Computer Graphics

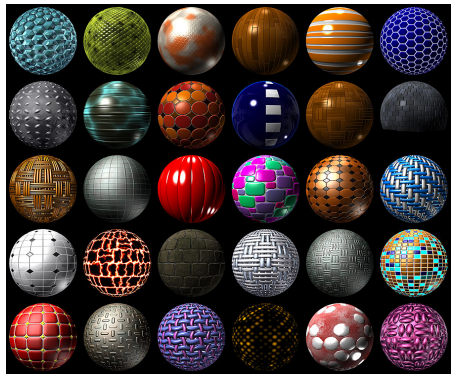
Procedural Methods - L-Systems Part 1

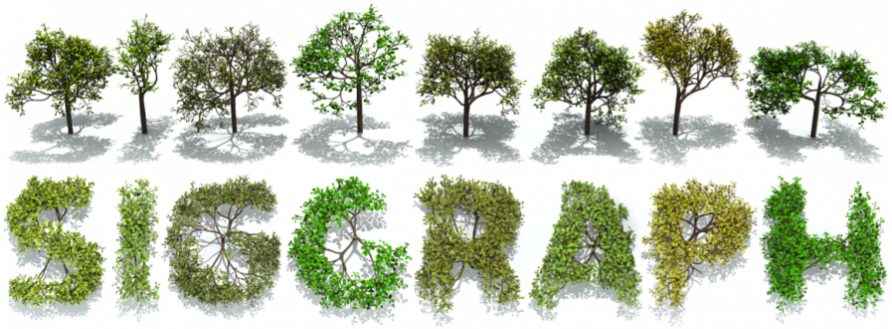
Mark Pauly

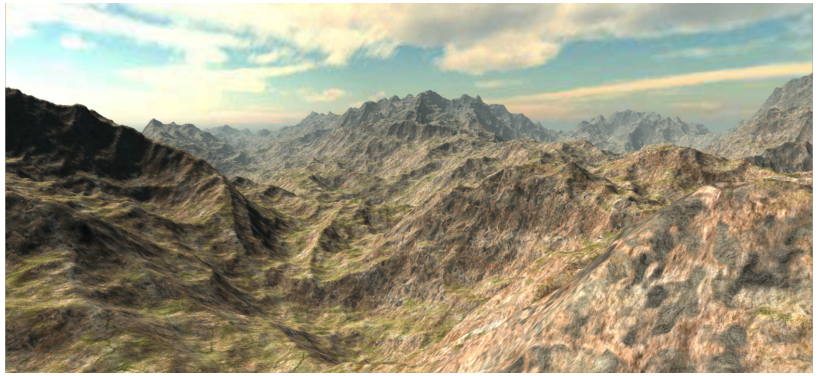
Geometric Computing Laboratory

Recap: Procedural Techniques

- Ubiquitous in graphics
  - texturing, modeling, animation, etc.







Recap: Procedural Approach

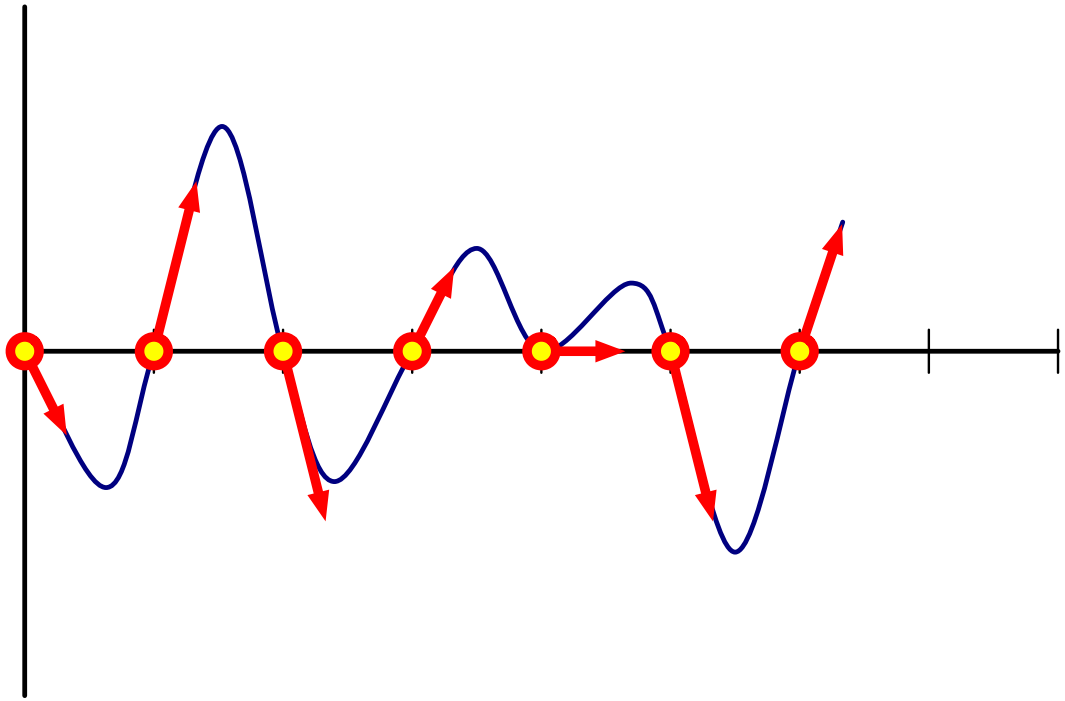
- Why?
  - automatic generation on the fly
  - compact representations
  - infinite detail
  - unlimited extent
  - parametric control
- Particularly suitable for models resulting from processes that are repeating, self-similar, or random
- Challenges: artistic control, debugging, efficiency

