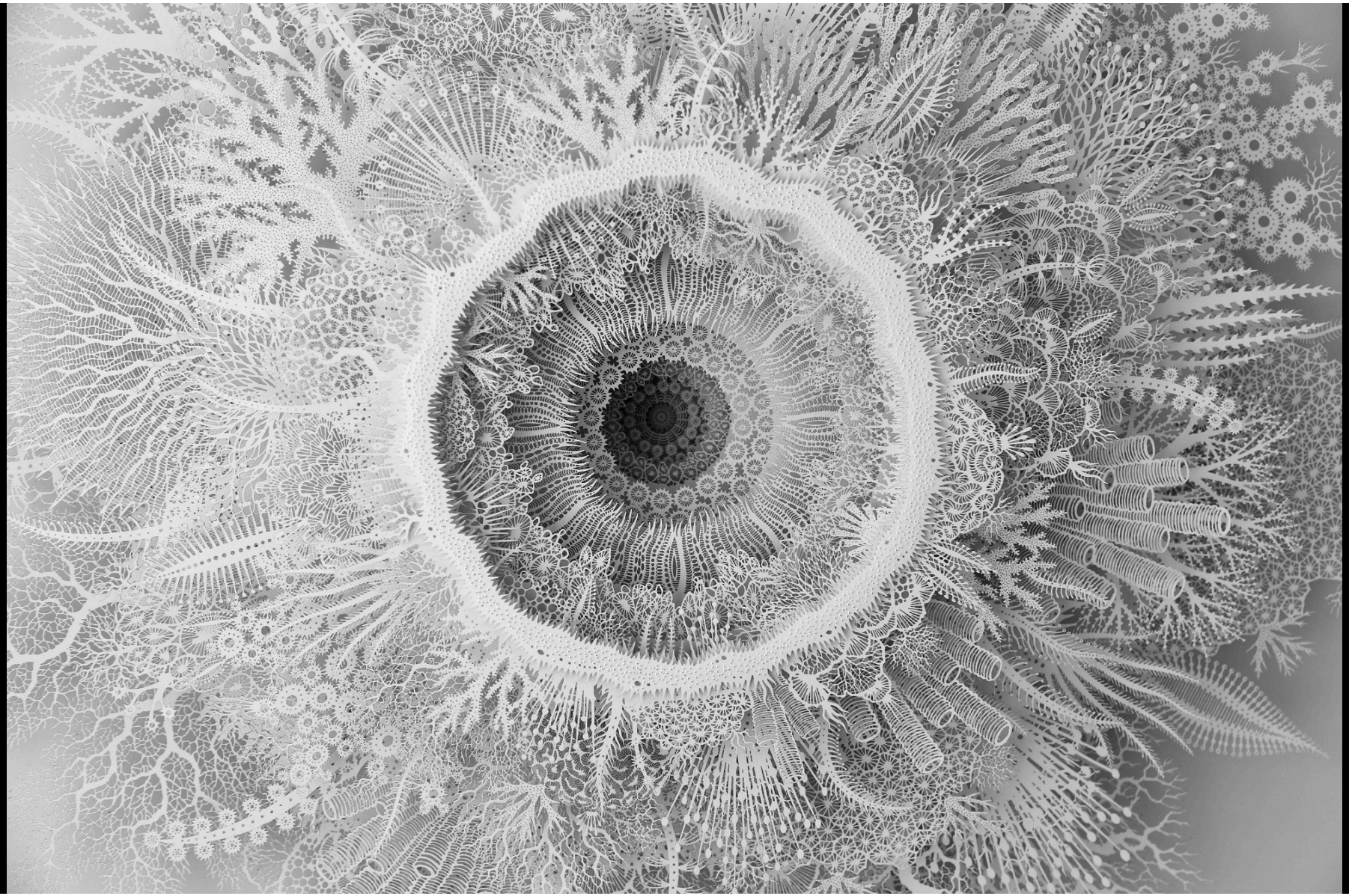
Computational Fabrication

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.cs.unm.edu/classes/Computational_Fabrication_Spring2021/

Weekly Artist: Rogan Brown

https://roganbrown.com/


Rogan Brown


Rogan Brown

Large Assignment 1:
Turtle Geometry & L-Systems

Modeling Plants

Project assignments graded on

Design
Craftsmanship, Code
Craftsmanship, Artifact
Documentation

Documentation:
Taking Good Photographs

Taking Good Photographs

Indirect natural light only
Turn off artificial lights
Turn off flash
Set up by a window
No direct light

Plain background
white or grey
Set up a paper background

Taking Good Photographs




document your entire process including problems

questions?

