NVIDIA Developer Tools
for
Graphics and PhysX

The NVIDIA Developer Toolkit

Content Creation

- FX Composer
  - mental mill Artist Edition
  - NVIDIA Shader Library
- Photoshop Plug-ins
- Texture Tools
- Melody
- PhysX Plug-Ins

Software Development

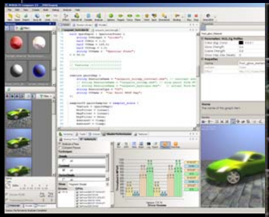
- Shader Debugger
- Direct3D SDK
- OpenGL SDK
- CUDA SDK
- Cg Toolkit
- NVSG
- PhysX SDK
- PhysX VRD

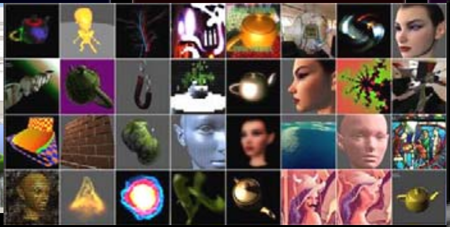
Performance

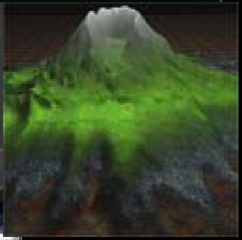
- PerfKit
  - PerfHUD
  - PerfSDK
  - GLExpert
  - NV PIX Plug-in
  - gDEBugger
- ShaderPerf
- PhysX Tools

Education

- Conference Presentations
- Whitepapers
- GPU Programming Guide
- Videos
- Books
  - *GPU Gems 3*
  - *GPU Gems 2*
  - *GPU Gems*
  - *The Cg Tutorial*







Agenda

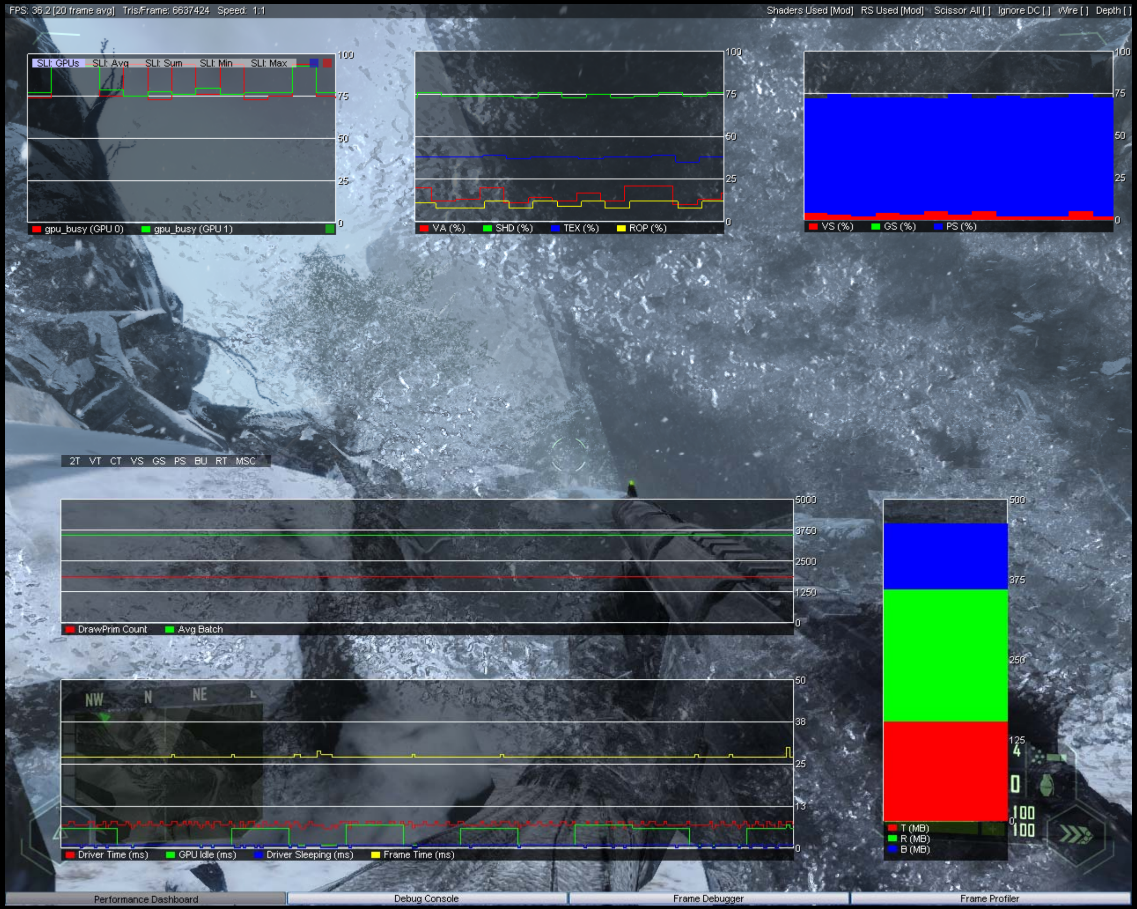
- PerfHUD 6.5
- FX Composer 2.0, Shader Debugger and ShaderPerf
- AgPerfMon
- VRD

NVIDIA PerfHUD 6.5

What’s new in PerfHUD 6.5?

- 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 and override
  - API Call data mining and analysis
  - Dependency view
- Usability Features
  - Save/Load DX10 frame captures
  - Rich use of Direct3D PerfMarkers (PIX)

PerfHUD: Performance Dashboard


FPS: 36.2 [20 frame avg]   Tris/Frame: 6637424   Speed: 1:1
Shaders Used [Mod]   RS Used [Mod]   Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]
SLI: GPUs   SLI: Avg   SLI: Sum   SLI: Min   SLI: Max
100
75
50
25
0
gpu_busy (GPU 0)   gpu_busy (GPU 1)
VA (%)   SHD (%)   TEX (%)   ROP (%)
VS (%)   GS (%)   PS (%)
2T   VT   CT   VS   GS   PS   BU   RT   MSC
5000
3750
2500
1250
DrawPrim Count   Avg Batch
500
375
250
125
T (MB)
R (MB)
B (MB)
50
38
13
Driver Time (ms)   GPU Idle (ms)   Driver Sleeping (ms)   Frame Time (ms)
Performance Dashboard
Debug Console
Frame Debugger
Frame Profiler