Recap: Noise Functions

- Function $\mathbb{R}^n \rightarrow [-1, 1]$, where $n = 1, 2, 3...$
- Desirable properties
  - No obvious repetition
  - Rotation invariance
  - band-limited
    - frequencies stay finite
    - more structure than white noise
  - efficient to compute
  - reproducible
- Fundamental “primitive” or building block of most procedural synthesis approaches

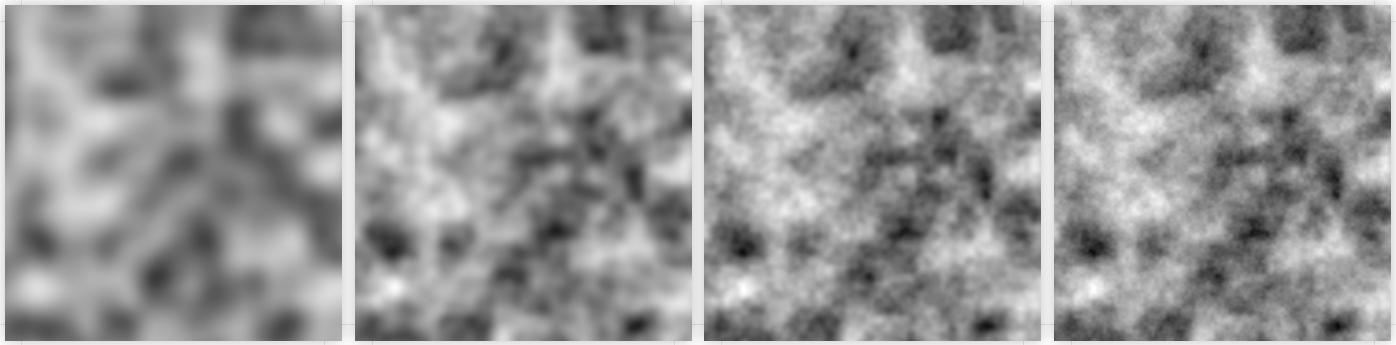
Recap: Classic Perlin Noise (1980s)

- Interpolate random gradients with Hermite interpolation

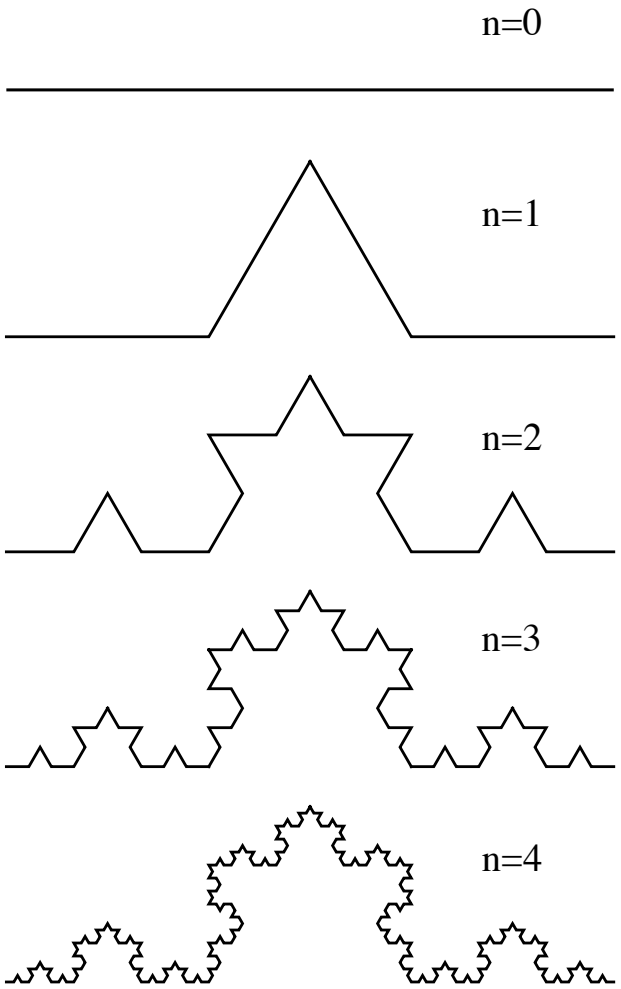


Recap: Fractal Brownian Motion (fBm)

- Spectral synthesis of noise function
  - Progressively smaller frequency
  - Progressively smaller amplitude
- Typically Perlin noise is used
- Each term in the summation is called an *octave*
- Each octave typically doubles frequency and halves amplitude



