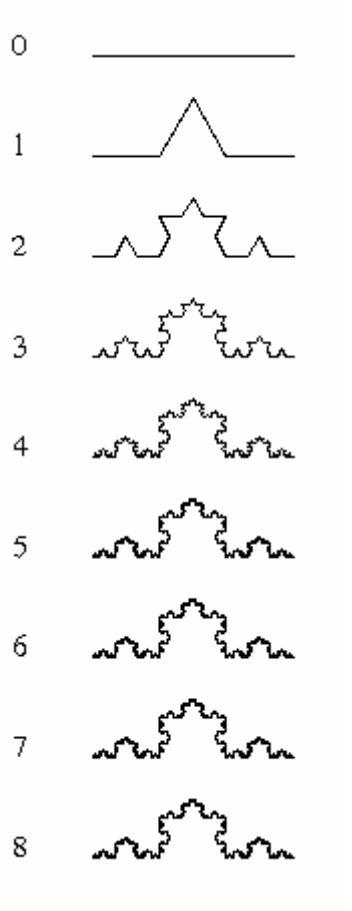
Fractals

- "Fractals possess structural self-similarity on multiple ... scales, meaning that a piece of a fractal will often look like the whole."
- powerful tool in computational geometry
- used in many modern graphics applications
  - rendering realistic looking scenery.
- fun, fun, fun!

Koch Curve

- A straight line is transformed:
  - split into three parts of equal size
  - middle part is removed
  - middle is replace with two lines of the same length (1/3 of original line).
- For each straight line in the transformed line, repeat the process.
- At some point the changes are no longer visible (if we keep our viewing scale constant).



Issues

- What is the length of a line of depth *d*?
  - Is there a limit to the length of a line (as *d* increases forever)?

- There are lots of other ways we could transform a line – this is just one interesting way...

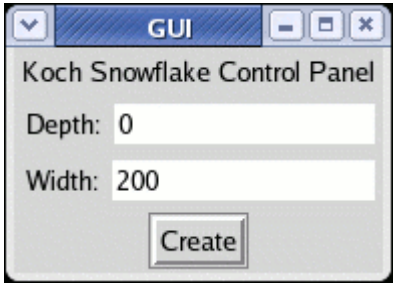
Scheme Code for Koch Curve

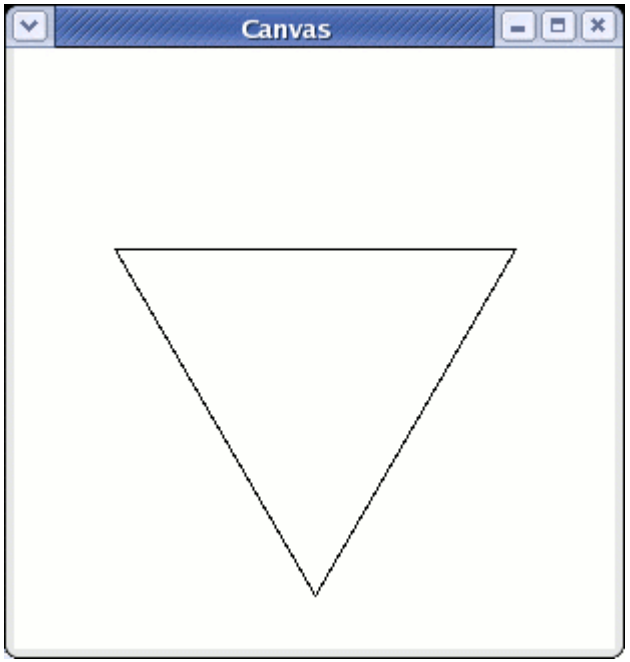
- Complete code is available via course home page.

