

NVIDIA®

Integrating Shaders into Your Game Engine

Bryan Dudash

NVIDIA

Developer Technology



Agenda

Why shaders?
 What are shaders exactly?
 Evolution of graphics
 Using Shaders
 High Level Shading Languages
 C++ side API and semantics
 Tools



Why Shaders?

- Pixel Shaders are the #1 feature that will visually differentiate next-gen titles
- Distinct materials
 - Great way to show detail without geometry
 - Not everything matte or plastic
 - Moving away from just Blinn/Phong
 - Custom light types
 - Volumetric lights
 - Not limited to OpenGL fixed pipeline



No Shaders vs Shaders





Flat texture, single texture, vertex lighting, no shadow

Bump mapped, multi texture, per pixel lighting, soft shadow

Doom 3 courtesy of id Software. All Rights Reserved.



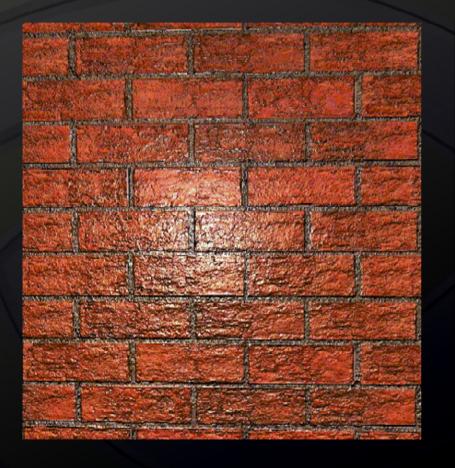
Per Pixel Lighting

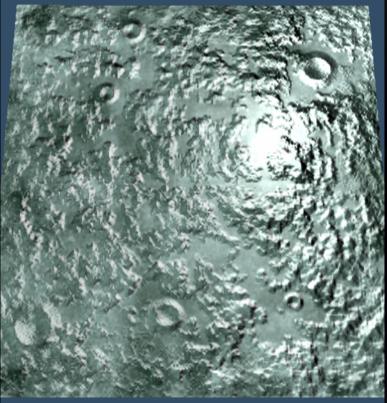
 Bump mapping / Normal Mapping / Per-Pixel Lighting are synonyms
 Blinn Diffuse Specular lighting
 With Tangent space Bump mapping

Instead of calculating lighting on a per-vertex normal, use a per-pixel normal instead

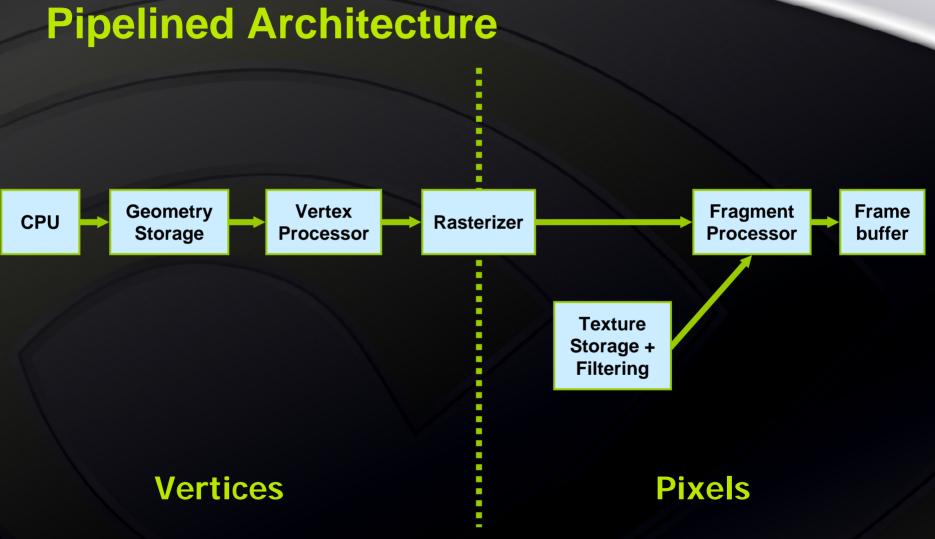


Two quads lit per pixel











What are Shaders?