Office hours (come talk to me!)

When: 11-12pm on Tuesday and Thursday, after class, or by appointment (email me)

Where: Hand and Machine Lab, Farris Engineering Center basement/ground floor. Room G390

Why: Get help with assignments, get help with your printer, career advice, conversation, whatever

Simulating Natural Growth with L-Systems

L-Systems

developed by Aristid Lindenmayer in 1964

Lindenmayer was a Hungarian botanist and biologist

developed as a way to understand plant growth

“Algorithmic Beauty of Plants” published in 1990

L-System

Mathematically a "grammar". Consists of:

Alphabet (V): The symbols in the grammar. Must be finite.

Axiom (ω): A starting word. Must be made from symbols in the alphabet.

Rules(P): Rules or "Productions" that define how symbols in the alphabet should be replaced in each iteration.

Iteration: In an iteration, all symbols in the current word are replaced according to the rules. Replacements happen in parallel.

L-Systems

A set of rules that turn a simple starting word (an “axiom”) into a more complex word by replacing symbols. A grammar.

Example:

axiom (starting word): F

rule: F —> F-F++F-F

3 iterations of the system:

F

F-F++F-F

F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F

L-Systems and Turtle Geometry

Visualize these patterns by translating them into actions carried out by the turtle. Each symbol is translated into an action

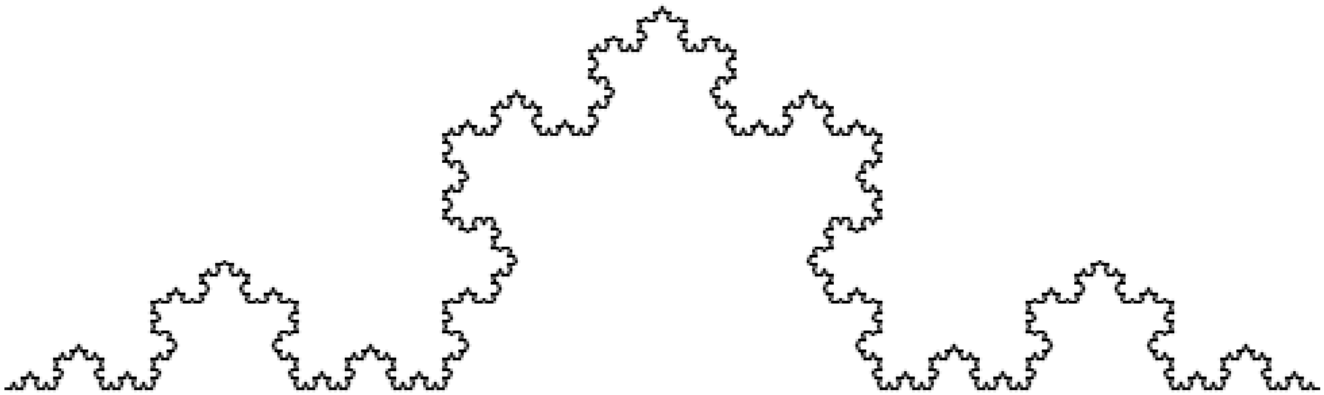
Example:

F = move forward an amount (100)
- = turn left an angle (60°)
+ = turn right an angle (60°)

F
F-F++F-F
F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F

L-Systems and Turtle Geometry

Koch curve



Some Questions
Shh...Raise Your Hand

angle = 60
what is the shape generated by this expression:
F++F++F ?