Recap: Fractal Dimension Example: Koch Curve



$$D = \frac{\log(N)}{\log(r)}$$

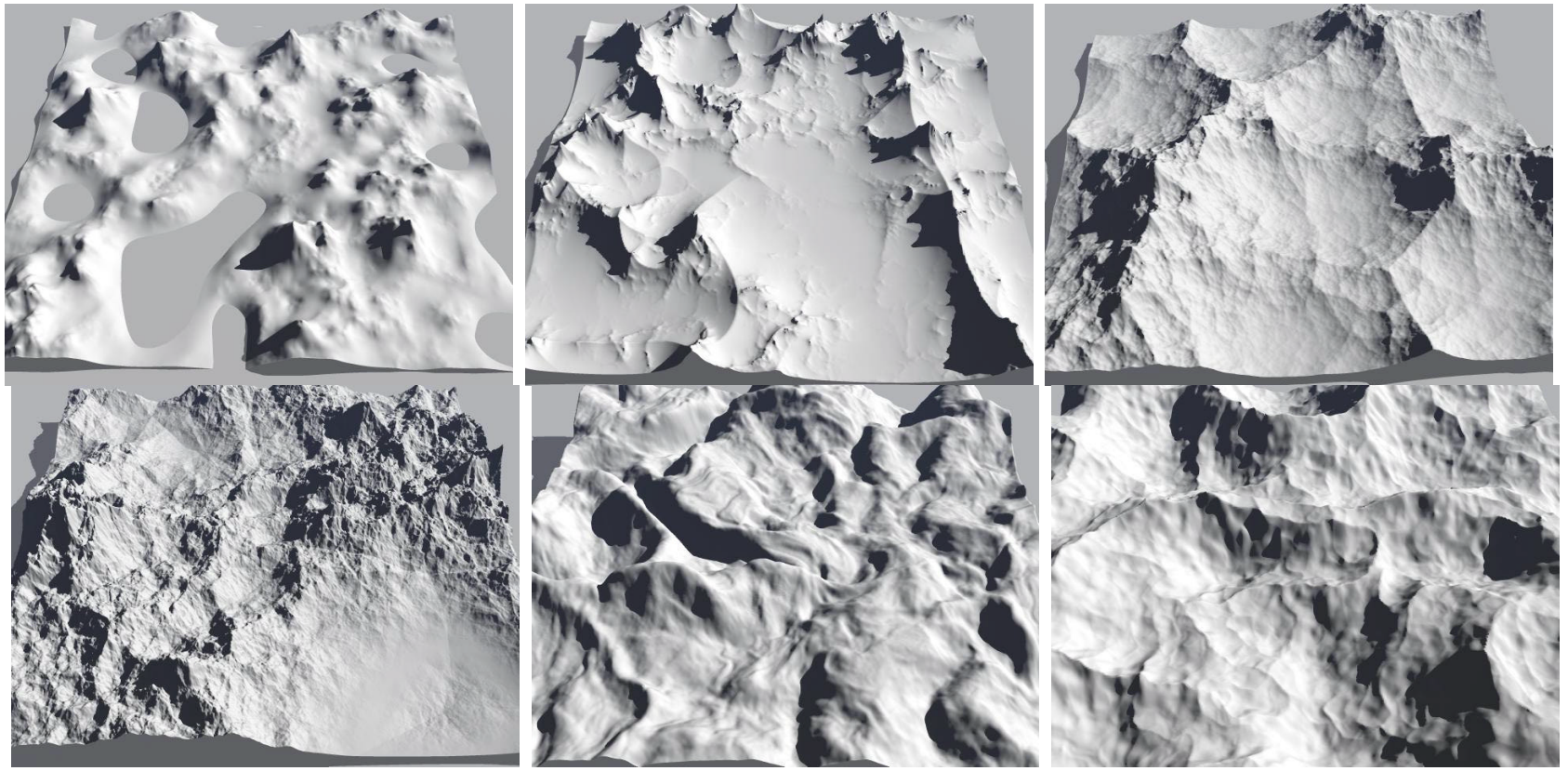
]

$$N = 4$$
$$r = 3$$
$$D = \log(4)/\log(3)$$
$$= 1.26185951\ldots$$

Recap: Multifractals

- Fractal system which has a different fractal dimension in different regions
- Heterogeneous fBM
  - Typical implementations *don't* just spatially vary the H parameter
  - One strategy: scale higher frequencies in the summation by the value of the previous frequency.
  - Many possibilities: heterogenous terrain, hybrid multifractal, ridged multifractal
    - See the Texturing & Modeling book [Ebert et al.] for details

Recap: Heterogeneous fBm



source: Ken Musgrave

Plants

L-System

- An *L-system* is a string rewriting system (semi-Thue grammar) invented by Aristid Lindenmayer (1968).
  - Conceived as a mathematical theory of plant development
  - Later shown to be a useful tool for graphics by Alvy Ray Smith (1984)
  - Similar to context-free grammars, but rules executed in parallel

Wikipedia

L-System

- An *L-system* is a string rewriting system (semi-Thue grammar) invented by Aristid Lindenmayer (1968).
  - Conceived as a mathematical theory of plant development
  - Later shown to be a useful tool for graphics by Alvy Ray Smith (1984)
  - Similar to context-free grammars, but rules executed in parallel
- L-System $\mathbf{G} = (V, \omega, P)$
  - Grammar on an alphabet of **symbols**, V, such as “F”, “+”, “-”.
  - **Production rules** P describe the replacement of a nonterminal symbol with a string of zero or more symbols.
  - Process is seeded with an **axiom** ω, an initial string

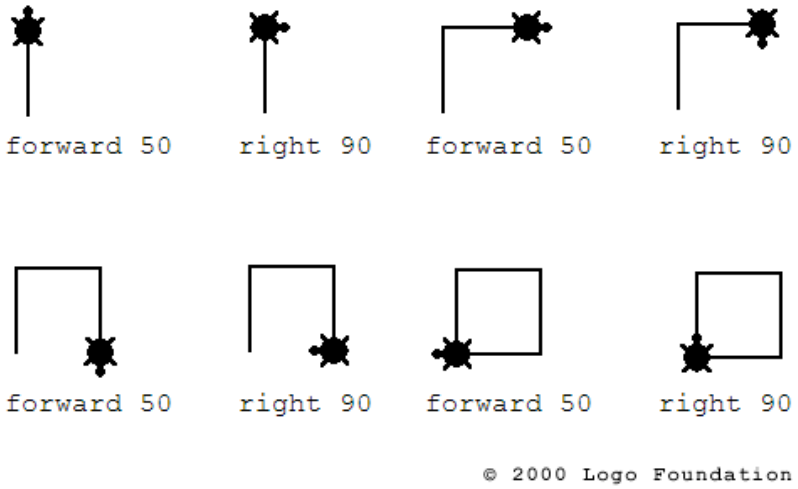
Wikipedia

L-System: Example

- Input:
  - $\omega : F$
  - $P : F \rightarrow F + F - F$
- Output:
  - level 0: F
  - level 1: $F + F - F$
  - level 2: $F + F - F + F + F - F - F + F - F$
  - etc.

