Coming to a Pixel Near You: Mobile 3D Graphics on the GoForce WMP

Chris Wynn

NVIDIA Corporation

What is GoForce 3D?

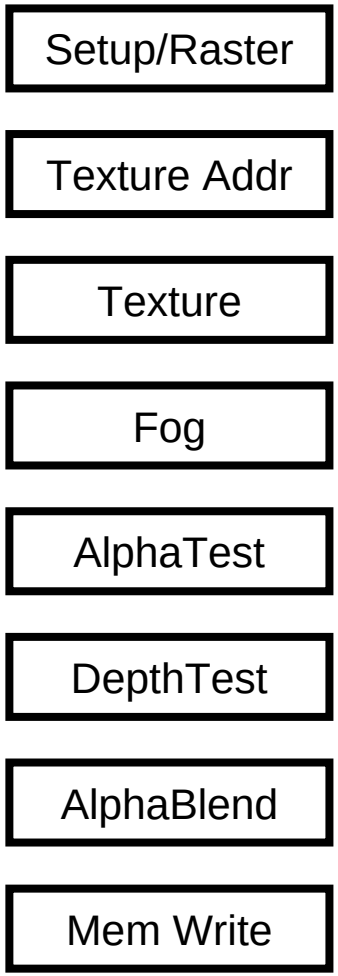
- Licensable 3D Core for Mobile Devices
- Discrete Solutions: GoForce 3D 4500/4800

- OpenGL ES / Direct3Dm compliant
- Low Power
- Integrated SRAM
- Up to VGA resolution
- Modern Feature Set

GoForce 3D 4800 Features

- Geometry Engine
- 16-bit color w/ 16-bit Z (40-bit color internal)
- Multi-texturing w/ up to 4 textures
- Bilinear / Trilinear Filtering
- Flexible Texture Formats (DXT1/4-bit/8-bit)
- Fully Perspective Correct
- Sub-Pixel Accuracy
- Per-Pixel Fog, Alpha Blending, Alpha-Test

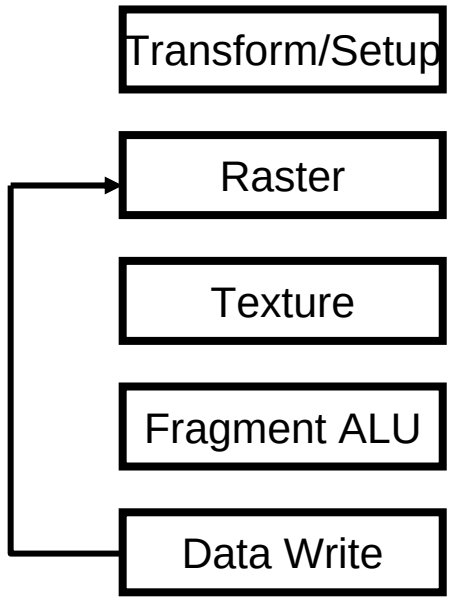
Traditional Architecture



- **Deep pipeline**
- **Always have to go through all stages**
- **Pipelines always clocking**
- **Fast, but too much power consumption**
- **~750mW per 100M pixel/sec**

(~200 pipe stages)

GoForce 3D: New Low-Power Architecture



(~50 pipe stages)

- **Flexible Fragment ALU**
- **Raster – fragment generation and loop management**
- **Pipelines only trigger on activity**
- **Low Power**
  - **< 50 mW per 100M pixel/sec**
  - **During actual gameplay**
- **Very scalable architecture**

