

**Overview of
Embedded OpenGL Presentation
to OpenGL_Japan
by Seaweed Systems, Inc.**
June 22, 1999

- Who is Seaweed?
- What is embedded OpenGL?
- Why embed OpenGL?
- Reality vs. embedded OpenGL
- Who wants it?
- Tanks and planes and choppers.
- Simulators
- Set-top boxes
- Other markets
- Byebye

Presented by: Robert Schulman
Seaweed Systems, Inc.

Who is Seaweed?

- Founded in 1993.
- A small company.
- Specializes in X Window System and OpenGL products for the embedded marketplace (pSOS and VxWorks).
- Level 3 OpenGL licensee.
- Partnered with 3Dlabs; Seaweed's OpenGL software is based on 3Dlabs' OpenGL ICD source.
- Interesting work experience items
 - Started with Unix in 1976
 - SRA of Tokyo experience:
 - † Installed first commercial Unix system (1980)
 - † Compiler and code tools (1981-1982)
 - † Helped with Sony News (1986-1987)
 - First non-Project Athena X server (V9) (1985)
 - Started Seaweed (1993)
 - Started OpenGL renderer work (1995)

What is embedded OpenGL?

- It is OpenGL 1.1. It is not subsetted.
- It is a GLX implementation.
- What is an embedded system? It's a system where the presence of the computer is not why you buy the system.
- What are the engineering characteristics of embedded systems? Hard to say...but they almost always run a "real-time" kernel which has been modified especially for embedded apps (no VM, no persistent storage, quick boot).
- What are examples of embedded OpenGL implementations?
 - Avionics control panels
 - Set-top box graphics
 - Navigation systems in cars
 - Portable, vehicular-based simulators

Why embed OpenGL?

- Why do *any* kind of graphics in an embedded system?
- How does a big API like OpenGL work on what is usually a small OS?
- Why not put a "real" OS under the OpenGL implementation?

Reality vs. embedded OpenGL

- Users of embedded systems don't need or want a "complete" graphics solution. They just want OpenGL. They don't want X+OpenGL or Windows+OpenGL or QuickDraw+OpenGL. They just want OpenGL. But you can't do "just" OpenGL because OpenGL is designed to rely on other graphic system services.
- Memory may be constrained.
- Memory will not be VM and there won't be a disk. So, display lists can't be near-infinite in size.
- Real estate. Typical embedded systems don't have room to spare. An OpenGL solution which takes 15 chips to implement will be very difficult to implement in an embedded system.
- Heat. Typical embedded systems don't have noisy, failure-prone fans. The less fans you have, the more heat becomes an issue.
- Buses. AGP is really a very fast PCI wire. Really interesting graphics chips will soon only be available in AGP for the PC marketplace. How can an embedded system talk to such chips if no AGP port is available for the chosen embedded CPU?

Who wants it?

- Users who need 3D rendering. Spinning logos, bird's-eye navigation systems, simulators.
- Users who need interesting "free" features from OpenGL.
 - Textures. Video or map images can be treated as textures and arbitrarily rotated, scaled, and translated. Very useful in military vehicles.
 - A widely deployed graphics standard. Avionics controls typically render only 2D graphics, but development occurs on Windows and Unix boxes with deployment on embedded systems.

Tanks and planes and choppers

- Two kinds of customers in the military vehicle market:
 - Controls
 - Imaging and maps
- For controls, think of what's in the cockpit of a 777. A brightly lit LCD display. In the F-22 (next-generation US fighter), for example, there will be 6 such displays. These displays are information-rich and heavily designed instruments.
- Off-the-shelf software packages aid in the development of such displays. These packages run on Irix, Windows, Solaris, and embedded systems.
- Using this off-the-shelf software, the displays are implemented using a meta language. The meta language is compiled, and the generated code is C language with lots of calls to OpenGL functions. The generated C code is then compiled and run on whatever platform is appropriate.
- Note that the display is *developed* on Irix-Windows-Solaris platforms and the display is *deployed* on the embedded system.
- The displays **must** be correct, responsive, informative, robust, and effective. That takes a lot of engineering.
- Note that the common theme is OpenGL.
- Note that the generated C code and its OpenGL calls don't take advantage of the 3D capabilities of OpenGL.

Tanks, ... *continued*

- For maps, users display scanned map data.
- Typical application would be to orient the map data in the direction of travel, overlay a flight or battle plan on top of the map data, and then overlay existing conditions (e.g. current position) on top of that.
- Video information is also used as a texture. Some graphics chip move YUV-format video information into the frame buffer by texturizing the image. Some radar applications want to texturize video information to create a radar-like display.

Simulators

- Simulators in the field can be useful. Can be used for:
 - Wire missile guidance rehearsal.
 - Rehearsal of deployment/attack to enhance familiarity with terrain.
 - "Out the window" scene simulation in an aircraft needed due to poor visibility.

Set-top boxes

- BGL == Broadcast GL.
- Various companies, chip, software, and integration are involved.
- BGL is proposing a subsetted OpenGL for use in set-top boxes.
- Subsetting strips out software rendering, display lists, color index mode. Also possibly eliminated are accumulation buffers, stencils, and perhaps lighting.
- Adding texture compression, anisotropic filtering, multitexturing and surfacing and compositing primitives.
- In set-top world, customers only care about effects and don't care about OpenGL. They want spinning logos, alpha blended graphics, textured everything, games, and so on.
- Easiest to deliver the necessary technology via OpenGL and not via Direct3D or something proprietary. Leverage existing expertise and markets.
- Aim for 0.5 Mbyte OpenGL DLL (shrunk from 1.5 Mbyte)!

Other markets

- Arcade games.
- Other games (e.g. slot machines)
- Cars

Byebye

- Embedded OpenGL can be just like desktop OpenGL.
- Needs less physical space, less power, less heat, diminished capabilities, less memory to implement.
- Some target markets so huge (set-tops) they can evolve OpenGL spec.
- Related sites:
 - Wind River Systems (RTOS vendor): www.wrs.com
 - Integrated Systems (RTOS vendor): www.isi.com
 - Virtual Prototypes (rapid prototyper): www.virtualprototypes.ca
 - Corypheaus Software (rapid prototyper): www.centricsoftware.com

Contact info

Seaweed Systems, Inc.
Robert Schulman
21225 NE 132nd Ct.
Woodinville, WA 98072
+1 (425) 895-1721 (phone)
+1 (425) 895-9442 (fax)
bob@seaweed.com
www.seaweed.com

Seaweed's representative in Japan:
kamo, inc.
Christopher R. Keener, Ph.D.
1-39-4 Onsen Kamiyamada-machi
Sarashina-gun Nagano-ken 389-0821
026 275 3850 (phone)
026 275 3851 (fax)
duck@kamoinc.com
www.kamoinc.com