User-defined vertex and fragment processing
 Custom animation, lighting, image processing, etc.

Ubiquitous platform & API support
 PCs, next-generation consoles, cellular phones
 Direct3D, OpenGL, OpenGL-ES

Programmed in C-like high level languages

- HLSL (Direct3D)
- GLSL (OpenGL)
- GLSL-ES (OpenGL-ES)
- Cg (OpenGL, OpenGL-ES)



Shader Taxonomy

Hardware functionality often described relative to Direct3D shader models 1 – 3

Newer shader models increase programmability

- SM 1: Fixed-point color blending, static dependent texturing, <= 16 operations</p>
- SM 2: Floating-point arithmetic, programmable dependent texturing, <= 64 operations</p>
- SM 3: Branching & subroutines, 1000s of operations



PC/DirectX Shader Model Timeline





Quake 3

Giants

Halo

Far Cry

UE3

All images courtesy of respective companies. All Rights Reserved.



DirectX 8, SM 1.x / OpenGL 1.4

Programmable vertex shaders

Up to 128 floating-point instructions

Programmable pixel shaders

- Up to 16 fixed-point vector instructions and 4 textures
- 3D texture support
- Up to 1 level of dependent texturing
- Advanced Render-to-Texture support

Example Hardware

GeForce 3, ATI Radeon 8500, XGI Volari V3, Matrox Parhelia



SM 1.x-era Game: Halo

Vertex shaders used to add fresnel reflection to ice
 Pixel shaders used to add glow to sun
 Render-to-texture used to distort pistol scope
 Dependent texturing used to animate & light water



Halo courtesy of Microsoft. All Rights Reserved.



DirectX 7 vs DirectX 8



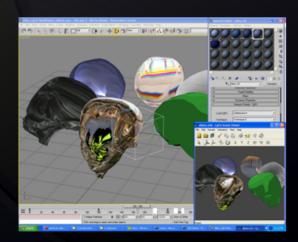
Halo courtesy of Microsoft. All Rights Reserved.

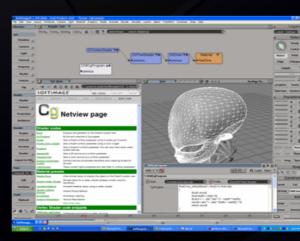


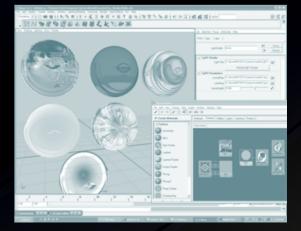
Cg – C for Graphics

High-level language designed for real-time shaders

Supported in major DCC apps (Maya, Max, XSI)
 What artists see in tool chain matches in-game result





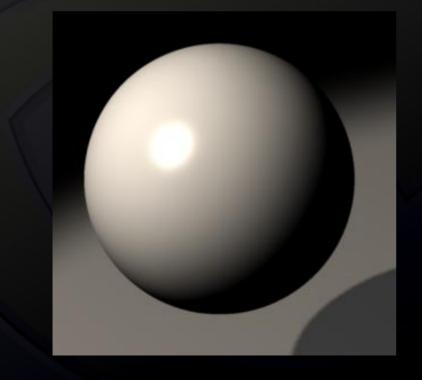




HLL vs Assembly

High-level source code

color.xyz = Ke + (Ka * globalAmbient) +
 Kd * lightColor * max(dot(L, N), 0) +
 Ks * lightColor * pow(max(dot(H, N), 0), shininess);
color.w = 1;