Turtle Graphics

- A state of the *turtle* is defined as a triplet (x, y, α)
  - Cartesian coordinates (x, y) represent the turtle's position
  - Angle α, called the heading, is interpreted as the direction in which the turtle is facing.



Wikipedia

Turtle Graphics

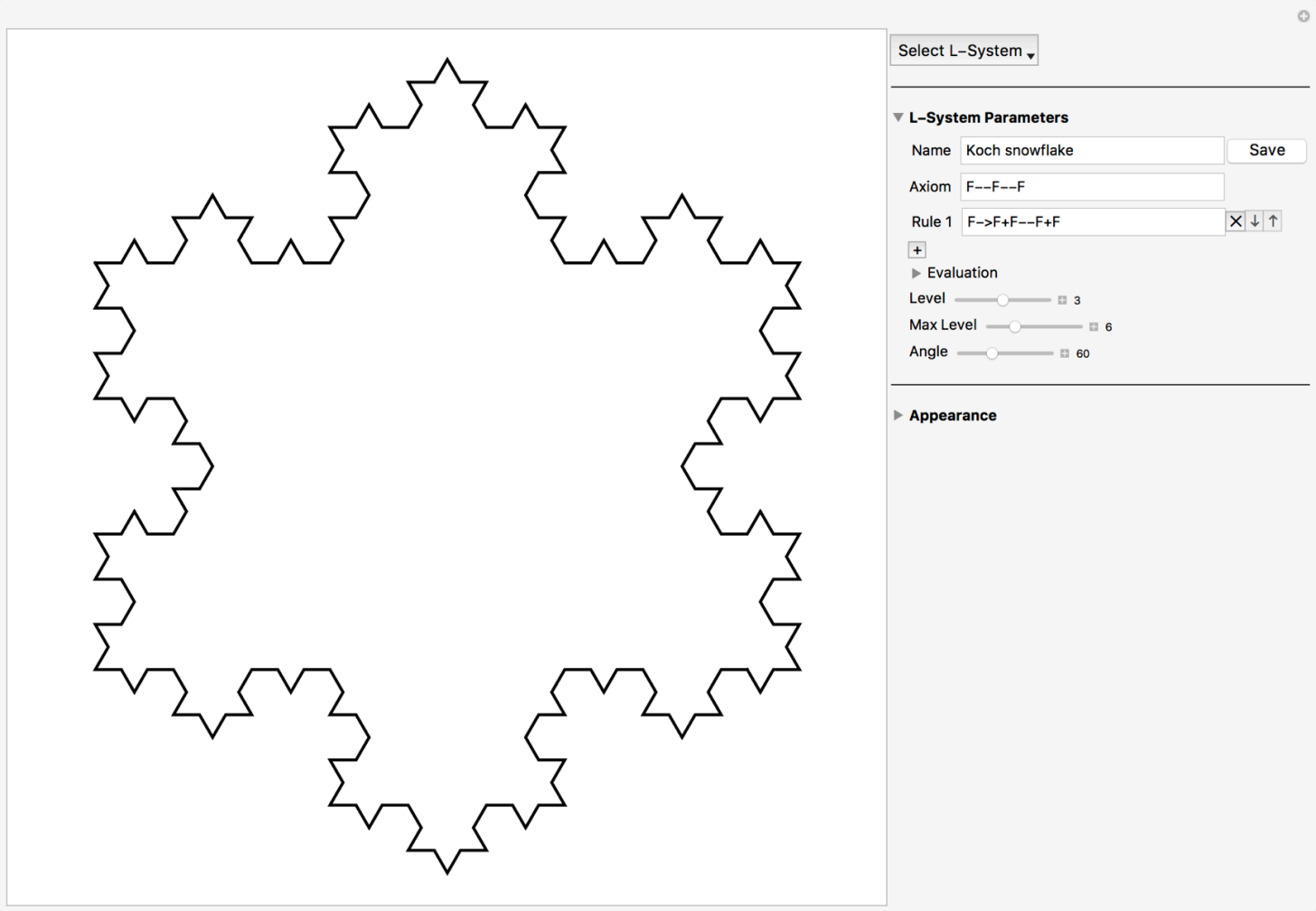
- A state of the *turtle* is defined as a triplet (x, y, α)
  - Cartesian coordinates (x, y) represent the turtle's position
  - Angle α, called the heading, is interpreted as the direction in which the turtle is facing.
  - The turtle also has a pen with attributes such as color, width, and up versus down.
- Turtle Game
- Logo programming language (1967) included turtle graphics