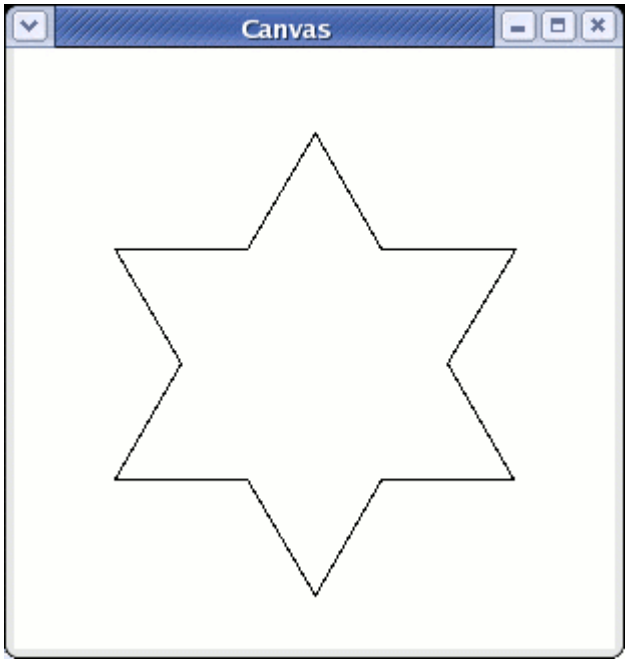
```
(define (koch-line start end)
  (cond
    [...too-small? start end ...
        (draw-solid-line start end 'black)]
    [else
      ... draw first 1/3rd using koch-line
      ... draw middle part using koch-line
      ... draw last 1/3rd using koch-line ]))
```

Need some threshold

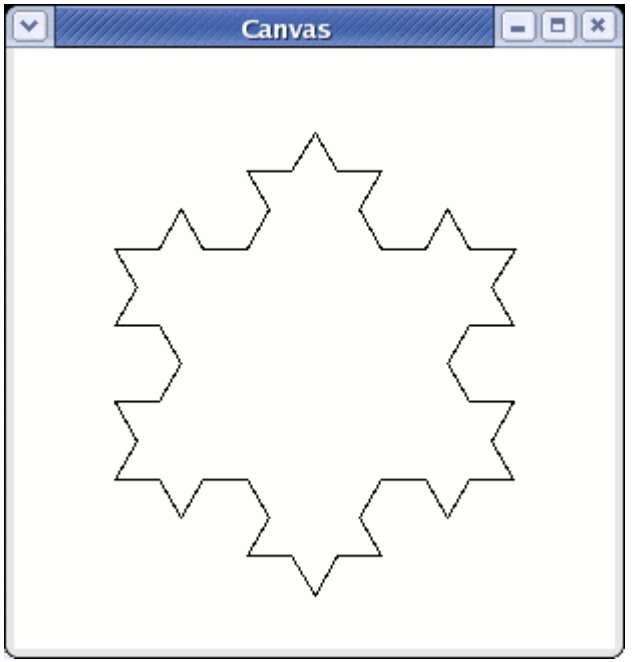
Koch Snowflake
Start with an equilateral triangle

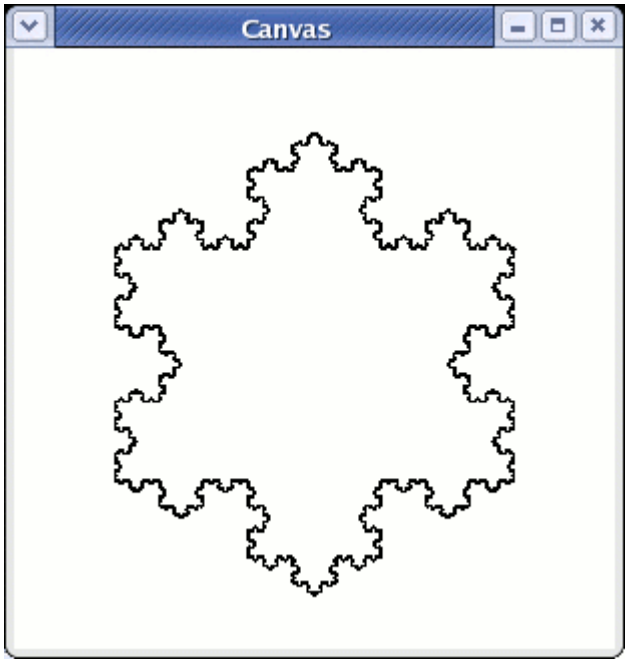






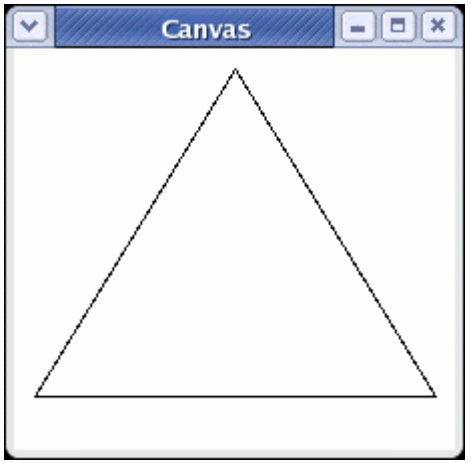
Koch Snowflake (cont.)