Assembly

ADDR R0.xyz, eyePosition.xyzx, -f[TEX0].xyzx; DP3R R0.w, R0.xyzx, R0.xyzx; RSOR R0.w, R0.w; MULR R0.xyz, R0.w, R0.xyzx; ADDR R1.xyz, lightPosition.xyzx, -f[TEX0].xyzx; DP3R R0.w, R1.xyzx, R1.xyzx; RSOR R0.w, R0.w; MADR R0.xyz, R0.w, R1.xyzx, R0.xyzx; MULR R1.xyz, R0.w, R1.xyzx; DP3R R0.w, R1.xyzx, f[TEX1].xyzx; MAXR R0.w, R0.w, {0}.x; SLER H0.x, R0.w, {0}.x; DP3R R1.x, R0.xyzx, R0.xyzx; RSOR R1.x, R1.x; MULR R0.xyz, R1.x, R0.xyzx; DP3R R0.x, R0.xyzx, f[TEX1].xyzx; MAXR R0.x, R0.x, {0}.x; POWR R0.x, R0.x, shininess.x; MOVXC HC.x, H0.x; MOVR R0.x(GT.x), $\{0\}$.x; MOVR R1.xyz, lightColor.xyzx; MULR R1.xyz, Kd.xyzx, R1.xyzx; MOVR R2.xyz, globalAmbient.xyzx; MOVR R3.xyz, Ke.xyzx; MADR R3.xyz, Ka.xyzx, R2.xyzx, R3.xyzx; MADR R3.xyz, R1.xyzx, R0.w, R3.xyzx; MOVR R1.xyz, lightColor.xyzx; MULR R1.xyz, Ks.xyzx, R1.xyzx; MADR R3.xyz, R1.xyzx, R0.x, R3.xyzx; MOVR o[COLR].xyz, R3.xyzx; MOVR o[COLR].w, {1}.x;



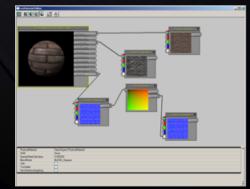
Impact of HLLs

Dramatic increase in shader adoption

Tens of games per year to hundreds

Shift in game development

- Shaders become content requirement, not tech feature
- "What do I want?", not "what can I do?"
- Gives control of the look of the game to artists





Unreal courtesy of Epic Games. All Rights Reserved.



DirectX 9, SM 2.0 / OpenGL 1.5

Floating point pixel processing

- 16/32-bit floating point shaders, render targets & textures
- Up to 64 vector instructions and 16 textures
- Arbitrary dependent texturing
- Longer vertex processing 256 instructions
- Multiple Render Targets up to 16 outputs per pixel
- Example Hardware
 - GeForce FX 5900, ATI Radeon 9700, S3 DeltaChrome



DirectX 9.0c, SM 3.0 / OpenGL 2.0

Unified shader programming model
 Pixel & vertex shader flow control
 Infinite length vertex & pixel shaders
 Vertex shader texture lookups

Floating-point filtering & blending

Geometry instancing

Example Hardware
GeForce 6800, GeForce 7800 GTX



SM 3.0-era Game: Unreal Engine 3

16-bit FP blending for high dynamic range lighting
 16-bit FP filtering accelerates glow and exposure FX
 Long shaders & flow control for virtual displacement mapping, soft shadows, iridescence, fog, etc.



Unreal Engine 3 courtesy of Epic Games. All Rights Reserved.



Using Shaders



"Effects"

Direct3D FX and CgFX

 ID3DXEffect or CGeffect

 Wrapper around pixel and vertex shaders
 Can Configure

 Target shader version
 Common case variables

 Can reference a library of shader functions
 Define multi-pass techniques



Semantics

Define any variable naming you want Semantics make sure constants get set

float4x4 wvp : WorldViewProjection;