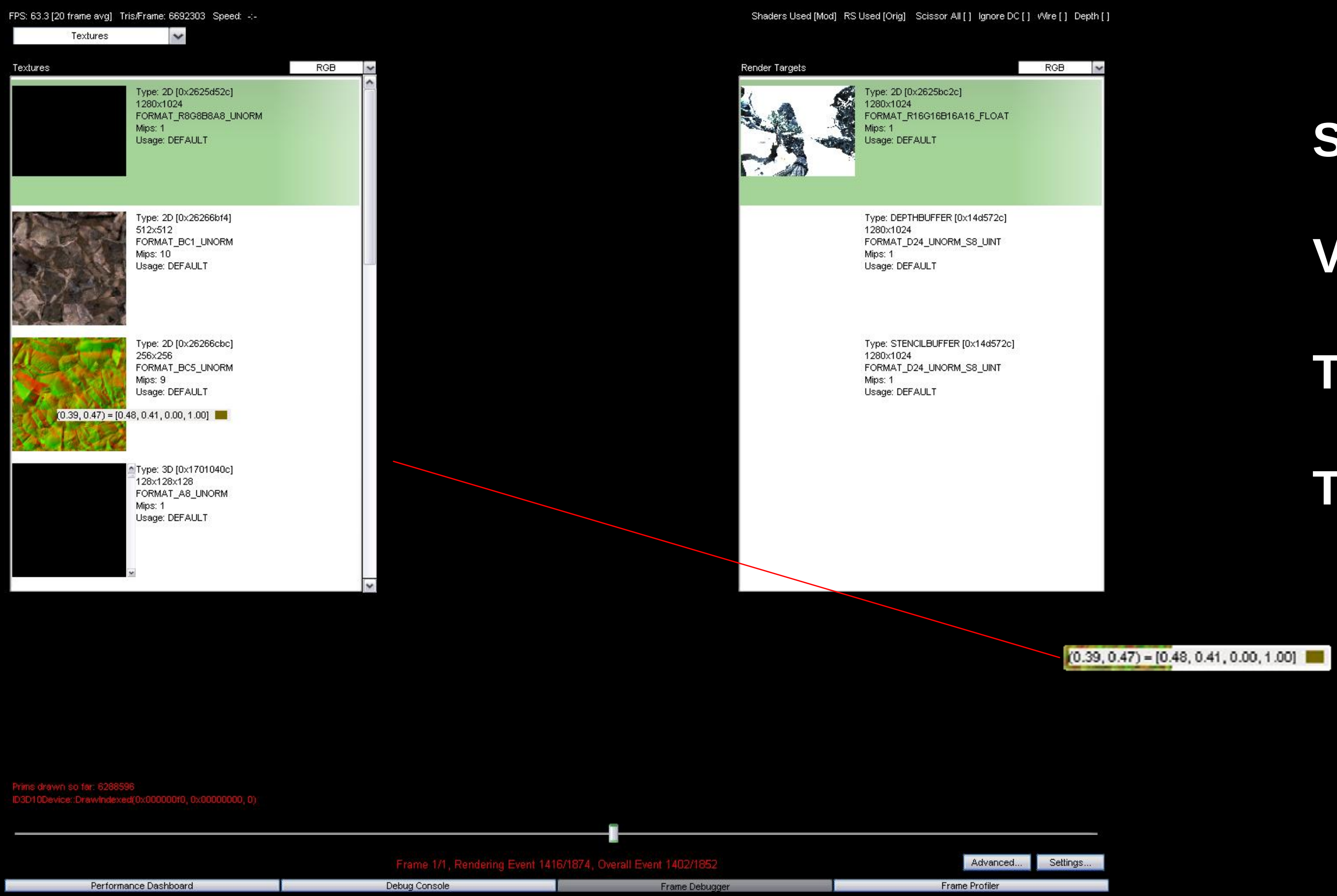
Graph GPU and driver data

Edit to suit your needs

SLI Graph for multi-GPU

API usage statistics

PerfHUD: Frame Debugger



```
FPS: 63.3 [20 frame avg]   Tris/Frame: 6692303   Speed:  -:-
Shaders Used [Mod]   RS Used [Orig]   Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]
```

Textures

```
Type: 2D [0x2625d52c]
1280x1024
FORMAT_R8G8B8A8_UNORM
Mips: 1
Usage: DEFAULT

Type: 2D [0x26266bf4]
512x512
FORMAT_BC1_UNORM
Mips: 10
Usage: DEFAULT

Type: 2D [0x26266cbc]
256x256
FORMAT_BC5_UNORM
Mips: 9
Usage: DEFAULT
(0.39, 0.47) = [0.48, 0.41, 0.00, 1.00]

Type: 3D [0x1701040c]
128x128x128
FORMAT_A8_UNORM
Mips: 1
Usage: DEFAULT
```

Render Targets

```
Type: 2D [0x2625bc2c]
1280x1024
FORMAT_R16G16B16A16_FLOAT
Mips: 1
Usage: DEFAULT

Type: DEPTHBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

Type: STENCILBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT
```

```
(0.39, 0.47) = [0.48, 0.41, 0.00, 1.00]
```

```
Prims drawn so far: 6288596
ID3D10Device::DrawIndexed(0x000000f0, 0x00000000, 0)

Frame 1/1, Rendering Event 1416/1874, Overall Event 1402/1852
```

Scrub through scene

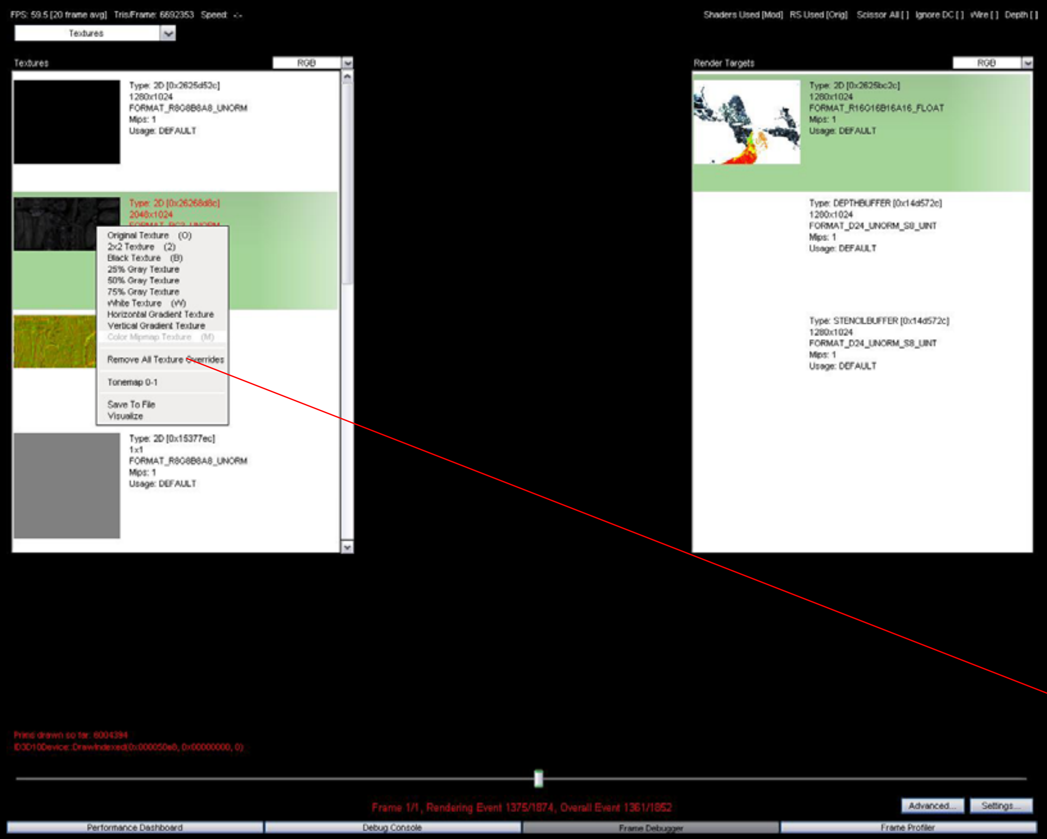
Visualize draw call info

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.

PerfHUD: Frame Debugger



FPS: 59.5 [20 frame avg]   Tris/Frame: 6692353   Speed: -.-

Shaders Used [Mod]   RS Used [Orig]   Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]

Textures

Type: 2D [0x2625d52c]
1280x1024
FORMAT_R8G8B8A8_UNORM
Mips: 1
Usage: DEFAULT

Type: 2D [0x26268d8c]
2048x1024

Type: 2D [0x15377ec]
1x1
FORMAT_R8G8B8A8_UNORM
Mips: 1
Usage: DEFAULT

Render Targets

Type: 2D [0x2625bc2c]
1280x1024
FORMAT_R16G16B16A16_FLOAT
Mips: 1
Usage: DEFAULT

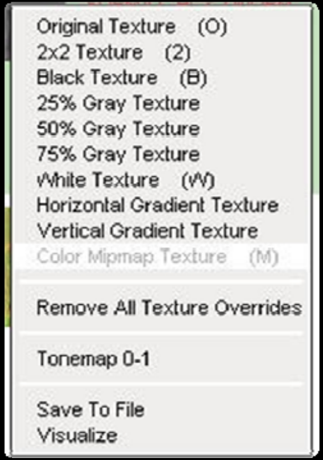
Type: DEPTHBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

