Lecture 13: Thursday, 16 October 1997

Announcements:
- Prof. McMillan speaks on VR, current & future
- Monday 3:45-5 in 34-101 (w/ refreshments)

RESOURCES:
- Each other (see course team page)
- Tools (see course tool page)
- Space (4Gb on imagery.mit.edu)
- Staff (talk to / email us)

Final Project discussion
- Timeline (L11 handout):
  - Team Formation
  - Proposal submitted
  - Feedback from Staff
  - Checkpoint 2-5: Meetings with staff
  - Project Writeup (Web document)
  - Project Presentation (Web slides)

Last Year's Projects

```
Distributed Rendering and Viewing Pipeline

Converting Inventor scene graph into Random Dot stereograms.

Distributed raytracing over a network.

A Java program for Image Morphing

A graphical flower that dances to music played in the CD drive.

An Implementation of Occluder Nodes as Applied to Quake

Physical Modeling of Coin Flips

Animation: Speeder Bike Chase on Endor

Water wavelet simulation controlled by sampled/MIDI audio data.

Distributed Graphics Computation and Rendering Pipeline

Escape from a room by solving interactive puzzles.

Animation of a Pinball machine/game.

Generation of comic-book style images from 3-d models

Real-time virtual fish tank in Java
```



```
A Machine Vision Object Tracker using the IndyCam

Light Field Rendering

Interactive Radiosity Renderer/Viewer

Java Raytracer

Real-time, 3D game simulation with overdraw reduction

An Environment for Interactive Object Manipulation

Interactive Plant Growth

Masquerade: 3D world with human-to-human expressive interaction

Interactive 3D Rube Goldberg Machine Simulator.

Rendering raindrops on the surface of a puddle.

An Emergent Ultimate Frisbee simulation.

One on One Basketball Animation

Dancing Robot Man with Motion Capture

Melting Ice Cube

Thin Soap Films or Soap Bubbles
```



```
Elevator/Bus Route/Highway Simulation

Roller Coaster Design / Physically Accurate Simulations

An illustrated, animated storybook

A robot arm shooting a basketball

Alien Abduction Animation on Alias

Modeling a distributed virtual environment

A realistic model and display of the human body

Analyze and implement a global illumination algorithm
```

Today:
- Project Scope
- Broad Categories
  - (Not just animation!)
- Some example topics
- Proposal specifics
  - What do we expect?

Project Scope

Teams of 3 or 4 people
- Six weeks of real team effort
- Effort per person: roughly 5-6 assts
- Related teams are OK, **but**
  - don't put other teams on your critical path
  - each team must show self-contained result

Project Components:
- Reasonable Scope/Goals
- Substantial Team Design Effort
- Individual Technical Mastery (Code, Tools)
- Individual Creative Expression
- Communication of Work, Lessons Learned

Deliverables:
- Source, binaries, presentation, slides
- Documentation for future students
- Feedback (about tools, staff, etc.)
- Artifacts suitable for 26-100 "6.837 night"
  - Images, videos, etc.
- All archived on the Web (via imagery)

Broad Project Areas

Modeling (Modeling *Tools*)
- Generate interesting model of "something"
- OR: Make a tool for generating interesting models
- Rotoscope, motion capture, facade, etc.

Animation. Requires:
- Plot, storyboard, modeling, keyframes, teamwork, planning
- Visualization
  - Explain complex concept with graphics
  - Start with, arrive at thorough understanding

Simulation
- Appearance: Sophisticated illumination/rendering
- Behavior: Physically-based modeling
- Critical: Study of actual phenomenology

Optimization
- Accelerate a classical algorithm
- Interactive ivscan? ivray? ivrad? ivview?
- Fast particle system update method?

Interaction/Collaboration
- Single-person, multi-person game
- Virtual environment/workspace
- Interesting new technique
- Use of IndyCam, sound, network, etc.

Modeling and Modeling Tools

Model N campus buildings (outside or inside!)
- Photographic textures, photoCD, etc.
- We can lend you digital cameras, tripods

Build a set of primitives, and workspace
- Basic 3D modeler, on top of Inventor
- Athena Tools: AC3D; Alias; ProEng; Sketch

Pattern builders:
- Stone walls, vegetation, building facades...
- Procedural modeling/texturing tools
- Athena Tools: RenderMan shader/renderer

Simulation

Illumination/rendering extensions
- Ray-Tracing:
  - Dispersion, refraction, area lights
  - Motion blur, depth of field, etc
  - Real camera model, match to photographs
- Radiosity:
  - Equilibrium of diffuse luminaires/reflectors
- Hybrid algorithms
  - View-independent, view-dependent effects

Visualization (as in FlyRay)
- Possibilities (speed knob; debug level)
- Visual debugging

Animation

Plot, storyboard, keyframes, ...