D3D SAS is standardized Supported by many applications FX Composer 3D Studio Max

OpenGL semantics standardized for CgFX in 1.4



Annotations

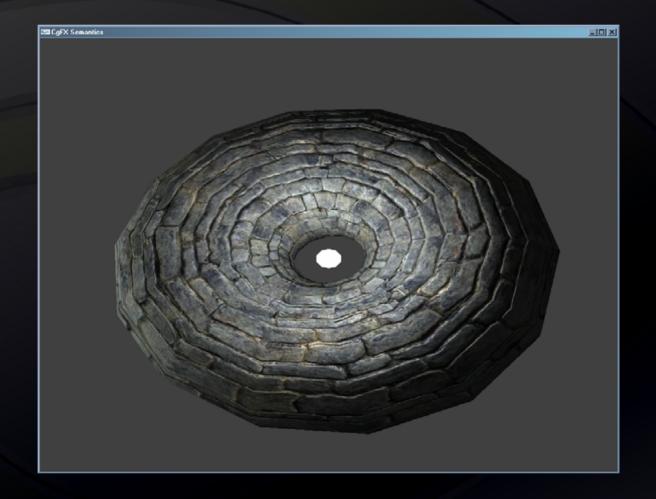
Custom data associated with any element of your HLSL or CgFX effect

```
sampler2D anisoTextureSampler <
    string file = "Art/stone-color.png";
> = sampler_state {
    generateMipMap = true;
    minFilter = LinearMipMapLinear;
    magFilter = Linear;
    WrapS = Repeat;
    WrapT = Repeat;
    MaxAnisotropy = 8;
};
```

Allows you to provide hooks to set per object data E.g. Used extensively by shader tools for UI controls



CgFX Semantics Demo





Demo Important Bits

- Tangent basis interpolated from vertex shader
- Single fragment shader for lighting
- An unsized array of light structures that is dynamically resized by the C++ side
- A handful of different light types that implement the light interface
 - Point light
 - Spot Light
 - Etc...
- Optional Bump mapping based on a constant



Single Lighting Function

- Sample Albedo map for base color
- Normalize interpolated vectors
 - Tangent space basis vectors
- Optionally perturb our normal based on a normal map
- Iterate over our lights and accumulate diffuse and specular
- Combine color and lighting values to produce final result



C++ Side

- Assign the light position through the effect given a handle to the variable
- Sets number of lights and light info based on program code dynamically
- Can also pick whether or not to use normal maps
 - When not using it, shader gets faster
- Any dynamic configuration can be represented as a uniform parameter or global constant

Shader Library

- Rather than writing each shader separately
- Code re-use is good!!
- Establish common interpolated values
 - Vertex to Fragment/Pixel program
 - e.g. At a base, COLOR0, TEXCOORD0 off limits
- Create a library of useful functions
 - Break everything out
 - Only costs compile time (can be preprocessed!)



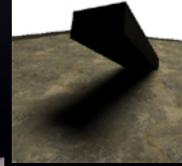




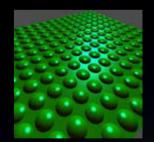
















Write with extensibility in Mind

- Quick hacks are for prototyping
- Same as regular code
- Establish guidelines for style
- Full preprocessor support
 - #ifdef #define etc
- Naming convention for techniques
- No Assembly!

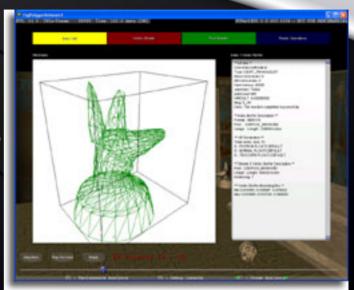


Performance

 CPU bound, or Pixel Shader
 NVIDIA's GPU Programming Guide
 NVIDIA provides a number of handy performance analysis tools
 NVShaderPerf
 NVPerfHUD

NVPerfKit

W T REPORTED AND A REAL	
to be a self-schole she being be to see a set of the schole of a self-	
Second provide as the Report of Second	
Sada bas Phayled Sough, Sana 1,81	
T II SHIT IN	8
by 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,	
E3 /9 #0 10	
NO MERCE	
21 2 2 3 2 3 2 3 2 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
丙间: 15.15.15.15.15.15.15.15.15.15.15.15.15.1	
問題: 注目	
2012 S 16.0	
2014 - 19 A.	
STORES AND ALL A	8
all and a	
Compile Softwards (1980) 2019; 11 Earlied Song Real address Services of the S.S. September 1.3.	
and the state of the second	
	1.1