Type: STENCILBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

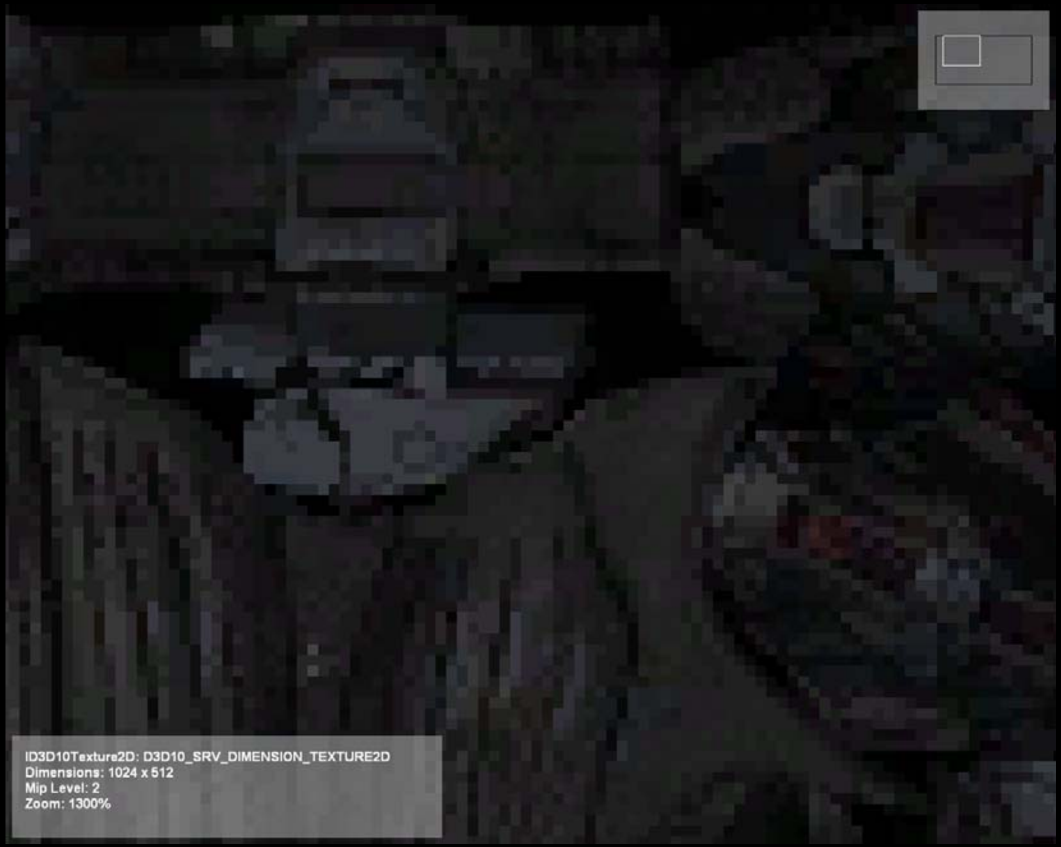
Prims drawn so far: 6004394
ID3D10Device::DrawIndexed(0x000050e0, 0x00000000, 0)

Frame 1/1, Rendering Event 1375/1874, Overall Event 1361/1852

Texture analysis: substitute precomputed textures
Controllable via Perf Markers



PerfHUD: Frame Debugger



```
ID3D10Texture2D: D3D10_SRV_DIMENSION_TEXTURE2D
Dimensions: 1024 x 512
Mip Level: 2
Zoom: 1300%
```

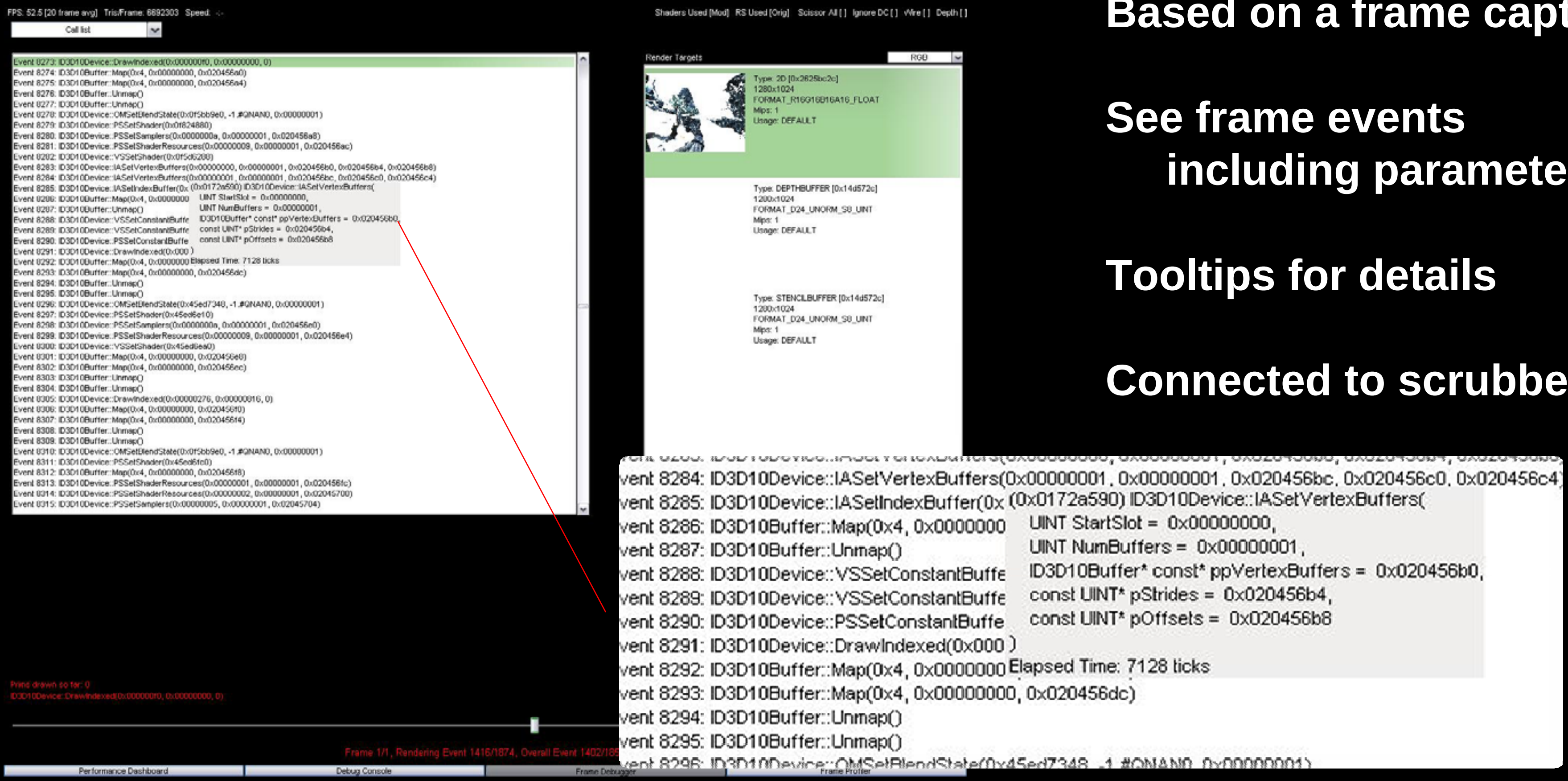
Crysis used with permission from Crytek. © Crytek GmbH. All Rights Reserved. Crysis and CryENGINE are trademarks or registered trademarks of Crytek GmbH in the U.S and/or other countries.

