Announcements

- Final project outline due November 19
- Second midterm exam: Tuesday November 23rd during lecture hours (2-3:20pm)
- Phi will do late grading for assignment 6 tomorrow (Friday): 2-3pm
- Please check Gradesource for accuracy. Homework assignments 1-6 and midterm should be complete.
Final Project

- Problem description is on-line
- Must be done in teams of two or three
- Application design description (min. 300 words) due Friday, November 19 at 4pm.
  Email to me: jschulze@ucsd.edu
- Project due to be presented on Friday, Dec 3rd, between 2 and 4pm
- No late submissions accepted
StarCAVE Tour

- Location: Atkinson Hall, 1st floor
- 18 Dell XPS PCs with Quad Core Intel CPUs
- CentOS 5.3 Linux
- Dual Nvidia Quadro 5600 graphics cards per node
- 34 JVC HD2k projectors (1920x1080 pixels): ~34 megapixels per eye
- Passive stereo with circular polarization filters
- 15 screens, ~8 x 4 feet each
- Floor projection
- Optical, wireless tracking system
- Software: COVISE
- Programming Language: C++

Tour Date:
- Friday, Nov 19, 4:00-5:00pm

Location:
Immersive Visualization Laboratory
1st floor Atkinson Hall
Turn right at main entrance
Lecture Overview

- Procedural Modeling
  - Concepts
  - Algorithms
- Shadow Volumes
Modeling

- Creating 3D objects/scenes and defining their appearance (texture, etc.)
- So far we created
  - Triangle meshes
  - Bezier patches
- Interactive modeling
  - Place vertices, control points manually
- For realistic scenes, need extremely complex models containing millions or billions of primitives
- Modeling everything manually is extremely tedious
Alternatives

- **Data-driven modeling**
  - Scan model geometry from real world examples
  - Use laser scanners or similar devices
  - Use photographs as textures
  - Examples
    - [http://www-graphics.stanford.edu/data/3Dscanrep/](http://www-graphics.stanford.edu/data/3Dscanrep/)
    - [http://www.tsi.enst.fr/3dmodels/](http://www.tsi.enst.fr/3dmodels/)
    - .ply file format reader
      - [http://w3.impa.br/~diego/software/rply/](http://w3.impa.br/~diego/software/rply/)

- **Procedural modeling**
  - Construct 3D models and textures with algorithms
**Procedural Modeling**

- Wide variety of techniques for algorithmic model creation
- Used to create models too complex (or tedious) for a person to build
  - Terrain, clouds
  - Plants, ecosystems
  - Buildings, cities
- Usually defined by a small set of data, or rules, that describes the overall properties of the model
  - Tree defined by branching properties and leaf shapes
- Model is constructed by an algorithm
  - Often includes randomness to add variety
  - E.g., a single tree pattern can be used to model an entire forest

[Deussen et al.]
Randomness

- Use some sort of randomness to make models more interesting, natural, less uniform, clean
- *Pseudorandom* number generation algorithms
  - Produce a sequence of (apparently) random numbers based on some initial seed value
- Pseudorandom sequences are repeatable, as one can always reset the sequence
  - E.g., if a tree is built using pseudorandom numbers, then the entire tree can be rebuilt by resetting the seed
  - If the seed is set to a different value, a different sequence of numbers will be generated, resulting in a (slightly) different tree
Recursion

- Repeatedly apply the same operation (set of operations) to an object
- Generate objects that are self-similar: fractals
  - Objects that look the same when viewed at different scales
- For example, the shape of a coastline may appear as a jagged line on a map
  - As we zoom in, we see that there is more and more detail at finer scales
  - We always see a jagged line no matter how close we look at the coastline
Lecture Overview

- **Procedural Modeling**
  - Concepts
  - Algorithms
- **Shadow Volumes**
Height Fields