Michiel Michiel <td< th=""><th></th><th>Available Countors</th><th>- Active Counters</th></td<>		Available Countors	- Active Counters
Counter Description The dec_static_linc_leadure : Time the pixel shader unit is statled waiting for a texture sead Settings	NMEDIA Weformance Monitor	GPU GPU GPU GPU GPU GPU GPU GPU	Here CPU Image: CPU Image: CPU Image: CPU Im
			a lexiture read
			Cel Falde



NVPerfHUD

What is NVPerfHUD?
How does it work?
Schedule



What is NVPerfHUD?

Stands for: PERFormance Heads Up Display
 Overlays graphs and dialogs on top of your application
 Interactive HUD



What is NVPerfHUD?

4 different types of HUD
 Performance Dashboard
 Debug Console
 Frame Debugger
 Frame Profiler (New in 4.0)



How to use it

Run your application with NVPerfHUD
 Use it as you normally do until you find:
 Functional problem: use the debugger
 Low FPS: use the profiler

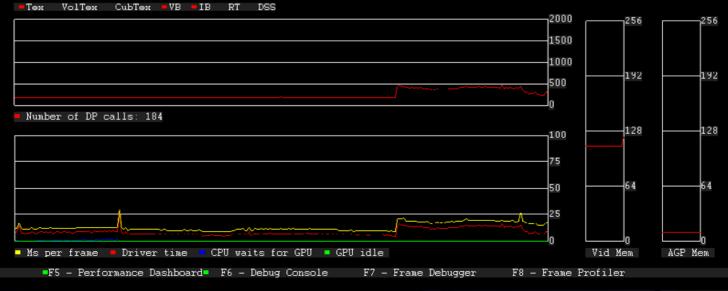
Performance Dashboard



FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.NProgram Files/Futuremark/3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.





1

Performance Dashboard



FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.\Program Files\Futuremark\3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.



-Tex VolTex CubTex -VB -IB RT DSS



1

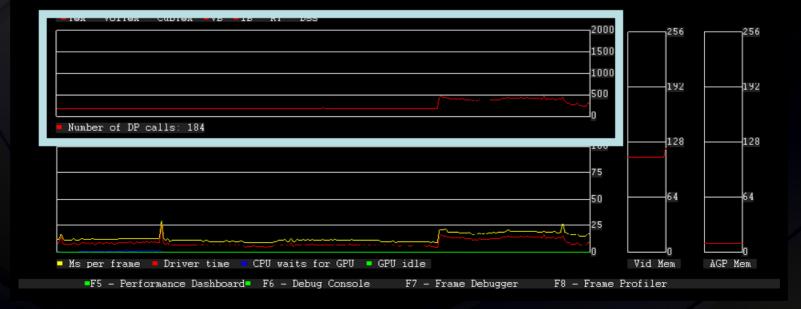


FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.NPogram Files/Futuremark/3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.



.



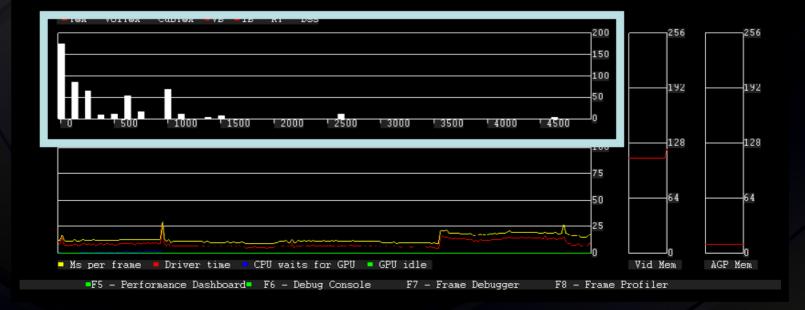


FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.NPogram Files/Futuremark/3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.



