Lecture 12: Tuesday, 19 October 1999

Announcements:
- Asst 6A and 6B (Ray Casting/Tracing) due F 5pm
  - Eye-ray and jittered eye-ray generation
  - Intersection: spheres, cones, cylinders, polygons
  - Utility functions (listed in `README`):
    - `ReflectionDirection()`
    - `TransmissionDirection()`
    - `etc.`
  - Turnin **6A and 6B** together as one set of files:
    `turnin -course 6.837 6 foo.C bar.C ...`

Final Project preparations:
- Requirements
- Deliverables
- Timeline
- Team Formation
- Resources

Requirements

Final Project consists of
- Some independent investigation
- Significant design experience
- Significant implementation effort
- Realistic engineering plan, milestones
- Credible overlap with 6.837

Our tasks include:
- Helping you "scope" your project
  - Six weeks of effort for three people
  - That's more than four person-months
- Reacting to your proposal with meaningful technical feedback
- Helping you prioritize milestones
  - Breadth-first, depth-first progress
- Weekly meetings to identify resources, break roadblocks, provide moral support

Deliverables

End-to-end deliverables:
- Written proposal, timeline & URL
  - 3-5 pages, with URL
- Implemented project
  - Source, inputs, outputs
- Written report & URL
  - 10-15 pages, with URL
- Presentation & demo
  - 15-20 minutes, in room 3-133
- Presentation slides & URL
- Video/audio artifact (to TAs)

Timeline

Final Project timeline:
- Team formation: ongoing (class project page)
- Brainstorming: today in class, and by email
- Friday, 29 October: teams, Written Proposals due
- Friday, 5 November: Checkpt. 2 with staff
  - Feedback on goals, milestones, tools
- Friday, 12 November: Checkpt. 3 with staff
- Friday, 19 November: Checkpt. 4 with staff
- Friday, 26 November: Thanksgiving Vacation
- → Thursday, 2 December: **Final Lecture** ←
  - Course summary; Using Graphics
- Friday, 3 December: Written Report (Checkpt. 5)
- Fri-Thu, 3-9 December: Team Presentations
  - F 3 Dec: 1-3pm in 3-133
  - M 6 Dec: 9-12am; 7-9pm in 3-133
  - T 7 Dec: 9-11am; 7-9pm in 3-133
  - W 8 Dec: 7-9pm in 3-133
  - R 9 Dec: 9-11am; 2-5pm; 7-9pm in 3-133

Team Formation

Teams of three students expected
- TAs will help match members if necessary

Use your course web pages to:
- Post your interests
- Link to sources of inspiration
- Look for like-minded teammates

Resources

Resources available to you:
- Each other (see course team page)
- Tools (see course tool page)
- Space (several Gb on /mit/imagery4)
- Staff (talk to, or email, staff)

Final Project discussion
- Timeline:
  - Team Formation
  - Proposal submission (Checkpoint 1)
  - Feedback from Staff
  - Checkpoints 2-5: Meetings with staff
  - Project Writeup (Web document)
  - Project Presentation (Web slides)
  - Artifact Transfer (to TA)

Projects from Previous Years

Distributed Rendering and Viewing Pipeline

Converting Inventor Scenes into Random Dot Stereograms

Distributed Raytracing over a Network

Image Morphing with Java

Dancing Graphical Flower

An Implementation of Occluder Nodes in 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

Meteor (Animation)

Blowing Curtain (Animation)

Interactive Parallel Radiosity

Acoustic Simulation (Concert Halls)

Trick or Treat--Gothic Animation

Luxo Jr. Animation Sequel

Dome of the Rock

3D Modeler for Kids

Precomputing Ray Tracer

Monk Animation (Douglas Adams)

Foosball Players Come to Life (Animation)

MIT Model & Animation

Relativistic Ray-Tracing

Human in Free-Fall

The Job (Animation)