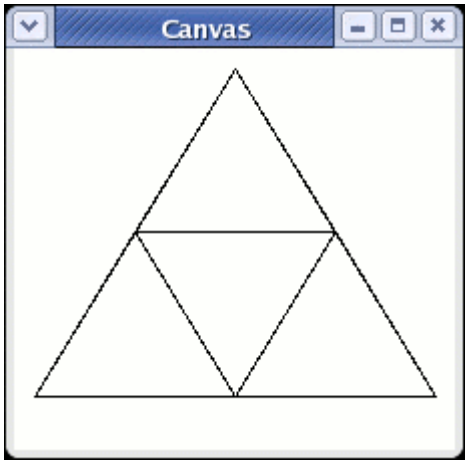




Sierpinski triangle

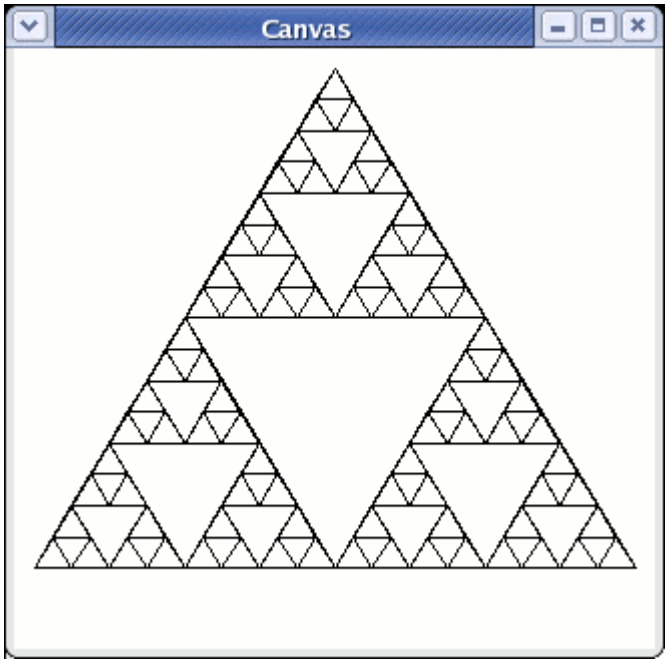
- Start with an equilateral triangle
- Connect the midpoints of the three sides.
  - this divides the original triangle in to four smaller equilateral triangles.
- rinse and repeat (recursively).