Wikipedia

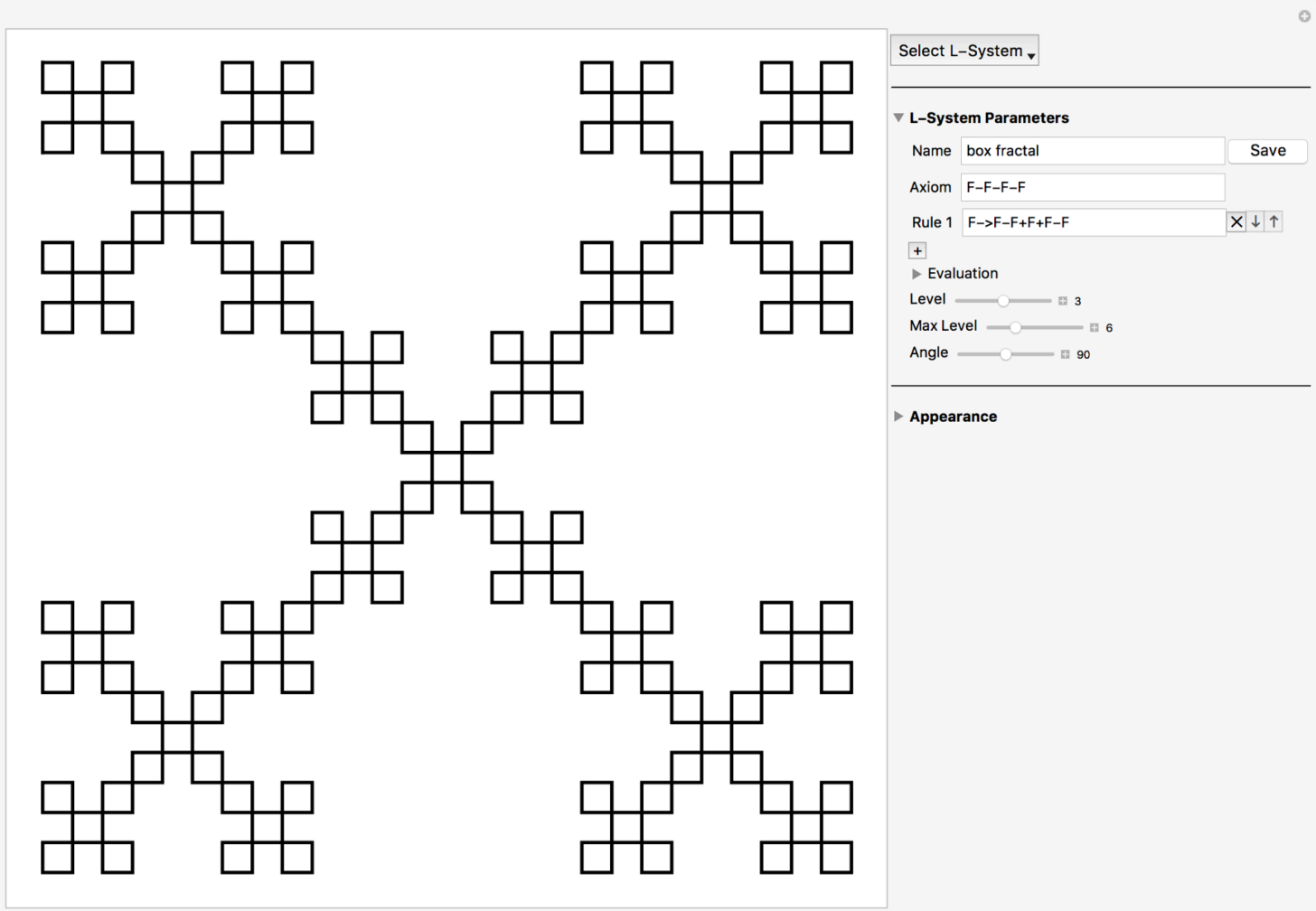
Graphical L-Systems

- L-System string can be interpreted as turtle graphics commands
- F: move forward a step of length d
  - The state of the turtle changes to (x', y', α) where $x' = x + d\cos(\alpha)$ and $y' = y + d\sin(\alpha)$
  - A line segment between points (x, y) and (x', y') is drawn
- $+$: Turn left by angle θ
  - The next state of the turtle is $(x, y, \alpha + \theta)$
- $-$: Turn right by angle θ
  - The next state of the turtle is $(x, y, \alpha - \theta)$
- Graphical L-System Web Applet

