



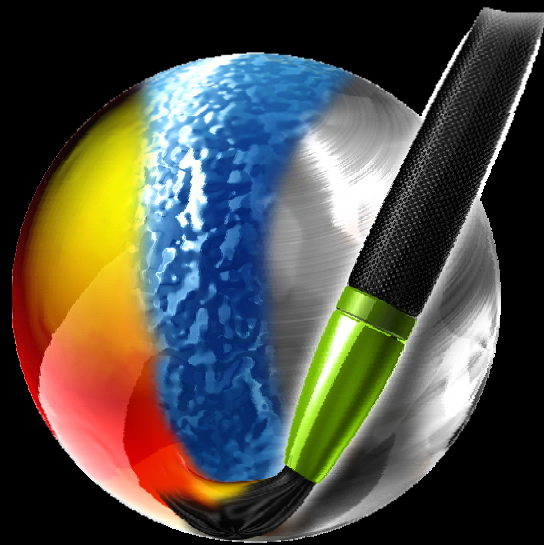
nVIDIA®

Developer Tools Showcase

Randy Fernando
Developer Tools Product Manager

NVISION 2008

FX Composer 2.5

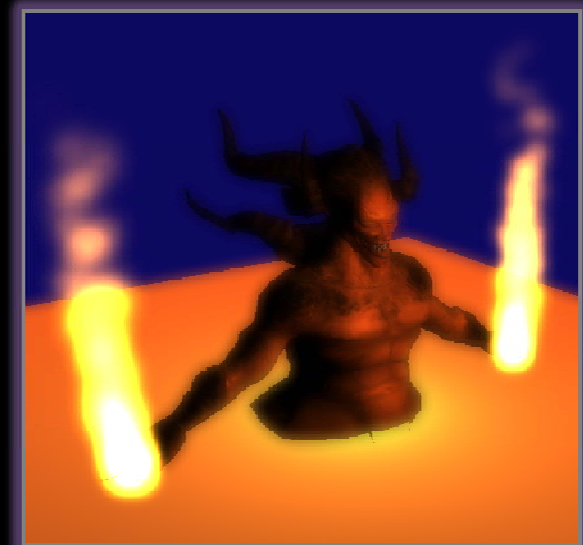


www.fxcomposer.com

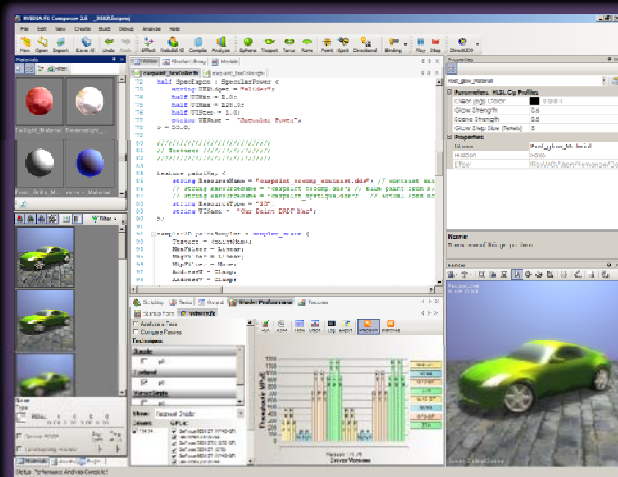
NVIDIA FX Composer 2.5

The World's Most Advanced Shader Authoring Environment

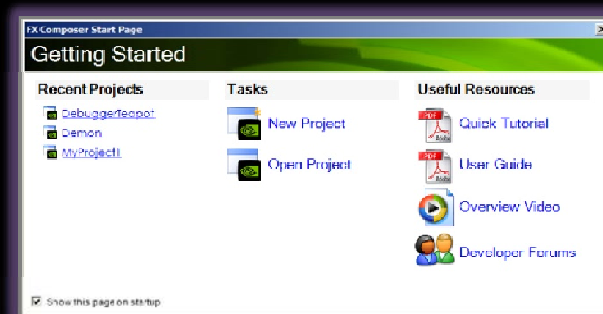
- DirectX 10 Support
- NVIDIA Shader Debugger Support
- ShaderPerf 2.0 Integration
- Visual Models & Styles
- Particle Systems
- Improved User Interface



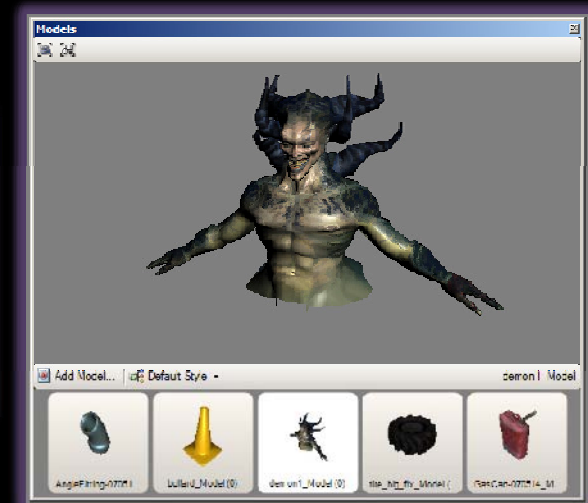
Particle Systems



350Z Sample Project

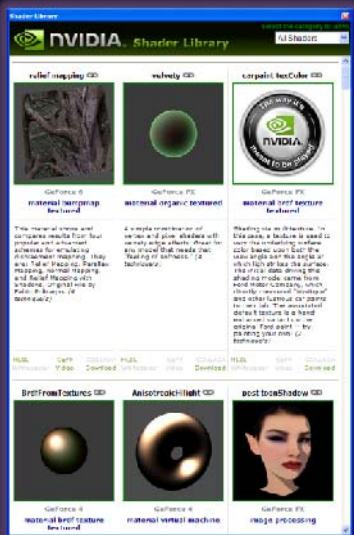


All-New Start Page

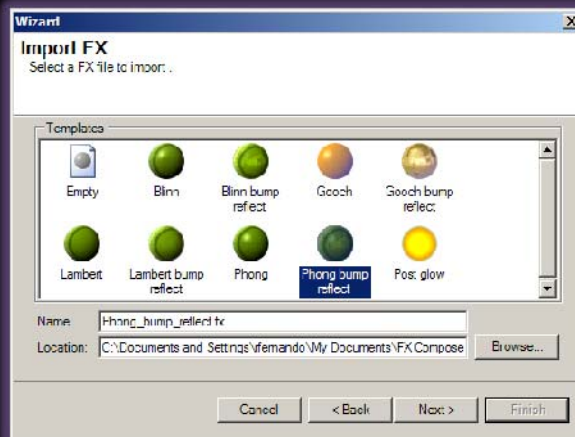


Visual Models & Styles

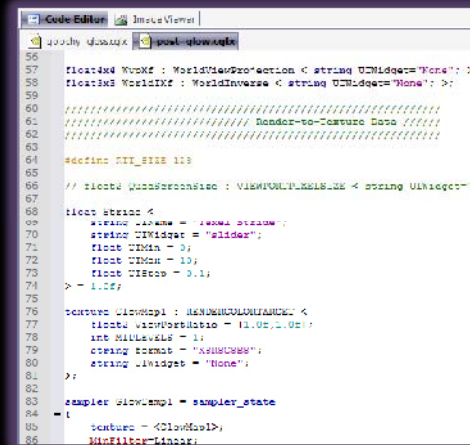
Other Major Features



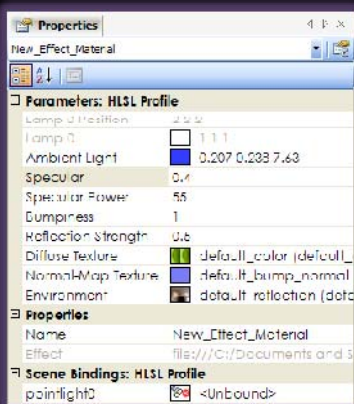
Shader Library
Hundreds of samples



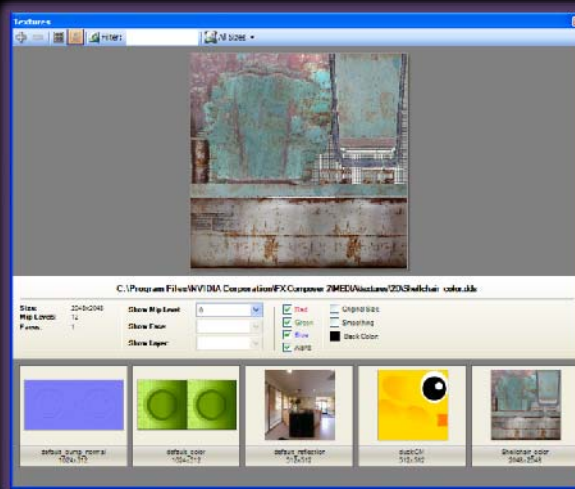
Shader Creation Wizard
Quickly create common shaders



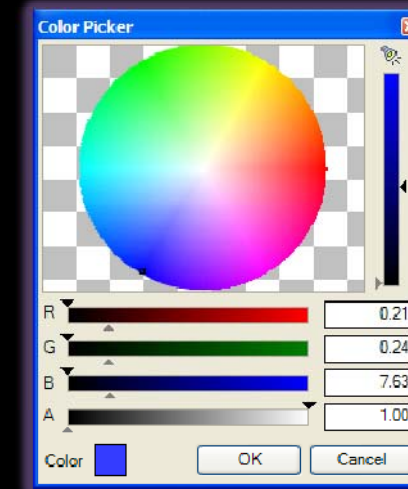
Code Editor
Full editor with assisted code generation



Properties Panel



Texture Viewer
View, organize, and apply textures



HDR Color Picker



Materials Panel

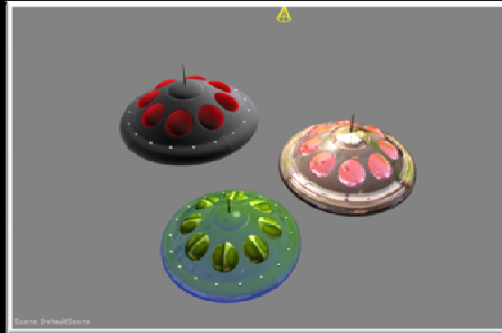
Even More Features

- **Automatic Light Binding**
- **Complete Scripting Support**
- **Support for DirectX 10**
(Geometry Shaders, Stream Out, Texture Arrays)
- **Support for COLLADA, .FBX, .OBJ, .3DS, .X**
- **Extensible Plug-in Architecture with SDK**
- **Customizable Layouts**
- **Semantic and Annotation Remapping**
- **Vertex Attribute Packing**
- **Remote Control Capability**

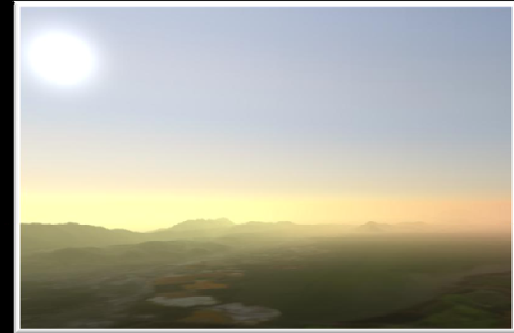
New Sample Projects



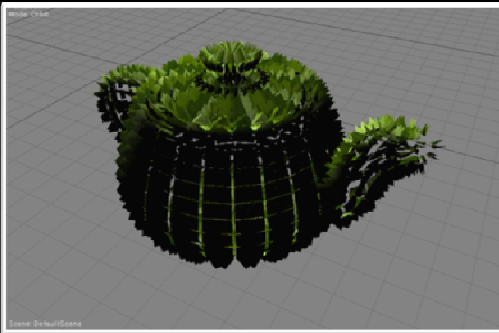
350Z



Visual Styles



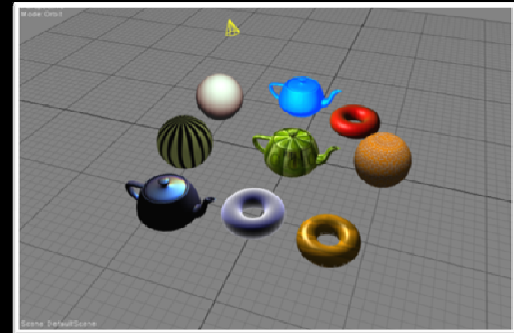
Atmospheric Scattering



DirectX 10



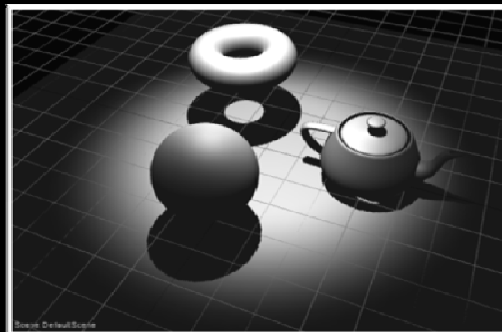
PCSS Soft Shadows



Materials



Post-Processing



Simple Shadows

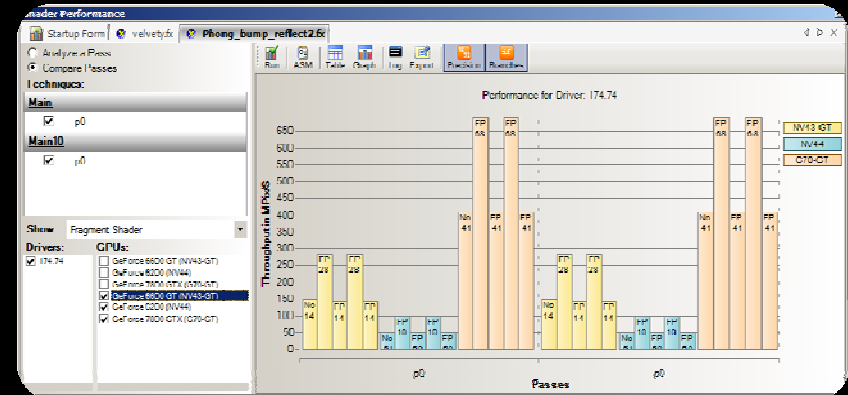


Ninja

ShaderPerf 2.0

Making shader performance tuning easy

- **New!** GeForce 8 Series support
- **New!** API for programmatic access
- **New!** Fragment program differencing
- Vertex and pixel shader analysis
(No OGL vertex shader analysis on GPUs older than GeForce 8 Series)
- Many supported shader formats
- Compare different techniques
- See shader throughput, register usage, instruction counts, and branching performance
- Integrated in FX Composer 2.5
- Also available as standalone command-line tool

