CS450/550 — Project Proposal — Due February 22, 1996

From February 22 to March 12, you will work on a project for this class. The project may take one of three forms:

- **Research Project**. You will choose a topic in computer graphics and read advanced textbooks or papers and write a short (8–10 page) paper describing what you learned.
- **Programming Project**. You will implement an enhancement to the graphics code that came with the textbook.
- **Modeling Project**. You will use either the textbook code, the IBM data explorer (DX) or some other graphics system to model and render a complex scene, possibly with animation.

For February 22, you should write a one-page description of your project that includes the following:

- **Project Description**. What kind of project is it? What will you do?
- **Expected Outcome**. What will you learn from doing this project?
- **Evaluation Criteria**. How should the resulting project be evaluated? Specify what level of outcome you think should receive each of the following grades: A, B, C.

On February 27, I will return your one-page description with any comments or modifications that I think are appropriate.

On March 12 and March 14, our classs time will be devoted to presentations of some of the projects. We will not have time for all of them, so on March 12, I will choose a random subset of the projects for presentation (in other words, *everyone* must prepare a presentation).

On March 14, you must turn in the following:

- **Research Projects** should turn in the written paper. It must include a bibliography listing what papers or textbooks you have read.
- **Programming Projects** should turn in a tar file of the revised code and some example programs that exercise this code. The example programs do not need to be complicated. You should also turn in a written summary of what changes or extensions you made to the graphics code from the book.
- **Modelling Projects** should turn in a tar file containing the source code for the project and image files of the output of the project (in tiff, gif, or mpeg format).

Here is a list of example research projects. You may choose one of these or propose your own project.

- **Advanced lighting methods**. Read about radiosity methods that directly handle conservation of energy.
- **Psychophysics of color**. Read more about how people perceive color. For example, you could read about the color matching experiments or about Edwin Land's retinex theory of color and other theories of color constancy.

- **Physics of light sources**. Why are there different colors of "white"? Why do they depend on the temperature and material of the light source? Why does this matter?
- **Color conversion.** Why does the CIE to RGB conversion system work? Is it exact or approximate? How do artists and printers convert between RGB and CYMK printing processes? How is the color gamut of a CYMK printing process computed?
- **Smooth shape representations.** Read the chapters in the textbook on splines and other smooth surface representations. Write a paper (or implement a program) that uses two or more such representations to represent the same object, and compare the results.

Here is a list of example programming projects.

- **Re-implement the code to use template classes**. Use the standard template library (or some other standard data structure library) to encapsulate the linked lists in the code.
- **Redesign and implement `xglib` to isolate the device-dependent functions.** For example, define a new class called `device_t` that has member functions for each of the device-oriented operations (setting pixels, colors, mouse operations, etc). Implement subclasses of `device_t` for 8-bit X color, 24-bit X color, TIGA, etc. Each of these subclasses could be defined in a separate file, and the makefile for a particular device would just compile and load the particular file it needed. This would avoid all of the `ifdef`'s that are currently in the code and make it much more portable.
- **Port the code to an SGI workstation with hardware graphics operations**. Some SGI workstations have hardware Z-buffers and hardware support for rotations and scaling. There are two SGI's in Dearborn Hall 311, and we could determine whether these have Z-buffering and modify the code to use it.
- **Implement transparency.** Modify the textbook code to permit transparent surfaces in the Z-buffer code. Demonstrate the transparency support with some interesting images.

Here is a list of example modeling projects.

- **Bouncing and colliding balls**. Implement an animation in which 2 or more balls bounce around the inside of a box and exhibit realistic dynamics (e.g., gravity, elastic collisions with each other and with the walls).
- **Ray tracing**. Construct an interesting scene with multiple light sources and specular objects and ray trace it to produce a nice picture.
- **Last Year's Projects**. Take a look at the Web page `http://www.cs.orst.edu/ tgd/classes/450/projects/mpegs.` for some MPEG movies from last year's class.
- **DX projects**. Take a look at the projects that have been done is CS418 at Cornell on the Web page `http://www.tc.cornell.edu/Visualization/contrib/cs418-sp94/cs418.html`. These are projects by students who had been using DX for a full semester, but they can give you some idea of what can be done in DX.