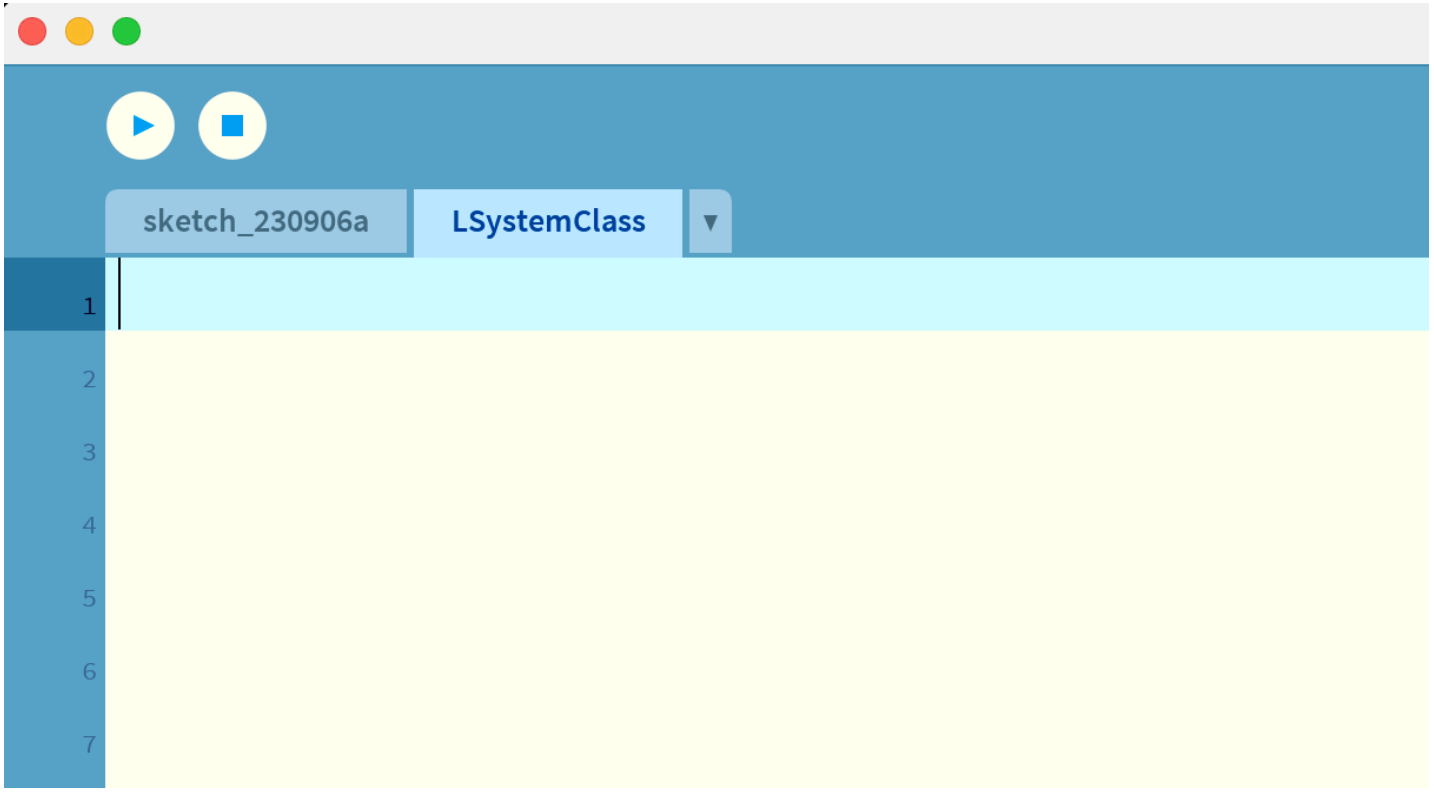
angle = 90
what is the the shape generated by this expression:
F-F-F-F ?

angle = 45
what is the expression for a zigzag?

questions?

Implementation in Code

First: add a new tab for an LSystemClass



LSystem variables and initialization

in the new tab

```
class LSystemClass {
  String axiom = "F";
  String [][] rules = { {"F","F-F++F-F"},
                        {"+","+"},
                        {"-","-"}};
  String computedWord;

  LSystemClass () {
    computedWord = axiom;
  }

}
```

Computing one iteration

What do we do in an iteration?

- Loop through all the characters in the current word
- For each character, check to see what the rule for it is
- Build up a new word that has each character replaced, according to the rules.

iteration method in LSystemClass

```
void iteration() {
  char s;
  String newWord = "";

  //loop through computedWord
  for (int i = 0; i<computedWord.length(); i++) {
    //for every symbol
    s = computedWord.charAt(i);
    //loop through rules to find a match
    for (int j=0; j<rules.length;j++) {
      if (s == rules[j][0].charAt(0)) {
         newWord = newWord + rules[j][1]; //replacement
         break;
      }
    }
  }
  computedWord = newWord;
  println(computedWord);
}
```

Run program and check output

back to main tab

```
import Turtle.*;
LSystemClass l;

void setup() {
  size(900,900);
  background(#2C9093);
  stroke(255);
  l = new LSystemClass();
  frameRate(1);
}

void draw() {
  l.iteration();
}
```

printed output, 3 iterations:

```
F-F++F-F
F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F
F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F-F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F
```

Now draw the System

What do we do to draw?