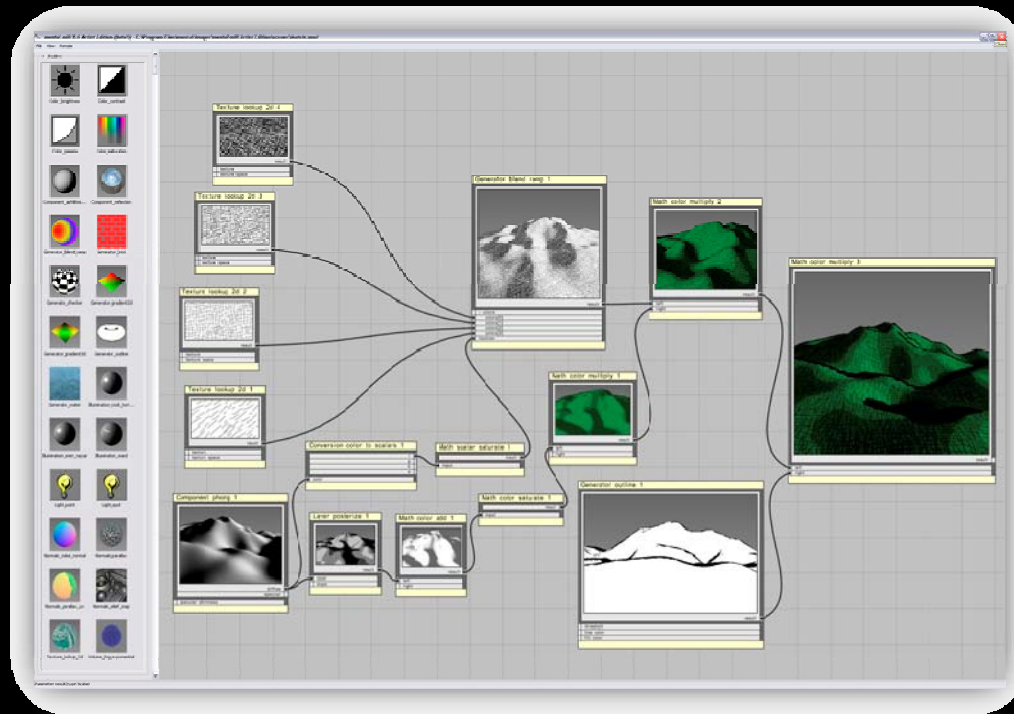


Performance for Driver: 174.74

GPU Model	Regs			Cycles			MIPs		
	Normal	FP16	FP32	Normal	FP16	FP32	Normal	FP16	FP32
Main 102p0									
NV43-GT (GeForce 6600 GT)	4	3	4	27	14	3.8	148	285	142
NV44 (GeForce 6200)	4	3	4	27	14	2.6	51	100	50
G70-GT (GeForce 7800 G1X)	1	2	4	26	15	7.5	112	738	417
Main 102p0									
NV43-GT (GeForce 6600 GT)	4	3	4	27	14	3.8	148	285	142
NV44 (GeForce 6200)	4	3	4	27	14	2.6	51	100	50
G70-GT (GeForce 7800 G1X)	4	3	4	26	16	2.6	412	688	412

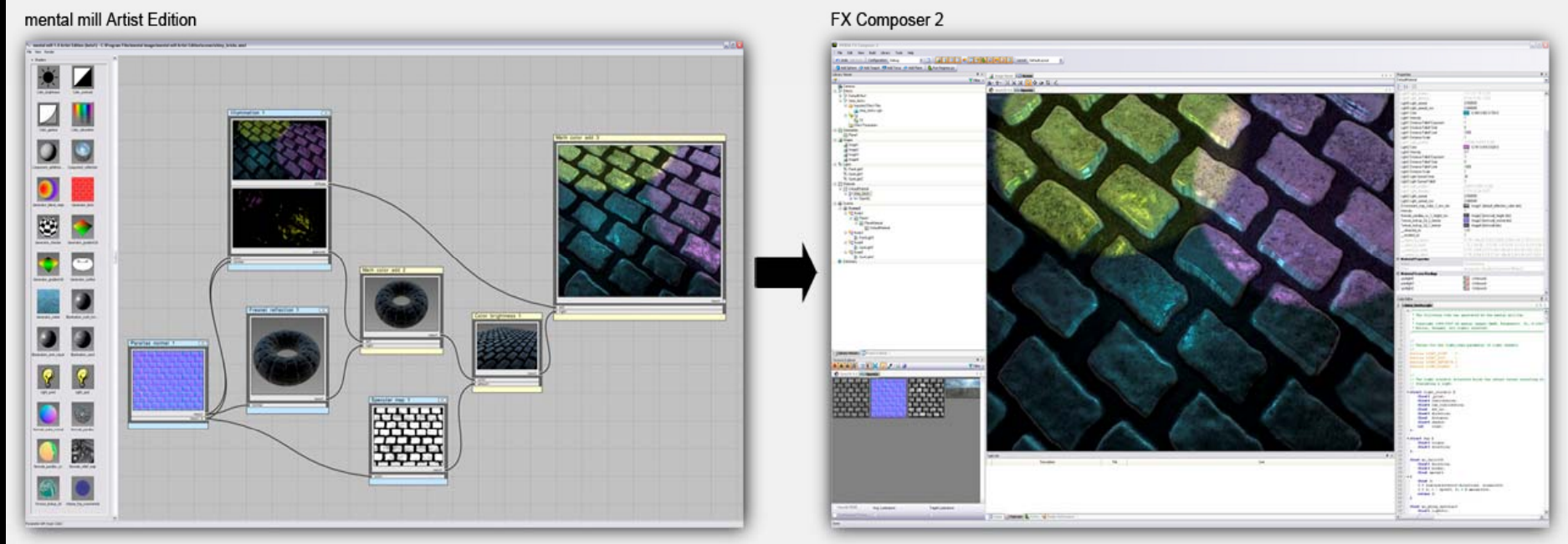
mental mill™ Artist Edition

- Enables artists to quickly and easily develop and prototype new looks within a fast, flexible GUI.

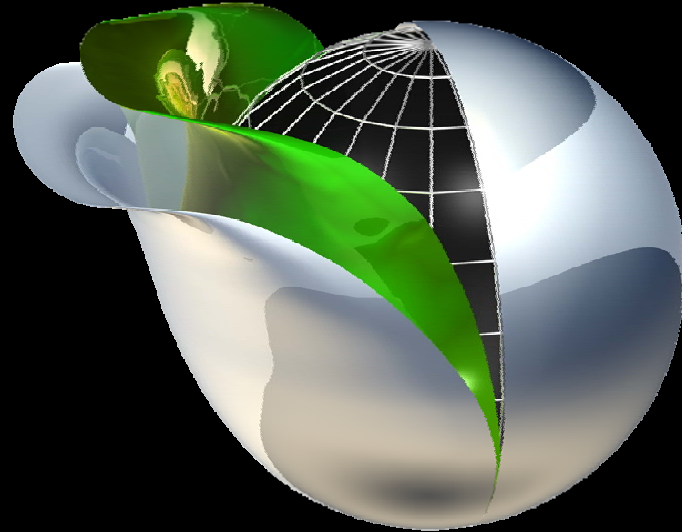


mental mill Artist Edition and FX Composer

- **The best of both worlds.** More power for artists plus optimization abilities for programmers and shader specialists.
- **Smooth interoperability with FX Composer.** A shader exported from mental mill can be seamlessly imported into FX Composer.
- **Better collaboration.** Dramatically reduced iteration time.