Developing for GoForce 3D

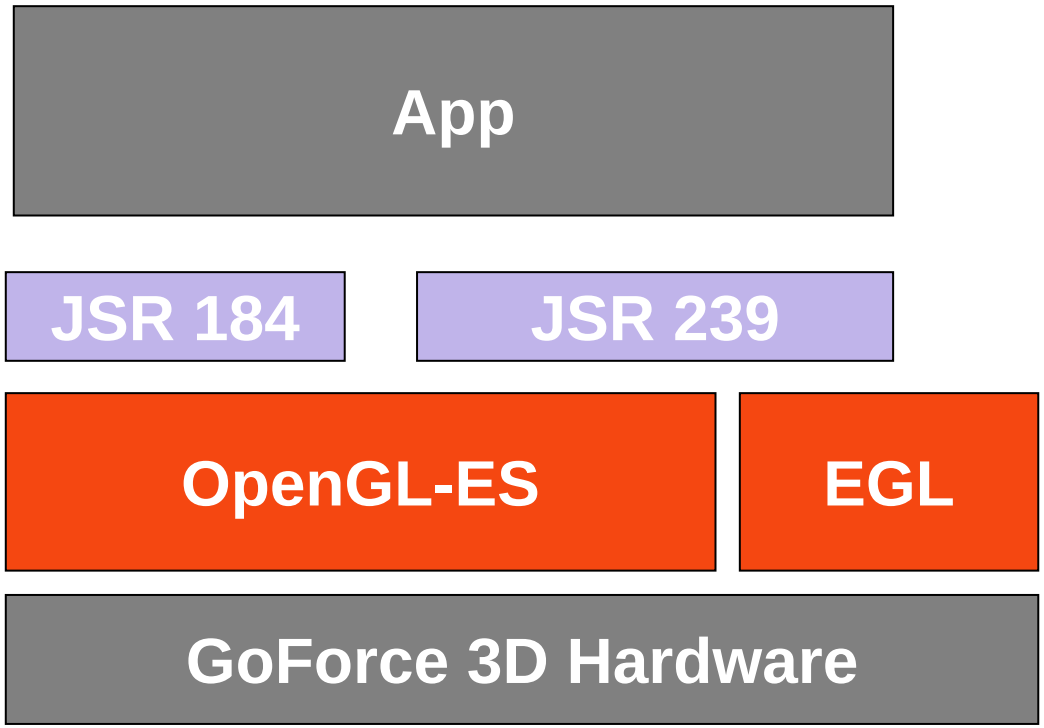
- Java Applications
- Native Programming Model
- Carrier / Middleware Models

Bubble
(NVIDIA Tech Demo)

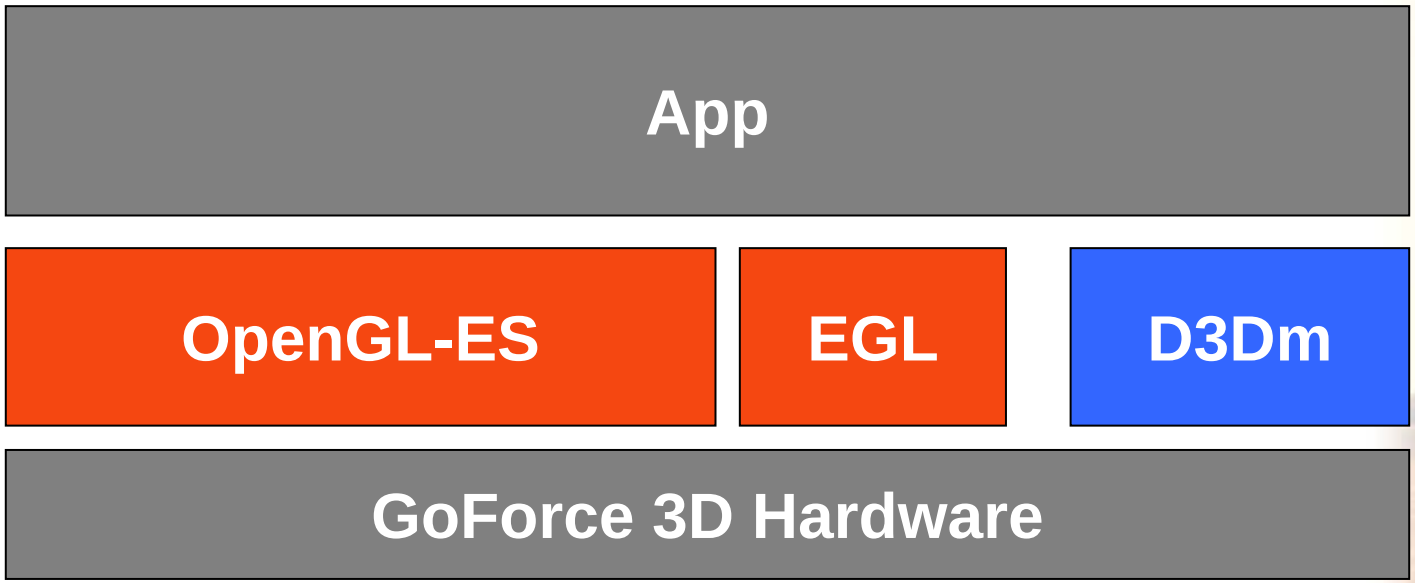
Sherman
(The Astonishing Tribe)