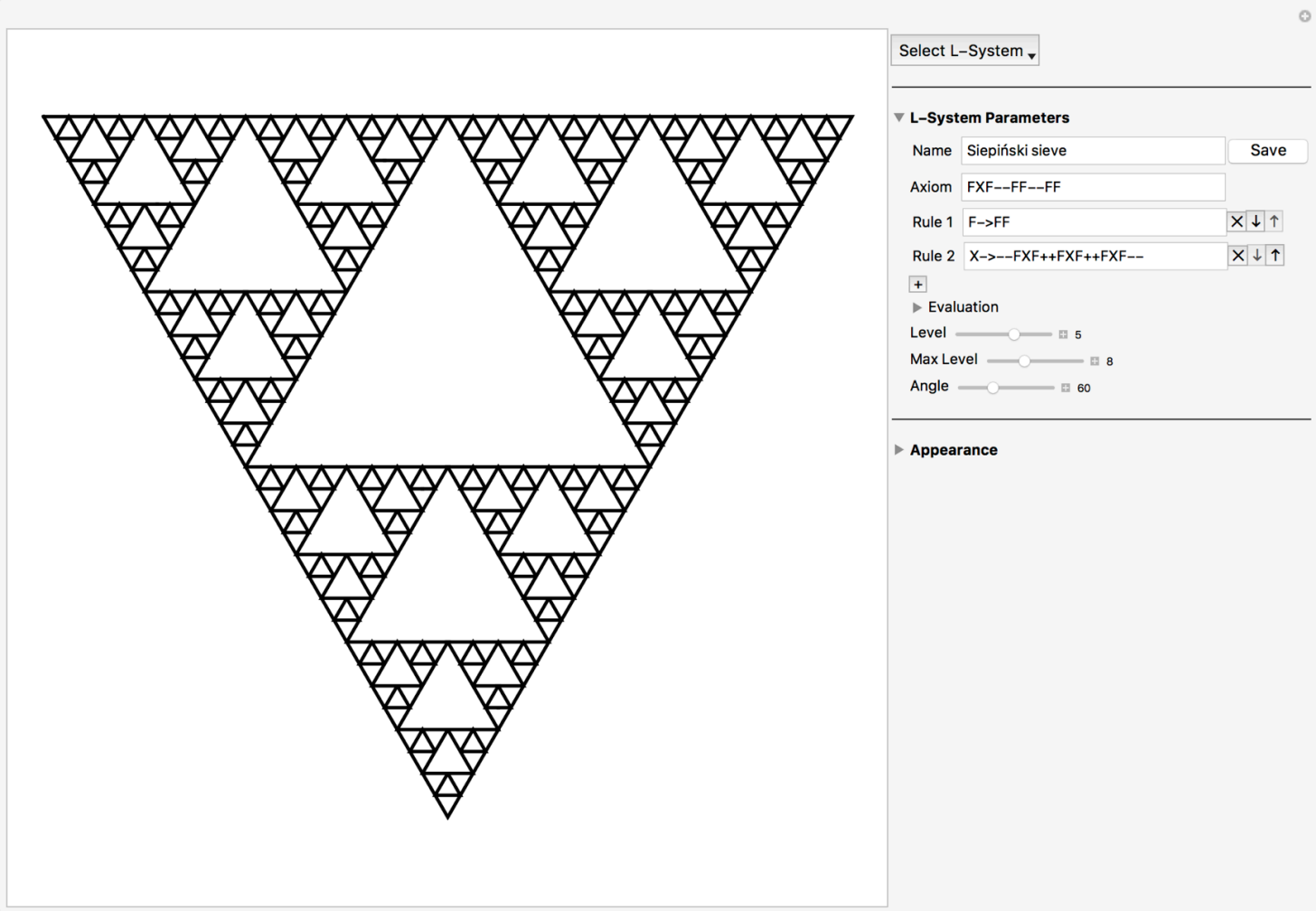
Examples


Select L–System
L–System Parameters
Name
Koch snowflake
Save
Axiom
F−−F−−F
Rule 1
F−>F+F−−F+F
Evaluation
Level
3
Max Level
6
Angle
60
Appearance

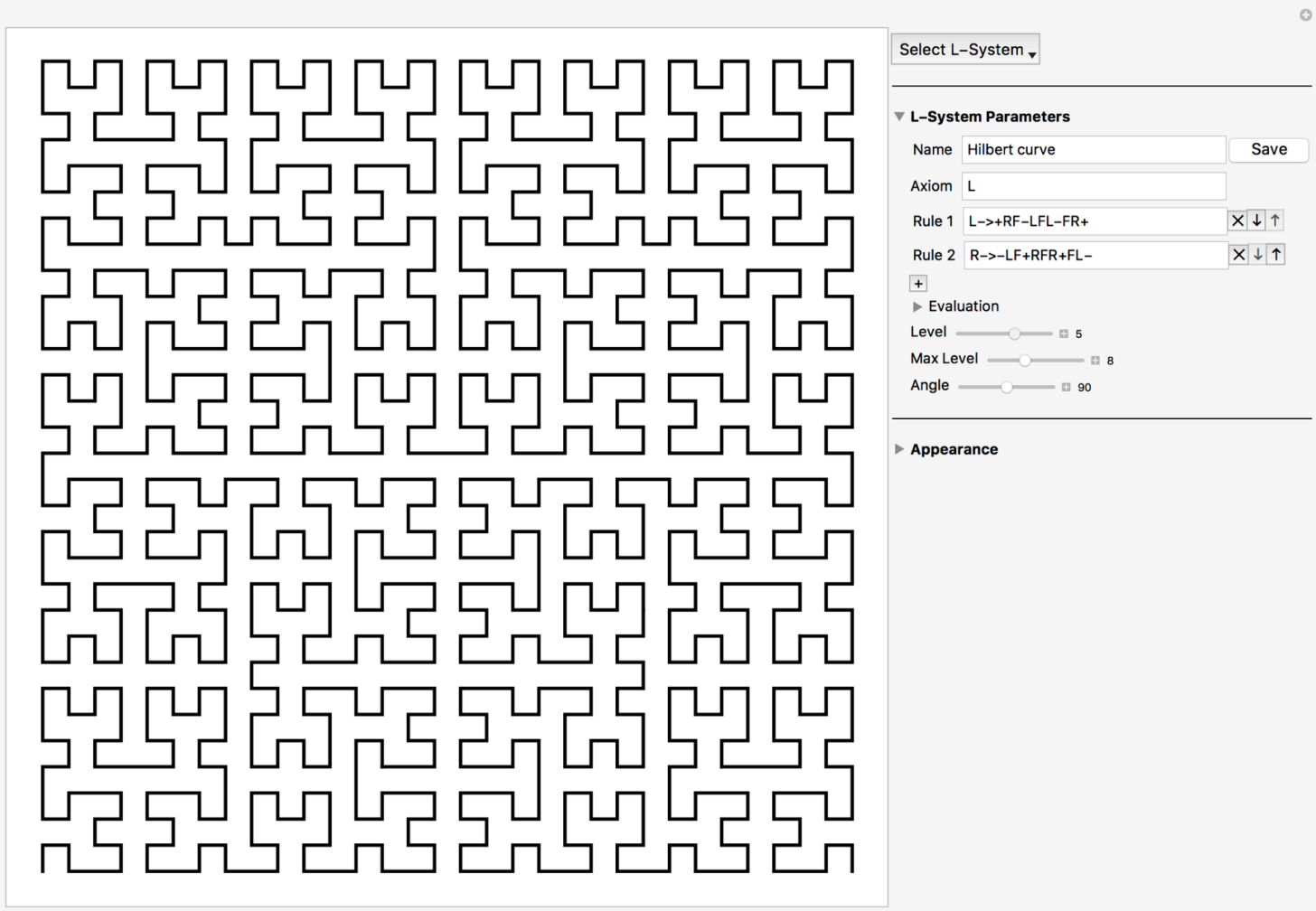
Examples


Select L–System
L–System Parameters
Name
box fractal
Save
Axiom
F–F–F–F
Rule 1
F–>F–F+F+F–F
Evaluation
Level
3
Max Level
6
Angle
90
Appearance