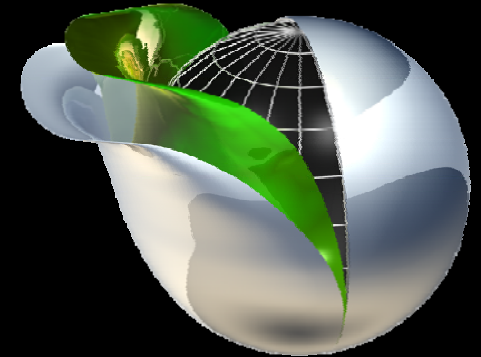
The NVIDIA Shader Debugger



www.shaderdebugger.com

NVIDIA Shader Debugger

Visualize your shaders, step by step



Broad Language Support

• HLSL10/9

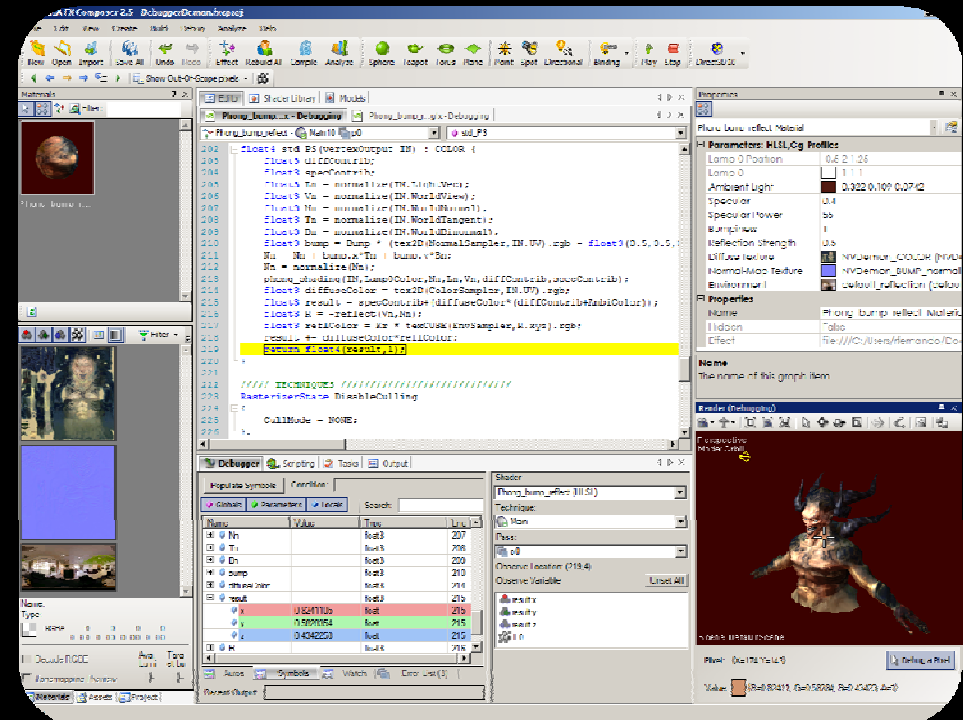
• CgFX

• COLLADA FX Cg

Step through shader source code

Visualize variables across your geometry

Plug-in for FX Composer 2.5



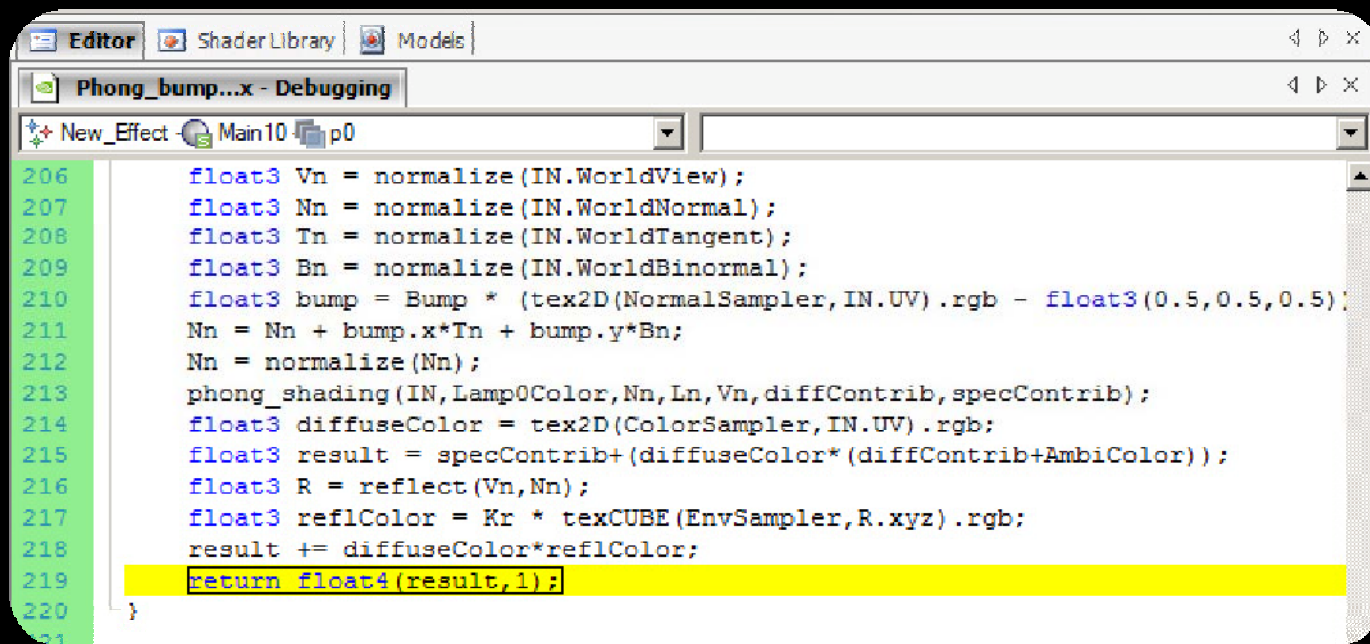
Run Control

Full control of shader execution

Next/Previous Statement

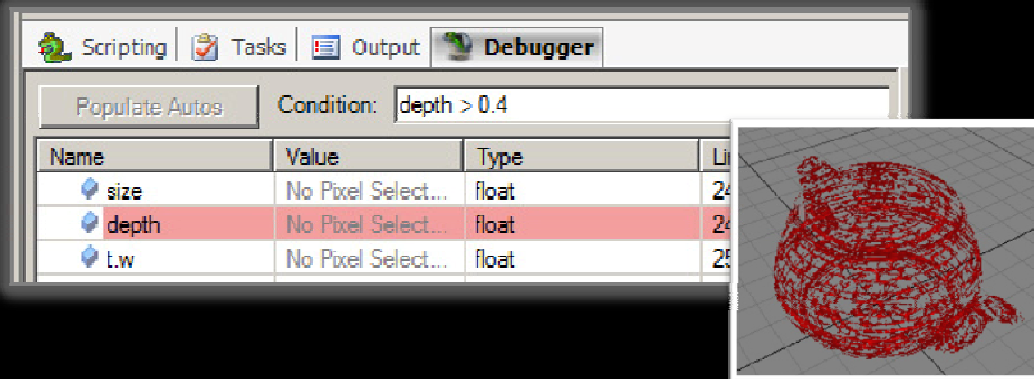
Run to Cursor

Run to Bookmark



```
206 float3 Vn = normalize(IN.WorldView);
207 float3 Nn = normalize(IN.WorldNormal);
208 float3 Tn = normalize(IN.WorldTangent);
209 float3 Bn = normalize(IN.WorldBinormal);
210 float3 bump = Bump * (tex2D(NormalSampler, IN.UV).rgb - float3(0.5, 0.5, 0.5));
211 Nn = Nn + bump.x*Tn + bump.y*Bn;
212 Nn = normalize(Nn);
213 phong_shading(IN, Lamp0Color, Nn, Ln, Vn, diffContrib, specContrib);
214 float3 diffuseColor = tex2D(ColorSampler, IN.UV).rgb;
215 float3 result = specContrib + (diffuseColor * (diffContrib + AmbiColor));
216 float3 R = reflect(Vn, Nn);
217 float3 reflColor = Kr * texCUBE(EnvSampler, R.xyz).rgb;
218 result += diffuseColor * reflColor;
219 return float4(result, 1);
220 }
```

More Shader Debugger Features

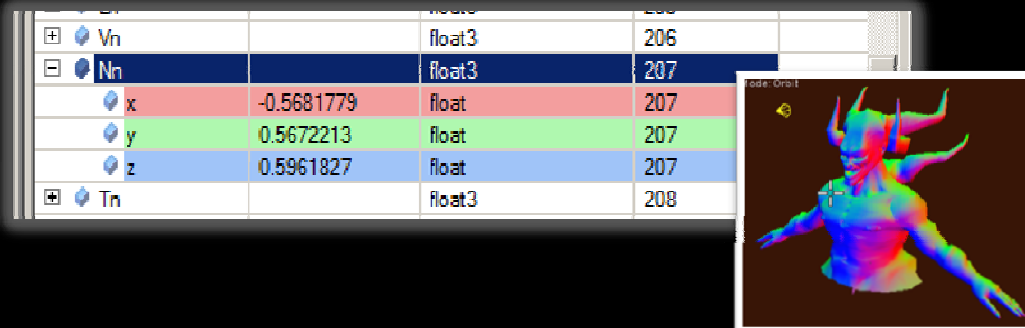


Populate Autos Condition: `depth > 0.4`

Name	Value	Type	Line
size	No Pixel Select...	float	24
depth	No Pixel Select...	float	24
t.w	No Pixel Select...	float	25

A 3D view of a teapot with a red wireframe overlay, indicating the result of a conditional fragment kill.

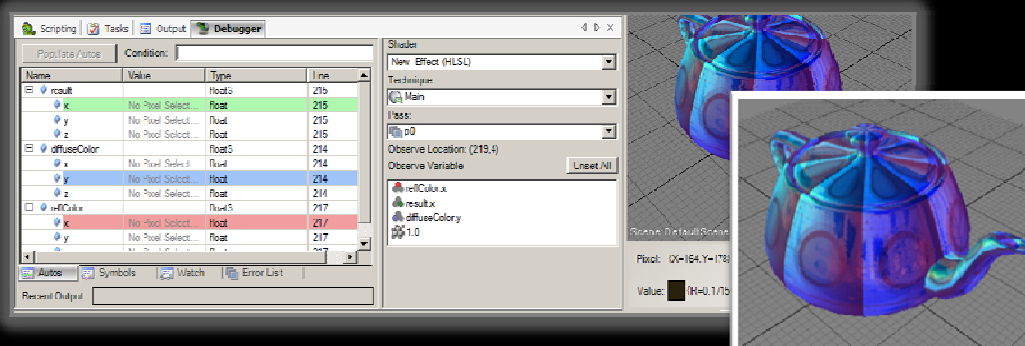
Conditionally Kill Fragments



Vn		float3	206
Nn		float3	207
x	-0.5681779	float	207
y	0.5672213	float	207
z	0.5961827	float	207
Tn		float3	208

A 3D view of a character model with a color gradient overlay, representing the visualization of a variable's value across the scene.

Visualize Any Variable



Populate Autos Condition:

Name	Value	Type	Line
result		float3	210
x	No Pixel Select...	float	215
y	No Pixel Select...	float	215
z	No Pixel Select...	float	215
diffuseColor		float3	214
x	No Pixel Select...	float	214
y	No Pixel Select...	float	214
z	No Pixel Select...	float	214
diffColor		float3	217
x	No Pixel Select...	float	217
y	No Pixel Select...	float	217

Shader: New Effect (HLSL) Technique: Main Pass: P0 Observe Location: (215.4) Observe Variable: Inset All

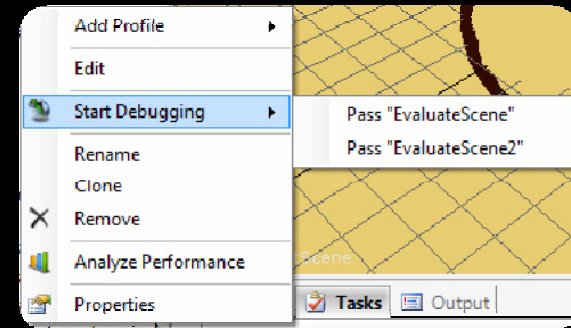
Pixel: (0=164,1=178) Value: (R=0,1/G=0,1/B=0)

A 3D view of a teapot with a blue and purple color mapping, representing a custom output mapping.

Create Custom Output Mappings

More Shader Debugger Features

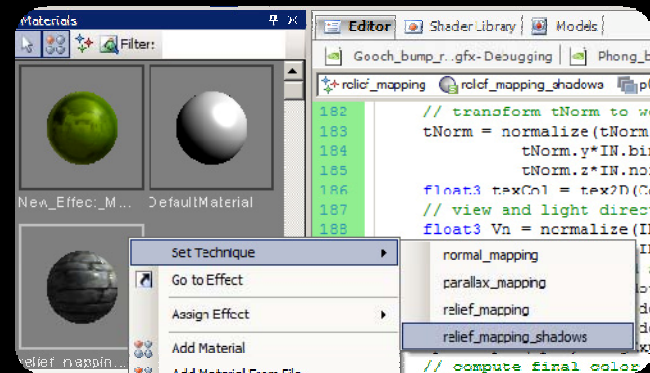
Debug multi-pass/full scene effects.



Vn		float3	206	
Nn		float3	207	
x	-0.5681779	float	207	
y	0.5672213	float	207	
z	0.5961827	float	207	
Tn		float3	208	

Examine pixels individually or in parallel.

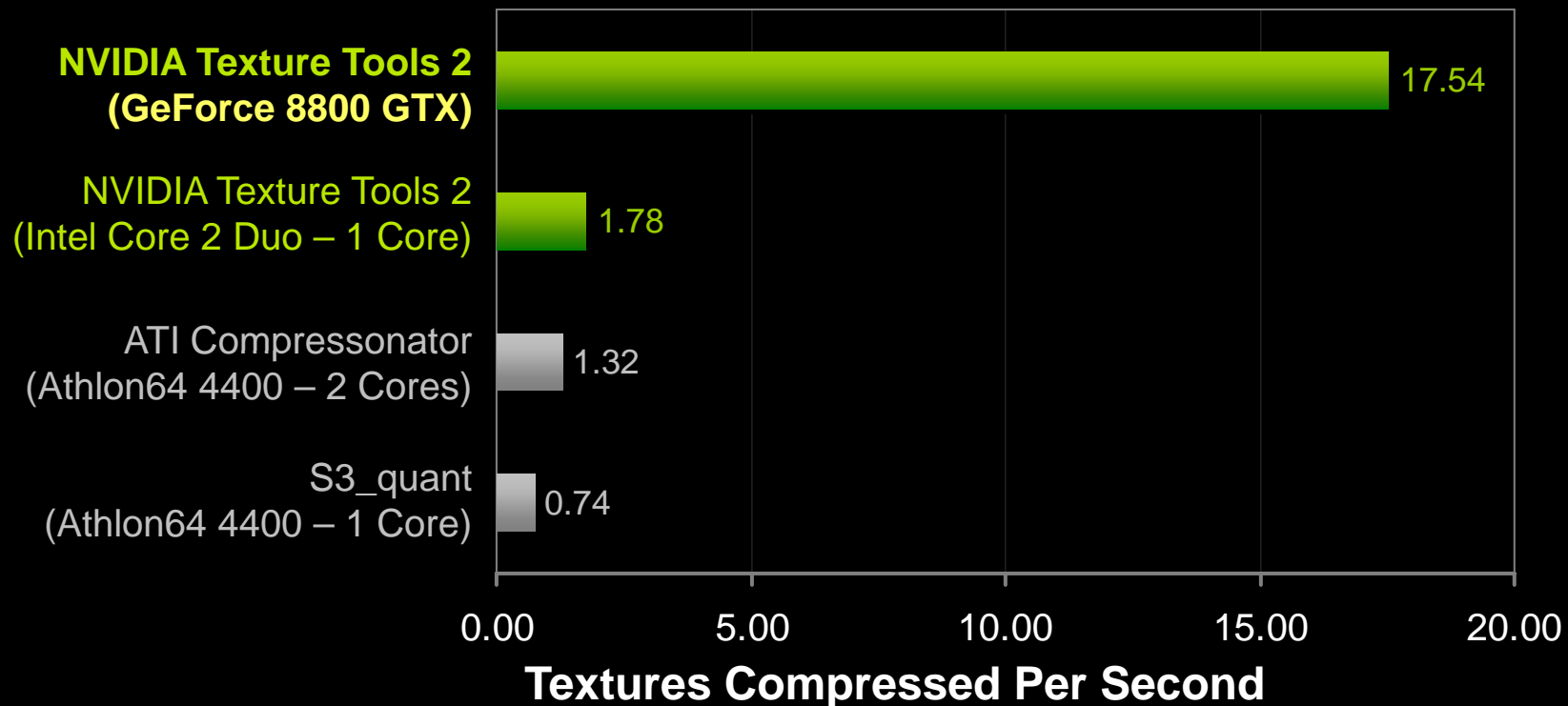
Pick any technique in your shader to debug.



