Procedural Generation
L-systems and Textures

ICS 162
Lecture #6
Procedural generation

Generate **interesting** geometry, texture, animation, etc. automatically (programmatically)

Why?

- Manpower: Generate content for very large virtual worlds without requiring an artist to create and place every single blade of grass, pebble, etc.

- Compression: Code to generate content is often much smaller than the content generated
L-Systems

- A model of growth and morphogenesis, based on formal grammars (set of rules and symbols)
- Introduced in 1968 by the Swedish biologist A. Lindenmayer
- Originally designed as a formal description of the development of simple multi-cellular organisms
- Later on, extended to describe higher plants and complex branching structures.
Self-similarity in Nature

• Approximate

• Only occurs over a few discrete scales (3 in this Fern)

• Self-similarity in plants is a result of developmental processes, since in their growth process some structures repeat regularly. (Mandelbrot, 1982)
Self-Similarity in Fractals

- Exact
- Example Koch snowflake curve
- Starts with a single line segment
- On each iteration replace each segment by \_
- As one successively zooms in the resulting shape is exactly the same
Snowflake curve rewriting example

generator

initiator

productions
L-Systems Grammar

• Begin with a set of “productions” (replacement rules) and a “seed” axiom
• In parallel, all matching productions are replaced with their right-hand sides
• Ex:
  – Rules:
    • B -> ACA
    • A -> B
  – Axiom: AA

  Resulting Sequence: AA, BB, ACAACA, BCBBCB, ...

• Strings are converted to graphic representations via interpretation as turtle graphics commands
A simple rewrite system

Example:
- Alphabet = \{a, b\}
- Rules = \{a \rightarrow ab, b \rightarrow a\}
- Axiom: \(b\)

Example of a derivation in a DOL-System
“Turtle” Interpretation of Strings

Turtle is at location \((x, y)\) and oriented in direction \(\alpha\)

- **F** Move forward a step of length \(d\). The state of the turtle changes to \((x', y', \alpha)\), where \(x' = x + d \cos(\alpha)\) and \(y' = y + d \sin(\alpha)\). A line segment between points \((x, y)\) and \((x', y')\) is drawn.
- **f** Move forward a step of length \(d\) without drawing a line. The state of the turtle changes as above.
- **+** Turn left by angle \(\delta\). The next state of the turtle is \((x, y, \alpha + \delta)\).
- **-** Turn left by angle \(\delta\). The next state of the turtle is \((x, y, \alpha - \delta)\).
Turtle Interpretation of Strings

\[ w: F+F+F+F \]
\[ p: F \rightarrow F+F-F-FF+F+F-F \]
\[ \text{Angle (}\delta\text{)} = 90^\circ \]

Quadratic Koch island

\[ n = 0 \quad n = 1 \quad n = 2 \]
L-Systems Example: Dragon Curve

- Axiom: $F_l$  \( \delta : 90 \text{ degrees} \)  \( n: 10 \text{ iterations} \)
- \( F_l \rightarrow F_l+F_r+ \)
- \( F_r \rightarrow F_l-F_r- \)
Types of L-systems

- **Context-free**: production rules refer only to an individual symbol
- **Context-sensitive**: the production rules apply to a particular symbol only if the symbol has certain neighbours
- **Deterministic**: If there is exactly one production for each symbol,
- **Stochastic**: If there are several, and each is chosen with a certain probability during each iteration
Bracketed L-systems

• To represent branching structures, L-systems alphabet is extended with two new symbols: [, ], to delimit a branch. They are interpreted as follows:

  [  Push (store) the current state \((x,y, \alpha)\) of the turtle onto a stack.

  ]  Pop (retrieve) a state from the stack and make it the current state of the turtle. No line is drawn, in general the position of the turtle changes
Turtle Interpretation of Bracketed Strings

\[ w: F \]

\[ p: F \rightarrow F[-F]F[+F][F] \]

Angle (\( \delta \)) = 60°

\[ n = 1 - 5 \]
3D L-Systems
3D Bracketed L-Systems
L-system Weeds
L-Systems for Plants

- L-Systems can capture a large array of plant species
- Designing rules for a specific species can be challenging
PovTree
Evolving L-Systems

Obtained by a fitness function considering symmetry only. And interactively mutating and recombining organisms.
Mutation

Symbol Mutation

F[+F]+(+F-F-F)-F[-F-F]

Block Mutation

FF[+FF][-F+F][FFF]F

F[+F]+(+F-F-F)-F[-F][-F-F]

FF[+FF][-F+F][-F]F
Recombination

Parents

Offspring

Results

Considering symmetry only

Considering branching points only

Considering phototropism, and symmetry

Considering phototropism only

Considering phototropism, symmetry and branching points

Considering symmetry only
Some others unexpected figures!

