

One of the key points in creating an effective computer science course is developing the projects students will do in the course. This paper describes a set of projects for a computer graphics programming course, with a particular focus on projects whose content is in the area of visualization in the sciences.¹ This focus serves a well-defined group of students whose needs have long been a concern to the SIGGRAPH community² and can also make the course a key factor in a computational science program. The projects are intended to allow the course to be both an effective service course for science students and a sound introduction to computer graphics for computer science students.

THE COURSE

As described elsewhere,^{2,3} the author believes that developments in graphics hardware and APIs allow creation of a new paradigm for the introductory computer graphics course. This course can now focus on graphical thinking, problem-solving, and communication as expressed through a high-level API, and it can provide an excellent background to students who would want to use computer graphics in their professional work. This new course is a half-year course that serves as both a service course for many other programs on a university campus and as an entry point for students who want to study computer graphics more deeply. Its only requirements are programming skills (as expressed by successful completion of the equivalent of the CS1 and CS2 sequence) and an ability to think geometrically and spatially in the student's field.

One of the key issues in creating any successful computing course, and one that often is overlooked in the literature on teaching computer graphics, is the projects students will do in the course. This is especially true of a course that offers significant content to non-computer-science majors, because the projects are the students' primary tool to encounter the subject matter.

PROJECT CONCEPTS FOR THE COURSE

The author suggests that a half-year course would include five programming projects. These projects are intended to develop from simple geometric thinking to more significant applications of the tools in the API. Note that these projects do not identify the API being used; it should be possible to do them with any of the current APIs that might be used in the course. To begin, let us outline the projects in very general terms, focusing on the topics to be considered without specific applications (these will follow in the subsequent section on science applications).

Project 1: Identify some appropriate geometry to be displayed using the API that supports the course. This requires proper initialization of the API, definition of the viewing environment, analysis of the geometry into appropriate primitives and use of simple instancing transformations, use of geometric primitives to display the geometric primitives, use of hidden-surface display, and use of color as needed for the geometry. A technique such as the OpenGL idle callback can be used to create some motion, even in this first project.

Project 2: Use the same or different geometry, but increase the sophistication of the rendering by adding shading and lights to the scene, illustrating ambient, diffuse, and specular lighting. Use multiple lights, colored lights, or spotlights to illustrate the effects they can create.

Project 3: Use the same or different geometry, but focus on simple interaction with the scene, adding keyboard, menu, and mouse callbacks and controls for the scene. If tools are available (such as the MUI system for OpenGL) add button or slider controls as appropriate. The approach to this project will depend on the amount of UI focus the instructor wants to have in the course, but this could be significant.

Project 4: Use the same or different geometry and add more features of modeling and/or rendering, including hierarchical modeling, clipping planes, alpha blending, fog, or texture mapping. This is a good place to discuss genuine 3D viewing with stereo pairs or Chromadepth color encoding,⁶ and this is very interesting to students. While it is tempting to use techniques for their own sake, this might be a good project to leave fairly open-ended to allow the student to choose (and justify) the techniques to be used to create a particular communication.

Project 5: A more advanced, "capstone" project would ask the student to use additional techniques noted in Project 4 that were not used there, as well as new techniques, such as evaluators to generate spline surfaces or object selection to allow students to expand their interactions with images. Again, this project could be built on the same or different geometry as earlier projects, as appropriate.

SOME SPECIFIC PROJECTS

The programming approach to the computer graphics course with science-focused projects offers an excellent opportunity to create a course that has unique value to students outside computer science. In particular, focusing on science topics can make this course effective for science students or, especially, computational science programs. This section describes some such projects and gives examples of the kinds of images each could create. Restricted space limits the number of projects that can be described here, so we describe only projects for mathematics, chemistry, and physics. All of these projects, as well as others, are described in more depth in the online materials noted at the end of this paper.

Project 1

- **Mathematics.** Display a surface defined by a function of two variables. This requires creation of a grid in the domain and corresponding points in 3-space in the surface, displaying the triangles derived from the domain grid, and coloring the surface in some way such as height.
- **Chemistry.** Read a standard molecule description file and create a display of the molecule with standard atom-coded colors. This requires the use of simple transformations to place the atoms, using pre-written functions to read molecular description files.
- **Physics.** Display a rectangular bar divided into a grid, with some grid areas held at constant hot or cold temperatures and with temperatures changing by a diffusion process. Because temperature can be encoded through either geometry or color, this project offers the opportunity to discuss ways to encode information in a graphics environment.

Project 2

- **Mathematics.** Add shading and lights to the surface viewing. This project could also focus on higher-dimensional or parametric surfaces, generalizing the first project.
- **Chemistry.** Add shading and lights to the molecular image, showing the atoms with appropriate highlights.
- **Physics.** Display a function defined as the sum of two linear or circular wave functions, with ambient, diffuse, and specular lighting. Use the simple animation of the first project to show the behavior of the wave functions over time.