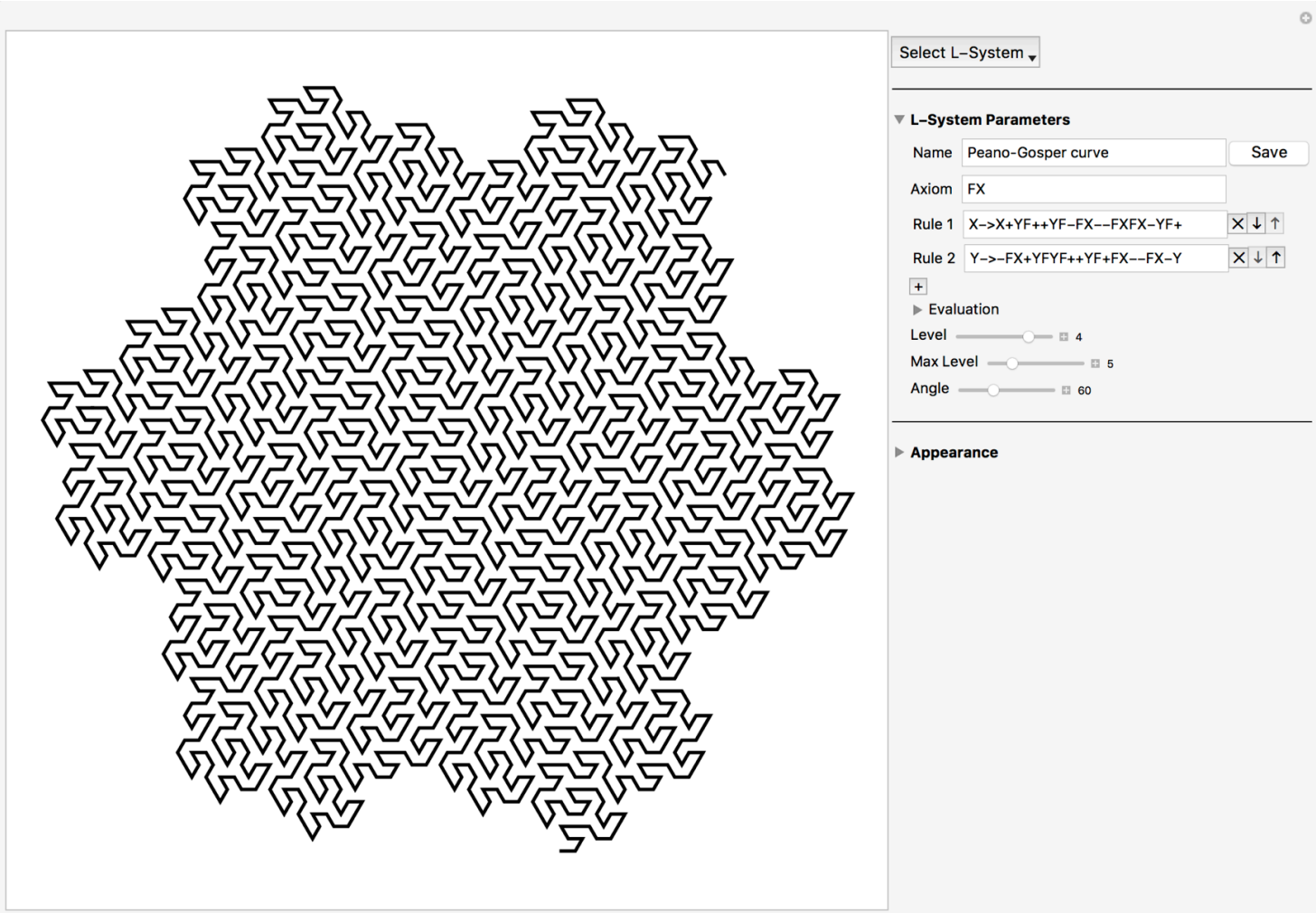
Examples


Select L–System
L–System Parameters
Name
Siepiński sieve
Save
Axiom
FXF––FF––FF
Rule 1
F–>FF
Rule 2
X–>––FXF++FXF++FXF––
Evaluation
Level
5
Max Level
8
Angle
60
Appearance

Examples


Select L-System
L-System Parameters
Name
Hilbert curve
Save
Axiom
L
Rule 1
L->+RF-LFL-FR+
Rule 2
R->-LF+RFR+FL-
Evaluation
Level
5
Max Level
8
Angle
90
Appearance

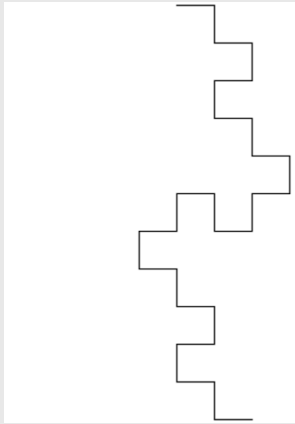
Examples


Select L-System
L-System Parameters
Name
Peano-Gosper curve
Save
Axiom
FX
Rule 1
X->X+YF++YF-FX--FXFX-YF+
Rule 2
Y->-FX+YFYF++YF+FX--FX-Y
Evaluation
Level
4
Max Level
5
Angle
60
Appearance

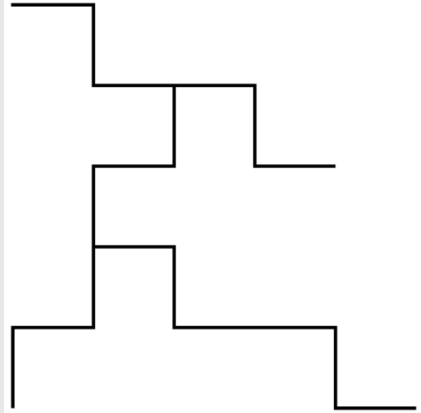
Quiz: Understanding L-Systems

- Which output does the following L-System produce at level 3?
  - $\omega : F$
  - $P : F \rightarrow F - F + F$
  - $\alpha : 90$