GPU-Accelerated Texture Tools

10x faster, high-quality texture compression

- GPU-accelerated via CUDA
- Support for DirectX 10 texture formats
- Includes complete source code
- Amazing performance without sacrificing quality



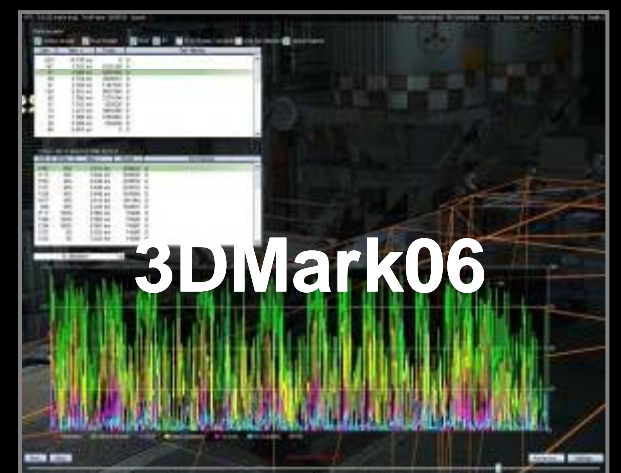
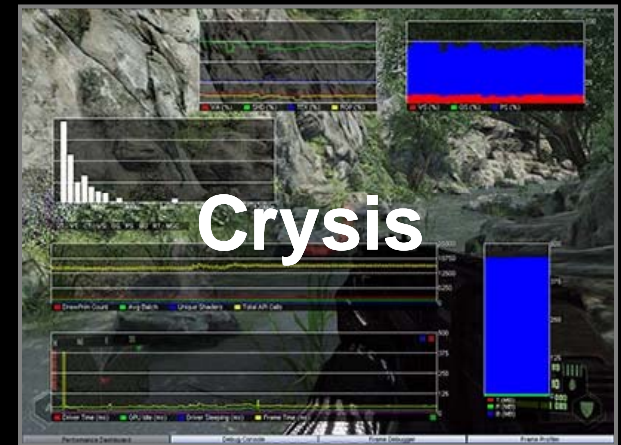


NVIDIA PerfKit 6

NVIDIA PerfHUD 6.1

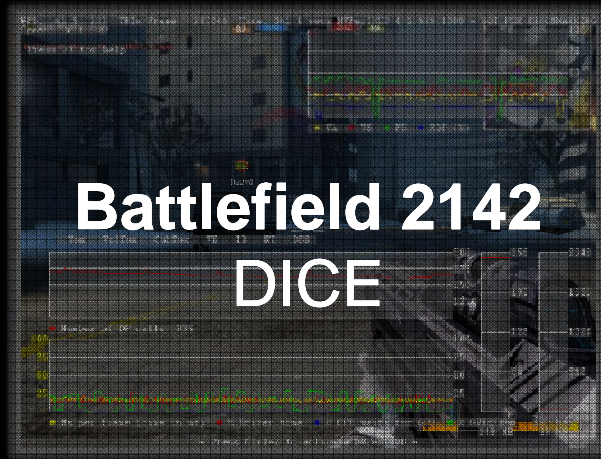
More convenient and powerful than ever

- G8X & GT200 support
- Comprehensive SLI support
- No More “Instrumented Driver”
- Frame Capture (coming soon)
- Advanced texture and render target debugging & analysis
- API Data Mining & Analysis
- Debug Break capability
- Many other convenient features...



Unreal Tournament 3 is copyright © 2007 Epic Games, Inc, Cary, N.C., USA. ALL RIGHTS RESERVED. Epic, Unreal, and Circle U logo are registered trademarks of Epic Games, Inc. in the United States of America and elsewhere. 3DMark06 used with permission from Futuremark corporation. 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.

Top Game Developers Use PerfHUD (Games shown with PerfHUD running on them)



Over 300 surveyed PerfHUD 5 users reported an **average speedup of 37%**, and as much as **400%!**

NVIDIA PerfKit 6: The World's Most Advanced GPU Performance Suite

PerfHUD 6

- New!** GeForce 8800GT, 9600GT, 9800GX2, GTX 280 support

- New!** Use stock NVIDIA drivers with PerfHUD (on Windows Vista)

- New!** Multi-GPU Support

- New!** More real-time signals, including SLI signals

- New!** API Call List

- New!** Draw Call Dependency Analysis

- New!** Advanced Texture Visualization

- New!** Texture Overrides

- Real-time performance analysis and debugging

- Automated bottleneck determination

PerfSDK

- New!** GeForce 8800GT, 9600GT, 9800GX2, GTX 280 support

- API for accessing GPU and driver counters

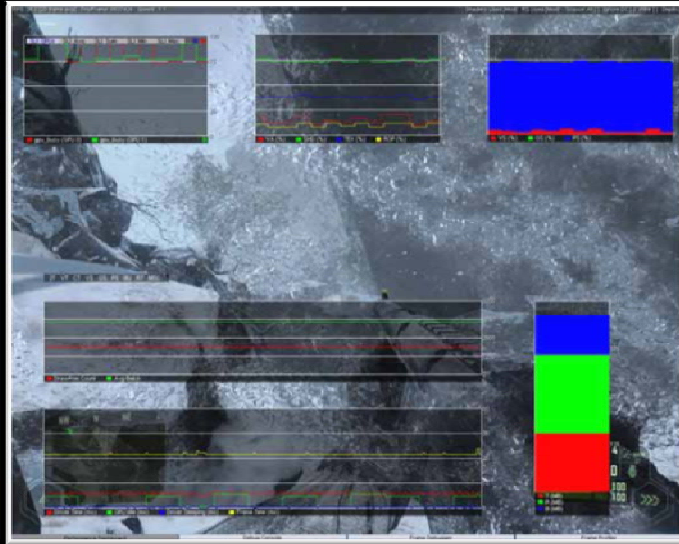
- Supports DirectX 9 & 10, as well as OpenGL

- NVIDIA Plug-in for Microsoft PIX for Windows

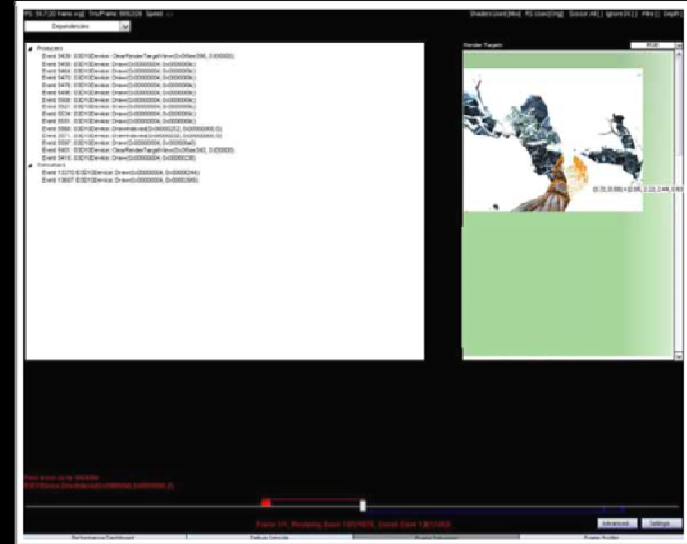
- GLExpert



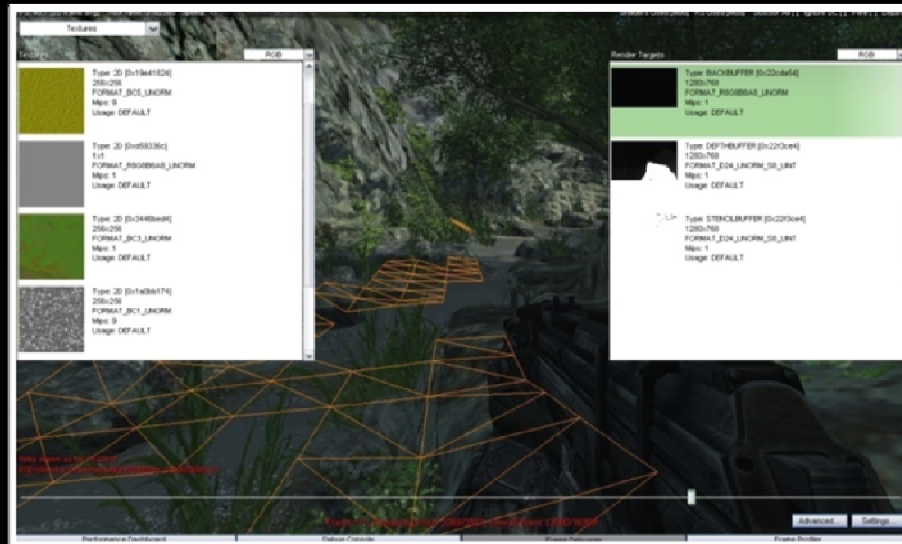
PerfHUD 6



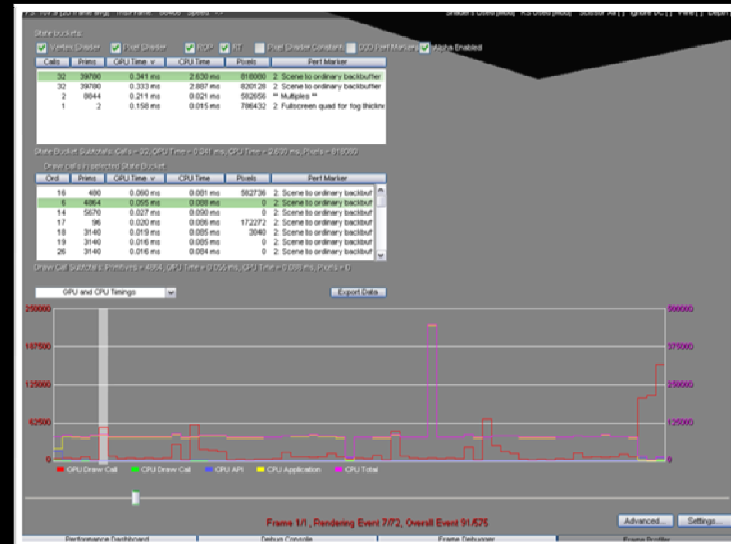
Multi-GPU Support



Advanced Data Mining



Powerful Texture Visualization



Combined CPU/GPU Timing Graph

Tons of other improvements!

- **Multiple, named Performance Dashboard Layouts**

- Save separate layouts for each of your monitoring scenarios.

- **Sampler State Editing**

- Edit and override any property of a sampler in your application

- **New Real-time Experiments**

- Minimize Geometry (Replaces all geometry with a single primitive)

- **Improved Compatibility and Stability**

- Rewritten interception layer requires less cooperation from the application.
- Now compatible with applications that include frame-limiting logic.
- Now compatible with applications which send non-deterministic graphics workloads.
- Many bug fixes and performance improvements.

- **Improvements to Frame Profiler**

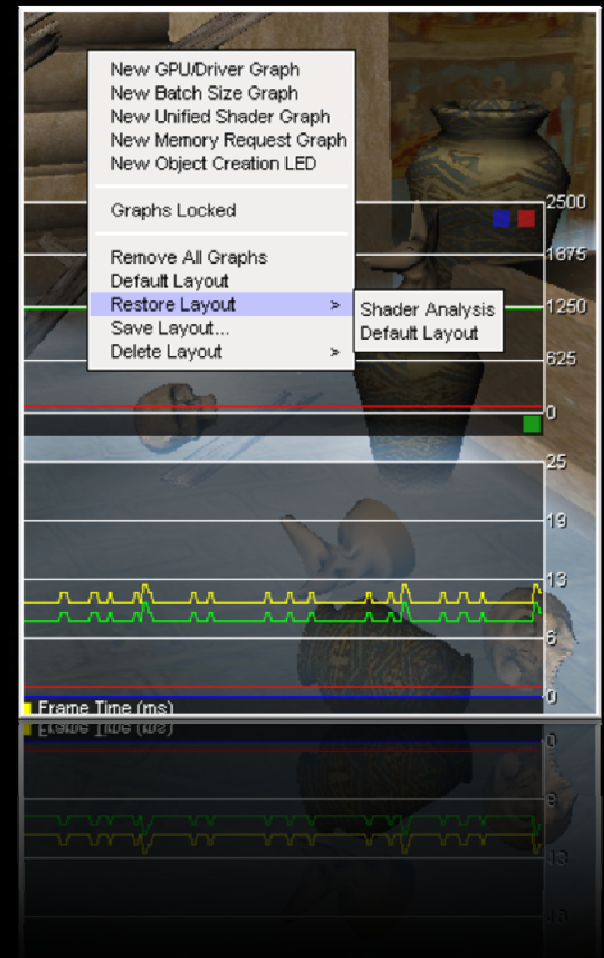
- New CPU/GPU Timings Graph lets you directly see and compare utilization in the CPU, the driver, and the GPU.

