Visualization is not a Game!

Hanspeter Pfister, MERL
pfister@merl.com

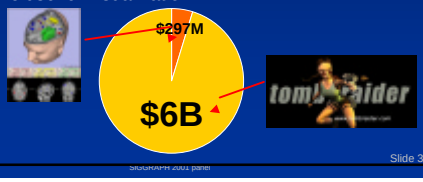
"With a little help from my friends…"

Lisa Avila-Sobierajski, Kitware
John Clyne, NCAR
Michael Halle, Surgical Planning Lab, B&W Hospital
Kevin Kreeger, Viatronix
Barthold Lichtenbelt, 3DLabs
Andy Vesper, TeraRecon
Ruediger Westerman, University of Aachen

All opinions expressed in this talk are of course my own!
Especially if you have a problem with them...

Trends

- Games drive the pace of graphics innovation on the PC.
  - New commodity hardware every 6 months.
  - New DirectX API every 12 months.
- Graphics and visualization applications must run on PCs to be commercially viable.
- Games directly influence what graphics hardware and APIs we use for visualization.



The Good, the bad & the ugly

- Thanks to games, we are getting some really good and very cheap graphics hardware.
- More programmability in the graphics pipe:
  - Vertex shaders, pixel shaders, multipass rendering.
- The visualization community is starting to do some cool stuff with this.
  - "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading", K. Engel et al., Graphics Hardware 2001.
  - "Volumetric Deformation on General Purpose Hardware", C. Rezk-Salama et al., Graphics Hardware 2001.

The Good, the bad & the ugly

- Commodity graphics cards have:
  - Missing or broken features.
    - High precision graphics pipe and framebuffer, overlay planes, texture border support, 3D texture mapping in hardware, pixel read/write operations, hardware accumulation buffers, support for stereo, multiple synchronized graphics pipes, multiple video formats, support for many graphics contexts, high-quality AA lines, indexed color in HW, streaming texture support (for video), hardware picking, large textures (> 512x512), deep pixels, etc.
  - Drivers tuned for Quake.
  - Severe image quality issues.
  - Unstable drivers.

The Good, the bad & the ugly

- Mind the "blue screen of death"…

The Good, the bad & the ugly

- APIs are driven by the needs of the game community.
- Game programmers have a very short event horizon.
  - They don't worry about the long term consequences of new API features.
  - Shaders = assembly programming (@#%$#!)
- Sound API principles are replaced with "extensions" or "capabilities".
  - OpenGL extensions are badly broken.
  - Who can keep up with D3D?
- Professional APIs (i.e., OpenGL) are in danger of commercial obsolescence.
  - D3D / Xbox vs. OpenGL / Quake

Here's the deal...

- Games and game programmers screw up our graphics hardware and APIs.
- The requirements for visualization are not met:
  - Image quality.
  - Scalability and stability.
  - Cross-platform availability of features.
  - A professional API built on sound principles.
- How do we change this?
  - Beg John McCarmack to use 64-bit pixels, multiple graphics contexts, and 3D textures in Quake IV!