Visualize any buffer
full screen

2D/3D/Cube/Arrays
Pan/Zoom

Change mipmap level

PerfHUD: Frame Debugger



```
FPS: 52.5 [20 frame avg]   Tris/Frame: 6692303   Speed: -:-
Call list

Event 8273: ID3D10Device::DrawIndexed(0x000000f0, 0x00000000, 0)
Event 8274: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456a0)
Event 8275: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456a4)
Event 8276: ID3D10Buffer::Unmap()
Event 8277: ID3D10Buffer::Unmap()
Event 8278: ID3D10Device::OMSetBlendState(0x0f5bb9e0, -1.#QNAN0, 0x00000001)
Event 8279: ID3D10Device::PSSetShader(0x0f824880)
Event 8280: ID3D10Device::PSSetSamplers(0x0000000a, 0x00000001, 0x020456a8)
Event 8281: ID3D10Device::PSSetShaderResources(0x00000009, 0x00000001, 0x020456ac)
Event 8282: ID3D10Device::VSSetShader(0x0f5d6208)
Event 8283: ID3D10Device::IASetVertexBuffers(0x00000000, 0x00000001, 0x020456b0, 0x020456b4, 0x020456b8)
Event 8284: ID3D10Device::IASetVertexBuffers(0x00000001, 0x00000001, 0x020456bc, 0x020456c0, 0x020456c4)
Event 8285: ID3D10Device::IASetIndexBuffer(0x (0x0172a590) ID3D10Device::IASetVertexBuffers(
Event 8286: ID3D10Buffer::Map(0x4, 0x0000000    UINT StartSlot =  0x00000000,
Event 8287: ID3D10Buffer::Unmap()               UINT NumBuffers =  0x00000001,
Event 8288: ID3D10Device::VSSetConstantBuffe    ID3D10Buffer* const* ppVertexBuffers =  0x020456b0,
Event 8289: ID3D10Device::VSSetConstantBuffe    const UINT* pStrides =  0x020456b4,
Event 8290: ID3D10Device::PSSetConstantBuffe    const UINT* pOffsets =  0x020456b8
Event 8291: ID3D10Device::DrawIndexed(0x000 )
Event 8292: ID3D10Buffer::Map(0x4, 0x0000000 Elapsed Time: 7128 ticks
Event 8293: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456dc)
Event 8294: ID3D10Buffer::Unmap()
Event 8295: ID3D10Buffer::Unmap()
Event 8296: ID3D10Device::OMSetBlendState(0x45ed7348, -1.#QNAN0, 0x00000001)
Event 8297: ID3D10Device::PSSetShader(0x45ed6e10)
Event 8298: ID3D10Device::PSSetSamplers(0x0000000a, 0x00000001, 0x020456e0)
Event 8299: ID3D10Device::PSSetShaderResources(0x00000009, 0x00000001, 0x020456e4)
Event 8300: ID3D10Device::VSSetShader(0x45ed6ea0)
Event 8301: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456e8)
Event 8302: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456ec)
Event 8303: ID3D10Buffer::Unmap()
Event 8304: ID3D10Buffer::Unmap()
Event 8305: ID3D10Device::DrawIndexed(0x00000276, 0x00000816, 0)
Event 8306: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456f0)
Event 8307: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456f4)
Event 8308: ID3D10Buffer::Unmap()
Event 8309: ID3D10Buffer::Unmap()
Event 8310: ID3D10Device::OMSetBlendState(0x0f5bb9e0, -1.#QNAN0, 0x00000001)
Event 8311: ID3D10Device::PSSetShader(0x45ed6fc0)
Event 8312: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456f8)
Event 8313: ID3D10Device::PSSetShaderResources(0x00000001, 0x00000001, 0x020456fc)
Event 8314: ID3D10Device::PSSetShaderResources(0x00000002, 0x00000001, 0x02045700)
Event 8315: ID3D10Device::PSSetSamplers(0x00000005, 0x00000001, 0x02045704)
```

```
Shaders Used [Mod]   RS Used [Orig]   Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]

Render Targets                                   RGB

Type: 2D [0x2625bc2c]
1280x1024
FORMAT_R16G16B16A16_FLOAT
Mips: 1
Usage: DEFAULT

Type: DEPTHBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

Type: STENCILBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT
```

```
Prims drawn so far: 0
ID3D10Device::DrawIndexed(0x000000f0, 0x00000000, 0)

Frame 1/1, Rendering Event 1416/1874, Overall Event 1402/185
```

```
vent 8284: ID3D10Device::IASetVertexBuffers(0x00000001, 0x00000001, 0x020456bc, 0x020456c0, 0x020456c4
vent 8285: ID3D10Device::IASetIndexBuffer(0x (0x0172a590) ID3D10Device::IASetVertexBuffers(
vent 8286: ID3D10Buffer::Map(0x4, 0x0000000    UINT StartSlot =  0x00000000,
vent 8287: ID3D10Buffer::Unmap()               UINT NumBuffers =  0x00000001,
vent 8288: ID3D10Device::VSSetConstantBuffe    ID3D10Buffer* const* ppVertexBuffers =  0x020456b0,
vent 8289: ID3D10Device::VSSetConstantBuffe    const UINT* pStrides =  0x020456b4,
vent 8290: ID3D10Device::PSSetConstantBuffe    const UINT* pOffsets =  0x020456b8
vent 8291: ID3D10Device::DrawIndexed(0x000 )
vent 8292: ID3D10Buffer::Map(0x4, 0x0000000 Elapsed Time: 7128 ticks
vent 8293: ID3D10Buffer::Map(0x4, 0x00000000, 0x020456dc)
vent 8294: ID3D10Buffer::Unmap()
vent 8295: ID3D10Buffer::Unmap()
vent 8296: ID3D10Device::OMSetBlendState(0x45ed7348, -1.#QNAN0, 0x00000001)
```

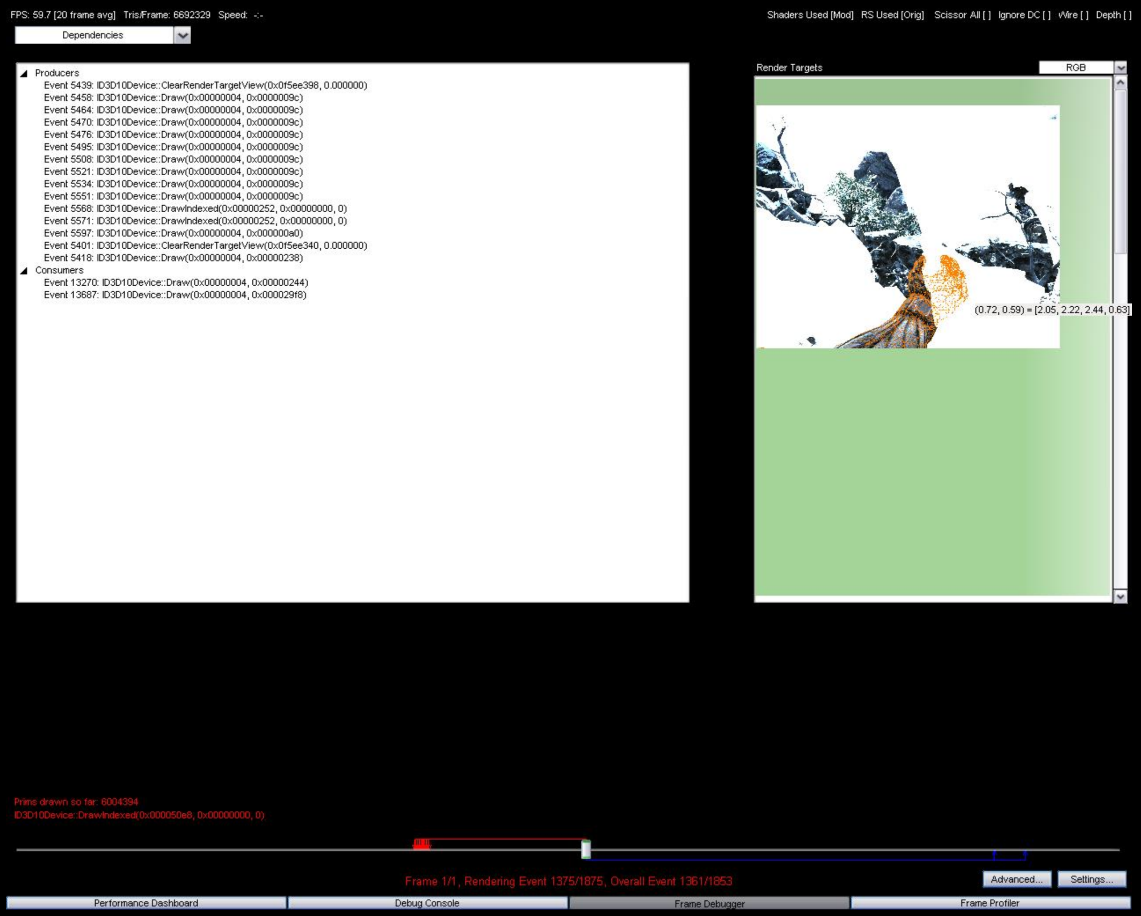
Based on a frame capture

See frame events including parameters

Tooltips for details

Connected to scrubber

PerfHUD: Frame Debugger



```
FPS: 59.7 [20 frame avg]   Tris/Frame: 6692329   Speed:  -:-
```

```
Shaders Used [Mod]   RS Used [Orig]    Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]
```

Dependencies

```
◢ Producers
    Event 5439: ID3D10Device::ClearRenderTargetView(0x0f5ee398, 0.000000)
    Event 5458: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5464: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5470: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5476: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5495: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5508: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5521: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5534: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5551: ID3D10Device::Draw(0x00000004, 0x0000009c)
    Event 5568: ID3D10Device::DrawIndexed(0x00000252, 0x00000000, 0)
    Event 5571: ID3D10Device::DrawIndexed(0x00000252, 0x00000000, 0)
    Event 5597: ID3D10Device::Draw(0x00000004, 0x000000a0)
    Event 5401: ID3D10Device::ClearRenderTargetView(0x0f5ee340, 0.000000)
    Event 5418: ID3D10Device::Draw(0x00000004, 0x00000238)
◢ Consumers
    Event 13270: ID3D10Device::Draw(0x00000004, 0x00000244)
    Event 13687: ID3D10Device::Draw(0x00000004, 0x000029f8)
```