- **User Interface Improvements**

- Navigation hotkeys for every major tool

- **Compatibility, stability, and reliability improvements**

- Extensive testing on a wide range of applications
- Minor bug fixes



How to Think of PerfHUD's Main Modes

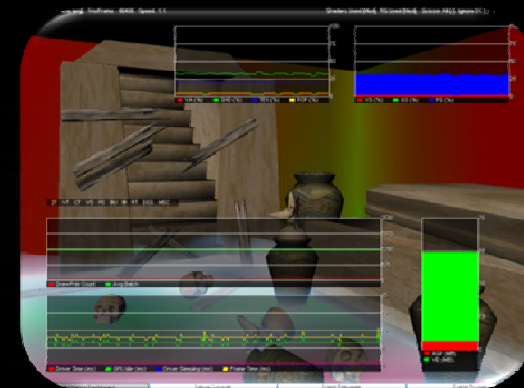
- **Performance Dashboard**
Real-time Holistic Analysis
- **Frame Debugger**
Rendering Debugging
- **Advanced Screens**
Shader and Render State Inspection and Modification
- **Frame Profiler**
Automated Bottleneck Detection
Per-Draw Call Performance Graphs

New! Advanced Texture Overrides

- **Override any texture in your application with a variety of useful texture overrides**
- **2x2 Texture: Reduces texture bandwidth usage by using the smallest texture possible.**
- **Black, 25% Gray, 50% Gray, 75% Gray, White, Horizontal gradient, Vertical Gradient: Each of these can be useful as debug input to your shaders.**
- **Color Mipmap Texture: Visualize your mip levels quickly and easily**
- **Use with shader edit and continue to quickly diagnose and correct bugs.**



Texture Context Menu



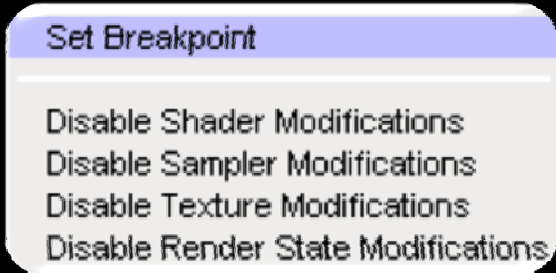
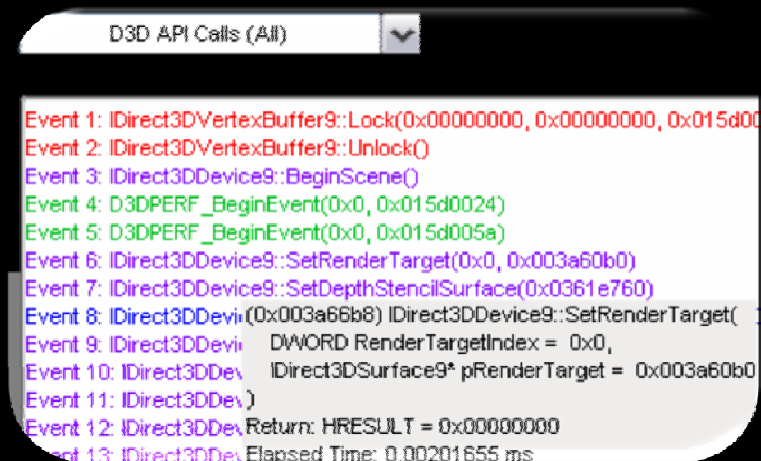
Mipmap Visualization

New! API Call List and Perf Event View

New! API Call List

A full list of all D3D API calls, colored by call type

One click jumps you to the corresponding draw call.



New! Perf Event List

Navigate your scene using D3D Perf Events to annotate sections of the frame.

Use Perf Events to selectively disable state overrides for portions of the scene.

Set a debug break (`_int 3`) to occur on a Perf Event to help debug your CPU code.

Shader Edit and Continue

- **Edit & Continue for:**
 - DirectX 9 HLSL and .fx
 - DirectX 10 HLSL and .fx
 - Vertex, geometry, and pixel shaders
- **Code editing**
 - Standard keyboard and mouse interaction
 - Search bar
- **Easily toggle between original and modified shaders**
 - Discard Current Edits
 - Restore Original Shader
 - Controlled via hotkey

```
{
    Sparkles_PSOut output;
    output.color = input.alpha * (texture_star.Sample(
    return output;
})

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// TECHNIQUES TECHNIQUES TECHNIQUES TECHNIQUES TECHNIQ
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

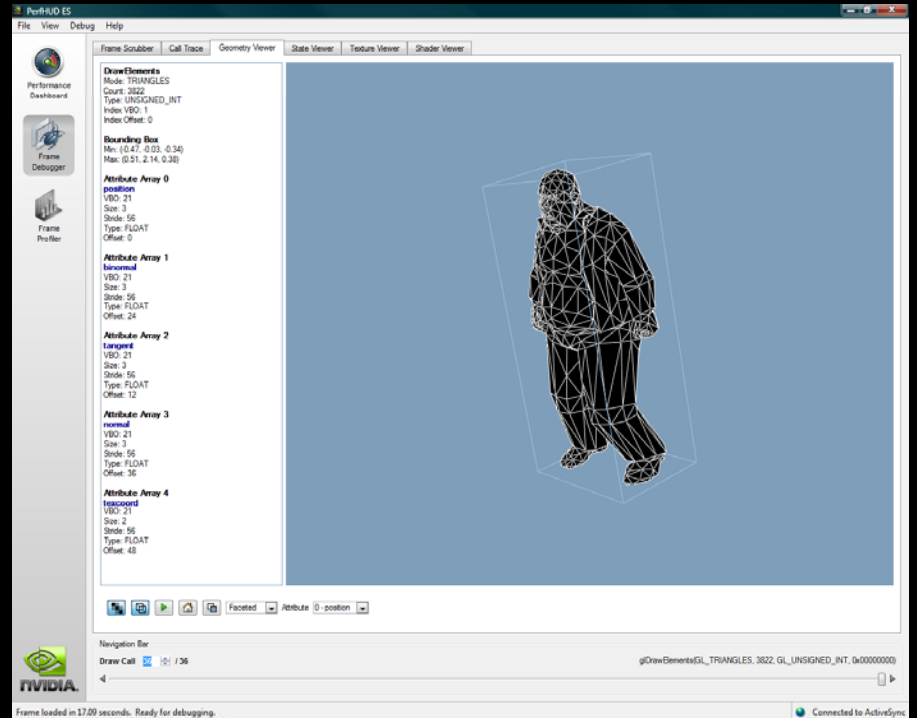
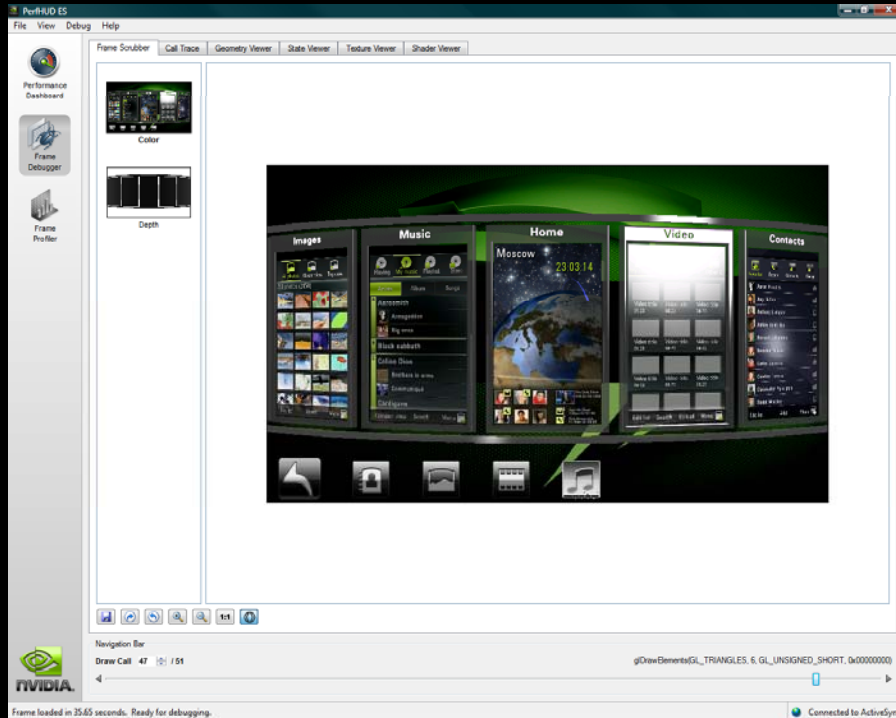
technique10 Sparkles_And_CarPaint
{
    pass ColorNDepth
    {
        SetBlendState(blendOFF, float4(1.0, 1.0, 1.0,
        SetRasterizerState( RStateMSAA );
        SetDepthStencilState( depthEnabled, 0 );
        SetVertexShader( CompileShader( vs_4_0, carpai
        SetGeometryShader(NULL);
        SetPixelShader( CompileShader( ps_4_0, carpain
        paintamp,
        paintamp
```

Editing Shaders

How is PerfHUD Different from Other Performance Tools?

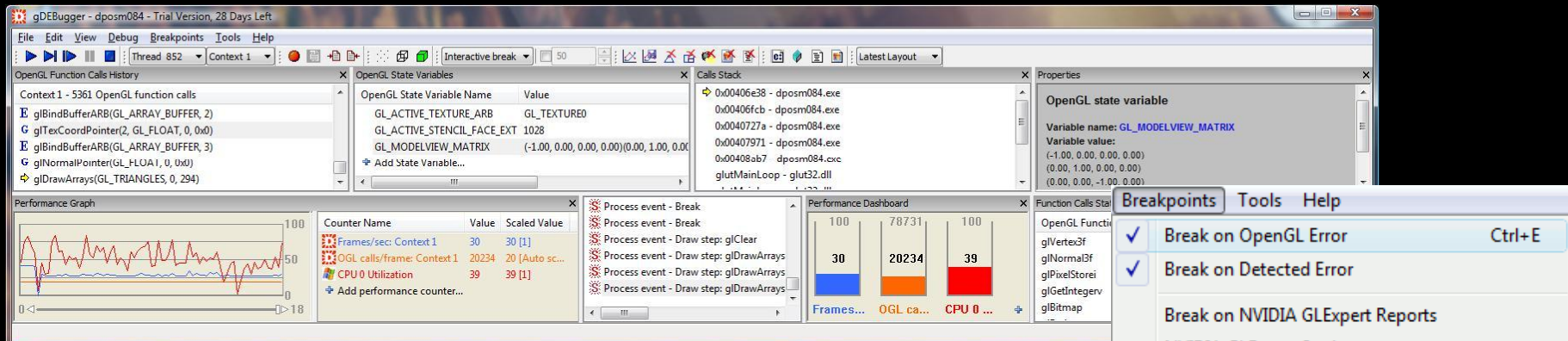
- **It works in real-time on your application.** Other tools require out of context, offline analysis. PerfHUD allows you to debug and tune your application in the most natural place: within your application!
- **One key press provides a list of draw calls (grouped by bottleneck and sorted by duration) to work on.** Running experiments and collecting data from individual pipeline units is difficult and easy to misjudge. PerfHUD automatically tells you exactly where your bottlenecks are so you can fix them quickly.
- **Real-time frame scrubbing.** Modern engines have thousands of draw calls per frame. PerfHUD lets you decompose the scene, stepping through each draw call to find any problems.
- **Edit-and-continue.** Modifying shader code and render states can be time consuming. PerfHUD allows you to make changes while your application is running – allowing you to quickly try our ideas and get immediate feedback.

PerfHUD ES



Similar functionality to PerfHUD PC for OpenGL ES
(For example, for Tegra development)

Graphic Remedy's gDEBugger



The screenshot displays the gDEBugger application window with several panels open. The 'OpenGL Function Calls History' panel shows a list of function calls such as `glBindBufferARB` and `glDrawArrays`. The 'OpenGL State Variables' panel lists variables like `GL_ACTIVE_TEXTURE_ARB` and `GL_MODELVIEW_MATRIX`. The 'Performance Graph' panel shows a line graph of performance metrics. The 'Performance Dashboard' panel displays a table of performance data:

Counter Name	Value	Scaled Value
Frames/sec: Context 1	30	30 [1]
OGL calls/frame: Context 1	20234	20 [Auto sc...]
CPU 0 Utilization	39	39 [1]

The 'Breakpoints' panel is open, showing options to break on OpenGL errors and detected errors. The 'NVIDIA GLEExpert Settings' dialog is also visible, with the following settings:

- Enable GLEExpert Integration:
- Break on GLEExpert Reports:
- Report messages from:
 - Vertex Unit:
 - Geometry Unit:
 - XFB Unit:
 - Raster Unit:
 - Fragment Unit:
 - ROP Unit:
 - Framebuffer Unit:
 - Pixel Unit:
 - Texture Unit:
 - Texture Objects:
 - Program Objects:
 - Framebuffer Objects:
 - Buffer Objects:
 - SLI:
- Message Detail:
 - OpenGL Errors:
 - Software Fallback:
 - Basic Information:
 - Detailed Information:
 - Performance Warning:
 - Quality Warnings:
 - Usage Warnings:

- OpenGL and OpenGL ES Debugger and Profiler
- NVIDIA PerfKit and GLEExpert integrated
- Shorten development time
- Improve application quality
- Optimize performance
- Find redundant function calls
- Supports Windows & Linux
- Discounted academic licenses available

<http://www.gremedy.com>

Graphic Remedy's gDEDebugger

The screenshot displays the gDEDebugger application interface, which is used for debugging graphics. It is divided into several main sections:

- Textures and Buffers viewer:** This is the largest window, showing a list of textures and buffers on the left. The main area contains a grid of data for a selected texture (2D Tex: 6 (unit 2, bound 2D)). The grid has columns labeled 18 through 29 and rows labeled with texture names and their components (e.g., Blue, Green, Red, Alpha). A 'Data view' window is overlaid on the grid, showing a detailed view of the data for a specific pixel. The 'Image view' on the right shows a preview of the texture data.
- Properties View:** Located at the bottom left, it shows the properties of the selected texture, including its dimensions (Width: 512 px, Height: 512 px), format (GL_RGBA), and type (GL_UNSIGNED_SHORT).
- Static Buffers:** A list of static buffers is shown on the left side of the main window. The selected buffer is 'Depth Buffer'. Its properties are shown in the 'Properties View' at the bottom, including its dimensions (Width: 348 px, Height: 348 px) and format (GL_DEPTH_COMPONENT).
- Current Pixel:** At the bottom of the main window, the 'Current Pixel' is highlighted in green. Its position is (x: 155, y: 67) and its value is 0.54201056. The 'Previous Pixel' is at (x: 112, y: 62) with a value of 0.30485824.
- Adjust 'Depth Channel' active range:** A slider at the bottom right allows adjusting the active range of the depth channel, currently set to 0 min and 1 max.

