Computer Graphics in Computing Curriculum 2001

Dr. Steve Cunningham

California State University Stanislaus

WSCG 2001

The overall context for this discussion is the undergraduate computer science curriculum

Any discussion of computer graphics in Computing Curriculum 2001 must be considered in this context

A brief review of Curriculum 91

- First of the standardized curricula that contained any computer graphics
- Recommendations had a limited number of knowledge units (KUs) in the general curriculum, but it was not clear how this would be provided
- A traditional course based on graphics algorithms and techniques was given as an optional elective

Computer graphics has changed since 1991

- Standards such as GKS and PHIGS have been superceded by industry APIs

- Developments in hardware have made it possible to do very fast, high-quality rendering on inexpensive desktops

- There is a great deal more experience with using graphics to accomplish important work

Computing Curriculum 2001

Ironman version of curriculum proposal is at

`http://www.acm.org/sigcse/cc2001/`

In Computing Curriculum 2001

- **Fewer hours for computer graphics in the core (5) than in Curriculum 91**
  - Smaller core generally
  - Other new content is competing for space
- **Computer graphics is part of a larger context that includes image processing and computer vision**
- **Core content is generally in broad, relatively early courses**

The focus in this presentation is
a stand-alone introductory
course in computer graphics

Whom do we serve?

- Future computing professionals
- Future graphics professionals
- Future professionals in other fields (service courses), especially computational science, whose importance is specifically recognized by CC2001

So for this introductory course we must ask:

- **What will contribute best to those who will integrate graphics into future applications?**

- **What will contribute best to those who will develop graphical systems?**

- **What will contribute best to those outside computer science who will use computing in their professions?**

These questions are in a context...

- There is little space in the undergraduate computer science program for multiple introductory courses

- There is little opportunity to introduce significant computer graphics in another part of the program

- We must create a course that will satisfy all three groups as well as possible

My conclusions are...

- **We should shift the focus of the first course from fundamental algorithms and techniques to API-based graphical problem-solving and communication**
- **We should include discussions of the issues that underlie modeling and rendering in computer graphics and build the motivation for modeling and rendering as done by the API**

My conclusions (cont)...

- We should build the course on a sturdy API and give students the opportunity to create some useful visual communication from their work
- We should include content in interaction and animation that is beyond the scope of a traditional course
- We should motivate the student to study further to get the details

The course we recommend for CC 2001

- Primary focus is on developing concepts and geometric thinking
- Work is expressed by a current API, but in principle is independent of that API
- Emphasis is on effective visual and graphical communication
- Include an application area to give students a set of content to express

This course...

- **Will be somewhat similar to courses being taught now at several universities**
- **Is sometimes controversial because it does not include the traditional focus on algorithms and techniques**
- **Is not yet fully supported by textbooks, although Hill 2nd edition could be used**

This course may be subsetted...

- **To be part of an overview course in visual computing that includes computer graphics and computer vision (synthesis and analysis)**
- **To be part of an introductory course in scientific visualization or computational science**

This course is a foundation for an advanced course

- That takes advantage of the graphics sophistication the students develop in the first course

- That includes the algorithms and techniques that are used in the API

- That develops additional advanced topics as appropriate for the research or focus of the instructor or institution

A potential example introductory course

- Developing a course now with a set of developing notes and materials at `http://www.cs.csustan.edu/~rsc/NSF/` that have been presented elsewhere; this work is supported by US NSF grant DUE-9950121

- Comments and suggestions are welcome at `rsc@cs.csustan.edu`