World War II Airplane Demo
(Futuremark Corporation)

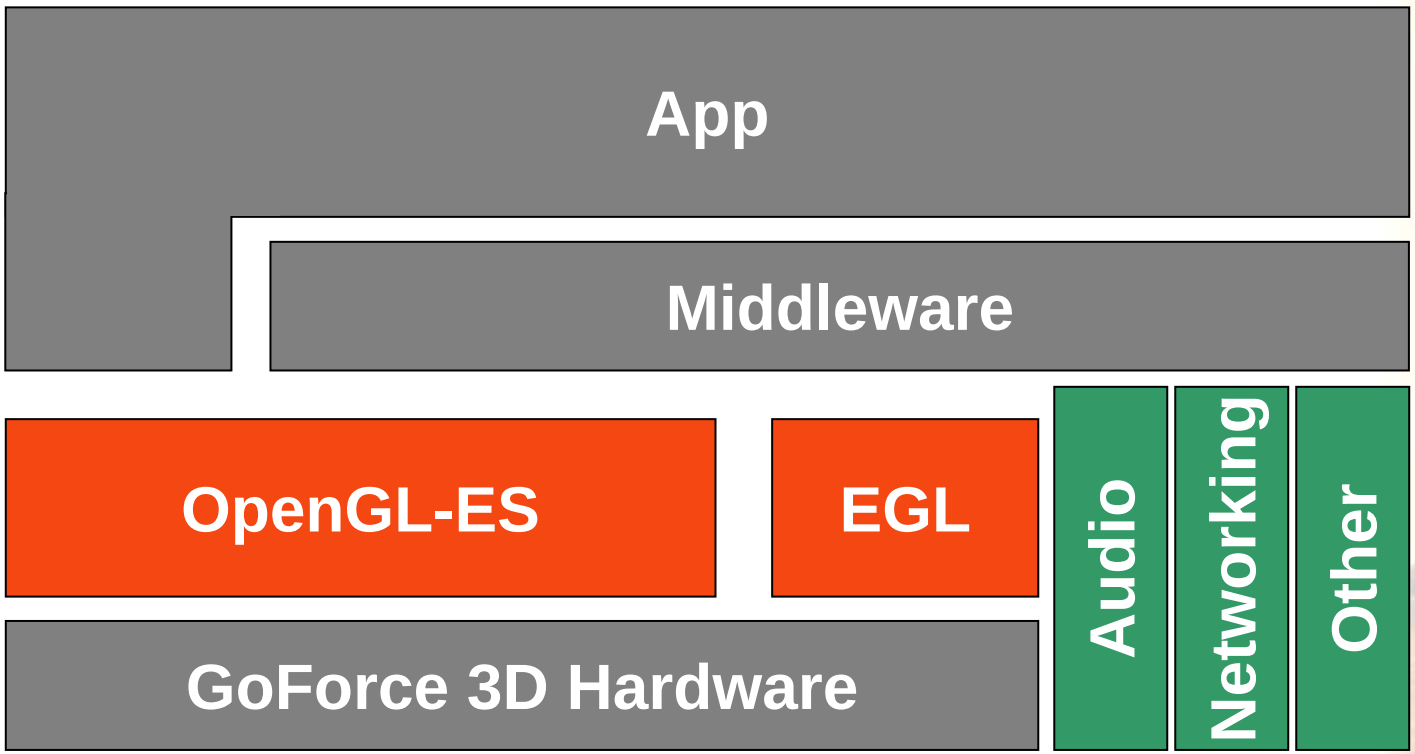
Java Programming Model



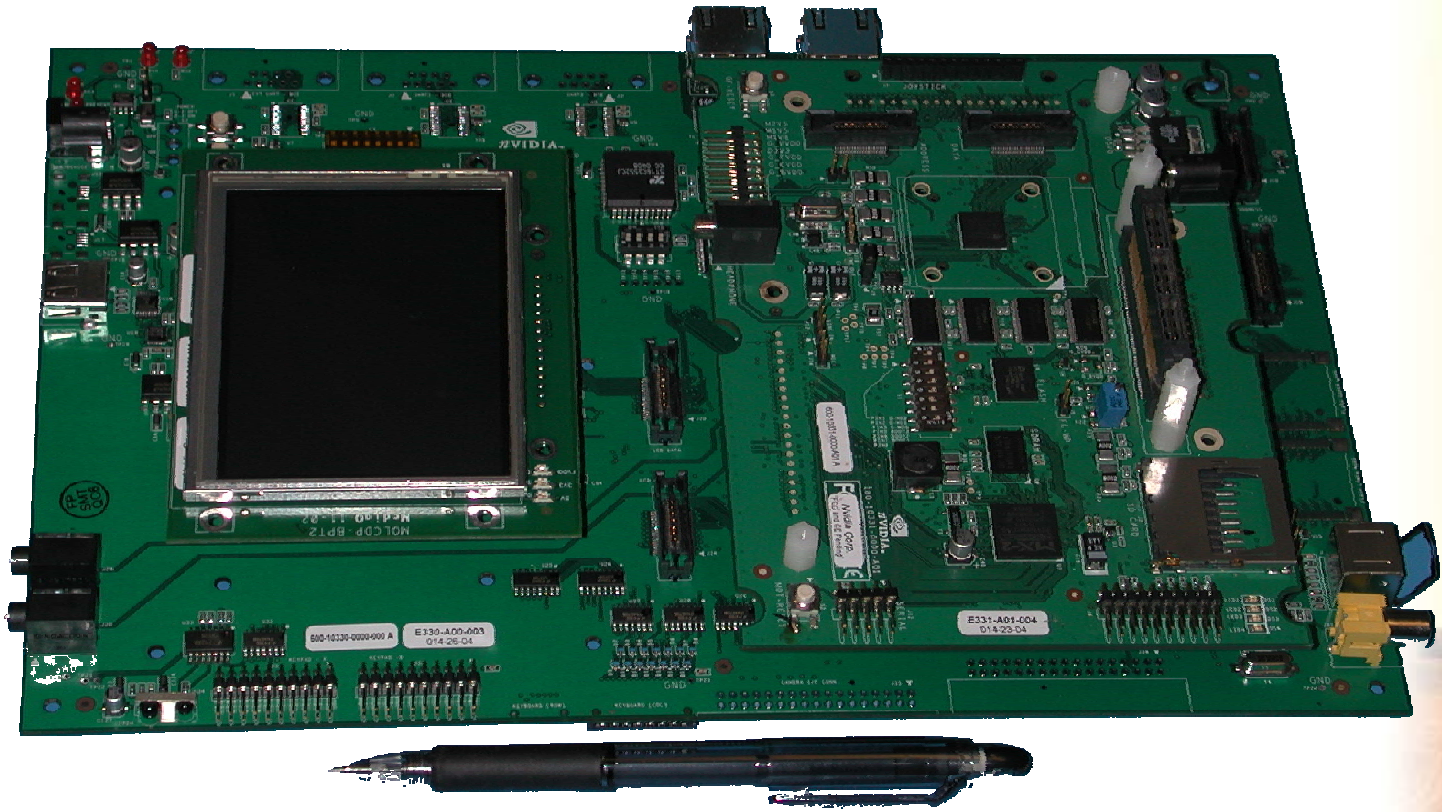
Native Programming Model



Middleware Programming Model



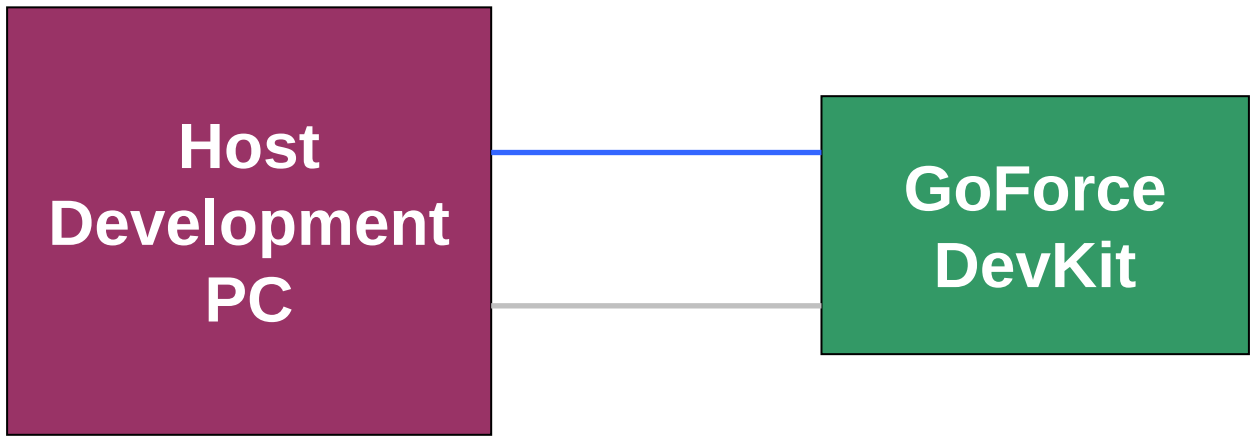
GoForce 3D Development Kit



What's Included?

- **Handset Hardware Environment**
  - **Freescale i.MX21 (ARM926EJ-S™ core)**
  - **GoForce 3D 4500 / 4800**
  - **2.2" or 3.5" QVGA (240x320) LCD**
  - **Joypad Input Device**
  - **Audio, SD, USB**
- **Multi-OS capable (Linux now, WinCE soon)**
- **Software (drivers, tools, libraries, sample code)**