SDK 10.5

The latest DirectX and OpenGL code samples

- Teaches developers how to take advantage of new DirectX 10-class GPUs
- Browser with clear code samples, detailed whitepapers, and videos



NVIDIA SDK 10.5
developer.nvidia.com

Search: GPU: Any

Direct3D Samples (13/19)

Category	GPU	Actions
Clippmaps Clippmaps are a feature first implemented on SGI workstations that allow mapping extremely high resolution textures to terrains. The original SGI implementation required highly specialized, custom hardware. The advanced features of the NVIDIA® GeForce® 8800 now permit the same algorithm using consumer hardware.	GeForce 8	Video Whitepaper Files Run
HDRRendering Demonstrates HDR rendering on the GeForce 8800 and DirectX10. The GeForce 8800 supports new features to facilitate next-gen HDR. There are new texture and render target formats, such as R11G11B10F, which has the same memory requirements as standard 32-bit RGBA. The sample also demonstrates anti-aliasing combined with HDR, including high-quality 16x Coverage Sample Anti-Aliasing.	GeForce 8	Video Files Run
Cloth Simulation This sample demonstrates how to simulate cloth on the GPU using DirectX 10. The cloth vertex positions are computed in several rendering passes by looping through the vertex and geometry shader stages using the stream output stage to stream the positions out of the geometry stage.	GeForce 8	Video Whitepaper Files Run
SolidWireframe Render the wireframe of a mesh using filled triangles, coloring only the fragments near the edges of the triangle. This technique produce high quality thick embossed lines on the mesh edges, without any z fighting issues. This technique is a DirectX10 implementation of the SIGGRAPH 2005 sketch "Single-Pass Wireframe Rendering". This technique can be easily added in an	GeForce 8	Video Whitepaper

Selected SDK 10.5 Direct3D Samples



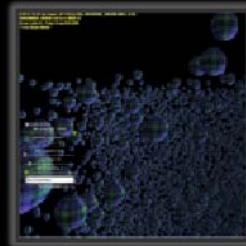
Tessellation



Screen-Space
Ambient Occlusion



PCSS
(Soft Shadows)



Instancing Tests



Parallax Mapping



Volume Light



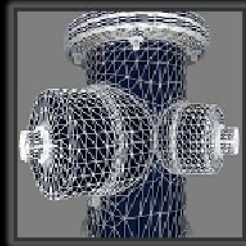
Stencil-Routed
K-Buffer



Advanced Skinning



Clipmaps



HQ Wireframe



Texture Arrays



MetaBalls



Smoke



Skinned Instancing



HDR Rendering



Soft Particles



Denoising



CSAA



Lightning



Rain



Fur



GPU Blendshapes



Perlin Fire



Soft Shadows

Selected SDK 10.5 OpenGL Samples



Tessellation



Cascaded Shadow
Maps



Dual Depth Peeling



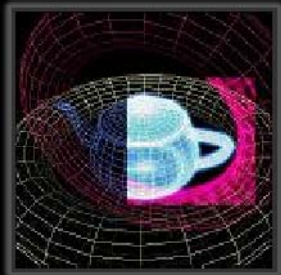
Compress
Normal-DXT



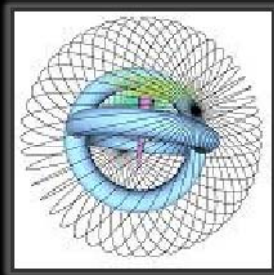
Compress YCoCg-
DXT



Texture Buffer
Object



Glow



HQ Antialiasing



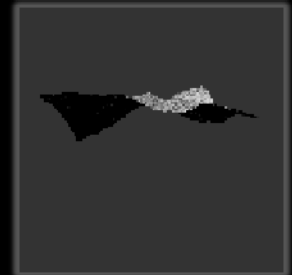
Christmas Tree



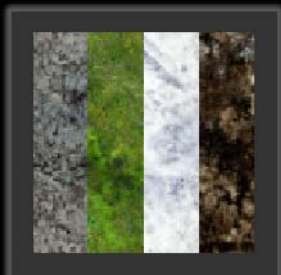
Render to 3D
Texture



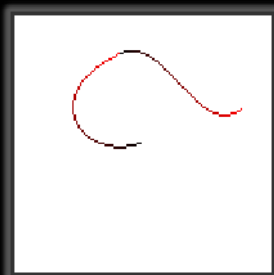
Render to 3D
Texture



Transform-
Feedback Fractal



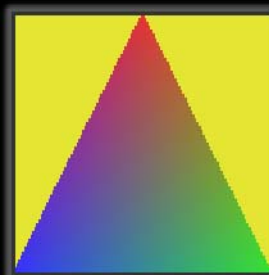
Texture Array



Geometry
Program



Framebuffer
Object



Depth Float



HDR



Isosurface

CUDA Toolkit & SDK

- Complete software development solution for programming CUDA-enabled GPUs

- Includes:

- Standard FFT and BLAS libraries
- C-compiler for the NVIDIA GPU
- Runtime driver

- SDK Samples:

- Parallel bitonic sort
- Matrix multiplication
- Matrix transpose
- Performance profiling using timers
- Parallel prefix sum (scan) of large arrays
- Image convolution
- 1D DWT using Haar wavelet
- OpenGL and Direct3D graphics interoperation examples

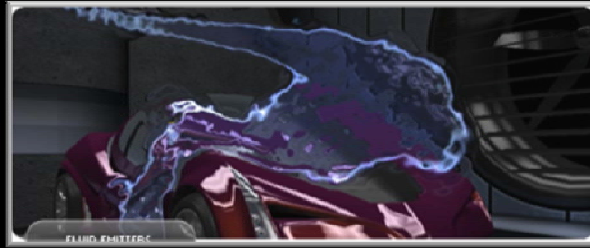


- CUDA BLAS and FFT library usage examples
- CPU-GPU C- and C++-code integration
- Binomial Option Pricing
- Black-Scholes Option Pricing
- Monte-Carlo Option Pricing
- Parallel Mersenne Twister (random number generation)
- Parallel Histogram
- Image Denoising
- Sobel Edge Detection Filter

