Powers of 10:
The Case for Changing the First Course in Computer Graphics

Steve Cunningham
California State University Stanislaus
San Diego Supercomputer Center

ACM SIGCSE 2000
Traditional Wisdom

• Computer graphics is a difficult subject that requires a student to know a lot of mathematics and to master highly technical algorithms

• All the computer graphics non-specialists need to know is how to run the right tools and applications
BUT...

• Computer graphics is **not necessarily** a difficult subject

• Computer graphics **need not** require a student to master a lot of mathematics and technical algorithms

• Scientists and other users **should** know what lies behind graphics just as a scientist should know what lies behind calculus
Goal: create a computer graphics course that serves a broad student audience and is still a sound computer science course
Whom does the course serve?

- Shift the emphasis from developing graphics specialists to developing a broad group of students with graphics skills
- Students can come from science and mathematics (my personal emphasis) or from many other disciplines, depending on the focus of the institution and department
- Computational science is a natural!
Whom does the course serve?

- High-Level Graphics Users
- App & Tool Developers
- Sys Dev
What is the new course like?

• The focus is on graphics programming instead of graphics theory, algorithms, and techniques

• The course uses a standard programming API, such as OpenGL, for its work

• The course lectures discuss graphics concepts, while the course projects allow the students to work in their individual specialty areas
What are the prerequisites?

• Sound programming skills, and an ability to see the geometry in the world around them
  – Programming skills means roughly B or better in two programming courses
  – Seeing geometry requires simple spatial abilities that don’t come from coursework but may be picked up from the students’ work in their fields, especially science/math
Course projects

Projects progress from very simple to more sophisticated and can cover many fields

- Simple geometry and color
- Lighting/shading, transformations, callbacks
- Event-driven programming, callbacks, user control, interface issues
- Clipping, transparency, texture maps, splines, ...
- Object selection and interaction with image parts
Examples of possible student projects
Who wins with this approach?

• Computer science wins because we serve our colleagues and our universities better (and we get a broader student base),

• Other students win because they get a good background in the computer graphics they will use for their professional work

• Computer science students win because they get useful professional skills
What’s the follow-on course?

• A second graphics course presents the traditional graphics algorithms and techniques. The graphics programming course allows this second course to move very quickly and to go much further than the traditional introductory course.

• Other courses could build on the intro course to serve other groups of students
Contact information:

• Email address is  rsc@sdsc.edu or cunningham@siggraph.org

• Draft materials are online at
http://www.cs.csustan.edu/~rsc/NSF/

This work is 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.