

An Evolving Approach to Computer Graphics Courses in Computer Science

Steve Cunningham
California State University, Stanislaus

Abstract

Recent developments in computer graphics APIs permit us to take a new approach to teaching a first graphics course that parallels a first computer programming course. This first graphics course, and a second course that could follow it, are briefly described and are placed in the context of current graphics courses.

New Opportunities in Graphics Courses

There are two reasons students study computer graphics: to be able to do computer graphics programming, either on the job or for personal work, or to understand computer graphics deeply in order to build a graphics-focused career. Few computer science programs can develop separate courses for these two kinds of students because the faculty and equipment resources needed are too scarce. Thus we must consider courses that serve both purposes.

The field of computer graphics has evolved to a point that allows us to respond to these dual needs. This response is a full year program in computer graphics that involves two half-year courses. In the first course, lasting perhaps 15 weeks in the U.S. system, the focus would be on computer graphics programming. At this point in our field, there's no use in considering 2D graphics; all significant graphics is 3D. This course would use a high-level platform-independent 3D API such as OpenGL or Java3D and would focus on the principles of developing a 3D model, displaying it, and interacting with it. The particular concepts that should be covered here are fundamental 3D geometry; geometric transformations; lighting and viewing; expressing geometry, materials, and views through the API; material properties; and details of managing programming with local systems and the API. This course would be similar to that described by Angel [ANG 97] and covered by Angel's book [ANG 97a], but would not try to include both high-level graphics programming and fundamental algorithms because it seems more productive to keep the focus on the capabilities provided by the API. The course would be quite distinct from a number of other introductory courses that have been described in the literature, such as [SCH 90] or [XIA 94] because it has the single focus on graphics programming.

The second course would focus on computer graphics concepts. This course covers the fundamental concepts and algorithms by which a graphics system creates images and is intended for the student who wishes to understand computer graphics in depth, to prepare for graduate study, or to pursue individual work

in the field. Topics in this course include the process of presenting 3D images on a 2D surface (projections and viewing; Z-buffers), developing and presenting realistic images (texture maps, environment maps, anti-aliasing, alpha channel, lighting models), and managing the viewing device (scan conversion, clipping). The second course is close to the concepts and algorithms course described in [BRO 88] and [OHL 86], frequently taught now as the first computer graphics course, but is not the kind of course in specialized techniques or systems that is sometimes considered as an advanced course [OWE 92].

An analogy might be useful. This analogy is the general programming environment, in which students learn programming through a high-level language (and the trend has been to move the level of the language upwards as the field has matured) and later learn the details of how program execution works. In a more diagrammatic form:

Computer Programming builds on: is: Problem Solving + Language Computer Organization

Graphics Programming builds on: is: Geometric Thinking + Language Fundamental Graphics Techniques

Thus the first course in computer graphics might be seen as a "high-level language" course in graphics, while the second might be seen as an "assembly language" or "system organization" analog in computer graphics.

Additional courses should complement these for any program with a concentration in computer graphics. A course in interaction covers event-driven programming and how one can interact with 3D environments, and might end with a consideration of interactive environments such as VRML. A course on animation would describe how one can describe and model motion and create image sequences that give good illusions of motion [WAT 92].

The First Course

The first course focuses on graphics programming with a graphics API as noted above. Several people in the United States now teach such a course, though it is not yet considered standard. An example of this course, based on OpenGL, is the undergraduate course taught by Lew Hitchner at Cal Poly State University, San Luis Obispo, California. This course is based on the Angel book [ANG 97a] and a book on SGI Inventor [WER 94]; it is described by online materials found at

<http://www.csc.calpoly.edu/~hitchner/CSC455.S98>

This course covers an introduction to graphics systems, graphics programming, input and interaction,

geometric objects and transformations, viewing, modeling (using Open Inventor), shading, the implementation pipeline, and discrete techniques. It is an advanced course and includes a mixture of graphics programming and fundamental techniques.