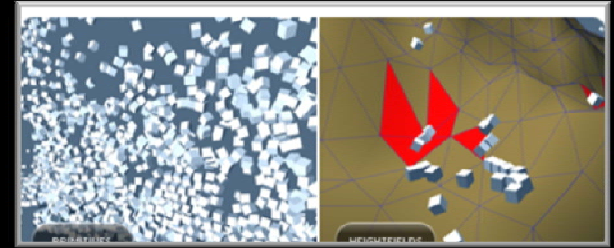
NVIDIA PhysX SDK



Cloth



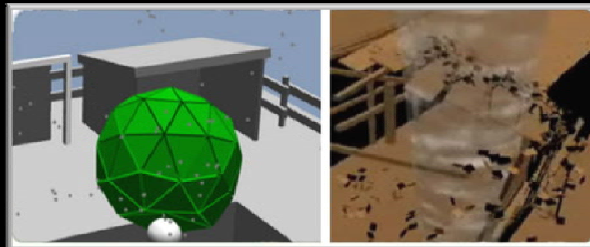
Fluid Simulation



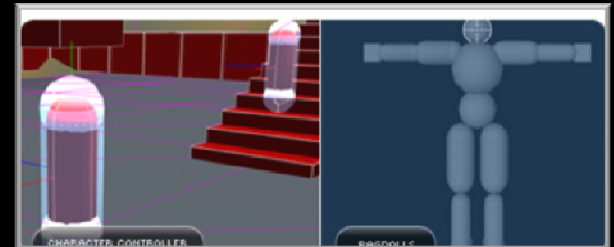
Particles



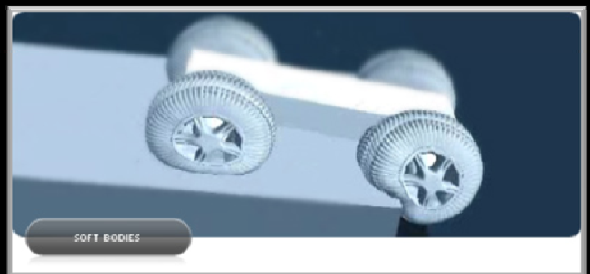
Vehicle Dynamics



Force Fields

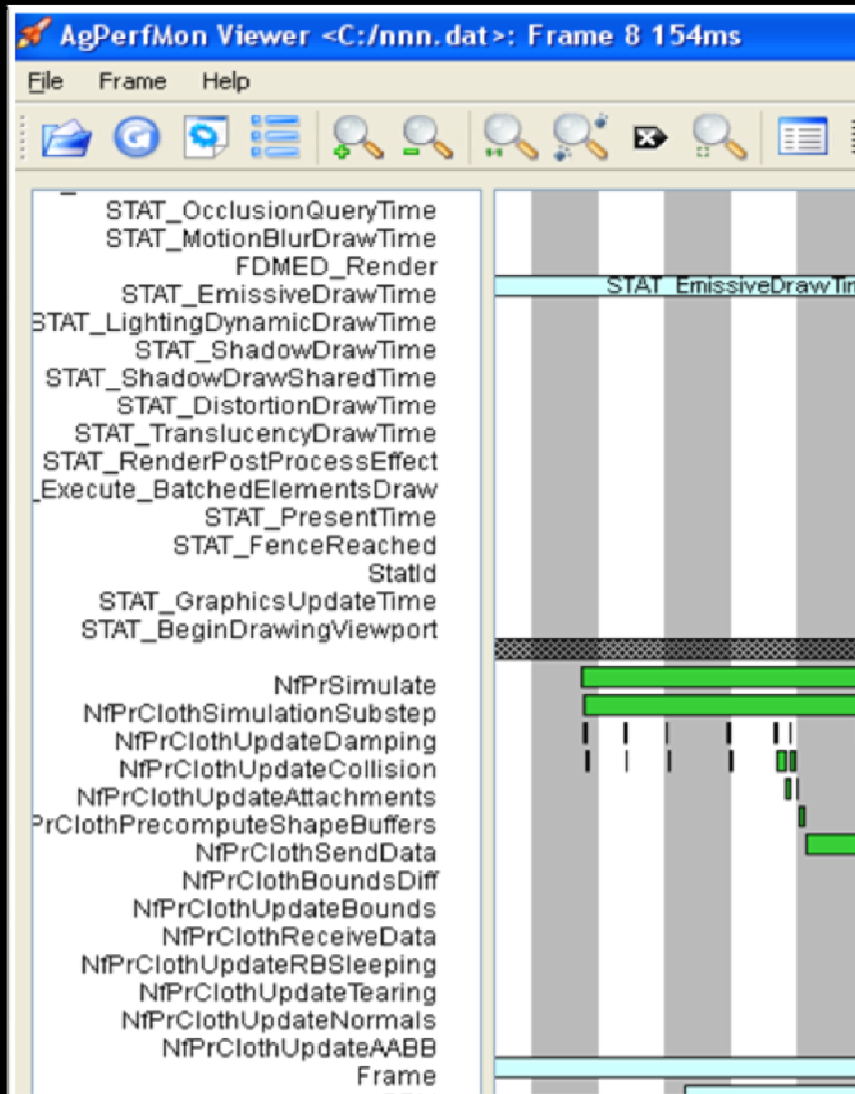


Ragdolls

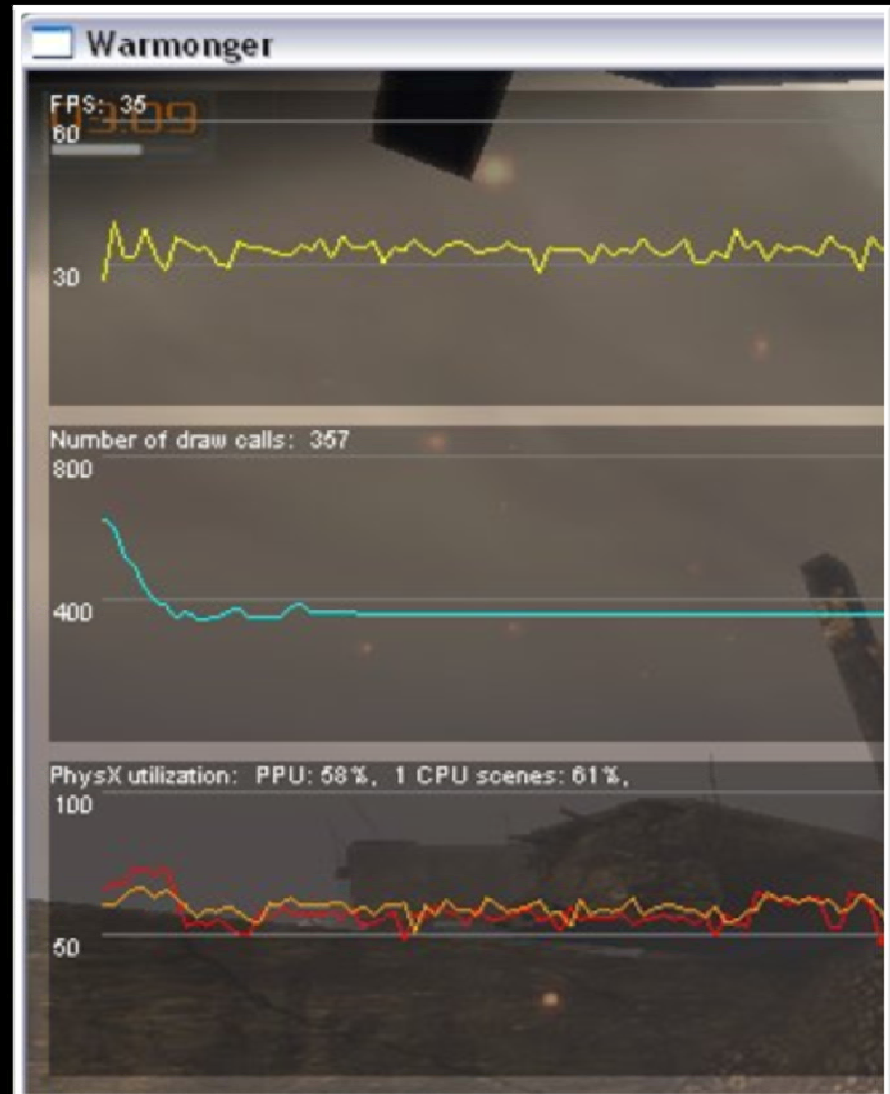


Soft Bodies

NVIDIA Platform Analyzer Components

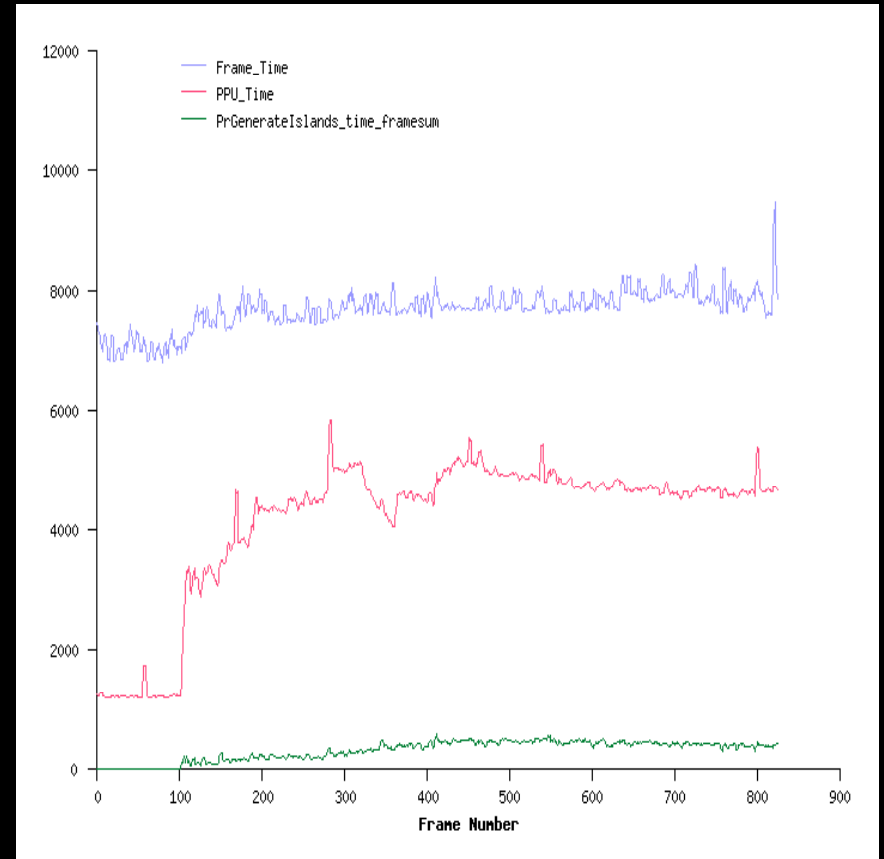
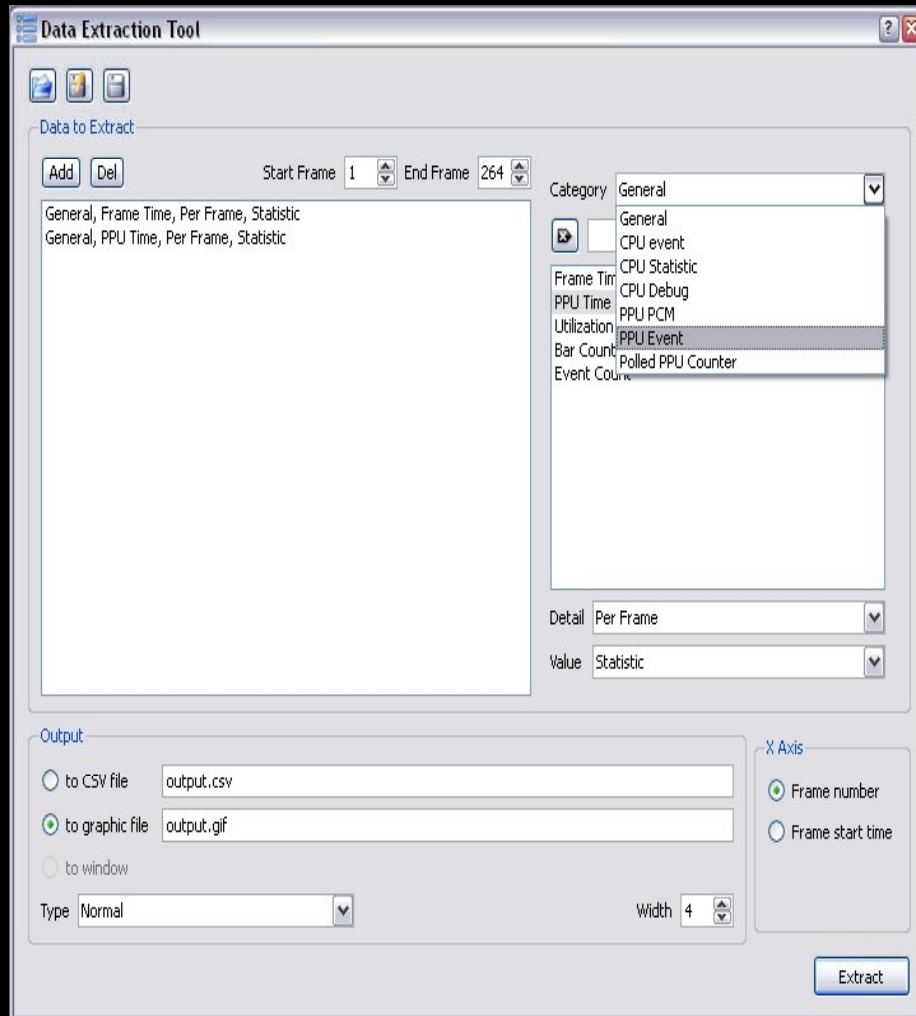


