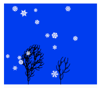
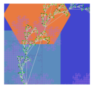
Teaching Computational Thinking with L-Systems and Turtle Graphics

Bruce A. Maxwell
Colby College



Images by Leah Perlmutter



Images by Martha Witick

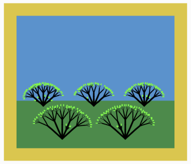


Image by Leah Perlmutter

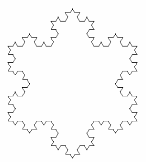
L-Systems

Lindenmayer-systems, or L-systems are a formal grammar created by Lindenmayer and Prusenkeiwicz to model the development of biological systems [2].

L-systems are the basis for many plant and tree models in computer graphics [1].

L-systems can produce strings that have a natural turtle graphics interpretation.

L-systems differ from regular grammars in that a substitution rule applies to all instances of a symbol simultaneously. This models cells dividing in parallel within an organism.



Example:
Alphabet: F, +, -
Base string: F++F++F++
Rule: 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

[1] Deussen et. al., “Realistic modeling and rendering of plant ecosystems”, SIGGRAPH ’98.
[2] P. Prusenkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*, Springer-Verlag, 1990.

Project 1

Introduces students to 1-rule **L-systems** with a small alphabet: forward (F), left (+), right (-), push ([), and pop (]).

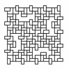
Students read the L-system parameters and rule from a **file** and write the code to implement **substitution**.

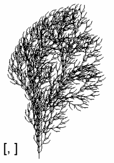
Students write a separate transformer module to **interpret** a string and execute turtle actions, including **pushing** and **popping** the turtle state from a **stack**.

Students have to create a simple scene containing several different types of L-systems.

Space filling curve

Alphabet: F, +, --
Base string: F-F-F-F
Rule: F -> FF-F+F-F-FF





Bushy Plant

Alphabet: F, +, --, [,]
Base string: F
Rule: F -> FF-[F+F+F]+[+F-F-F]

Project 2

Image by Leah Perlmutter

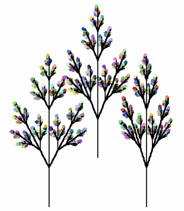
Students re-implement the L-system representation using a **class** instead of a list of lists.

The new class implements **multi-rule L-systems**, which requires a new substitution algorithm and the ability to **copy** information from **nested lists**.

The transformer module becomes a class and **abstracts** the turtle commands so students can create scenes by orienting and placing objects created from L-systems, which now have a larger alphabet.

Node-based plant

Alphabet: F, +, --, [, [, X, L
Base string: X
Rules:
X -> F[+XL][-XL]FXL
F -> FF



Project 3

Students use **dictionaries** to implement **stochastic L-systems**. Stochastic L-systems permit a replacement rule to choose randomly from multiple possible strings. They have to write a new substitution algorithm that integrates random choice.

Students implement a Shape **parent class** that stores the information required to interpret a string as turtle commands and knows how to draw itself using the Transformer class.

Students create several **child classes** that override the constructor to use a different string.

Stochastic Tree

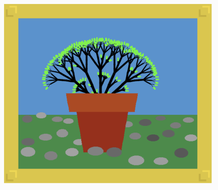
Alphabet: F, +, --, [, [, X, L
Base string: X
Rules:
X -> F[+X]F[-X]+XL or F[+X]+XL or F[]-X]+XL or F[+X]FF[-X]F+XL
F -> FF

Project 4

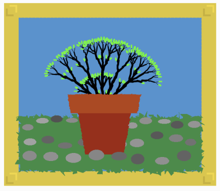
Introduces students to **non-photorealistic rendering [NPR]**, using **abstraction** to override the turtle’s forward command.

The Transformer class can draw an object according to a style. The style parameter controls the action of the forward command.

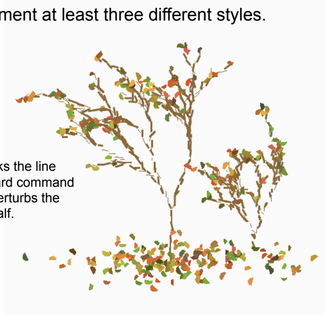
Students implement at least three different styles.



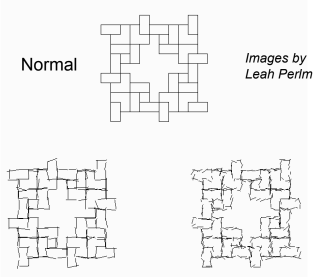
Normal



Jittered



Jitter style that breaks the line created by the forward command into two parts and perturbs the endpoints of each half.



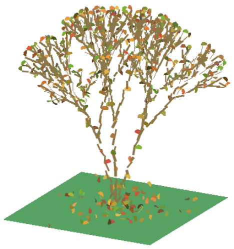
Images by Leah Perlmutter

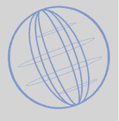
Project 5

Students expand their Lsystem, Transformer, and Shape classes to make use of a **3D turtle package** developed by the author on top of Tkinter.

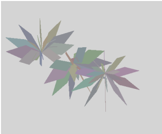
Because of the prior design, students are able to easily generate 3D shapes and draw them in different NPR styles.

The underlying structure is identical. The shape classes define a 3D shape using a string. The Transformer class interprets the string into graphics commands. The forward command implements the NPR settings, and the 3D turtle does the drawing.



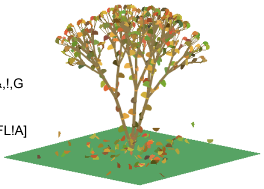


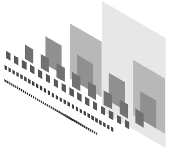
Images by Martha Witick



Stochastic Tree

Alphabet: F, +, --, [, [, X, L,A,/,&,!,G
Base string: A
Rules:
A -> [&FL!A]//////[&FL!A]///////[&FL!A]
F -> S/////GF
S -> GFL





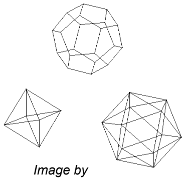


Image by Leah Perlmutter

All assignments and lab exercises are available at http://cs.colby.edu/maxwell/courses/cs151/S09 as projects 8 through 12