Performance Tools Agenda

- Eric Preisz: Optimizing Marble Blast Ultra
- Introducing PerfHUD 6.0
- PerfKit
  - PerfSDK: GPU & Driver Performance Data
  - Graphic Remedy’s gDEBugger
- ShaderPerf
- FX Composer 2.5: Shader Debugger
Optimizing
Marble Blast Ultra
By: Eric Preisz
The Game Pipeline

CPU
- AI
- Phys.
- Culling
- Game
- DirectX
- Drivers
- System Mem
- Non-Local Vid Mem

GPU
- Primitive Assembly
- V/G/F Shader
- Rasterization
- Vertex Assembly
- Texture
- Video Memory

PCle
Levels of Optimization

System - % of hardware utilization
Application - Class/function/algorithm
Micro - Line by line optimizations.
Simple Mistakes - Resource Allocations
GPU Busy % for the best FPS increase
GPU System Level Performance Dashboard

- VA(%) ~5%
- Shader(%) ~55%
- Texture(%) ~45%
- ROP(%) ~25%
- VS(%) ~5%
- GS(%) ~0%
- PS(%) ~55%
State Bucket = 4.125 ms, 307,695 pixels
Draw Call = 3.247 ms, 257,662 pixels
Problem: Inexpensive geometry, low number of draw calls, high number of expensive pixels.....

Solution: Render scene twice and disable color writes on the first draw!
GPU Application Level

State Bucket = 4.826 ms, 365,928 pixels
Draw Call = 0.834 ms, 125,868 pixels
Still very GPU Bound (A),
Still very shader and texture bound (B)…
Now we are limited by a FFP call. By using the “scrubber” we determine that the call renders the glow buffer quad to the screen.
Moving on to a texture optimization
A StretchRect optimization reduces texture work, making pixel optimizations even more valuable.
A driver optimization increases GPU Utilization.
What’s next...

- Investigate more pixel optimizations – Scissor test confirms.
- Compressed texture usage – forcing 2x2 gives ~5 frame improvement.
- SLI Optimizations
- Investigate CPU optimizations, on the 8800 GPU utilization is only 25%
Thanks Eric!!!
Introducing PerfHUD 6.0!

- Unified Driver on Vista: use any release driver!
- Comprehensive SLI Support
  - Graphs for SLI specific data
  - Insight into SLI performance gotchas
- Powerful new debugging features
  - Texture visualization modes
  - API Call data mining and analysis
  - Shader visualization
- Usability Features
  - All new hot key support
  - Rich use of Direct3D PerfMarkers

© NVIDIA Corporation 2008
PerfHUD 6: Performance Dashboard

Graph GPU and driver data
Edit to suit your needs
SLI Graph for multi-GPU API usage statistics

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.

© NVIDIA Corporation 2008
PerfHUD 6: Frame Debugger

Scrub through scene
Visualize draw call info
Textures and RTs
Tooltips on buffers

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.

© NVIDIA Corporation 2008
PerfHUD 6: Frame Debugger

Texture analysis: substitute precomputed textures
Controllable via Perf Markers

Texture analysis in the PerfHUD 6 interface.
PerfHUD6: Frame Debugger

Buffer Visualization

Visualize any buffer
full screen
2D/3D/Cube/Arrays
Pan/Zoom
Change mipmap level

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
### PerfHUD6: Frame Debugger

#### API Call List

- See frame events including parameters
- Tooltips for details
- Connected to scrubber

<table>
<thead>
<tr>
<th>Event</th>
<th>API Call Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Frame Capture Event</td>
</tr>
<tr>
<td>0001</td>
<td>API Call Event 1</td>
</tr>
<tr>
<td>0002</td>
<td>API Call Event 2</td>
</tr>
<tr>
<td>0003</td>
<td>API Call Event 3</td>
</tr>
</tbody>
</table>

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
PerfHUD 6: Frame Debugger

Draw Call Dependencies

Show producers & consumers dependencies for each call
These can hurt single GPU and SLI performance

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
PerfHUD 6: Adv Frame Debugger

Vertex Assembly

Geometry Preview
Vertex and index buffer setup

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
PerfHUD 6: Adv Frame Debugger
Vertex, Geometry and Pixel Shaders

Edit & Continue Shaders
Visualize input textures
Constants
Sampler overrides

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.

© NVIDIA Corporation 2008
PerfHUD 6: Adv Frame Debugger

Display and modify all render state settings
Render targets displayed

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
PerfHUD 6: Frame Profiler
One button bottleneck determination

All draw calls profiled
Draw calls grouped by State Buckets: multiply performance optimizations
Multiple result graphs

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.

© NVIDIA Corporation 2008
PerfHUD 6: Adv Frame Profiler

Same advanced features now in the profiling context

Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.
PerfHUD 6

A taste of things to come!

- Better control via PerfMarkers: add them now!
- API time graph
- More performance hints: VSync on, windowed mode, event queries, not all render targets used, VBs not managed, etc.
- Subtotals in Frame Profiler
- Break (_int 3) on draw call
- 32bit apps on 64bit OSs
PerfKit: Features

**PerfSDK**
- Real time performance information in your game
- Driver data, GPU counters, etc.
- Simplified Experiments for easy bottleneck analysis
- Simple API, code samples and helper classes

**GLExpert**
- Detailed feedback on pipeline setup
- SLI performance feedback
- Warnings for software fallback
- VBO/FBO performance information

**Microsoft PIX for Windows plugin**
- GPU & driver counters alongside PIX data
Graphic Remedy’s gDEBugger

OpenGL and OpenGL ES Debugger and Profiler
- NVIDIA PerfKit and GLEExpert integrated
- Shorten development time
- Improve application quality
- Optimize performance
- Texture/buffer viewer
- Windows XP & Vista, Linux too!
- Discounted academic licenses available

http://www.greemdy.com

© NVIDIA Corporation 2008
Graphic Remedy’s gDEBugger
ShaderPerf 2.0

- Determine shader performance
  - Compare with shader cycle budgets
  - Test optimization opportunities
- Automated regression analysis
- Integrated in FX Composer
  - Interactive GUI
  - Artists/TDs code expensive shaders
  - Achieve optimum performance
- GeForce 8 series performance data
- Beta 2.0 available now!
FX Composer
Shader Authoring Made Easy!

- DirectX 10 backend
- Shader Debugger
- GeForce 8 Series Shader Performance
- Full-featured code editor
- Shader creation wizard with templates
- Integration with online Shader Library
- Materials panel to organize materials

© NVIDIA Corporation 2008
FX Composer
HLSL10 Shader Debugging!

© NVIDIA Corporation 2008
Questions?

Online: downloads, videos, etc.

http://developer.nvidia.com/PerfKit
http://developer.nvidia.com/PerfHUD
http://developer.nvidia.com/ShaderPerf
http://developer.nvidia.com/FXComposer