Approaches to Teaching Introductory Computer Graphics

Chair
Maria M. Larrondo-Petrie, Florida Atlantic University

Panelists
Jack Bresenham, Winthrop University
Cary Laxer, Rose-Hulman Institute of Technology
John Lansdown, Centre for Electronics Arts and Middlesex University, UK
G. Scott Owen, Georgia State University

The panelists will discuss various approaches to teaching an introductory computer graphics course. They represent a wide spectrum of the discipline, offering courses emphasizing systems, engineering, mathematics, science, art design and animation. Sample course syllabi, textbook recommendations, software packages and suggested projects will be available.

This past summer at an Undergraduate Faculty Enhancement Workshop in Computer Graphics sponsored by the NSF and the ACM SIGGRAPH Education Committee, it became apparent that there were widely different viewpoints on the content and methodology for teaching the introductory computer graphics course.

The panelists will discuss various approaches to teaching an introductory computer graphics course. They represent a wide spectrum of the discipline, offering courses emphasizing in systems, engineering, mathematics, science, art design and animation. Included in the discussion will be conventional approaches, beginning with two-dimensional concepts and introducing three-dimensional concepts at the end of the course; an approach that focus on generic graphics pipeline elements and system specification/implementation; a new approach that emphasizes image synthesis based on Physics; and approaches that emphasize art design and animation. Software packages, both commercial and public domain, used to teach computer graphics will be described. The courses described are taught at large public universities as well as small private colleges. The equipment used varies from 8088 personal computers to Sun workstations. Sample course syllabi, textbook recommendations, and suggested projects will be available.

Jack Bresenham: *A "systems" approach*

Dr. Bresenham emphasizes a fundamental, systems programmer orientation in teaching introductory computer graphics basics at Winthrop University [Bresenham 92]. His students are expected to understand generic alternative reference models, attribute interactions, transformation concepts, and pixel space post-rastering clipping. His experience as a member of the X3H3.3 graphics standards committee for CGI specification, and more recently as a professor of computer science motivates him to stress understanding of basic principles in an introductory course [Schaller 92].

In convering specific graphics vocabulary, concepts, mathematics, and algorithms, he attempts to emphasize the WHY as much as the HOW. Analysis necessary to assure a consistent, complete architecture for a system specification or implementation is a primary objective [Ajuha, Bresenham 90, Earnshaw 85]. Students should avoid misinterpretations such as incorrect reference point transformations specified in the initial CGI proposed standard (ISO/DP9636) [Bono 88, Bresenham 90]; the final standard issued in 1991 did finally get anisotropic transformations done correctly. Working from basic principles, students are expected to be able to explain apparent anomalies such as why three thick line segments with one end point common will likely be drawn by a PostScript implementation with a quite different visual appearance dependent upon order of drawing sequence.

Students write a small 2-D graphics processing program that interprets geometric, attribute, and control commands then renders pixel by pixel the picture specified. Winthrop University is a 5000 student, state supported, teaching institution. The equipment for past graphics courses has been Leading Edge PC clones with 8088's. For Fall semester 1994, Winthrop should have 80486's in computer science student labs and most faculty offices. With the improved facilities, more compute-intensive graphics applications will become feasible.

John Lansdown: *Teaching and learning computer graphics in art and design*

For some years the Centre for Electronic Arts has been involved in teaching artists and designers the use of computers and computer graphics. Most of this work has been at Master's level. However, it is Faculty policy that all undergraduates be computer literate in appropriate application packages and the Centre contributes to teaching these too. As is traditional in skills-based art and design education in the UK, teaching and learning strategies in the Centre rely much more on over-the-shoulder instruction, projects and private study rather than formal lectures. Group working is encouraged (indeed, in most of the Centre's courses, compulsory).

Although we do not formally base our teaching and learning strategies on the van Hiele model of learning (Burger & Shaughnessy 1986), our experience suggests that the ideas underlying that model might have validity in this area. It is, for example, apparent that the levels of learning are sequential and ordered yet continuously connected and that each level has its own terminology and language of discourse. Thus, to ensure student understanding of the concepts of computer use, especially computer graphics programming, it is necessary to provide an environment to enable students incrementally to build up levels of learning. It would be interesting to discuss the forms that this environment might take.

It is clear that current commercial packages (at least of the sort we are able to afford in sufficient numbers for 90 Master's students and 2300 undergraduates) do not always meet the needs of those doing art and design. Our students seem always to want to do things that such packages can't do, especially in means of interacting with images. This is why our longest established Master's courses require students to learn programming. It is a credit to students' motivation and effort that, by this process, they can usually realise their sometimes difficult objectives. We need to examine the means and methods of making such realisation easier.

