Experimenting with Grammars to Generate L-Systems
April 10, 2007

Prof. Susan Rodger
Computer Science Dept
What is Computer Science?

• Computer science is no more about computers than astronomy is about telescopes. - Edsger Dijkstra

• Computer science is not as old as physics; it lags by a couple hundred years. However this does not mean that there is significantly less on the computer scientist’s plate than on the physicist’s: younger it may be, but it has had a far more intense upbringing! - Richard Feynman
Computer Science is a young discipline

- First computer science department formed in 1962

Samuel D. Conte
Professor emeritus of computer science and mathematics was the founding department head of the nation's first computer science program. October 1962.
What is Computer Science?

- Artificial Intelligence

Roomba

Mars Rover

CMU’s Sandstorm
What is Computer Science?

- Animation
What is Computer Science?

• The Organization of Data, and Searching
What is Computer Science?

- Medicine, Genomics
What is Computer Science?

- Devices
Computer Science and Programming

• Computer Science is more than programming
  – Called *Informatics* in many countries
  – Elements of both science and engineering
  – Elements of math, physics, cognitive science, music, art, and many other fields

• To some programming is an art, to others a science, to others an engineering discipline.
Why is Programming Fun?

• What delights may its practitioner expect as a reward
  – First is the sheer joy of making things.
  – Second is the pleasure of making things that are useful.
  – Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts.
  – Fourth is the joy of always learning.
  – Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thoughtstuff.

Fred Brooks
For those of you who have not Programmed:
Programming via Animation and 3D Virtual Worlds

• Introduce the programming language Alice
What Is Alice?

• A modern programming tool
  – 3-D graphics
  – 3-D models of objects

• Animation
  – Objects can be made to move around virtual world
    • a story simulation
    • video game
Objects in Alice

- Objects already exist
- Objects have parts
Alice Demo

• Alice is available for free:
  www.alice.org

• Alice handout to build a sample world
  www.cs.duke.edu/csed/alice/
  click on Alice Materials for Duke Femmes event
Now onto … L-Systems

- Model biological systems and create fractals
- Similar to Chomsky grammars, except all variables are replaced in each step, not just one!
- Successive strings are interpreted as strings of render commands and displayed graphically
English Grammar

- `<sentence> → <subject> <verb> <direct obj>`
- `<subject> → <noun> | <article> <noun>`
- `<verb> → hit | ran | ate`
- `<direct obj> → <article><noun> | <noun>`
- `<noun> → Fritz | ball`
- `<article> → the | an | a`

- Variables (shown in `< >`) are replaced by right side of arrow
Example: Derive a sentence

• `<sentence>` → `<subject> <verb> <direct obj>`
  → `<noun> <verb> <direct obj>`
  → Fritz `<verb> <direct obj>`
  → Fritz hit `<direct obj>`
  → Fritz hit the `<noun>`
  → Fritz hit the ball
Parts of an L-System (a type of grammar)

• Defined over an alphabet
• Three parts
  – Axiom (starting place)
  – Replacement rules (replaces all variables at once)
  – Geometric rules (for drawing)
    • g means move forward one unit with pen down
    • f means move forward one unit with pen up
    • + means turn right by the default angle
    • - means turn left by the default angle
Example – lsys-samp1

- Axiom

- Replacement Rules

- Geometric Rules

NOTE: Must use spaces as separator between symbols
Example – lsys-samp1 (cont)

• Derivation of strings

X

gggX+Y

ggggggX + Y + g

gggggggggX+Y+g+g

gggggggggggggX+Y+g+g+g

Note: replace both X and Y each time
More Geometric rules

• %  change direction 180 degrees
• ~  decrement the width of the next lines
• [  save in stack current state info
• ]  recover from stack state info
• {  start filled in polygon
• }  end filled in polygon
Example – lsys-samp2
Example – lsys-samp2 (cont)

\[ g[\sim+Yg]gX \]

\[ g[\sim+++Yg]gg[\sim+Yg]gX \]

\[ g[\sim++++Yg]gg[\sim+++Yg]gg[\sim+Yg]gX \]

\[ \ldots \]
Example - tree
Example – tree rendered
Stochastic Tree

- Add a rule $T \rightarrow T$

- Now there is a choice for $T$, draw a line or don’t
Same Stochastic L-System

- Rendered 3 times, each at 8\textsuperscript{th} derivation
JFLAP

- JFLAP is available for free:
  www.jflap.org
- JFLAP was developed by many Duke undergraduates over many years, has many other parts to it for studying theoretical computer science concepts
- JFLAP is downloaded in over 160 countries.
- Duke School of Environment uses L-systems to model pine needles in Duke Forest
Exercise 1

• Write an L-system for the picture below.
• Symbols needed are: g, + and one variable.
• Distance of the line is 100, rendering at 1 draws the first line, each additional render draws another line.
Exercise 2

• Write an L-system for the picture below.
• Symbols may need: g and +
• Distance is set to 10, angle to 90, first rendering draws smallest square, additional render draws next larger square
Exercise 3

• Write an L-system for the picture below.
• Symbols may need: g, %, +
• Distance set to 15, angle set to 45, side of square is length 30, first diagonal line is 60
• 1\textsuperscript{st}, 2\textsuperscript{nd} and 6\textsuperscript{th} renderings shown
Exercise 4

• Write an L-system for the picture below (this is a sample tree to focus on branching, don’t look at the tree from before).

• Symbols may need: g, +, -, [ ]

• angle set to 30, distance set to 20

• 3rd rendering shown
Exercise 5

• Write an L-system for the picture below.
• Symbols may need: g, +, -, [ ]
• Angle set to 90, distance set to 15
• Shows 1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd} renderings