Procedural Generation

Dr. Baldassano

chrisb@princeton.edu

Yu's Elite Education

Last class recap: Game Playing

- Game playing is “adversarial search” – we no longer aim for the best goal, but only the best goal our opponent will allow us to reach
- Basic algorithm: minimax
- Speed-ups:
  - Exact: Alpha-beta pruning, equivalent states
  - Approximate: Evaluation heuristics, approximate pruning

Homework: 2-move TTT

- Modify Tic-Tac-Toe program such that each player takes two turns at a time
- How do we change the minimax procedure? Does the game still end in a draw?

Creative algorithms

- Most problems we've talked about have well-defined correct answers
- The challenge has been to find efficient and correct algorithms
- Today we'll talk about the opposite: algorithms that give "creative" outputs, where the "right" answer is much more subjective

Procedural generation

- These algorithms define a set of rules (a procedure) for generating some kind of artistic output
- Called “procedural generation”

Maze generation

- Want to generate a simple maze – enter bottom left, exit top right, no loops
- Remember back to graph class – what would we call this kind of maze?
  - Spanning tree!
- Some ideas:
  - Use Prim's with random edge weights
  - Randomly add edges that don't form loops
    - Breadth-first (choose among all edges)
    - Depth-first (keep branching from same edge for as long as possible)
- http://bost.ocks.org/mike/algorithms/#maze-generation

Generating sequences

- Often we want to generate a sequence of outputs, where each output depends on previous ones
- For example, say we want an algorithm to write a story - it will need to pick a sequence of words, where each word is related to previous words
- Simplest model: each word is chosen probabilistically based on the previous word

Markov chain

- A sequence where each output depends just on the previous output is called a Markov chain
- For text, we look at the last word we output, and then randomly pick the next word based on how common that two-word combination is
- Example: current sentence is "I went to the"
  - Next word we pick is more likely to be "park" than "for", since "the park" is a common combination while "the for" is not

Coding a Markov chain

- We need to set the probability of every two-word English phrase - how can we set all these parameters??
- Machine learning to the rescue!
- Take a massive amount of text, and just count up how many times we see each two-word phrase

Simulating nature

- For video games and art, we often want to simulate things that exist in nature, like plants or organisms
- We want to capture their "organic" structure, but be able to generate an infinite number of examples

Drawing Trees

- Let's start with a 2D tree - what would be a procedure for generating a branching tree?
- One approach:
  - Start with several circles, each of which has a direction of travel
  - Draw circles in their direction of travel, plus some noise
  - Randomly decide to create copy of a circle with a perpendicular direction of travel
  - http://inconvergent.net/generative/hyphae/

L-systems

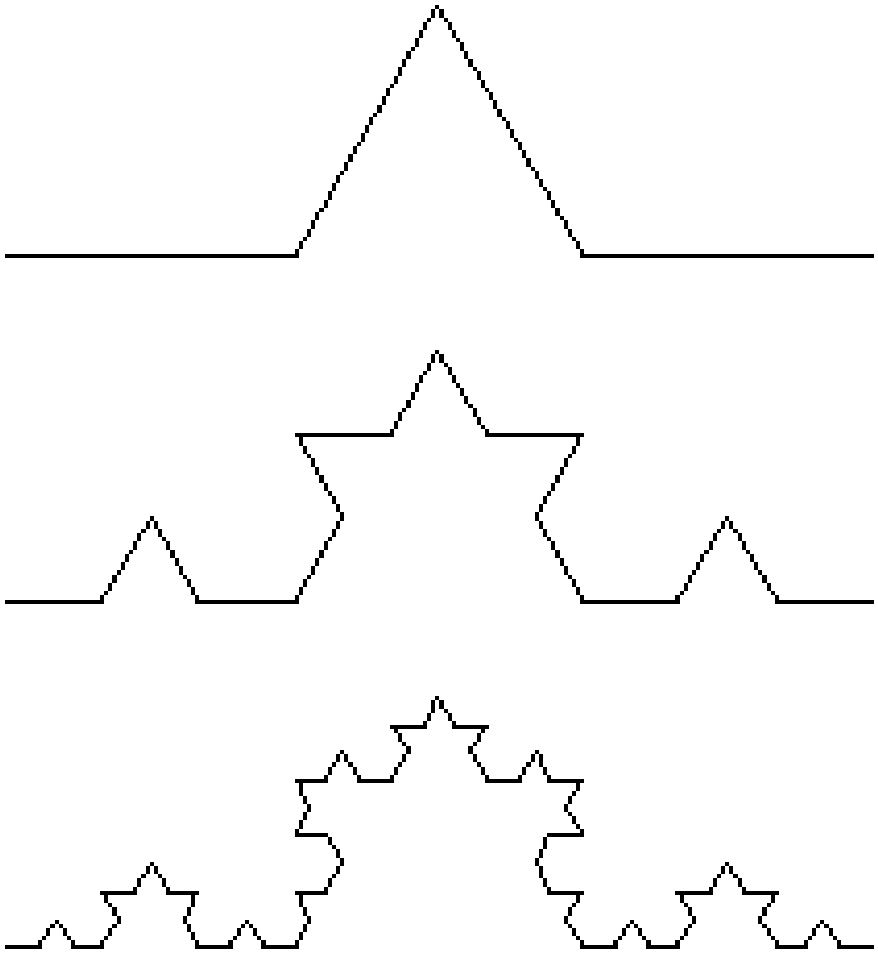
- Many procedural systems can be described as Lindenmayer systems (L-systems)
- Start with some base state, called an "axiom"
- Keep applying some "production rules" that transform the state in some way
- Define how to draw a state

Example L-system

- Axiom: F
- Production rule: F -> F+F--F+F
- Drawing: F is line segment, + rotates CW, - rotate CCW

- States:
  - F
  - F+F--F+F
  - F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

Example L-system



More L-systems

- http://mathforum.org/advanced/robertd/lsys2d.html
- http://www.allenpike.com/modeling-plants-with-l-systems/
- http://josauder.github.io/procedural_city_generation/

- Let's try!

Homework: Higher-order Markov chains

- In a 2nd-order Markov chain, the next state depends on the previous two states (same for 3rd, 4th,…)
- Try different order chains at http://projects.haykranen.nl/markov/demo for the Alice in Wonderland input text
- What happens for low vs high orders?
- (Note that this Markov chain generates each letter, not each word)