Event Viewer

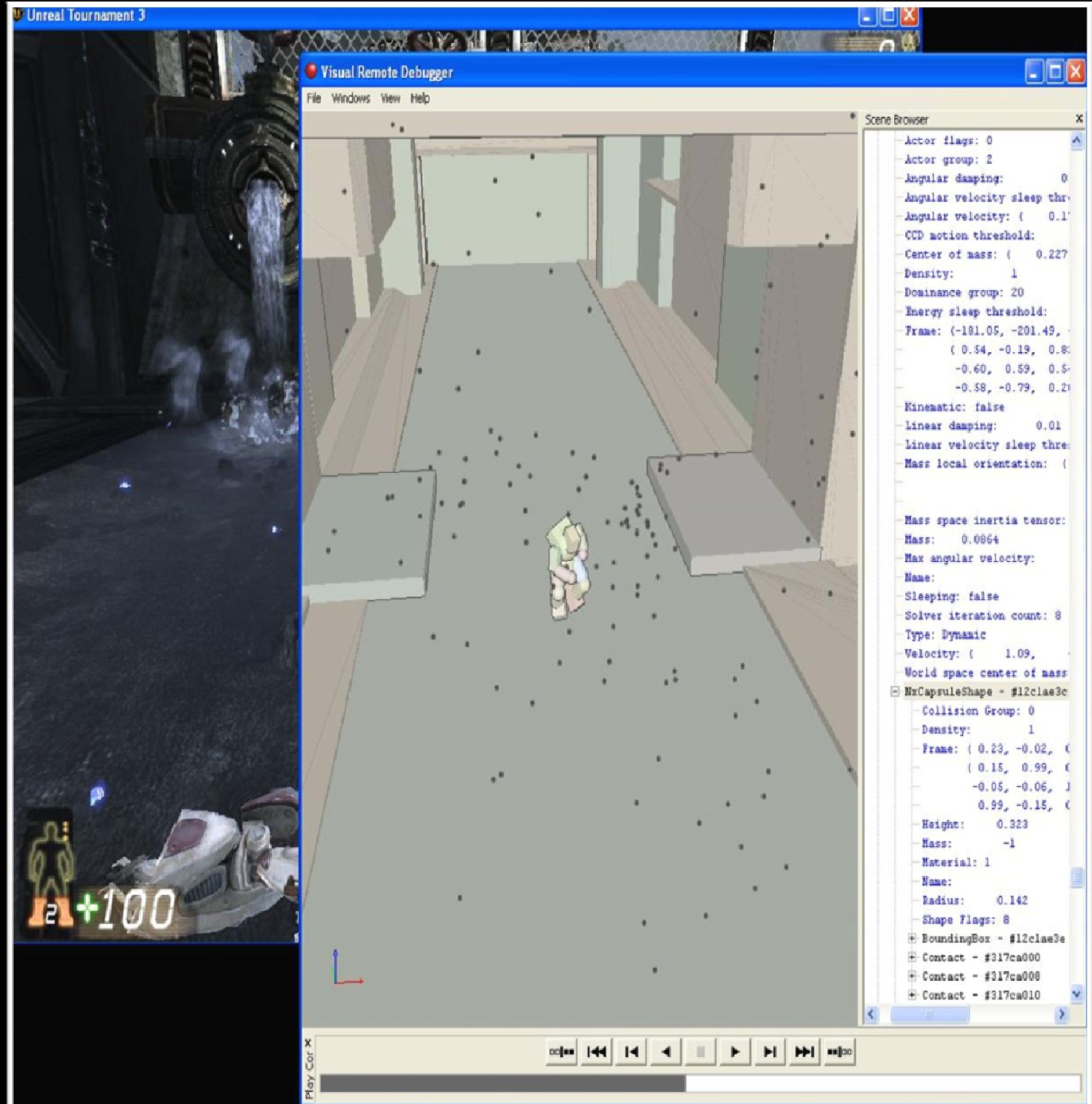


Real-Time Performance Monitoring

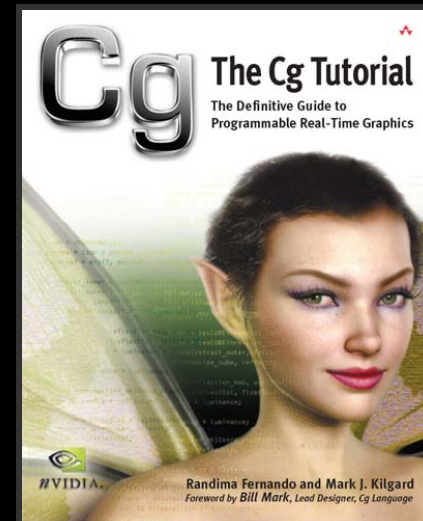
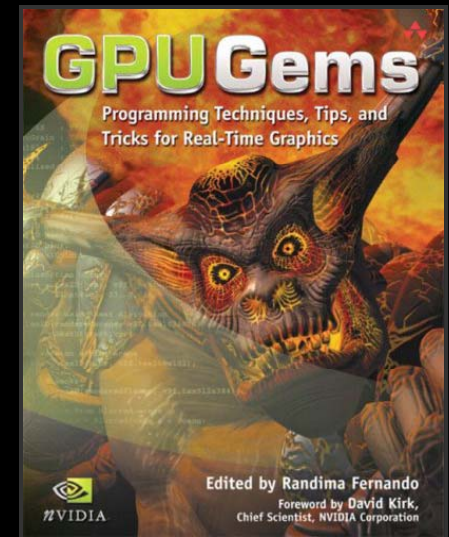
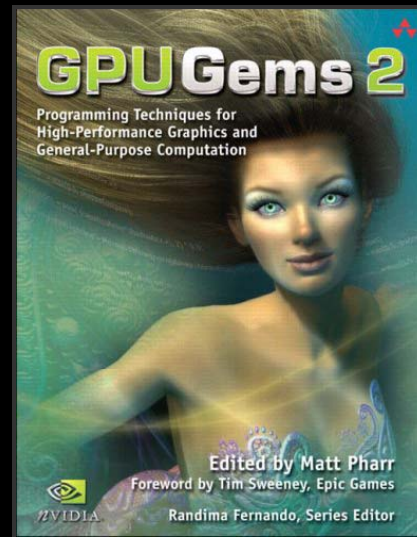
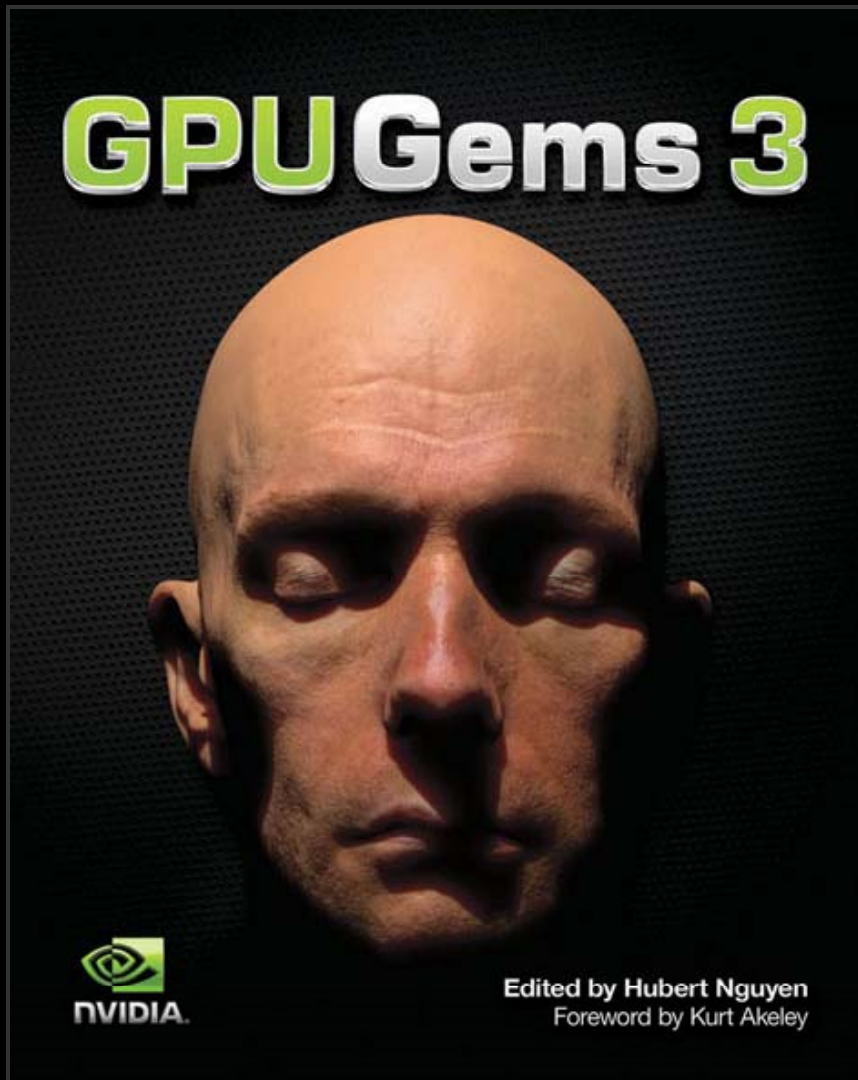
Data Extraction Tool



Visual Debugger



Award-Winning Books



Game Developer Magazine Front Line Awards

GPU Gems 3 Wins in Books (2007)

PerfHUD 5 Finalist in Programming (2007)

Our products have won several prestigious Front Line Awards* from Game Developer Magazine:

PerfHUD	2007 Finalist, 2006 Finalist – Programming
GPU Gems 3	2007 Winner – Books
FX Composer	2005 Finalist, 2004 Finalist – Art Tool
GPU Gems 2	2005 Finalist – Books
SDK	2004 Winner – Programming
GPU Gems	2004 Winner – Books
Cg	2002 Winner – Art

* The Front Line Awards recognize those software and hardware companies whose tools enable faster and more efficient game creation for advancing the state of the art. www.frontlineawards.com



The NVIDIA Software Improvement Program (SIP)

- **Goal: Better meet developer needs**
 - Simplify common usage patterns
 - Expand on popular features
 - Highlight powerful but underutilized features
 - Remove unwanted features
- **Instant Feedback allows user to press F4 and directly send a comment to NVIDIA**
- **Developer opts in to the SIP to participate – we encourage you to do so**
- **Only information about product features used is gathered, as well as GPU and driver version.**
- **No personally identifiable data, shaders, textures, scripts, geometry, or information about other applications is ever collected.**

NVIDIA Developer Zone

Developer Forums
developer.nvidia.com/forums

RSS feed at developer.nvidia.com

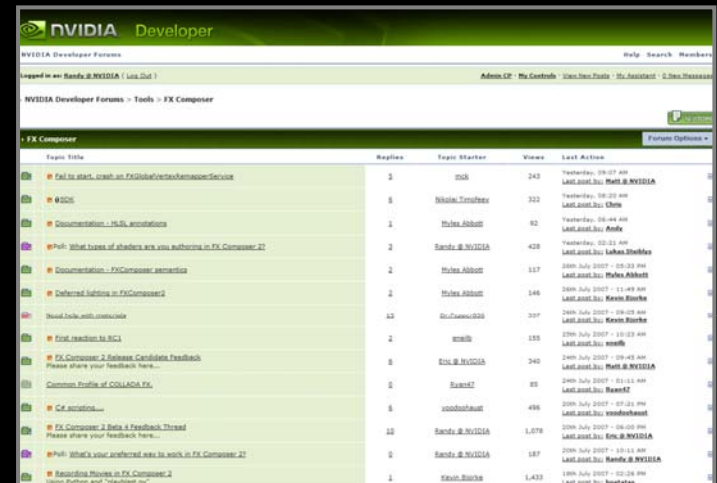
Registered Developer Program

Pre-Release Drivers

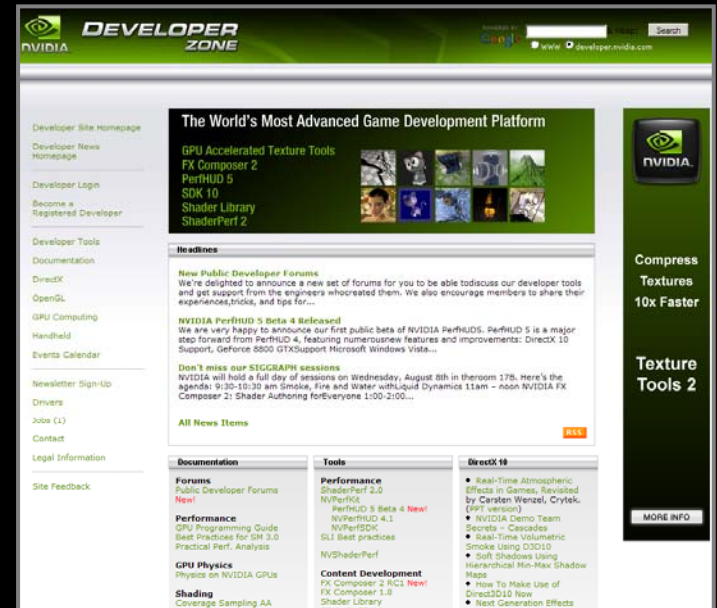
Early Access to Developer Tools

Secure Online Forums & Bug Submission

Apply from developer.nvidia.com



Topic Title	Replies	Topic Starter	Views	Last Action
Fail to start_creat on FXShaderVertexShaderService	1	DS	243	Yesterday, 09:27 AM LAST POST BY: Pete @ NVIDIA
WGL	5	Nicola Tomazini	312	Yesterday, 06:23 AM LAST POST BY: Chris
Documentation - HLSL annotations	1	Wiles Abbott	82	Yesterday, 06:44 AM LAST POST BY: Wiles Abbott
Post: What types of shaders are you authoring in FX Composer 2?	2	Sandy @ NVIDIA	428	Yesterday, 02:32 AM LAST POST BY: Lukas Dethlefs
Documentation - FXComposer parameters	2	Wiles Abbott	117	28th July 2007 - 05:23 AM LAST POST BY: Wiles Abbott
Deferred shading in FXComposer2	2	Wiles Abbott	146	28th July 2007 - 11:49 AM LAST POST BY: Kevin Everts
Need help with HLSL	13	GuillaumeD	307	28th July 2007 - 09:28 AM LAST POST BY: Kevin Everts
FX Shader Perf 2	2	msb	155	28th July 2007 - 10:23 AM LAST POST BY: msb
FX Composer 2 Release Candidate Feedback Please share your feedback here...	6	Eric @ NVIDIA	340	28th July 2007 - 09:45 AM LAST POST BY: Pete @ NVIDIA
Common Profile of COLLADA FX	6	Evant?	88	28th July 2007 - 01:11 AM LAST POST BY: woodchuck
FX Composer 2 Beta 4 Feedback Thread Please share your feedback here...	10	Sandy @ NVIDIA	1,078	28th July 2007 - 06:00 AM LAST POST BY: Pete @ NVIDIA
Post: What's your workflow like to work in FX Composer 2?	0	Sandy @ NVIDIA	187	28th July 2007 - 08:11 AM LAST POST BY: Pete @ NVIDIA
Reporting Issues in FX Composer 2 How to report and troubleshoot...	1	Wiles Abbott	1,433	18th July 2007 - 02:28 PM LAST POST BY: Wiles Abbott



The World's Most Advanced Game Development Platform

GPU Accelerated Texture Tools
FX Composer 2
PerfHUD 5
SDK 10
Shader Library
ShaderPerf 2

Headlines

New Public Developer Forums
We're delighted to announce a new set of forums for you to be able to discuss our developer tools and get support from the engineers who created them. We also encourage members to share their experiences, tricks, and tips for...

NVIDIA PerfHUD 5 Beta 4 Released!
We are very happy to announce our first public beta of NVIDIA PerfHUD 5. PerfHUD 5 is a major step forward from PerfHUD 4, featuring numerous new features and improvements: DirectX 10 Support, GeForce 8800, GTX Support, Microsoft Windows Vista...

Don't miss our SIGGRAPH sessions!
NVIDIA will hold a full day of sessions on Wednesday, August 8th in room 178. Here's the agenda: 9:30-10:30 am: Smoke, Fire and Water with Liquid Dynamics 11am - Aeon NVIDIA FX Composer 2: Shader Authoring for Everyone 1:00-2:00...

All News Items

Documentation

- Forums: Public Developer Forums **New!**
- Performance: GPU Programming Guide, Best Practices for SM 3.0, Practical Perf. Analysis
- GPU Physics: Physics on NVIDIA GPUs
- Shading: Coverage, Sampling, AA

Tools

- Performance: ShaderPerf 2.0, NVPerfHUD, PerfHUD 5 Beta 4 **New!**, NVPerfHUD 4.1, NVPerfSDK, SLI Best Practices, NVShaderPerf
- Content Development: FX Composer 2 RC1 **New!**, FX Composer 1.8, Shader Library

DirectX 10

- Real-Time Atmospheric Effects in Games, Revised by Carsten Wenzel, Crytek. (Perf. version)
- NVIDIA Demo Team Secrets - Classics
- Real-Time Volumetric Smoke Using D3D10
- Soft Shadows Using Hierarchical Min-Max Shadow Maps
- How To Make Use of DirectX10 Now
- Next Generation Effects

Compress Textures 10x Faster
Texture Tools 2

MORE INFO

Useful Links

- [Developer Web Site](#)
- [Developer Tools Forums](#)

- [SDK 10](#)

- [PerfHUD](#)
- [PerfHUD Overview Video](#)

- [FX Composer](#)
- [FX Composer Overview Video](#)

- [PhysX](#)

- [CUDA Zone](#)