.



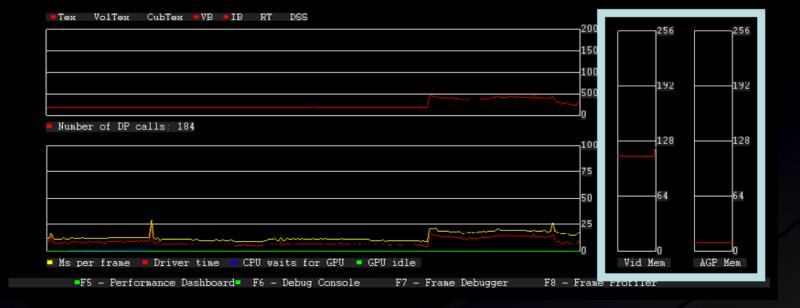


FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.NPogram Files/Futuremark/3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.



.





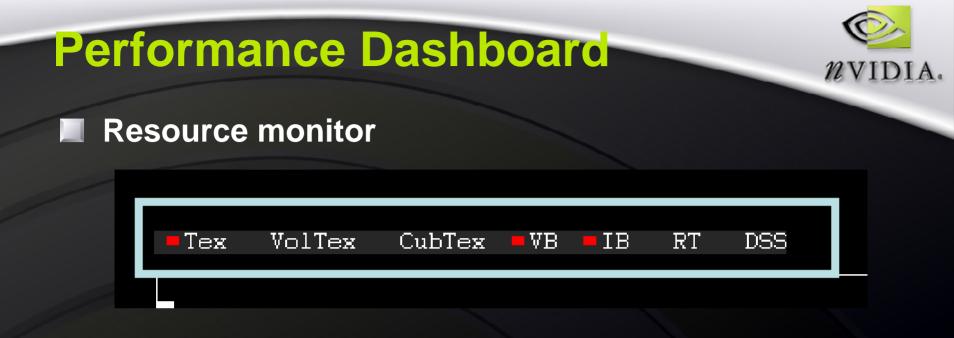
FPS: 52.3 TRIs∕Frame: 339400 Time: 28.7 secs Speed: ▶ 1.000 Press F1 for help

> NVPertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.NPogram Files/Futuremark/3DMark03.exe = Handshake with application OK. = WARNING: Forcing NON PURE device = DirectX *RETAIL* runtime detected. = : NVPMAPI found. enabling extended functionality.



1





Resources monitored

- **Textures**
- Volume Textures
- Cube textures
- Vertex Buffers
- Index buffers
- Stencil and depth surfaces





339400 Time: 28.7 secs

Press F1 for lelp

NVFertHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name. C.\Program Files\Futuremark\3DMark03.exe

Ilandshake with application OK.
 WARNING: Forcing NON PURE device
 DirectX *RETAIL* runtime detected.

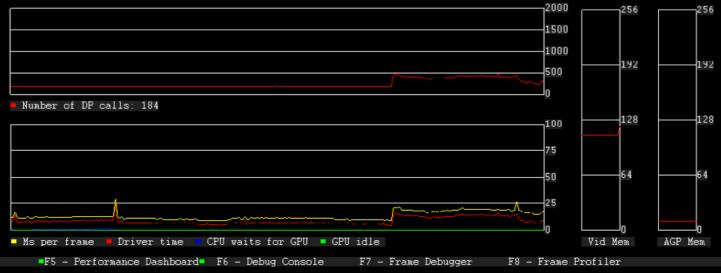
e

- NVPMAPI found, enabling extended functionality.