Render Targets: RGB

```
(0.72, 0.59) = [2.05, 2.22, 2.44, 0.63]
```

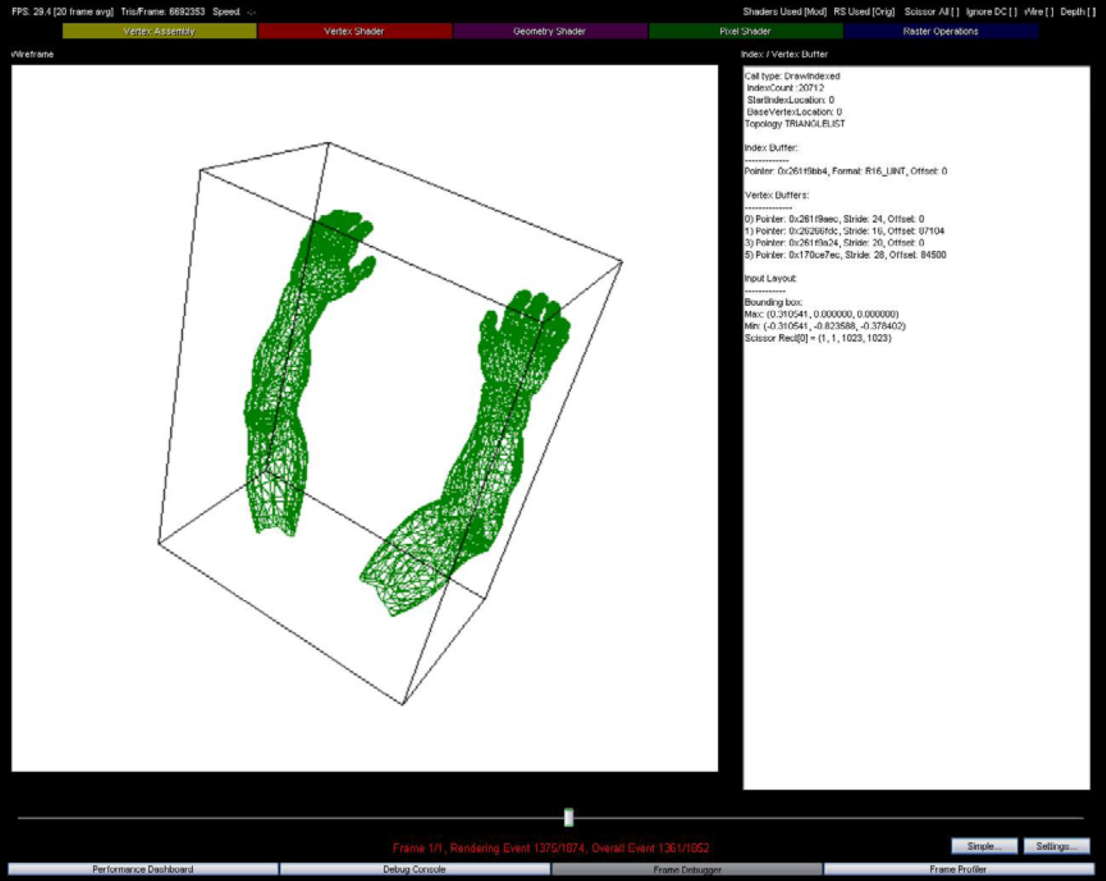
```
Prims drawn so far: 6004394
ID3D10Device::DrawIndexed(0x000050e8, 0x00000000, 0)
```

```
Frame 1/1, Rendering Event 1375/1875, Overall Event 1361/1853
```

- Show producers & consumers dependencies for each call

- These can hurt single GPU and SLI performance

PerfHUD: Advanced Frame Debugger



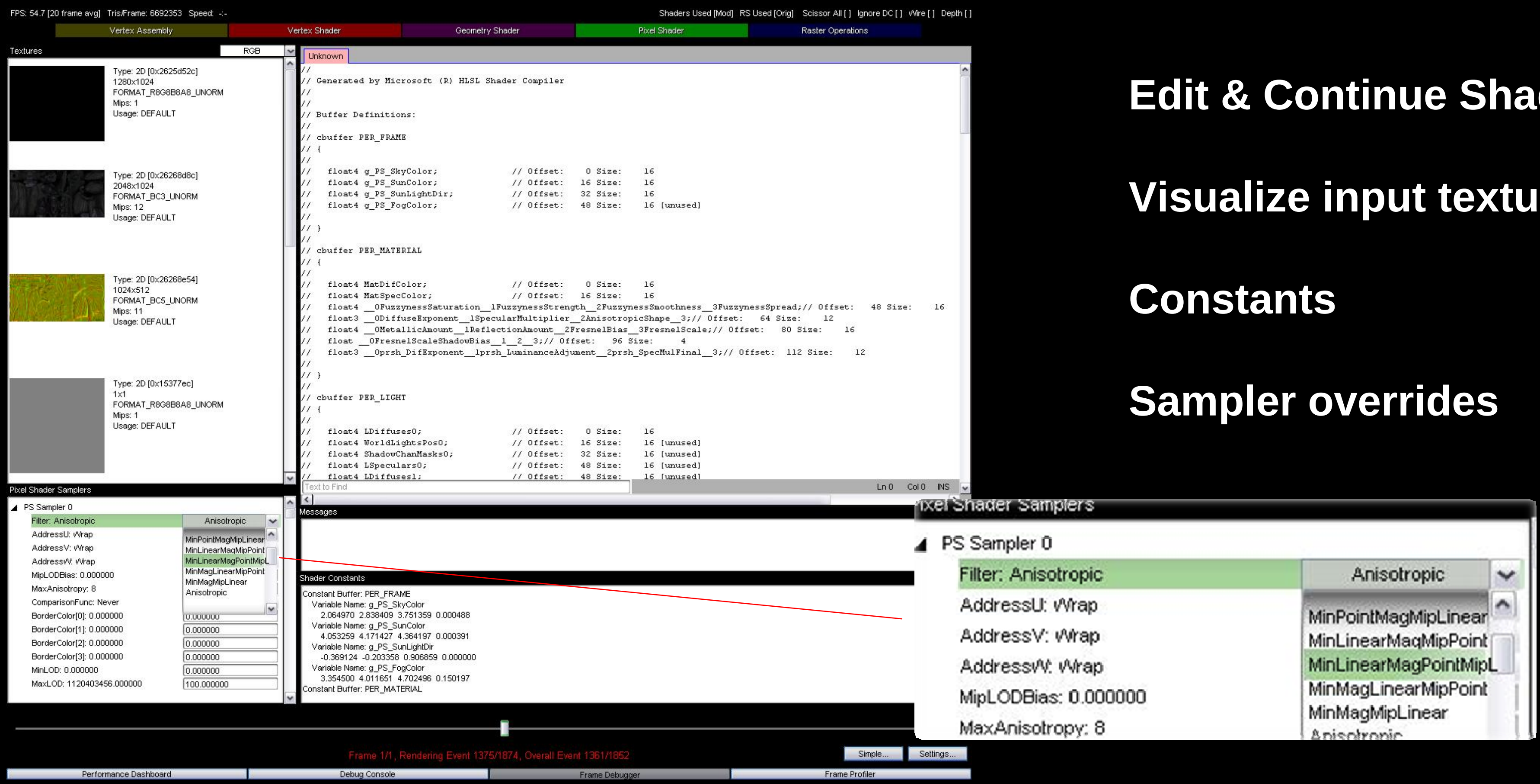
```
Call type: DrawIndexed
 IndexCount :20712
 StartIndexLocation: 0
 BaseVertexLocation: 0
Topology TRIANGLELIST

Index Buffer:
-------------
Pointer: 0x261f9bb4, Format: R16_UINT, Offset: 0

Vertex Buffers:
--------------
0) Pointer: 0x261f9aec, Stride: 24, Offset: 0
1) Pointer: 0x26266fdc, Stride: 16, Offset: 87104
3) Pointer: 0x261f9a24, Stride: 20, Offset: 0
5) Pointer: 0x170ce7ec, Stride: 28, Offset: 84500

Input Layout:
------------
Bounding box:
Max: (0.310541, 0.000000, 0.000000)
Min: (-0.310541, -0.823588, -0.378402)
Scissor Rect[0] = (1, 1, 1023, 1023)
```

```
Frame 1/1, Rendering Event 1375/1874, Overall Event 1361/1852
```

Geometry Preview

Vertex and index buffer setup

PerfHUD: Advanced Frame Debugger



FPS: 54.7 [20 frame avg] Tris/Frame: 6692353 Speed: -:-

Shaders Used [Mod] RS Used [Orig] Scissor All [ ] Ignore DC [ ] Wire [ ] Depth [ ]