- Loop through all the characters in the word
- For each character, take a turtle action

add size and angle variables to class

```
class LSystemClass {
  String axiom = "F";
  String [][] rules = { {"F","F-F++F-F"},
                        {"+","+"},
                        {"-","-"}};
  String computedWord;
  float size, angle;

  LSystemClass (float size, float angle) {
    this.size = size;
    this.angle = angle;
    computedWord = axiom;
  }
```

initialize variables in constructor (LSystemClass method)

add drawLSystem method to LSystemClass

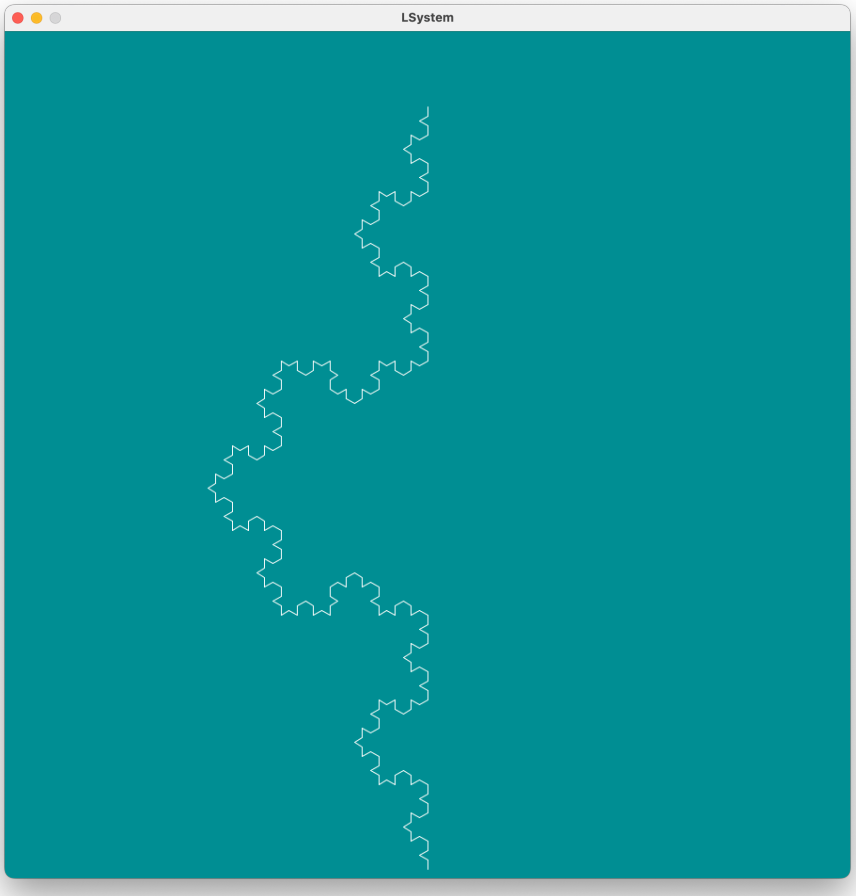
```
void drawLSystem(Turtle t) {
  //loop through computedWord and draw each symbol
  char s;
  for (int i=0;i<computedWord.length();i++) {
    s = computedWord.charAt(i);
    if (s=='F')
      t.forward(size);
    else if (s=='+')
      t.right(angle);
    else if (s=='-')
      t.left(angle);
  }
}
```

back to main tab

```
import Turtle.*;
LSystemClass l;
Turtle t;

void setup() {
  size(900,900);
  background(#2C9093);
  stroke(255);
  l = new LSystemClass(10,60);
  t = new Turtle(this);
  frameRate(1);
}

void draw() {
  t.setX(width/2);
  t.setY(height-10);
  background(#2C9093);
  l.drawLSystem(t);
  l.iteration();
}
```



Add scaleFactor variable to keep drawing on the screen

main tab

```
import Turtle.*;
LSystemClass l;
Turtle t;

void setup() {
  size(900,900);
  background(#2C9093);
  stroke(255);
  l = new LSystemClass(10,60,1);
  t = new Turtle(this);
  frameRate(1);
}

void draw() {
  t.setX(width/2);
  t.setY(height-10);
  background(#2C9093);
  l.drawLSystem(t);
  l.iteration();
}
```

