by Richard S. Wright Jr.

OpenGL & Mobile Devices

Real-time 3D graphics for handheld devices

When it comes to games, visualization, and even video-stream processing, OpenGL is the de facto standard for cross-platform real-time 3D graphics. So when I first heard that Ope... was being adapted for things like cell phones and PDAs, I was, well, ske... spoiled by workstation-class 3D graphics, going back to sof... the size of a note card was underwhelming, t...

Richard *writes science visualization an... softw...*

Results

Finally, I had my first 3D model textured, lit, and spinning on the screen. Not content with the same old example, Full Sail's Ed Womack dug up a better-looking model and I placed a sky background behind his game model of a tri-plane. Figure 1 is the result, running live on the Gizmondo. The plane model is lightweight, only 977 triangles, and I wa[s] getting a frame rate of just over 67 fps.

On small screens, you don't need a lot of geometry for nice-looking models. Us[ing] low poly models, 3D games can easily get playable frame rates. Also, in my examp[le I] was using floating-point math. There is no native floating-point support on the A[RM] CPU, and I knew that I was falling back to floating-point emulation, and that biti[ng the] bullet and going to fixed point could have dramatic performance benefits. I c[ouldn't] have been more wrong.

Float Versus Fixed

Switching to fixed-point math is probably the biggest single change in m...

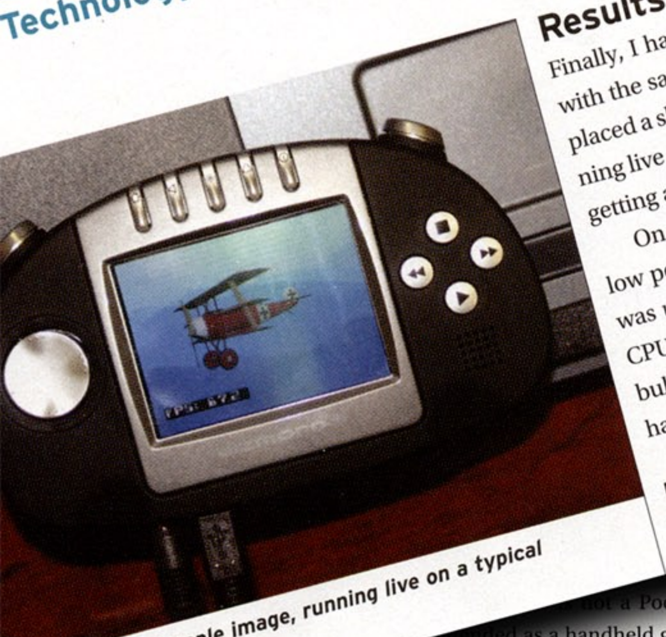

Figure 1: Sample image, running live on a typical mobile device.

...not a Pocket PC or PDA—it was ...ended as a handheld gaming device—so there are a few hurdles most PDA developers are spared. For instance, I had to use a special "Developers" version of the USB driver to form a regular partnership with the device and copy files back and forth to protected areas. There was also a special program that had to be integrat-

...dering. Dave ...OpenGL ES Game Develop-... was invaluable, covering many aspects ...OpenGL ES game development, and NVidia provided the headers and libraries:

```
#include <gl/egl.h>
#include <gl/egltypes.h>
#include <gl/gl.h>
```

for bringing in the EGL interface and OpenGL ES functions.

The EGL interface works much like desktop binding interfaces, in that you need handles to the display