Direct specification
- Simple animation language (repeatable)
- Hand-built script of object articulations/poses
- Use of `uid_foo` sources for objects

Direct manipulation
- `gview`, manipulate, write (tedious)

In-betweening in Inventor
- `Engines, Calculators`
- Code against Inventor run-time library

Use Alias, RenderMan on Athena. **Planning !**

Physically-based modeling

Gravity, EM forces, wind, heat, etc.
Auto-alignment, disjointness (UI techniques)
Water; ice; fire; collapse; etc.
Particle systems, spring networks...

Optimization

Accelerate classical algorithm by exploiting:
- Temporal coherence
- Spatial coherence

What is the first thing to do when optimizing?
Rendering pipeline
- Smart z-buffer, primitive tagging, etc.

Ray Tracing
- Voxel-based ray propagation
- Reusing radiance samples
- Lazily-built environment maps (on-line QTVR)

Radiosity
- Fluid interface
- Convergence speed
- Environment editing

Interaction Techniques

Of course, overlap with modeling, etc.
Games: (not 3D Tetris, Doom, Descent etc.)
Fluid Simulation/Rendering systems (*not* just algs)
Constraint-based Interaction techniques

Resources

`/usr/share/src/Inventor/`
- `demos/`
- `examples/`
- `samples/`
- `tools/`
- `GamesCD/`

ProEngineer from Parametric Technology
- `add parametric; proeng`
- Online help system `proguide`

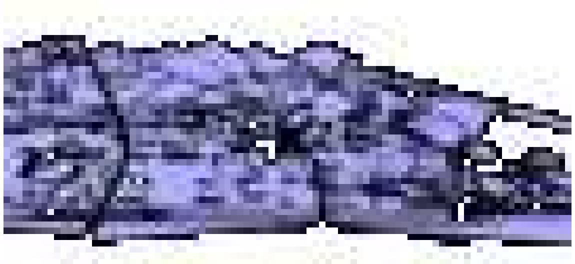
ac3D, Public domain 3D modeler
- Described on course Tools page

Alias modeler (ditto)
Renderman renderer
- `add renderman; set_renderman; set_demos`

Scripting support in ivscan, ivray
- Single-step, continuous motion; camera scripting
- Inventor, Scan Converter, Ray Traced rendering
- Human-readable, editable format
  - (or, can generate programmatically)

Modeling

Model MIT Campus (exteriors); photomap textures



MIT Office of Facilities and Management:
`http://web.mit.edu/ofms-space/www/`
`http://whereis.mit.edu/doc/`
`http://ortho.mit.edu` (b&w orthos)
color orthophotos (by arrangement w/ graphics group)

Model MIT Campus (interiors); photomap textures
arguably harder 3D modeling task

Or... model some other space

Figure Modeling

Convincing model of articulated human figure
Reasonable dynamics for motion (walk, run, etc.)
Could also do some part of figure (e.g., face)
Perhaps model underlying bones, musculature
Analogous: animals, plants
Phenomenology!

Parametric Modeling

All uid_foo programs used `argparse()`.
Modify to generate web interface to programs
Architect to allow continuous selection?
Used for last year's, this year's, future classes!

Detail Modeling

Procedural approach:
Stone walls
Mosaics
Example: Dome of the Rock

Dome of the Rock

Proposal Specifics

Web document, send us URL (for now, host locally)
State your project title, team, abstract (1 paragraph)
Brief (3-5 page) writeup of project goals, including
- **Problem** you'll solve, effect you'll achieve, etc.
- Proposed **design**, **including division of labor**
- **Tools** you will use, from where, on what platform(s)
- **Milestones** for checkpoints 1-5, presentation

Your document must:
- Have title, team, abstract
- Have four sections (Goals, Design, Tools, Milestones)
- Respect reality
  - Pose realistic goals
  - State fallback positions

We will:
- Give detailed written & individual feedback
- Assign each project to one matching TA
- Schedule weekly meetings, one per team per TA