This first computer graphics course could be given earlier than the Hitchner course or most other computer graphics courses. The course could be motivated by computer graphics applications in areas such as science, engineering, education, mathematics, business, statistics, architecture, or entertainment, depending on the students and the university. Lectures could discuss graphics systems as expressed through the API, geometry, simple modeling, transformations, and simple lighting. Student projects would create graphics for the application area of the course's focus. Students who complete this first course would be prepared to do graphics programming as part of their overall programming work, but would not necessarily have a deep understanding of all the techniques and issues involved in the field. Thus those students who wanted to go on in graphics would need to take the second course to lay the groundwork for the area.

The Second Course

As noted above, the second computer graphics course serves students who want a deeper understanding of graphics principles and techniques. It focuses on details of creating and manipulating graphics images, and would probably use a very basic graphics system built directly on raw pixel access. It covers the material typically seen in the concepts course as represented by [FOL 96] and as described in a number of sources such as [BRO 88], including scan converting and filling polygons, clipping and visible line and surface processing, spatial subdivision, implementing viewing environments, curve and surface construction, and shading and texturing, but not the introduction to geometry and other beginning concepts usually found in those courses. Student projects are very similar to those of the current computer graphics principles course, but because early introductory projects are no longer needed, more comprehensive and interesting projects are possible. An interesting feature of the second course is that students would understand principles of geometry, modeling, lighting, and materials from the first course, so the second course can move much more quickly into techniques. Students completing this second course would have a solid grounding in the field and would be prepared for a graduate program in computer graphics or for work in a firm in the graphics industry.

Students could also go from this second course to study particular topics in depth or to begin research work. These advanced courses could affect the content of the second course, but the second course should avoid the specialization described in [OWE 94] and remain as general as possible so students can go on to a wide range of future studies.

The Challenge

The course we propose as the introductory course is not yet supported by a textbook or a body of experience. It will be challenging to design a course, develop a set of projects and examples, and to create textbooks. This will likely take a few years, but it seems to be the right time to take this direction for the field.

References

- [ANG 97] Angel, Edward, "Teaching a Three-Dimensional Computer Graphics Class Using OpenGL," *Computer Graphics* 31 (3), August 1997, 54-55
- [ANG 97a] Angel, Edward, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Addison-Wesley, 1997
- [BRO 88] Brown, Judith R., Robert P. Burton, Steve Cunningham, Mark Ohlson, "Varieties of Computer Graphics Courses in Computer Science," Proceedings of SIGCSE 88, *SIGCSE Bulletin* 20(1), 1988, 313
- [FOL 96] Foley, van Dam, Feiner, and Hughes, *Computer Graphics Principles and Practice*, 2nd ed in C, Addison-Wesley, 1996
- [OHL 86] Ohlson, Mark R., "The Role and Position of Graphics in Computer Science Education," Proceedings of SIGCSE 86, *SIGCSE Bulletin* 18(1), 1986, 232-237
- [OWE 92] Owen, G. Scott, "Teaching Computer Graphics Using RenderMan," Proceedings of SIGCSE 92, *SIGCSE Bulletin* 24(1), 1992, 304-308
- [OWE 94] Owen, G. Scott, Maria M. Larrondo-Petrie, and Cary Laxer, "Computer Graphics Curriculum: Time for a Change?" *Computer Graphics* 28(3), August 1994, 183-185
- [SCH 90] Schweitzer, Dino, "Ray Tracing: a Means to Motivate Students in an Introductory Graphics Course," Proceedings of SIGCSE 90, *SIGCSE Bulletin* 22(1), 1990, 157-161
- [WAT 92] Watt, Alan and Mark Watt, *Advanced Animation and Rendering Techniques, Theory and Practice*, Addison-Wesley, 1992
- [WER 94] Wernecke, Josie, *The Inventor Mentor*, Addison-Wesley, 1994
- [XIA 94] Xiang, Zhigang, "A Nontraditional Computer Graphics Course for Computer Science Students," *Computer Graphics* 28(3) August 1994, 186-188

Department of Computer Science
California State University Stanislaus
Turlock, CA 95382 USA
rsc@csustan.edu