- **Documentation (2D/3D)**
- **3rd party support (Java VM w/ JSR184 available)**

What to Expect?



- **Null Modem Cable – boot sequence**
- **Crossover Ethernet Cable – console / remote debug**
  - **GNU Tools**
  - **eMbedded Visual C++ 4.0 Tools**

From PC to Mobile: Things to Be Aware Of

- **CPU Usage**
  - No floating point or integer division – avoid these
  - Simplify physics, collision, visibility
  - Small caches – avoid non-coherent algorithms
  - Avoid CPU-assisted driver paths (lighting)

- **System Memory Bandwidth**
  - Cell Phones and PDAs – low system memory bandwidth
  - Monitor/Conserve System BW

Alternative Lighting Strategies

- **Fake Phong highlights using Multitexture**
- **Pre-Computed Vertex Lighting**

Stuntcar Extreme
(Images Courtesy of Fathammer)

Conserving Bandwidth: Geometry

- **Use only the amount of geometry required (LODs)**
- **Minimize amount of data per vertex**
  - **Use packed ARGB (not floating point ARGB)**
  - **Don't specify Z or W if you don't need it**

- **Use multi-texturing instead of multi-pass rendering**

Example: Applying 2 textures

single-pass w/ multi-texture: pass 1 – ($xyzs_1t_1s_2t_2$)
two rendering passes: pass 1 – ($xyzs_1t_1$) pass 2 – ($xyzs_2t_2$)