Project 3

- **Mathematics.** Add keyboard control for rotations, clipping, or other mathematical operations. This would include discussions of the kinds of interaction or control appropriate for mathematical operations. Add other kinds of control for parameters if the surface is defined as a parametric function.
- **Chemistry.** Add keyboard-controlled rotations and menu selections to the molecular display. This would allow discussions of the kind of control appropriate for viewing objects in space. Because of the time it might take to display fairly complex molecules, the third project should also add display lists to improve performance on the display.
- **Physics or Chemistry.** Create a simulation of molecules moving randomly in a cube, and periodically report the number of times a molecule strikes the side of the cube in a step of the simulation. This value simulates pressure, and allows the student to test the gas laws: the pressure varies linearly with the number of molecules present and inversely with the volume of the cube.

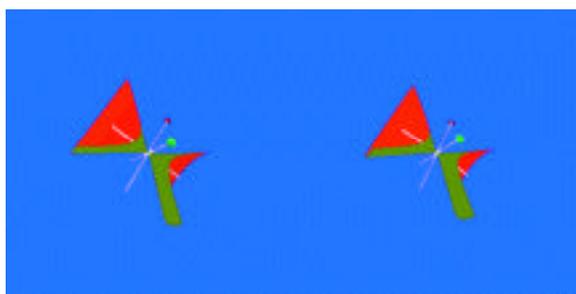
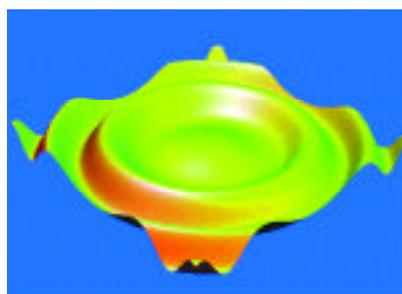


Figure 1: (a) A mathematical surface with light and materials modeling (Project 2).



(b) A stereo pair of conic sections that the reader can resolve without external aid. (Project 4).

Have the student track the motion of one molecule to see the nature of Brownian motion, allow rotations of the space so the user can see the entire space, and add controls that change the number of molecules present or the volume of the space to allow the user to test the simulation.

Project 4

- **Mathematics.** Create a display showing the quadratic curves as actual conic sections, using menus and other controls for managing the clipping planes that create the sections and creating coordinated stereo pairs to see the conic sections in 3D.
- **Chemistry.** Add transparency and alternative views of the molecule. The alternative views could add a clipping plane with keyboard control of a front-and-back motion on the plane, Chromadepth texture mapping to create real 3D viewing, stereo pairs, or texture-map information onto individual atoms, though this may be very slow on some machines.
- **Physics.** Create a display of a function of three variables with a physical basis, such as temperature, static force, or gravity in some spatial region. This could be an implicit surface or a color-coded display of function values in a clipping plane in the region. Allow the user to interact with the display to move the implicit surface or clipping plane, and have the project include some conclusions about the nature of the function.

Project 5

- **Mathematics.** Display a spline surface with user-selectable, movable control points. This requires students to manage both evaluators and object selection and to deal with moving objects in 3D.
- **Chemistry.** Add selection of an individual atom so that the atom can be manipulated or information on it can be returned. Alternately, the project could display vibration modes for a selected molecule component using vibration information provided separately.
- **Physics.** Create a simulation of a multiple-body problem with simple numerical integration and build a time-based animation of the situation. Use simple controls to allow the user to move around in the animation space, and use Chromadepth or stereo pairs to allow the user to experience the animation in real 3D space. Add selection to allow the user to identify an individual body and to use interaction techniques to provide real-time steering of the simulation by changing the velocity or direction of the moving body.

These figures illustrate some of the projects described above. The figures start with mathematics projects, then move to chemistry, and then to physics, and the captions identify which project is being illustrated. Note that some of the projects, especially Projects 3 and 5, do not lend themselves well to figures because their focus is an action, not an image.