1

Tex VolTex CubTex VB IB RT DSS





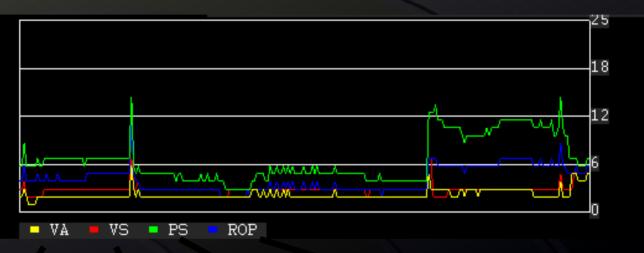
Speed control



NVPerfHUD version: 4.0.321.1500 NVIDIA driver version: 6.14.10.7772 App name: C:\Program Files\Futuremark\



The simplified graphics pipeline







Schedule

Beta: AugustRelease : September



NVPerfKit Performance Analysis Toolkit

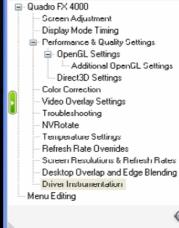
Complete Performance Instrumentation Solution

- Instrumented Driver
- NVIDIA Developer Control Panel (NVDevCPL)
- NVIDIA Plug-in for Microsoft PIX for Windows
- Direct access to performance counters via PDH
 - Support for PerfMon, Intel® VTune™, gDEBugger, and more
 - Access to performance signals inside your applications
- Includes code samples for OpenGL and Direct3D
- Opt-in security mechanism prevents unauthorized analysis



NVPerfKit Instrumented Driver

- Provides GPU and Driver Performance Counters
- Supports OpenGL and Direct3D
- Supports SLI Counters
- Requires GeForce FX or later
 - Significantly more counters available on GeForce 6 Series and later...



General Adapter Monitor Troubleshoot Color Management SQ Quadro FX 4000

Plug and Play Monitor and NVIDIA Quadro FX 4000 P...