→ **single-pass up to 30% reduction in data**

Vertex Cache Optimization

- **What is a Vertex Cache?**
  - **Place for the WMP to store Vertex Data**
  - **If a vertex is already in the cache, WMP uses the cached copy**
- **How to utilize this?**
  - **Use triangle strips or fans**
  - **Or Use Indexed Triangle Lists**
    - **Optimize for 32- or 16-entry LRU Vertex Cache**

Conserving Bandwidth: Texture

- GoForce 3D 4800 has 1280K SRAM
  - Fast but scarce resource
  - On-chip texture memory is 1280K – Frame Buffer

- Ex. QVGA (320x240) 16-bit color double buffered w/ Z:
  - 450K Frame buffer
  - 830K Texture memory available

Working within a Memory Budget

- **Fit Working Set of Textures in Vidmem(!)**
  - **Use compact texture formats**
  - **EXT_texture_compression_dxt1 (4-bit/texel)**
  - **Use smallest resolutions possible**
  - **Avoid allocating full-screen buffers**
- **If budget exceeded, sort by texture**

The Benefits of DXT1

Bubble: Each scene uses 8 textures

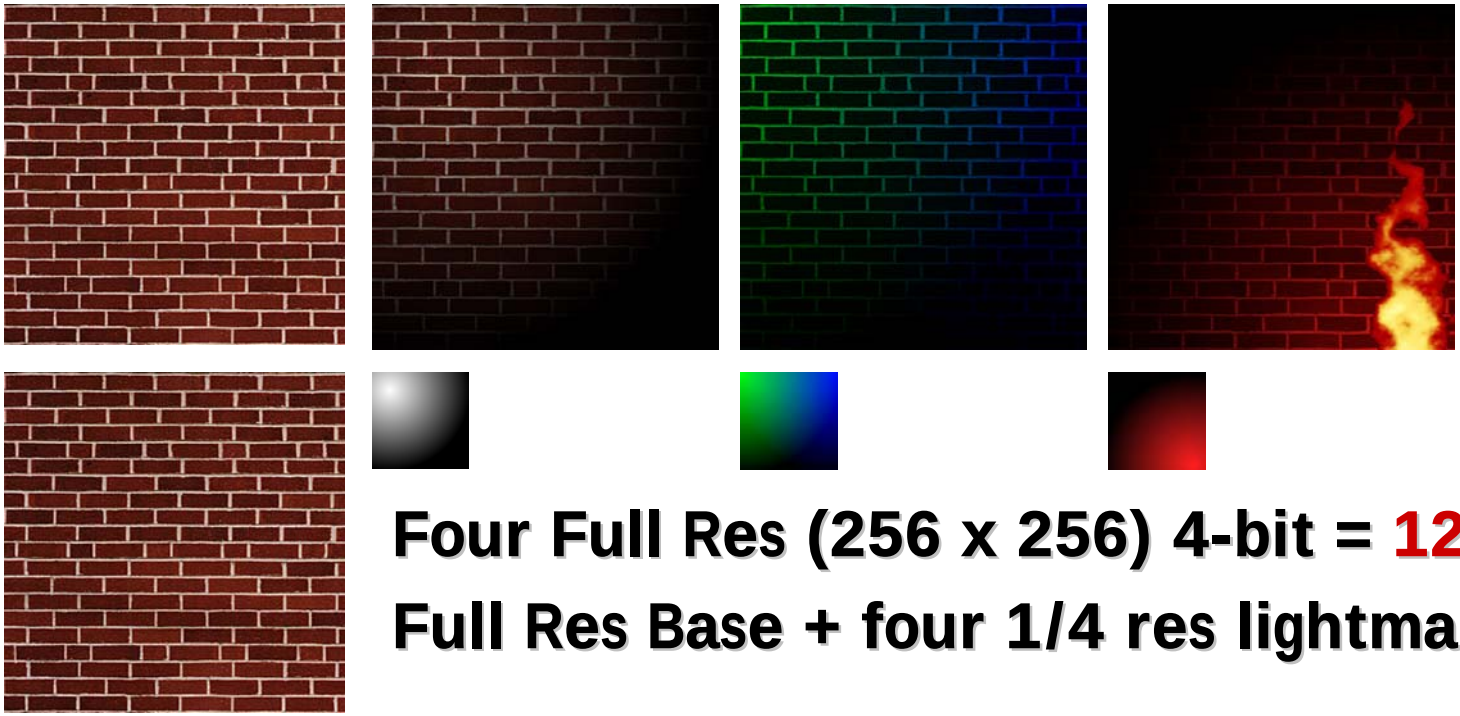
- 2 – 256x256 textures (mip-mapped)
- 6 – 256x256 textures (non-mipmapped)