Cary Laxer: *An "engineering" approach*

Rose-Hulman Institute of Technology is a small, predominantly undergraduate, engineering and science college. It attracts some of the brightest students in the country for their college education. Rose-Hulman trains its students to be problem solvers.

The introductory computer graphics course is taught primarily to junior and senior computer science majors and minors, although students studying computer engineering, electrical engineering,

SIGGRAPH '94, July 24-29, Orlando, Florida
© ACM 1994 ISBN: 0-89791-667-0 ...$5.00

mathematics, and physics are frequently enrolled. The prerequisites for the course are courses in algorithm and program design and linear algebra. Hill's *Computer Graphics* is the text for the course. A lab of twenty Sun-3 workstations supports the programming projects. Students program in C, using a locally-developed interface to Xwindows.

The course approach is to have students develop their own library of graphics routines. Students are taught algorithms for drawing lines and curves, clipping, and area filling. They must implement these algorithms and then use them to render an image. Since Rose-Hulman is on the quarter system with ten-week terms, the course focuses heavily on two-dimensional concepts (the students do a two-dimensional ray tracing project) with a brief exposure to three-dimensional concepts during the last week of the term. This background prepares the students well for the Advanced Computer Graphics course, which covers the breadth of three-dimensional computer graphics.

G. Scott Owen: *A "scientific" approach*

The hardware available for the class has progressed from the Apple II to 80486 based systems with reasonable color raster graphics capabilities. Yet, most instructors still teach Computer Graphics as if they were using Apple IIs. The image synthesis component of Computer Graphics has traditionally been taught as a collection of ad-hoc techniques. Students are taught how to turn on a pixel, draw lines, polygons, two and three dimensional modeling transformations, viewing transformations and finally, perhaps a simple empirical shading model. Most of the course is spent on 2D topics, with a little 3D at the end. There is not underlying theory behind the methods so there is no conceptual framework for the students.

In contrast to this approach, other scientific disciplines, such as chemistry or physics, teach students a basic underlying theory that provides a conceptual framework for the different techniques and approximations. Teaching image synthesis from a physical science viewpoint provides this underlying theory and conceptual framework.

In his current course, Dr. Owen starts from the basic physic of the interaction of light with surfaces and derives simple local (the Phong model) and global illumination (ray tracing) models. The programming assignments are to produce reflection, and texture mapping. After covering this he spends the rest of the time discussing scan line graphics, comparing it with ray tracing. The students do assignments using the Pixar Renderman system. Thus, most of his course is on 3D graphics with only a brief amount of time spent on 2D graphics. As an added benefit, the students are able to produce some very nice images, comparable to what they see in the media.

References

Ahuja, D. V., "An Algorithm for Generating Spline-like Curves", *IBM Systems Journal*, Vol. 7, No. 3 & 4, 1968, pp. 206-217.

Ahuja, D. V., Coons, S. A., "Geometry for Construction and Display," *IBM Systems Journal*, Vol. 7, No. 3 & 4, 1968, pp. 188-205.

Bono, P. R. and Arnold, D. B., *CGM and CGI: Metafile and Interface Standards for Computer Graphics*. Springer-Verlag, New York, 1988.

Bresenham, Jack, "Computer Graphics & University Curricula," *Jahrestagung Conference Proceedings* (Zurich-Örlikon, Switzerland, July 17, 1992), Swiss Computer Graphics Association and the University of Zurich MultiMedia Laboratory.

Bresenham, Jack, "Attribute Considerations in Raster Graphics," *Computer Graphics Techniques: Theory and Practice.* David Rogers and Rae Earnshaw, eds. Springer-Verlag, New York, 1990, pp. 9-41.

Burger WF and Shaughnessy JM (1986) "Characterising the van Hiele levels of development in geometry," *Journal for Research in Mathematical Education* (17)1, pp 31-48

Earnshaw, Rae, ed. *Fundamental Algorithms for Computer Graphics.* Springer-Verlag, New York, 1985. (paperback printing, 1991).

Owen, G. Scott, "Teaching Image Synthesis as a Physical Science," *Computers & Graphics*, 18:3, March 1994.

Owen, G. Scott, "Teaching Computer Graphics Using Renderman," *Proc. of 23rd SIGCSE Technical Symposium on Computer Science Education,* ACM, March 1992.

Schaller, Nan, panel chair, "Graphics Education for Computer Science," SIGGRAPH '92 Conference Proceedings (Chicago, Illinois, July 26-31, 1992), ACM SIGGRAPH, New York, 1992, pp. 410-411.

Upstill, Steve, *The Renderman Companion*, Addison-Wesley, 1990.