (Technically, you should think of the triangle as an infinite set of points in the plane, namely all the points inside the triangle or on its boundary.) Then, shrink the triangle to half its former width and height, make three copies of it, and arrange them in a triangle. Then repeat.



But that's not a fractal yet. It becomes a fractal after you iterate an infinite number of times.

The Sierpinski triangle has some strange properties. First, although it looks solid enough, its area is zero. You see, every time we iterate the process, we reduce the triangle's area by a factor of 0.75. After an infinite number of iterations, there's no area left.

Second, it's not exactly a two-dimensional object, because its area is zero. But it's more than one-dimensional, because the total length of all its edges is infinite. So what is its dimensionality? The Sierpinski triangle has dimension

    log 3 ~ 1.585.   (The wiggly "=" means "approximately equals.")
       2   ~   (That's the base-2 logarithm of 3, aka log 3 / log 2.)

Where did this crazy dimension come from? Well, suppose you have a piece of paper with a one-dimensional curve on it, like a line. If you photocopy the paper at 2x magnification, the line gets twice as long. But if the paper also has a square on it, the area of the square gets four times bigger, because its height and width are each multiplied by 2. Since a square is two-dimensional, its area increases by a factor of 2^2 = 4. If you could photocopy a cube, its volume would increase by a factor of 2^3 = 8.

What if the paper has a Sierpinski triangle on it? When you photocopy a Sierpinski triangle at 2x magnification, you get 3 copies of the original Sierpinski triangle. So its "measure" (which is not just length, but not quite area) increases by a factor of 3. Sure enough, because the triangle is (log_2 3)-dimensional, we can calculate that its measure increases by a factor of

     (log_2 3)
    2   = 3.

How would you write code to draw a Sierpinski triangle? Well, the bad news is you can never draw one exactly, because they're "infinitely complex." The good news is that if you iterate the process above a finite number of times--say, 10 or 12--your eye won't be able to tell you didn't do infinity iterations. So we'll use recursion to visit smaller and smaller triangles. The base case happens when the recursion goes deep enough--so there are 10 or 12 copies of the recursive method on the stack. For the base case, we'll just draw a (normal) triangle.

The parameters for the following code include the x-coordinates of the left and right vertices of the triangle's bottom edge; the y-coordinates of the bottom edge and top vertex; and a number "n" that says how many more iterations need to be done before the recursion bottoms out.

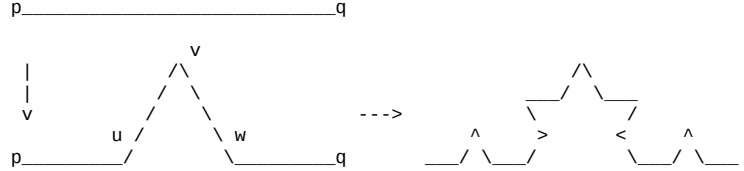
```
  public static void sierpinski(int left, int right, int down, int up, int n) {
    if (n == 0) {
      /* Base case:  draw a triangle. */
      int[] vertices = {left, down, right, down, (left + right) / 2, up};
      canvas.fillPolygon(vertices, Color.red);
    } else {
      /* Make 3 smaller triangles. */
      int leftRight = (left + right) / 2;
      int leftish = (3 * left + right) / 4;
      int rightish = (left + 3 * right) / 4;
      int downUp = (down + up) / 2;

      sierpinski(leftish, rightish, downUp, up, n - 1);
      sierpinski(left, leftRight, down, downUp, n - 1);
      sierpinski(leftRight, right, down, downUp, n - 1);
    }
  }
```

Observe that this code calls itself three times: once for the top triangle, once for the lower left triangle, and once for the lower right triangle.

The Koch Curve

A different way to approach fractals starts with a line segment, which is one-dimensional, rather than a triangle, which is two-dimensional. But we'll still end up with an object whose dimension lies between one and two.

Imagine a line segment in the plane, connecting a start point p to an end point q. Then, suppose we replace the line segment with four line segments all of the same length, as follows. (All the angles are 60 or 120 degrees.)



Each line segment is _directed_, so we know which side of the line segment to add the bump on. The directed line segment (p, q) is replaced by the four directed line segments (p, u), (u, v), (v, w), and (w, q), all having the same length. (The deleted segment (u, w) also has the same length, and uvw is an equilateral triangle.)

Then you replace each directed line segment with four smaller directed line segments. After you iterate this procedure an infinite number of times, you have a _Koch_curve_. The Koch curve is one of the earliest fractals in the mathematical literature, dating to a 1904 paper by Helge von Koch.

Even more famous is the _Koch_snowflake_, which is three Koch curves glued together in a ring, with the bumps sticking out. You get it by starting with an equilateral triangle, instead of just one line segment. You can also generate a _Koch_antisnowflake_ this way, which is three Koch curves glued together in a ring, with the bumps sticking in.

The Koch curve is as weird as the Sierpinski triangle. First, it has infinite length, because every iteration increases the length by a factor of 4/3. After an infinite number of iterations, the length is infinite.

Benoit Mandelbrot wrote a paper entitled "How Long Is the Coast of Britain?" in which he pointed out that coastlines are like fractal curves--the closer you look, the more sinuous they appear. The more you magnify, the more coastal length you find hiding in grains of sand, until you're down to the size of molecules and there is no longer a clearly defined line at all. So it really isn't possible to measure the length of a coast in a meaningful way.

The second strange fact is that the Koch curve has dimension log_3 4 ~ 1.26, because if you photocopy it at 3x magnification, you get four copies of the original curve.

The third strange fact is that the Koch curve is continuous everywhere, but differentiable nowhere. In other words, there is no point on the Koch curve where you can draw a tangent line, or say what the slope is.

In 1872, Karl Weierstrass made mathematical history by demonstrating the first continuous function that is nowhere differentiable. This and related discoveries by Weierstrass were a major driving force in the development of real analysis. Unfortunately, Weierstrass' function is not intuitive--it's an infinite series of sine functions. Koch invented his snowflake to create an intuitive, geometric example. It may look like just a fun toy, but Koch's snowflake holds in it the nub of modern analysis.

The Levy C Curve

In lab, you'll write recursive code to draw a C-shaped curve invented by Paul Levy. It's like the Koch curve, except that the iteration rule is even simpler. You just replace each segment with two segments meeting at a right angle. All the angles are 90 or 180 degrees.