Stars

Animals

Candlestick

Rockets
Environmentally sensitive L-systems

• Example from *Synthetic Topiary*, by P. Prusinkiewicz, M. James & R. Měch

(terminate a branch by deleting all the modules from the current point to the end of this branch)
Synthetic Topiary
Grammar Based Representation of Transmission Towers

Evolutionary approach was applied to the Inverse Problem, i.e., the identification of a grammar that generates a predetermined tower.

Real world towers translated into the grammar language.
Procedural Buildings

Generative Modeling Language (GML)
“A Generative Theory of Shape”, Michael Leyton. (available online)
Procedural Furniture

Parametric modeling allow you to change the size and shape of various parts of an object continuously.
Instancing and Replication

or.. How to make a complete forest. (First try)

Noise can tell us where to plant trees.
L-system can tell us how to build a three-dimensional tree.

But do we make one tree and copy it, or do we make lots of different trees?
Some times **instanc-ing** can look pretty good. Pines trees all look similar.
In other cases, **instancing** does not produce enough visual variation. This also depends on the placement algorithm.
CityEngine (Y.I.H. Parish & P. Müller)

- Uses maps of population density, land-water boundaries, etc. to create a road network using Open L-systems
- Spaces between blocks subdivided into lots for individual buildings
- Buildings generated using parametric stochastic L-systems, detail with procedural texturing technique
L-Systems for Cities

- Start with a single street
- Branch & extend w/ parametric L-System
- Parameters of the string are tweaked by goals/constraints
- Goals control street direction, spacing
- Contraints allow for parks, bridges, road loops
- Once we have streets, we can form buildings with another L-System
- Building shapes are represented as CSG operations on simple shapes
Highway generation goals

- Population density
  - Highways connect peaks on population density map
  - End of segment shoots rays across map; each ray is sampled and sample points look up population value
  - “Fitness” of ray is the sum of the population values along a ray multiplied by the inverse distance of the corresponding point on the ray
  - Highway generation continues in the direction of the “fittest” ray
Road generation goals

• Street patterns
  – Configurable using street pattern control maps
    New York:
      • Streets form blocks by restricting the length of a block and the angles that streets follow
    – Paris
      • Concentric rings around a central point, connected by short radial streets
    – San Francisco
      • Tries to reduce the length of non-contour streets; uses gradient of elevation map
Road generation constraints

• Modify attributes (length, angle) of road segments in response to surroundings
• Make sure roads do not cross water and any other places you’d expect not to find them
• Creates intersections with other roads
L-systems can be used to describe many objects where the structure changes iteratively, such as city blocks.

http://www.youtube.com/watch?v=yI5YOFR1Wus
Division into Lots

• City divided into “blocks”

• Blocks subdivided into allotments

– Blocks assumed to be convex and rectangular

→ Concave allotments forbidden
Building Generation L-System modules:

- Transformation modules
- Extrusion module
- Branching and Terminating modules
- Geometric templates

Final shape is ground plan transformed by L-System
Semi-automatic Building texture generation

→ Facades show 1 or more overlayed grid structures, with most cells having the same function

→ Grid cells are influenced by surrounding cells

→ Irregularities mostly affect entire rows or columns
Map the resulting layout to:

- A procedural texture
- An image map
- Another nested layer or layerstack
L-Systems: Further Readings

• Algorithmic Botany
  – Covers many variants of L-Systems, formal derivations, and exhaustive coverage of different plant types.
  – [http://algorithmicbotany.org/papers](http://algorithmicbotany.org/papers)

• PovTree
Texture

- Texture is “stuff” (as opposed to “things”)
- Characterized by spatially repeating patterns

radishes  rocks  yogurt
Texture Synthesis

• Goal of Texture Synthesis: create new samples of a given texture
• Many applications: virtual environments, hole-filling, texturing surfaces
Political Texture Synthesis!

Bush campaign digitally altered TV ad

President Bush’s campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.