Type: 2D [0x2625d52c]
1280x1024
FORMAT_R8G8B8A8_UNORM
Mips: 1
Usage: DEFAULT

Type: 2D [0x26268d8c]
2048x1024
FORMAT_BC3_UNORM
Mips: 12
Usage: DEFAULT

Type: 2D [0x26268e54]
1024x512
FORMAT_BC5_UNORM
Mips: 11
Usage: DEFAULT

Type: 2D [0x15377ec]
1x1
FORMAT_R8G8B8A8_UNORM
Mips: 1
Usage: DEFAULT

```
//
// Generated by Microsoft (R) HLSL Shader Compiler
//
//
// Buffer Definitions:
//
// cbuffer PER_FRAME
// {
//
//   float4 g_PS_SkyColor;              // Offset:    0 Size:    16
//   float4 g_PS_SunColor;              // Offset:   16 Size:    16
//   float4 g_PS_SunLightDir;           // Offset:   32 Size:    16
//   float4 g_PS_FogColor;              // Offset:   48 Size:    16 [unused]
//
// }
//
// cbuffer PER_MATERIAL
// {
//
//   float4 MatDifColor;                // Offset:    0 Size:    16
//   float4 MatSpecColor;               // Offset:   16 Size:    16
//   float4 __0FuzzynessSaturation__1FuzzynessStrength__2FuzzynessSmoothness__3FuzzynessSpread;// Offset:   48 Size:    16
//   float3 __0DiffuseExponent__1SpecularMultiplier__2AnisotropicShape__3;// Offset:   64 Size:    12
//   float4 __0MetallicAmount__1ReflectionAmount__2FresnelBias__3FresnelScale;// Offset:   80 Size:    16
//   float __0FresnelScaleShadowBias__1__2__3;// Offset:   96 Size:     4
//   float3 __0prsh_DifExponent__1prsh_LuminanceAdjument__2prsh_SpecMulFinal__3;// Offset:  112 Size:    12
//
// }
//
// cbuffer PER_LIGHT
// {
//
//   float4 LDiffuses0;                 // Offset:    0 Size:    16
//   float4 WorldLightsPos0;            // Offset:   16 Size:    16 [unused]
//   float4 ShadowChanMasks0;           // Offset:   32 Size:    16 [unused]
//   float4 LSpeculars0;                // Offset:   48 Size:    16 [unused]
//   float4 LDiffuses1;                 // Offset:   48 Size:    16 [unused]
```

Ln 0 Col 0 INS

PS Sampler 0
Filter: Anisotropic — Anisotropic
AddressU: Wrap
AddressV: Wrap
AddressW: Wrap
MipLODBias: 0.000000
MaxAnisotropy: 8
ComparisonFunc: Never
BorderColor[0]: 0.000000 — 0.000000
BorderColor[1]: 0.000000 — 0.000000
BorderColor[2]: 0.000000 — 0.000000
BorderColor[3]: 0.000000 — 0.000000
MinLOD: 0.000000 — 0.000000
MaxLOD: 1120403456.000000 — 100.000000

MinPointMagMipLinear
MinLinearMagMipPoint
MinLinearMagPointMipL
MinMagLinearMipPoint
MinMagMipLinear
Anisotropic

```
Constant Buffer: PER_FRAME
  Variable Name: g_PS_SkyColor
    2.064970  2.838409  3.751359  0.000488
  Variable Name: g_PS_SunColor
    4.053259  4.171427  4.364197  0.000391
  Variable Name: g_PS_SunLightDir
    -0.369124  -0.203358  0.906859  0.000000
  Variable Name: g_PS_FogColor
    3.354500  4.011651  4.702496  0.150197
Constant Buffer: PER_MATERIAL
```

Frame 1/1, Rendering Event 1375/1874, Overall Event 1361/1852

ixel Shader Samplers
PS Sampler 0
Filter: Anisotropic — Anisotropic
AddressU: Wrap
AddressV: Wrap
AddressW: Wrap
MipLODBias: 0.000000
MaxAnisotropy: 8
MinPointMagMipLinear
MinLinearMagMipPoint
MinLinearMagPointMipL
MinMagLinearMipPoint
MinMagMipLinear
Anisotropic

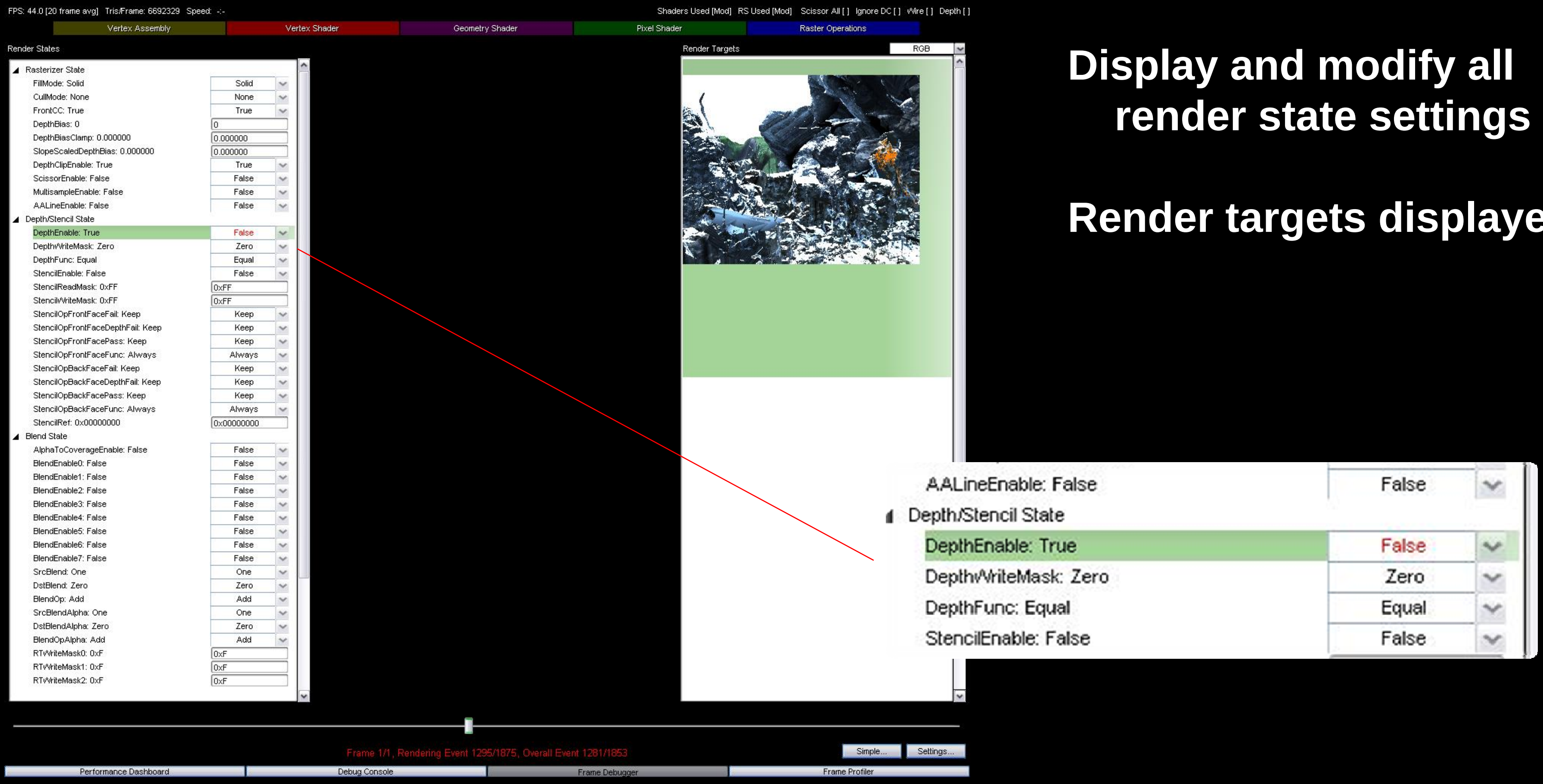
Edit & Continue Shaders

Visualize input textures

Constants

Sampler overrides

PerfHUD: Advanced Frame Debugger



FPS: 44.0 [20 frame avg] Tris/Frame: 6692329 Speed: -:-

Shaders Used [Mod] RS Used [Mod] Scissor All [ ] Ignore DC [ ] Wire [ ] Depth [ ]