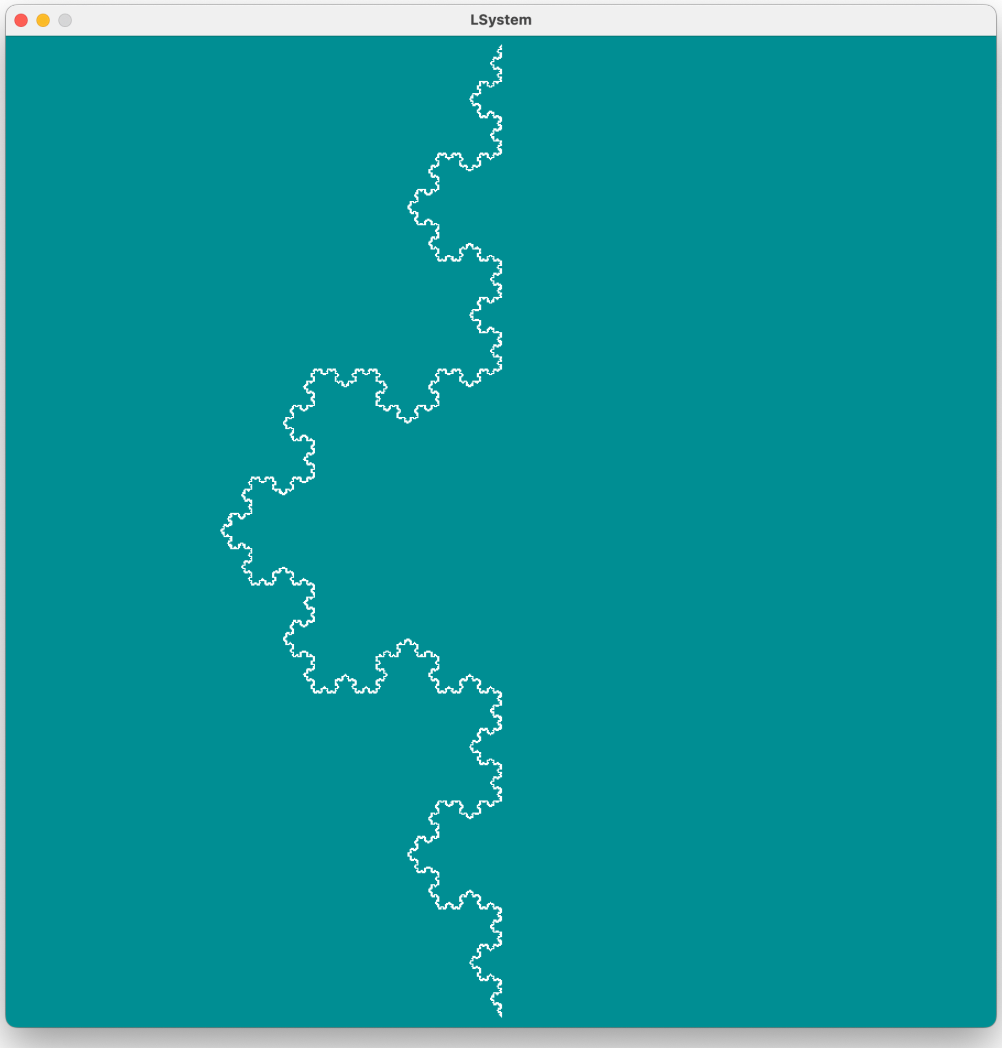
LSystemClass

```
class LSystemClass {
  String axiom = "F";
  String [][] rules = { {"F","F-F++F-F"},
                        {"+","+"},
                        {"-","-"}};
  String computedWord;
  int iterations;
  float size, angle, scaleFactor;

  LSystemClass (float size, float angle, float scaleFactor) {
    this.size = size;
    this.angle = angle;
    this.scaleFactor = scaleFactor;
    computedWord = axiom;
    iterations = 0;
  }

  void iteration() {
    char s;
    String newWord = "";
    size = size*scaleFactor;
    //loop through computedWord
    for (int i = 0; i<computedWord.length(); i++) {
      //for every symbol
      s = computedWord.charAt(i);
      //loop through rules to find a match
      for (int j=0; j<rules.length;j++) {
        if (s == rules[j][0].charAt(0)) {
           newWord = newWord + rules[j][1]; //replacement
           break;
        }
      }
    }
    iterations++;
    computedWord = newWord;
    println(iterations +": " +computedWord);
  }

  void drawLSystem(Turtle t) {
    //loop through computedWord and draw each symbol
    char s;
    for (int i=0;i<computedWord.length();i++) {
      s = computedWord.charAt(i);
      if (s=='F')
        t.forward(size);
      else if (s=='+')
        t.right(angle);
      else if (s=='-')
        t.left(angle);
    }
  }
}
```


LSystem

questions?

Notice: Fractal Structure

Play with angle variable

See book for other rules to explore!