3D Drilling for Tumors in MRI Scans Using PHANToM

Dali Animation

Flag modeling \& Animation

3D Morphing

Fluid Simulation Haptics

Evolution of Movement

Tron Gone Wrong (Animation)

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

Project Scope

Teams of three 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
- Substantial implementation effort
- Communication of Work, Lessons Learned

Deliverables:
- Source, binaries, presentation, slides
- Documentation for future students
- Artifacts suitable for 26-100 "6.837 night"
  - (to be held during IAP/January 2000)
  - Images, models, brief 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 a good plot/concept, and:
- Storyboard, models, keyframes, teamwork, planning

Visualization (Science/Math)
- 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

Algorithm Profiling & 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 camera, sound, network, etc.

Modeling and Modeling Tools

Model existing spaces (outside or inside!)
- Photographic textures, photoCD, etc.

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

Generative modeling/texturing techniques
- 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
Algorithm Visualization
  Possibilities (speed knob; debug level)
  Visual debugging

Animation

Plot, storyboard, keyframes, ...
Direct specification
  Simple animation language
  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/`

SIGGRAPH proceedings (LCS Reading Room)
ProEngineer from Parametric Technology
- `add parametric; proeng`
- Online help system `proguide`

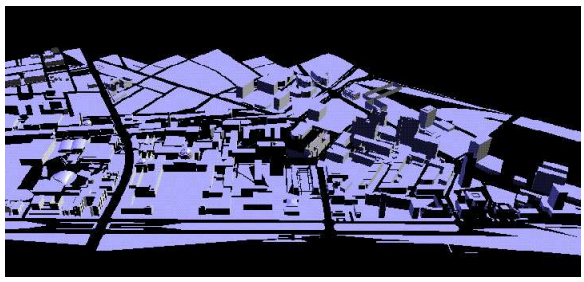
`ac3D`, Public domain 3D modeler
- Described on course Tools page

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

Campus exteriors with photomapped 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)

Campus interiors with textures, detail geometry
- arguably harder 3D modeling task

Or... model any other space of your choice

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

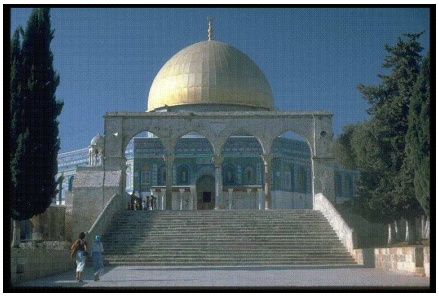
Analogous: animals, plants
Study phenomenology!

Parametric Modeling

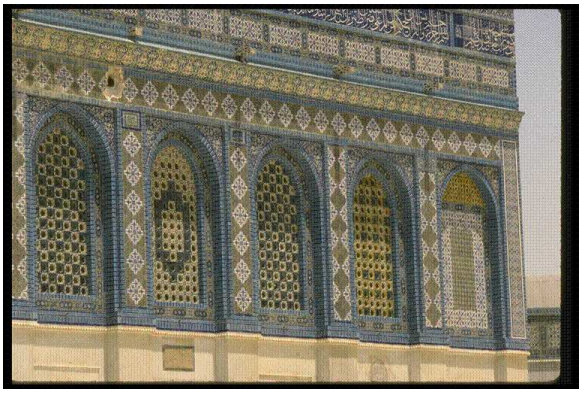
All uid_foo programs used `argparse()`.
Modify to generate web interface to programs
  Architect to allow continuous selection?
  Can interface be extracted from `arg_parse()`?
  Useful for this and future classes !

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**
  Resources you will use, 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
    Have clear weekly milestones
    State fallback positions
We will:
  Give detailed written & individual feedback
  Match each project to an appropriate TA
  Schedule weekly meetings, one per team per TA

Grading

Project is 40% of your total course grade
This 40% is broken into five parts:
- Initial proposal; checkpoint meetings; pacing (20%)
- Final written report (20%)
- Final team presentation & Demo (20%)
- Overall quality of project (30%)
- Project artifact (10%)