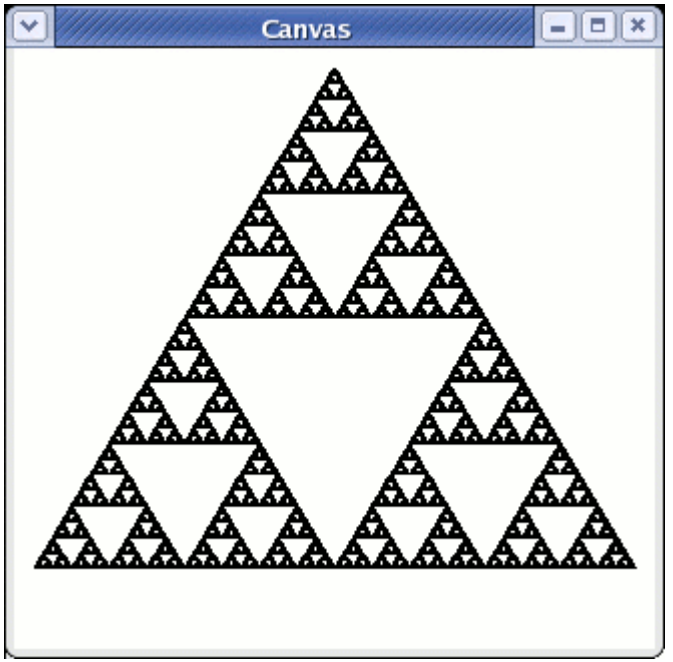




More Sierpinski triangles

Levels 5 and 10



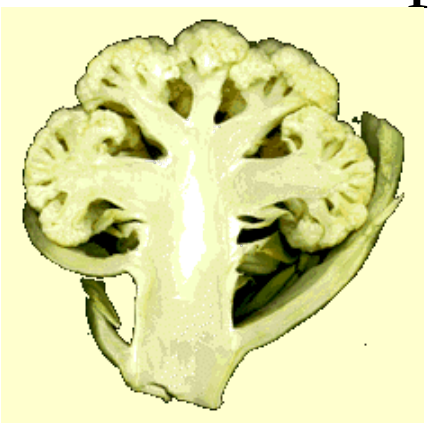


Sierpinski code

```
(define (sierpinski a b c)
  (cond
    [(too-small? a b c) true]
    [else
      (local ((define a-b (mid-point a b))
              (define b-c (mid-point b c))
              (define c-a (mid-point a c)))
        (and
          (draw-triangle a b c)
          (sierpinski a a-b c-a)
          (sierpinski b a-b b-c)
          (sierpinski c c-a b-c)))]))
```

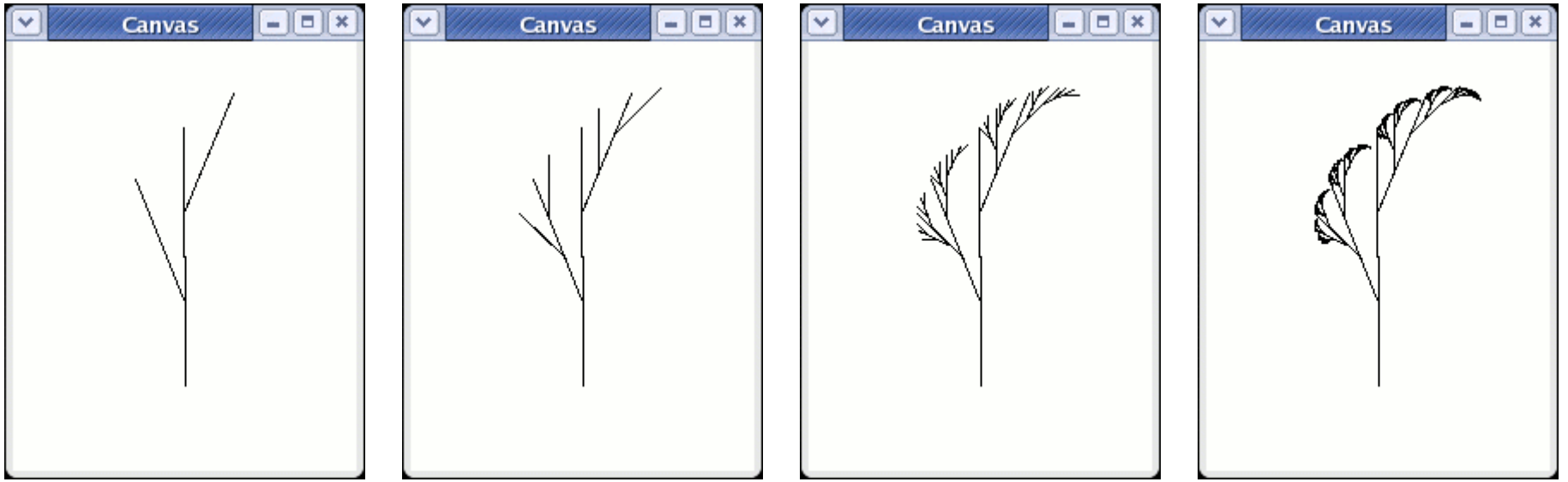
Fractals and Nature

- There are numerous examples of fractals found in nature, some obvious examples:
  - ferns
  - cauliflower



- Some of the best algorithms used for generating realistic scenery are based on fractals.
  - some *randomness* usually included