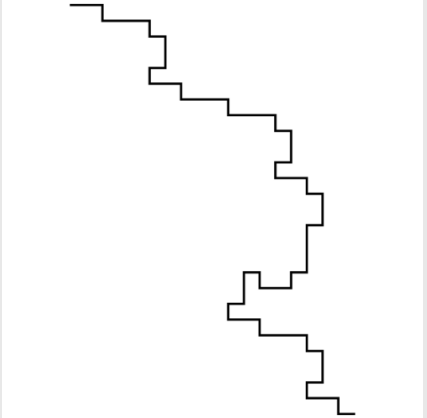
A:



B:



C:



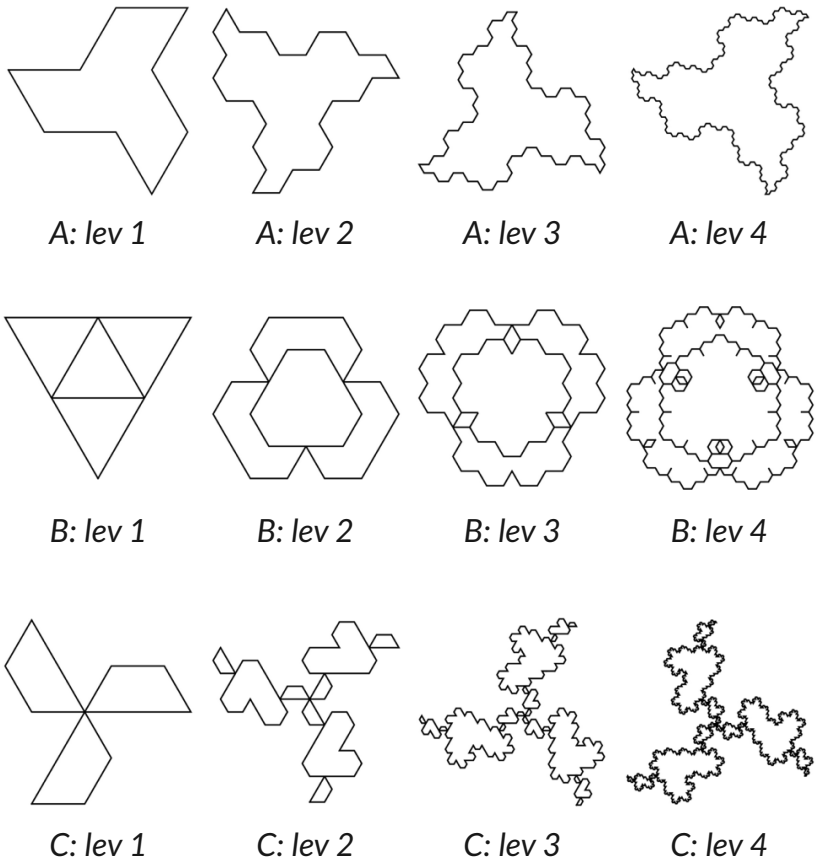
Quiz: Understanding L-Systems

- Which L-System rule produces which output for axiom $F++F++F$ and angle 60 ?
  1. $F \rightarrow F+F+F--FF$
  2. $F \rightarrow -F+F-F+$
  3. $F \rightarrow ++F-F-F$

A: 1B, 2C, 3A

B: 1A, 2B, 3C

C: 1C, 2A, 3B



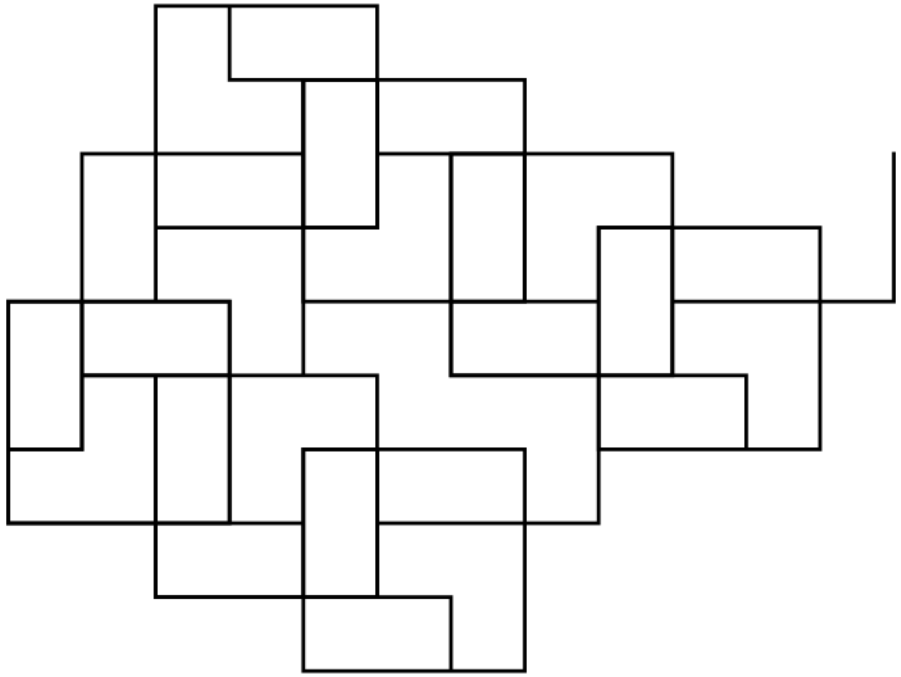
Quiz: Forward Problem

- Which rule produces the following output given axiom F at level 5 and angle 90?

A: $F \rightarrow F - F + F$

B: $F \rightarrow F + FF$

C: $F \rightarrow FFF + FF$



Quiz: Forward Problem

- Which L-System produces the following output given axiom F at level 3 with angle $\alpha = 120$?

A: $F \rightarrow F + F - F + F$

B: $F \rightarrow F + F - F$

C: $F \rightarrow F + F - F + +F$