Rasterizer State	
FillMode: Solid	Solid
CullMode: None	None
FrontCC: True	True
DepthBias: 0	0
DepthBiasClamp: 0.000000	0.000000
SlopeScaledDepthBias: 0.000000	0.000000
DepthClipEnable: True	True
ScissorEnable: False	False
MultisampleEnable: False	False
AALineEnable: False	False
Depth/Stencil State	
DepthEnable: True	False
DepthWriteMask: Zero	Zero
DepthFunc: Equal	Equal
StencilEnable: False	False
StencilReadMask: 0xFF	0xFF
StencilWriteMask: 0xFF	0xFF
StencilOpFrontFaceFail: Keep	Keep
StencilOpFrontFaceDepthFail: Keep	Keep
StencilOpFrontFacePass: Keep	Keep
StencilOpFrontFaceFunc: Always	Always
StencilOpBackFaceFail: Keep	Keep
StencilOpBackFaceDepthFail: Keep	Keep
StencilOpBackFacePass: Keep	Keep
StencilOpBackFaceFunc: Always	Always
StencilRef: 0x00000000	0x00000000
Blend State	
AlphaToCoverageEnable: False	False
BlendEnable0: False	False
BlendEnable1: False	False
BlendEnable2: False	False
BlendEnable3: False	False
BlendEnable4: False	False
BlendEnable5: False	False
BlendEnable6: False	False
BlendEnable7: False	False
SrcBlend: One	One
DstBlend: Zero	Zero
BlendOp: Add	Add
SrcBlendAlpha: One	One
DstBlendAlpha: Zero	Zero
BlendOpAlpha: Add	Add
RTWriteMask0: 0xF	0xF
RTWriteMask1: 0xF	0xF
RTWriteMask2: 0xF	0xF

Frame 1/1, Rendering Event 1295/1875, Overall Event 1281/1853

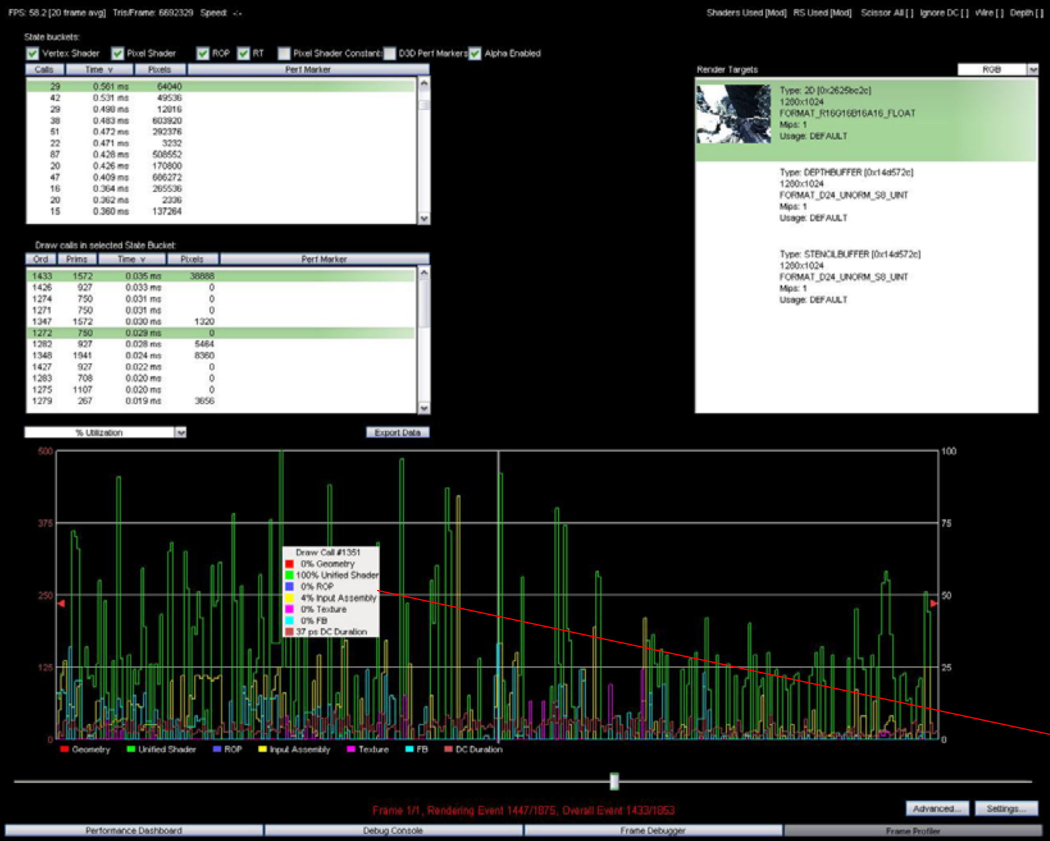
AALineEnable: False	False
Depth/Stencil State	
DepthEnable: True	False
DepthWriteMask: Zero	Zero
DepthFunc: Equal	Equal
StencilEnable: False	False

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: Frame Profiler



FPS: 58.2 [20 frame avg]   Tris/Frame: 6692329   Speed: -:-

Shaders Used [Mod]   RS Used [Mod]   Scissor All [ ]   Ignore DC [ ]   Wire [ ]   Depth [ ]

State buckets:

☑ Vertex Shader ☑ Pixel Shader ☑ ROP ☑ RT □ Pixel Shader Constant: □ D3D Perf Markers ☑ Alpha Enabled

Calls	Time v	Pixels	Perf Marker
29	0.561 ms	64040	
42	0.531 ms	49536	
29	0.490 ms	12016	
38	0.483 ms	603920	
51	0.472 ms	292376	
22	0.471 ms	3232	
87	0.428 ms	508552	
20	0.426 ms	170800	
47	0.409 ms	606272	
16	0.364 ms	265536	
20	0.362 ms	2336	
15	0.360 ms	137264	

Draw calls in selected State Bucket:

Ord	Prims	Time v	Pixels	Perf Marker
1433	1572	0.035 ms	38888	
1426	927	0.033 ms	0	
1274	750	0.031 ms	0	
1271	750	0.031 ms	0	
1347	1572	0.030 ms	1320	
1272	750	0.029 ms	0	
1282	927	0.028 ms	5464	
1348	1941	0.024 ms	8360	
1427	927	0.022 ms	0	
1283	708	0.020 ms	0	
1275	1107	0.020 ms	0	
1279	267	0.019 ms	3656	

Render Targets   RGB

Type: 2D [0x2625bc2c]
1280x1024
FORMAT_R16G16B16A16_FLOAT
Mips: 1
Usage: DEFAULT

Type: DEPTHBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

Type: STENCILBUFFER [0x14d572c]
1280x1024
FORMAT_D24_UNORM_S8_UINT
Mips: 1
Usage: DEFAULT

% Utilization

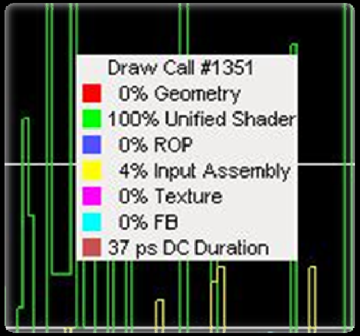
Draw Call #1351
🟥 0% Geometry
🟩 100% Unified Shader
🟦 0% ROP
🟨 4% Input Assembly
🟪 0% Texture
🟦 0% FB
🟥 37 ps DC Duration

Frame 1/1, Rendering Event 1447/1875, Overall Event 1433/1853

All draw calls profiled

Draw calls grouped by State Buckets: multiply performance optimizations

Multiple result graphs



Draw Call #1351
🟥 0% Geometry
🟩 100% Unified Shader
🟦 0% ROP
🟨 4% Input Assembly
🟪 0% Texture
🟦 0% FB
🟥 37 ps DC Duration

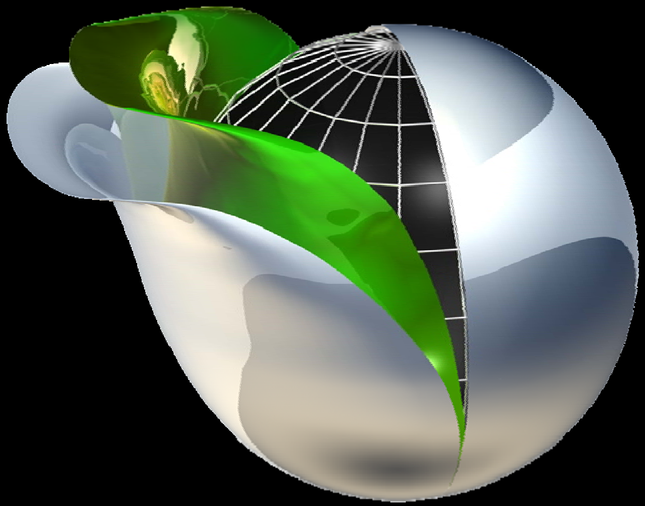
More on PerfHUD 6.5

- 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
- Save/Load DX10 frame captures (Ctrl+S in the Frame Debugger)

