A Project-Oriented Graphics Course

Robert G. Strader
Stephen F. Austin State University
rstrader@sfasu.edu

Introduction

Many students entering college who choose to major in computer science do so because of their exposure to graphical environments (including games). Although it is not always the primary focus of a graphics course, development of a simple game or similar project prepares students for developing and producing graphically oriented products. While the focus of the graphics course presented in this paper is one of traditional graphics topics, user interface principles, and the extension of background topics (i.e., data structures, algorithms, architecture, etc.) to graphical environments, a major term project is required. The primary emphasis of the project is on developing a small product rather than just completing a class exercise.

The next section discusses the course outline and an overview of the project. The following section presents information on activities required to support such a project. Next, the project framework is presented along with observations. Finally, concerns for future enhancements are addressed.

Course Outline

The computer graphics course presented here is a senior-level elective with traditional background requirements including data structures, algorithm analysis, and mathematics (trigonometry and matrix algebra are recommended). The course format has been fairly traditional, with topics including an introduction, hardware, software, rendering, modeling, theory, and advanced topics as time permits. The current course format is the result of an evolution from a traditional 2D-based course with some 3D at the end to a 3D-centric course.

The current 3D approach to the course grew out of experiences of the SIGGRAPH 97 conference,[1] where Edward Angel[2] presented a course model that indicated students could achieve 3D rather quickly. Since 3D had been difficult to achieve in previous course offerings, his outline was adopted, and in particular his suggested series of exercises[3] utilizing the "maze" theme (three assignments) was implemented in the 1997 course with success. Students were able to accomplish 3D projects by midterm.

As in most successful situations, it was noted that the difficulties "moved." Students now struggled with lighting, texturing, and more advanced topics. An enhancement for the 1999 course included an extension of the "maze" series to include five assignments in addition to the project. The extensions were a lighting/shading lab and a texture lab. Additional activities (homework) were assigned to introduce OpenGL, practice modeling techniques, and provide theory-based problems. A major goal of the course was to complete a project. The timeframe for the project was approximately one-half of the semester, although project requirements were given on the first day of the course. A project proposal was due at midterm, an intermediate report was required at approximately three-quarters of the way through the semester, and the final report was due before the end of the term.

Project Support

The laboratory assignments were designed to develop facility in the use of OpenGL, explore reasoning and representation using geometric objects, and to provide a small "engine" that might be useful for the project. Each assignment depended on completion of the previous assignment through the first three assignments. Each student assignment was captured on videotape and given to the student for review. Some of these were played in class and evaluated by all of the students.

The first assignment was to draw a 2D maze on the screen and recursively remove the walls to produce a connected maze.[3] A hedge was required to encompass the maze, and although the entrance side was chosen at random, the exit was forced to the opposite side. This exercise provided some OpenGL programming experience. including simple primitive and attribute manipulation. It also introduced students to an algorithmic technique that would be useful later. In addition, careful reasoning about special situations (boundaries and adjacencies) provided the opportunity to develop and extend reasoning skills. Much of the difficulty associated with completing the lab involved these reasoning problems and helped reinforce the idea that core concepts developed in previous courses were essential even in "graphics programming." Most students completed this assignment successfully, although some did not complete the recursive removal portion and had to "back-patch" a temporary solution.

The second assignment was to add a "rat" to the maze[3] and then manually traverse the maze. This involved positioning and movement of an object through the maze. Graphically, this exercise provided students with the opportunity to erase and redraw (back buffer) objects and to develop a simple collision-checking algorithm (again, somewhat more traditional programming). In addition, working with interactive events utilizing the keyboard and mouse in a window provided an introduction to interface issues (for example, no middle mouse button, etc.). Many of the students were successful in completing this exercise, although the heaviest class attrition occurred at this stage.

The third assignment was to develop a 3D "rat" view of the maze.[3] This changed the 2D lines to 3D surfaces and introduced a perspective view. Motion was achieved by changing views (no smooth motion). In addition, the original 2D window was retained as a map of the maze, requiring the students to control multiple windows. Again, most of the students were successful, although there were some interesting problems (for example, left and right mix-up, slow operation due to swapping buffers too often, etc.). A sample display is shown in Figure 1.