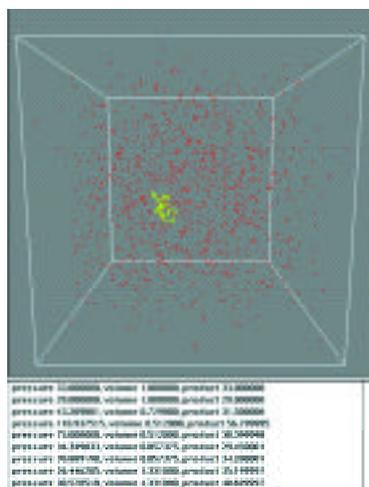
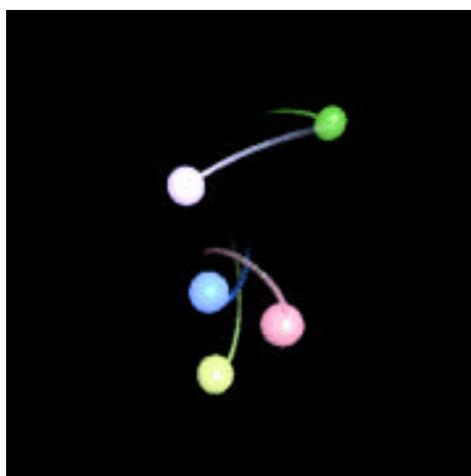


Figure 2: (a) Simulation of the gas laws with sample text output from a query (Project 1).



(b) An image of a molecule with Chromadepth coloring added (Helvetane) (Project 4).

CONCLUSIONS

This approach to the beginning graphics course is oriented toward the science student and fits the growing use of visualization in the sciences, but it should be possible to focus the projects towards many other fields to support visualization in these fields. Versions of this course have been taught twice (at two different universities) with significant student interest and success, although it has so far been difficult to reach significant numbers of science students. Some stability in the course content and some marketing of the course to the sciences are needed to increase interest. Conversations about and presentations of this course to a number of people in computer science, computational science, and the sciences lead the author to believe that this approach is worth sharing with the graphics community in spite of the relative lack of experience, because other fields are eager to gain the advantages of learning about graphics without having to master fundamental graphics principles and without having to rely on pre-written graphics applications.

The project materials described here are being developed with funding from the National Science Foundation. These materials include descriptions of projects for each of the sciences, source code for the examples, supplements on various graphics topics, and pointers to example materials from various sources. As additional materials are developed, they will be available at www.cs.csustan.edu/~rsc/NSF/

and other pages linked from there, and the final results will eventually be available at the online SIGGRAPH and Eurographics sites and at appropriate courseware repositories.

References

1. Brown, J., Cunningham, S., & McGrath, M. (1990). Visualization in science and engineering education. In G. M. Nielson & B. Shriver (Eds.), *IEEE Tutorial: Scientific Visualization*. IEEE Computer Society.
2. Cunningham, S. (2000, February) Re-inventing the introductory computer graphics course: Providing tools for a wider audience. *Computers & Graphics*.
3. Cunningham, S. (2000). Powers of 10: The case for changing the first course in computer graphics. *Proceedings of the SIGCSE 2000 Technical Symposium on Computer Science Education*.
4. Reports of the Graphics & Visualization 99 Workshop are to appear in *Computer Graphics and Computer Graphics Forum* and are available online at www.eg.org/WorkingGroups/GVE/GVE99 and www.education.siggraph.org/conferences/GVE99/.
5. Domik, G. *Report on visualization curriculum*. www.uni-paderborn.de/fachbereich/AG/agdomik/visualisierung/vis-report/
6. Chromadepth glasses and other information are available from Chromatek Inc: www.chromatek.com/

Credits

This work is partially supported by National Science Foundation grant DUE-9950121. All opinions, findings, conclusions, and recommendations in this work are those of the author and do not necessarily reflect the views of the National Science Foundation. The author would also like to thank the San Diego Supercomputer Center for supporting this work.

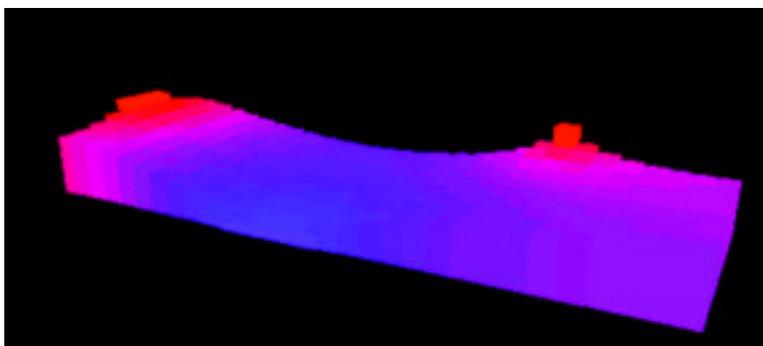
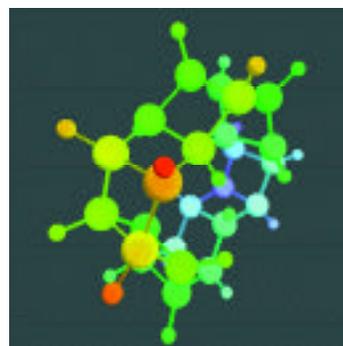


Figure 3: (a) A heat simulation in a bar (Project 1).



(b) An N-body simulation (Project 5).