A Simple Tree-drawing Fractal



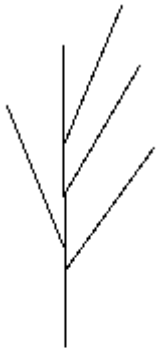
Drawing a Forest

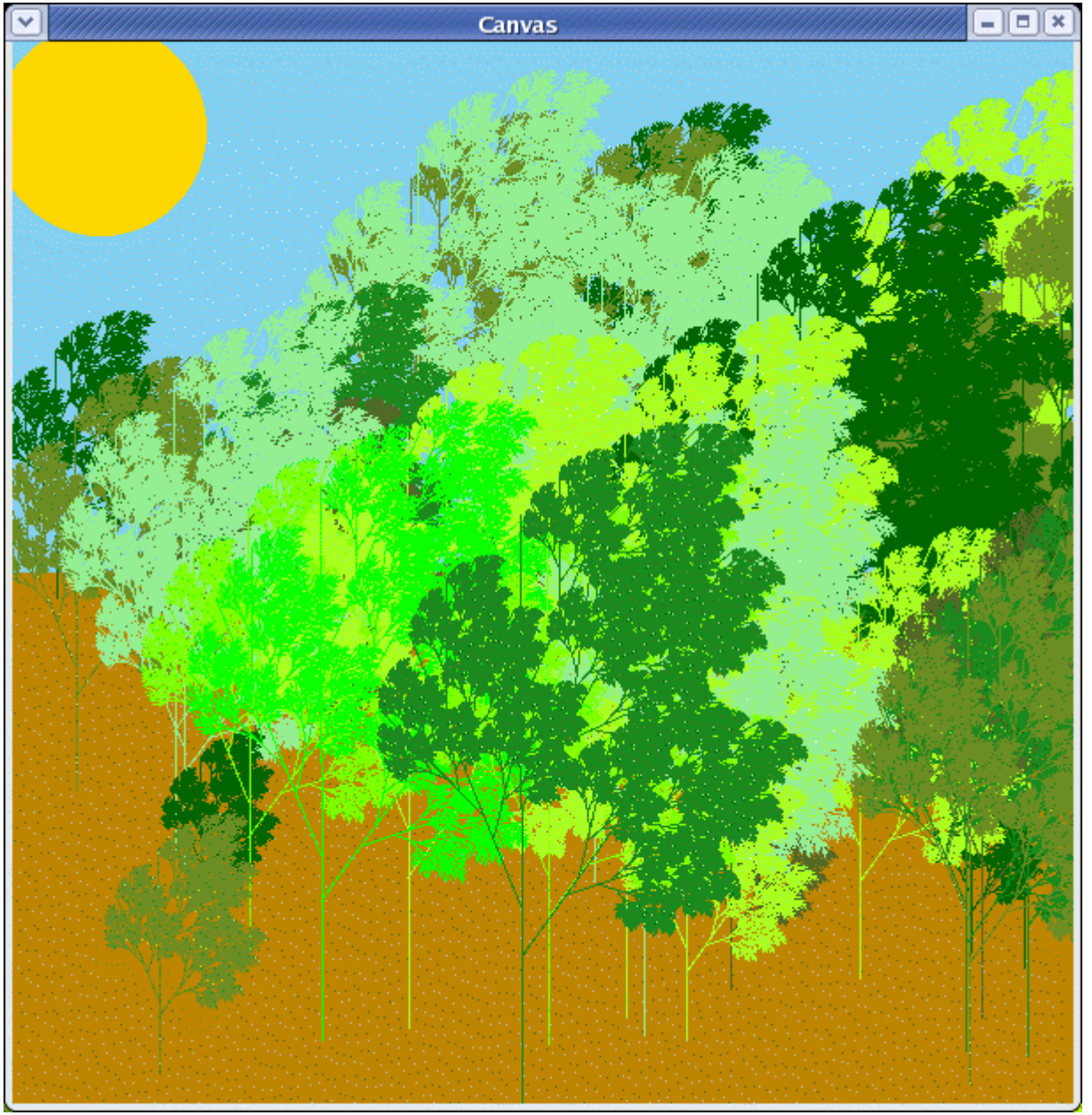
- generate a bunch of random trees
  - random sizes, positions, colors
- draw some background
  - sky, ground, sun
- Sort the trees by distance from observer (y coordinate)
- Draw trees in sorted order
  - farther away done first

A Forest

- 40 trees

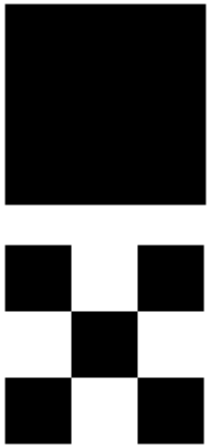
- trees are slightly more complex than the previous example.





Fractal Exercise

- Start with a square.
- divide into 9 equal parts and draw every other square (recursively).
- something like a checkerboard...



Ideas:

  - colors!
  - Vary the threshold used to stop the recursion..

Problems similar to Fractal Generation

- Similarity is in the code needed to solve the problems...
- Parsing
  - pattern matching, pattern detection
- Binary Search (function optimization)
  - continuous domain (we already looked at binary search in a discrete domain).
- Other numeric programming techniques.