- Landscapes are often constructed as *height fields*
- Regular grid on the ground plane
- Store a height value at each point
- Can store large terrain in memory
  - No need to store all grid coordinates: inherent connectivity
- Shape terrain by operations that modify the height at each grid point
- Can generate height from grey scale values
  - Allows using image processing tools to create terrain height
  - → Extra credit in Homework Assignment #2
Fractals

- Fractal: Fragmented geometric shape that can be split into parts, each of which is (at least approximately) a reduced-size copy of the whole

- Self-similarity

- Demo: Aros Fractals
  http://www.arosmagic.com/redblue/arosmagic.com/fractals/

From Wikipedia
Fractal Landscapes

- Random midpoint displacement algorithm
  - Recursively subdivide triangles
  - Displace edge midpoints with fractal formula
  - Reduce size of displacement as triangles get smaller
  - Similar for quadrilaterals
Fractal Landscapes

- Add textures, material properties; use nice rendering algorithm
- Example: Terragen Classic (free software)
  [http://www.planetside.co.uk/terragen/](http://www.planetside.co.uk/terragen/)[http://www.planetside.co.uk/gallery/f/tg09]
L-Systems

- Developed by biologist Aristid Lindenmayer in 1968 to study growth patterns of algae
- Defined by grammar

\[ G = \{ V, S, \omega, P \} \]

- \( V \) = alphabet, set of symbols that can be replaced (variables)
- \( S \) = set of symbols that remain fixed (constants)
- \( \omega \) = string of symbols defining initial state
- \( P \) = production rules
Sierpinski Triangle

- **Variables:** A, B
  - Draw forward
- **Constants:** +, -
  - Turn left, right by 60 degrees
- **Start:** A
- **Rules:** $(A \rightarrow B-A-B)$, $(B \rightarrow A+B+A)$

2 iterations 4 iterations 6 iterations 9 iterations
Fractal Fern

- **Variables:** $X$, $F$
  - $X$: no drawing operation
  - $F$: move forward
- **Constants:** $+,$ $-$
  - Turn left, right
- **Start:** $X$
- **Rules:**
  \[(X \rightarrow F-[(X)+X]+F[+FX]-X),(F \rightarrow FF)\]
- **Stochastic L-system**
  - If there is more than one production rule for a symbol, randomly choose one
Fractal Trees

- Recursive generation of trees in 3D
  
  http://web.comhem.se/solgrop/3dtree.htm

- Model trunk, branches as cylinders

- Change color from brown to green at certain level of recursion

Fractal tree  Sierpinski tree
Algorithmic Beauty of Plants

- Online book on algorithmic beauty of plants by Prusinkiewicz
  
  http://algorithmicbotany.org/papers/#abop

Buildings, Cities

Pascal Mueller
[http://www.vision.ee.ethz.ch/~pmueller/publications.html]
Demonstration: Procedural Buildings

- fr-041: debris. by Farbrausch, 2007
- 179 KB ZIP file
- http://www.farbrausch.de/
Lecture Overview

- Procedural Modeling
  - Concepts
  - Algorithms
- Shadow Volumes
Shadow Volumes

Light source

Surface outside shadow volume (illuminated)

Shadow volume (infinite extent)

Surface inside shadow volume (shadowed)

Eye position (note that shadows are independent of the eye position)

Shadowing object

Partially shadowed object
Shading With Shadow Volumes

- Many variations
- Stencil shadow volumes
  - Classic algorithm
  - Hard shadows
- Here, two-pass algorithm for approximate soft shadows
  - Very simple and inaccurate, but often plausible enough
- Many more complicated and more accurate variations exist
Shadow Volume Construction

- Need to generate shadow polygons to bound shadow volume
- Extrude silhouette edges from light source

Extruded shadow volumes
Shadow Volume Construction

- Needs to be done on the CPU
- Silhouette edge detection
  - An edge is a silhouette if one adjacent triangle is front facing, the other back facing with respect to the light
- Extrude polygons from silhouette edges
To be continued after midterm...

- Because we have not fully covered shadow volumes, they are not going to be part of the material for the midterm.
Next Lecture

- Second Midterm Exam