Teaching of Computer Graphics

Vincent Boyer
www.ai.univ-paris8.fr/~boyer

I am an assistant professor at the computer science department of the University of Paris 8. I started my research in 1996 to work on a PhD in Image Synthesis. My topic was the efficient paint box. I have proposed a new model for color shading and improved fundamental algorithms of computer graphics. Then I focused on rendering and my actual work concerns Non-Photorealistic Rendering (some of these publications are available on my web site).

In Paris 8 we teach courses in computer graphics, but our main concern is that the content must frequently change due mainly to hardware evolutions. Our first courses have been "classical" ones, basic algorithms in computer graphics, based on the well-known "Computer Graphics Principles and Practice" (Foley, Van Dam, Feyner, Hugues) book. Due to time constraints the 3D part was generally reduced.

But the expectations of the students were very different and we have been obliged to adapt our courses. Indeed students prefer to create quickly a doom-like or more generally a software to move a camera in a 3D scene. Moreover it seems that theses courses can not be the first ones. It is as if you read "The art of computer programming" (D. Knuth) before writing your first program. So we decided to first teach the basics in 3D and pipeline of the modern graphic cards is explained. We use OpenGL as graphic library and we explain the use of matrices, the computation of the transformation matrices, lighting and hidden-surface removal... Using it the students must design programs to pass the evaluation. Some of theses pedagogic programs are 3D games. The limitations of this approach come naturally and are detected by the students during the development of their projects. They compare generally the frame rate of their applications to the frame rate of games ! Moreover we show how video games are built and it is obvious for students that graphic designer, sound designer, level designer and game programmer are very different. As an extension, courses on geometric algorithms,

modeling, rendering, basic algorithms and GPU programming are provided. In these, we explain and compare different methods. For example the shadow mapping is discussed on the GPU programming course and methods using GPU or stencil buffer can be compared; Gouraud, Phong and other shading algorithms, anti-aliasing and 2D primitives are explained in basics algorithms course.

It seems that programmed computer graphics applications is very stimulating for students due to the interactive applications created. Moreover the OpenGL machine allows us to explain the modern graphic cards and their limits. Critical algorithms like scan-line are then introduced easily. This can be viewed as "Create your own rendering system" with first an explication on what exists and then extensions to explain how it can improved and approach video game.