0K

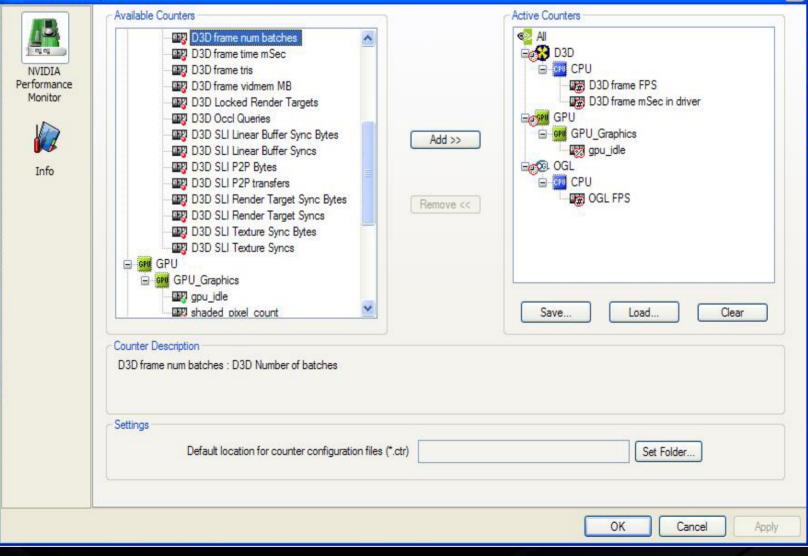
Cancel /

NVPerfKit NVIDIA Developer Control Panel



×

🧟 NVIDIA Developer Control Panel



NVPerfKit PDH Counters in PerfMon

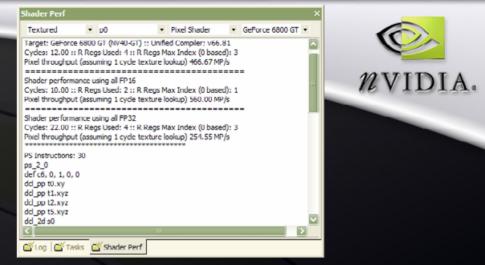


🙀 Performance							
i File Action ⊻iew Favorites ← → E III 2	<u>W</u> indow	<u>H</u> elp					_ & ×
Console Root		1 74 6) 🖸 🖬 🗎 ·	+ X 😵 🖻 🛍 🙆	2 🙁	\$ 2	
■ Werformance Logs and Alerts	100 80 60 40 20 0	Last	34.968 Ave	rage 62.919 M	linimum	23.944 Maximum Duration	
	Color	Scale	Counter	Instance	Parent	Object	Computer
		1.000 1.000 1.000 1.000 1.000	GPU Counter Value GPU Counter Value OGL Counter Value OGL Counter Value OGL Counter Value OGL Counter Value	% gpu_idle % primitive_count % OGL % driver waiting % OGL % pushbuf used OGL FPS OGL frame mSec sleeping OGL frame time mSec	GPU0 GPU0 CPU CPU CPU CPU CPU	NVIDIA GPU Performance NVIDIA GPU Performance NVIDIA OpenGL Driver NVIDIA OpenGL Driver NVIDIA OpenGL Driver NVIDIA OpenGL Driver NVIDIA OpenGL Driver	<pre>\\SDOMINE-P \\SDOMINE-P \\SDOMINE-P \\SDOMINE-P \\SDOMINE-P \\SDOMINE-P \\SDOMINE-P</pre>

PerfMon already shows CPI reported by the GPU hardwa irect3D Driver.

NVShaderPerf

Same technology as Shader Perf panel in FX Composer



Analyze DirectX and OpenGL Shaders
 HLSL, GLSL, Cg, !!FP1.0, !!ARBfp1.0, VS1.x, VS2.x, VS3.x, PS1.x, PS2.x, PS3.x, etc.

Shader performance regression testing on the entire family of NVIDIA GPUs, without rebooting!

NVShaderPerf	- 🗆 🗙
mad_pp r5, r3.x, r5, c0 lrp_pp r3, r2.x, r5, c2 lrp_pp r2, r0.x, r3, r4 lrp_pp r0, r1.x, r2, c2 mu1_pp r0, r0, v0 mov_pp oC0, r0	
// approximately 32 instruction slots used (6 texture, 26 arithmetic)	
Target: GeForce 6800 GT (NU40-GT) :: Unified Compiler: v66.81 Cycles: 14.00 :: R Regs Used: 3 :: R Regs Max Index (0 based): 2 Pixel throughput (assuming 1 cycle texture lookup) 400.00 MP/s	
C:\Program Files\NUIDIA Corporation\NUIDIA NUShaderPerf>nvshaderperf -v 0 -a 0 -t brix -p p0 shaders\brix.fx	a NV4
Running performance on file shaders\brix.fx ***********************************	****
Target: GeForce 6800 Ultra (NV40) :: Unified Compiler: v66.81 Cycles: 14.00 :: R Regs Used: 3 :: R Regs Max Index (0 based): 2 Pixel throughput (assuming 1 cycle texture lookup) 457.14 MP/s	
C:\Program Files\NUIDIA Corporation\NUIDIA NUShaderPerf>	-



Conclusion

Use high-level shading languages Use FX files and Semantics Either CgFX or D3D FX Use our tools Tons of free tools Tons of free examples **Treat Shaders like C++ code** Good design can save tons of time in making your game look amazing!



Questions

http://developer.nvidia.com

http://developer.nvidia.com/CgTutorial

Email: bdudash@nvidia.com

The Source for GPU Programming

developer.nvidia.com

- Latest News
- Developer Events Calendar
- Technical Documentation
- Conference Presentations
- GPU Programming Guide
- Powerful Tools, SDKs and more ...



VIDIA

Join our FREE registered developer program for early access to NVIDIA drivers, cutting edge tools, online support forums, and more.

developer.nvidia.com

©2004 NVIDIA Corporation. NVIDIA, and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation. Nalu is ©2004 NVIDIA Corporation. All rights reserved.