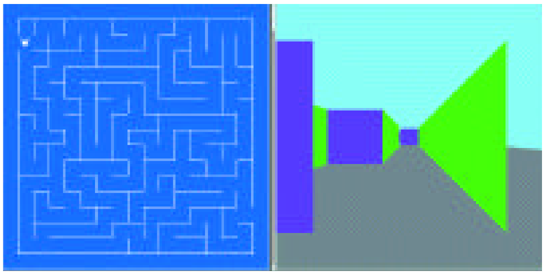

Figure 1. 2D map/3D view.

The fourth assignment was to add lighting and at least one shadow to the 3D portion of the maze. Most students tried to accomplish this through material properties, which generated "soft" shadows. Some students created an object (for example, cheese at the end of the maze) to produce a "hard" shadow, and the remainder created "hard" shadows using the walls of the maze. An interesting visual effect was produced when some students produced a darker color at the top of the wall than at the bottom.

The final assignment was to add texture to the 3D maze. Several students used the sample checkerboard pattern[4] presented in class for their texture. One student developed a utility to capture textures from files and place them in the program as a data object. A sample "final" maze view is shown in figure 2.

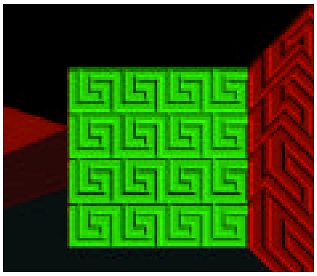

Figure 2. Shadow and texture.

The Project

The project encompassed the last half of the semester, although it was assigned on the first day so that students were aware of the project and could begin thinking about what they might do. Also, it was suggested that students work in teams to give them the opportunity to discuss common interests. It would not be fair, however, to consider that the project covered the entire term since until laboratory assignment three was completed, many students did not believe that they could work in 3D.

The project assignment was divided into three parts: a proposal, an intermediate report with a working prototype, and a final report with the working project. In support of the project, three information resources were utilized. The first was an invited speaker from the art department, who presented some design fundamentals and illustrated them with examples. Most of the students found this information helpful in their project design. The second was a presentation from a marketing department professor on product design, development, and related issues. Although it was more difficult to incorporate, students found this presentation interesting and, as a class, requested a copy of all of the projects (which was provided with labels, executables, source code, and documentation on a CD-ROM at the end of the term). The third information resource was a set of handouts from written and online references concerning story telling and writing. The success of this material was mixed, since it was given late in the course timetable and without a class guest presentation. Each of these topics was introduced to explore and support a non-programming aspect of the project.

The proposal covered the project outline, a general design, an implementation timetable and sample screens or views. In many ways, this was the most challenging part of the project. Several students encountered difficulty in choosing a topic, and in general, all had trouble with the scope of the project (it was either too limited because they were not confident about working in 3D, or it was too ambitious because of the commercial products they were accustomed to using). Only a few students had any working code at proposal time. Sample projects included games, walkthroughs, and stories. All of these projects utilized techniques and code segments developed in the lab exercises.

The intermediate report included an overview, a preliminary user guide, an issues section discussing design or implementation problems that might limit the scope of the project, an updated timetable, user testing, and a working prototype. This report was critical to the final success of the project. In particular, if the project was off track or too incomplete, feedback from this report provided the student with some guidance for correction and limited refinement of the project goals.

The final report was a small package including documentation and media. The report included an introduction, a complete user guide, a project summary including time spent, algorithms utilized, limitations and unaccomplished objectives, references, and appendices with listings, artwork, and support files. The media contained online documentation, source and executable code, and a soft copy of the report.

Observations

The course was successful in introducing students to "graphics programming" and a project-product approach. Based on faculty analysis and student feedback, future course offerings will need to increase the emphasis on modeling, provide additional support for non-technical issues (while retaining both the theoretical and implementation content currently in the course), incorporate an object approach, and encompass additional advanced techniques.

References

1. SIGGRAPH Conference Grants for Educators Program, 1997.
2. Angel, E. (1997). Teaching a three-dimensional computer graphics class using OpenGL. *Computer Graphics, 31* (3), 54-55.
3. Angel, E. (2000). *Interactive computer graphics* (2nd ed.). New York: Addison Wesley.
4. Woo, M., Neider, J., Davis, T., Shreiner, D. (1999). *OpenGL programming guide* (3rd ed.). New York: Addison Wesley.