RGB565 (16-bits/texel) = 786432 + 349524 = 1.08Mb

DXT1 (4-bits/texel) = 196608 + 87381 = 0.27Mb

DXT1 is High-Quality and 25% the cost of RGB565

Leveraging Multi-Texture



Four Full Res (256 x 256) 4-bit = 128kb

Full Res Base + four 1/4 res lightmaps = 40kb

- **Store diffuse maps in lower resolution and use multitexture to save memory**

OpenGL ES 1.0 Primer

- Roughly OpenGL 1.3
- Removes
  - Display List
  - glBegin/glEnd
  - Texgen
  - Environment Maps
  - Evaluators
- Adds
  - Fixed Point type/entry points
  - Byte type more universal

Direct3Dm Primer

- Largely based on Direct3D8
  - No Vertex or Pixel Shaders
- Float and Fixed-Point (16.16) types
- Texture Coordinate Generation
- Up to 4-way multi-texture
  - “Cascading” texture blending like OpenGL ES 1.0
- Additional Caps Bits
- Profiles – collection of caps bits required for a given profile

Interested in Developing for GoForce 3D 4800?

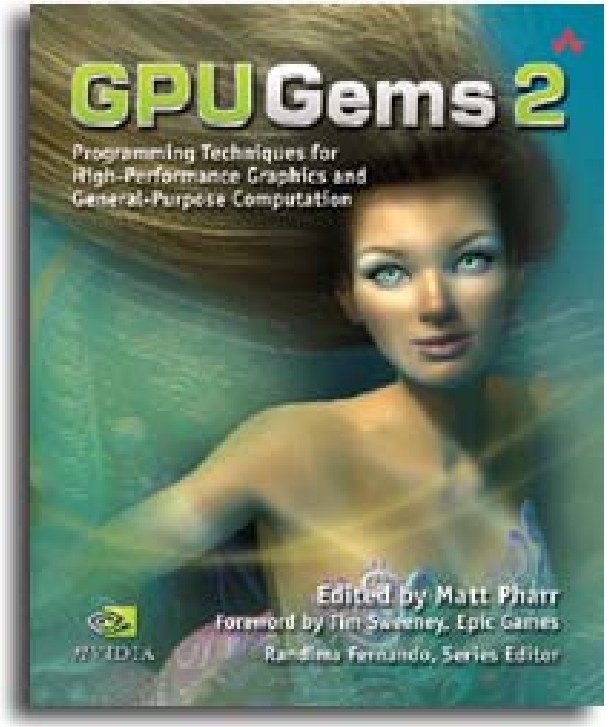
- **Hardware DevKits available now!**
- **Register for NVIDIA Handheld Developer Program**

  http://developer.nvidia.com
- **Email**

  handset-dev@nvidia.com

GPU Gems 2

Programming Techniques for High-Performance Graphi and General-Purpose Computation



- **880 full-color pages, 330 figures, hard cover**
- **$59.99**
- **Experts from universities and industry**

"The topics covered in *GPU Gems 2* are critical to the next generation of game engines."

— Gary McTaggart, Software Engineer at Valve, Creators of Half-Life and Counter-Strike

"*GPU Gems 2* isn't meant to simply adorn your bookshelf—it's required reading for anyone trying to keep pace with the rapid evolution of programmable graphics. If you're serious about graphics, this book will take you to the edge of what the GPU can do."

—Rémi Arnaud, Graphics Architect at Sony Computer Entertainment