Original photograph
The Challenge

• Need to model the whole spectrum: from repeated to stochastic texture
Statistical modeling of texture

- Assume stochastic (random) model of texture where the appearance of a pixel can be guessed based on the color of pixels in some neighborhood.
Text Synthesis

• [Shannon, ’48] proposed a way to generate English-looking text using N-grams:
  – Use a large text to compute probability of each letter given N-1 previous letters
  – Starting from a seed repeatedly sample to generate new letters
  – Also works for whole words

WE NEED TO EAT CAKE
Mark V. Shaney (Bell Labs)

• Results (using alt.singles corpus):
  – “As I've commented before, really relating to someone involves standing next to impossible.”
  – “One morning I shot an elephant in my arms and kissed him.”
  – “I spent an interesting evening recently with a grain of salt.”

• Notice how well local structure is preserved!
  – Now let’s try this in 2D...
Efros & Leung Algorithm

- **Input image**
- **Candidate pixel and comparison region**
- **Completed portion (grey)**
- **Output image**
Efros & Leung Algorithm

• Assume Markov property
  – Building explicit probability tables infeasible
  – Instead, we search the input image for all sufficiently similar neighborhoods and pick one match at random
Varying Window Size

input
Varying Window Size
Synthesis Results

french canvas

raffia weave
More Results

white bread

brick wall
Failure Cases

Growing garbage

Verbatim copying
Block based algorithm

- Pick size of block and size of overlap
- Synthesize blocks in raster order
- Search input texture for block that satisfies overlap constraints (above and left)
- Paste new block into resulting texture
  - blend together the overlap nicely by finding a clean cut
Failures
(Chernobyl Harvest)
Next week:

Moving models from Maya into games
- Level of detail, mesh simplification, texture maps, compression...
MEL Scripting Intro

• MEL is Maya’s built in scripting language
  – Allows programmatic access to underlying functions in Maya
  – Almost all of Maya’s own user interface is built on top of MEL
  – Allows you to build custom tools that interact with Maya
    • High demand in animation and game industry for technical artists
Accessing MEL

• Click on “script editor” icon in lower right-hand corner
  – Command history shows MEL commands that have been executed. Try creating a sphere and note the command that pops up in the history window. *This is very useful!*
  – Enter a command in the MEL window. Highlight text and hit Ctrl+Enter to execute it.
  – Create scripts, save them, add them to custom “shelf”
MEL commands

Example:

//create a sphere or radius 12
polySphere –r 12;

//move it distance 10 along x-axis
move –r 10 0 0
Online help:

polySphere

Go to: Synopsis, Return value, Related, Flags, MEL examples.

Synopsis


polySphere is immutable, queryable, and editable.
The sphere command creates a new polygonal sphere.

Return value

string/ Object name and node name.
In query mode, return type is based on queried flag.

Related

polyCone, polyCube, polyCubes, polyCylinder, polyPlane, polyTorus

Flags

axis, constructionHistory, createUIs, name, object, radius, subdivisionsX, subdivisionsY, subdivisionsZ, texture

<table>
<thead>
<tr>
<th>Long name (short name)</th>
<th>Argument types</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis (-ax)</td>
<td>list of lists of lists</td>
<td>C C C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This flag specifies the primitive axis used to build the sphere.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q: When queried, this flag returns a float.</td>
<td></td>
</tr>
<tr>
<td>radius (-r)</td>
<td>list</td>
<td>C C C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This flag specifies the radius of the sphere.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C: Default is 0.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q: When queried, this flag returns a float.</td>
<td></td>
</tr>
<tr>
<td>subdivisionsX (-sx)</td>
<td>int</td>
<td>C C C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This specifies the number of subdivisions in the X direction for the sphere.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C: Default is 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q: When queried, this flag returns an int.</td>
<td></td>
</tr>
</tbody>
</table>
MEL Variables

int $var = 1;

float $var = 3.123;

string $var = “sphereNumber1”;

int $nums[];               //dynamic array

int $vals[5]={1,2,3,2,1};  //arrays are 0 indexed
MEL commands

When you create a new bit of geometry it is given a name automatically.

polySphere –r 12 –n “mysphere”

Saving return values from commands:
string $retval[] = `polySphere –r 12`
Printing out messages

print $var;

print ("a string ="+$var+"\n");

Useful for debugging!
Listing and Selecting Nodes

//list & select
ls -tr; //list transform nodes
ls -s; // list out shape nodes
ls -sl; // list currently selected objects

string $selectednodes = `ls -sl`;
select -cl; //clear any selection
select -r $name; //select the specified node