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

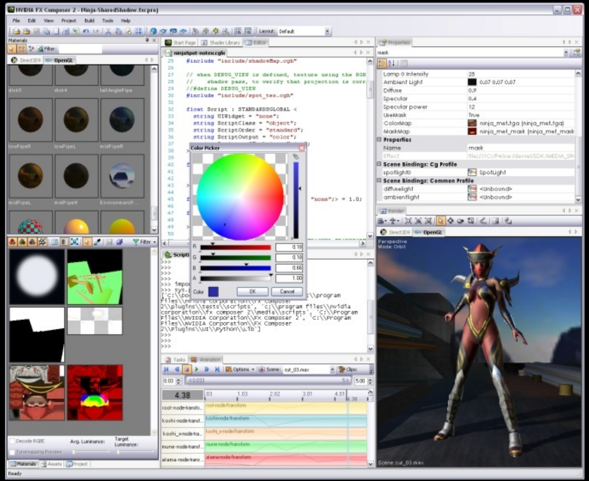
FX Composer 2.5, Shader Debugger and ShaderPerf

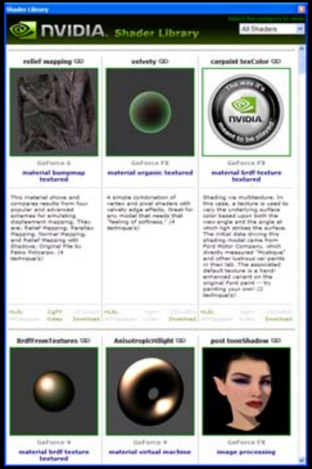


FX Composer

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 FX Composer 2 - Ninja-SharedShadow.fxproj
Color Picker
OK
Cancel
Properties
Scene Bindings: Cg Profile
Scene Bindings: Common Profile
Perspective
Materials
Assets
Project
Ready



relief mapping

GeForce 6

material bumpmap textured

This material shows and compares results from four popular and advanced schemes for simulating displacement mapping. They are: Relief Mapping, Parallax Mapping, Normal Mapping, and Relief Mapping with Shadows. Original File by Fabio Policarpo. (4 technique/s)

velvety

GeForce FX

material organic textured

A simple combination of vertex and pixel shaders with velvety edge effects. Great for any model that needs that "feeling of softness." (4 technique/s)

carpaint texColor

GeForce FX

material brdf texture textured

Shading via multitexture. In this case, a texture is used to vary the underlying surface color based upon both the view angle and the angle at which ligh strikes the surface. The initial data driving this shading model came from Ford Motor Company, which directly measured "Mystique" and other lustrous car paints in their lab. The associated default texture is a hand-enhanced variant on the original Ford paint -- try painting your own! (2 technique/s)

BrdfFromTextures

GeForce 4

material brdf texture textured

AnisotropicHilight

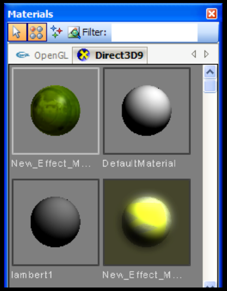
GeForce 4

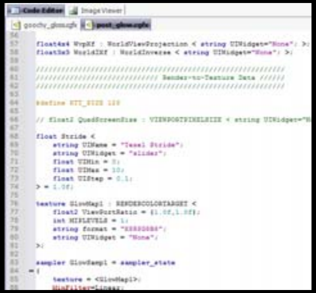
material virtual machine

post toonShadow

GeForce FX

image processing

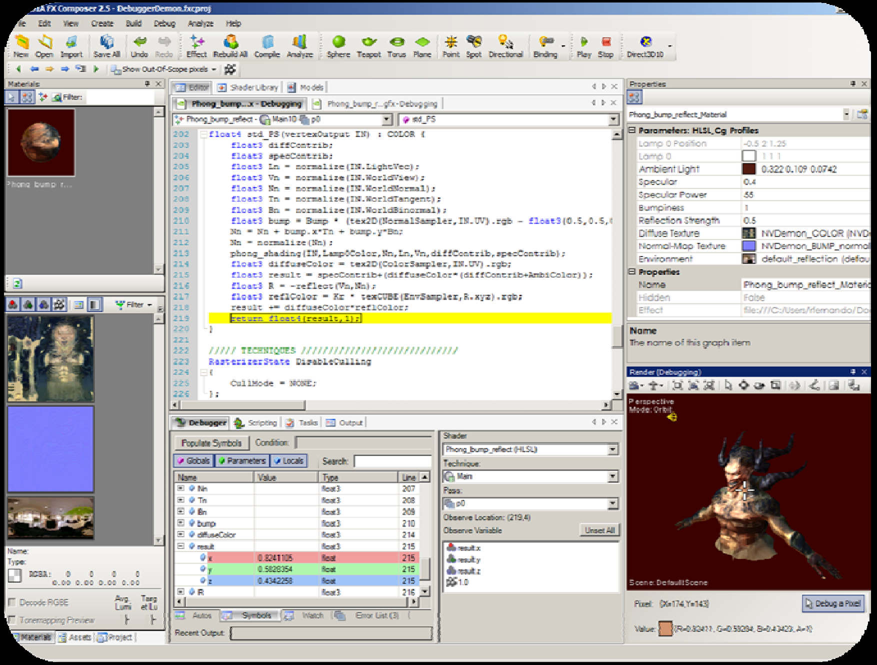




```
float4x4 WvpXf : WorldViewProjection < string UIWidget="None"; >;
float3x3 WorldIXf : WorldInverse < string UIWidget="None"; >;

////////////////////////////////////////////////////////////////////
////////////////////////////////// Render-to-Texture Data //////
////////////////////////////////////////////////////////////////////

#define RTT_SIZE 128

// float2 QuadScreenSize : VIEWPORTPIXELSIZE < string UIWidget="N

float Stride <
    string UIName = "Texel Stride";
    string UIWidget = "slider";
    float UIMin = 0;
    float UIMax = 10;
    float UIStep = 0.1;
> = 1.0f;

texture GlowMap1 : RENDERCOLORTARGET <
    float2 ViewPortRatio = {1.0f,1.0f};
    int MIPLEVELS = 1;
    string format = "X8R8G8B8";
    string UIWidget = "None";
>;

sampler GlowSamp1 = sampler_state
{
    texture = <GlowMap1>;
    MinFilter=Linear;
```

Shader Debugger

- 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



```
float4 std_PS(vertexOutput IN) : COLOR {
    float3 diffContrib;
    float3 specContrib;
    float3 Ln = normalize(IN.LightVec);
    float3 Vn = normalize(IN.WorldView);
    float3 Nn = normalize(IN.WorldNormal);
    float3 Tn = normalize(IN.WorldTangent);
    float3 Bn = normalize(IN.WorldBinormal);
    float3 bump = Bump * (tex2D(NormalSampler,IN.UV).rgb - float3(0.5,0.5,
    Nn = Nn + bump.x*Tn + bump.y*Bn;
    Nn = normalize(Nn);
    phong_shading(IN,Lamp0Color,Nn,Ln,Vn,diffContrib,specContrib);
    float3 diffuseColor = tex2D(ColorSampler,IN.UV).rgb;
    float3 result = specContrib+(diffuseColor*(diffContrib+AmbiColor));
    float3 R = -reflect(Vn,Nn);
    float3 reflColor = Kr * texCUBE(EnvSampler,R.xyz).rgb;
    result += diffuseColor*reflColor;
    return float4(result,1);
}

///// TECHNIQUES /////////////////////////////
RasterizerState DisableCulling
{
    CullMode = NONE;
};
```

Name	Value	Type	Line
Nn		float3	207
Tn		float3	208
Bn		float3	209
bump		float3	210
diffuseColor		float3	214
result		float3	215
x	0.8241105	float	215
y	0.5828354	float	215
z	0.4342258	float	215
R		float3	216

Shader: Phong_bump_reflect (HLSL)
Technique: Main
Pass: p0
Observe Location: (219,4)
Observe Variable: result.x, result.y, result.z, 1.0

Properties: Phong_bump_reflect_Material

Parameters: HLSL_Cg Profiles	
Lamp 0 Position	-0.5 2 1.25
Lamp 0	1 1 1
Ambient Light	0.322 0.109 0.0742
Specular	0.4
Specular Power	55
Bumpiness	1
Reflection Strength	0.5
Diffuse Texture	NVDemon_COLOR (NVD
Normal-Map Texture	NVDemon_BUMP_normal
Environment	default_reflection (defau
Properties	
Name	Phong_bump_reflect_Materia
Hidden	False
Effect	file:///C:/Users/rfernando/Do

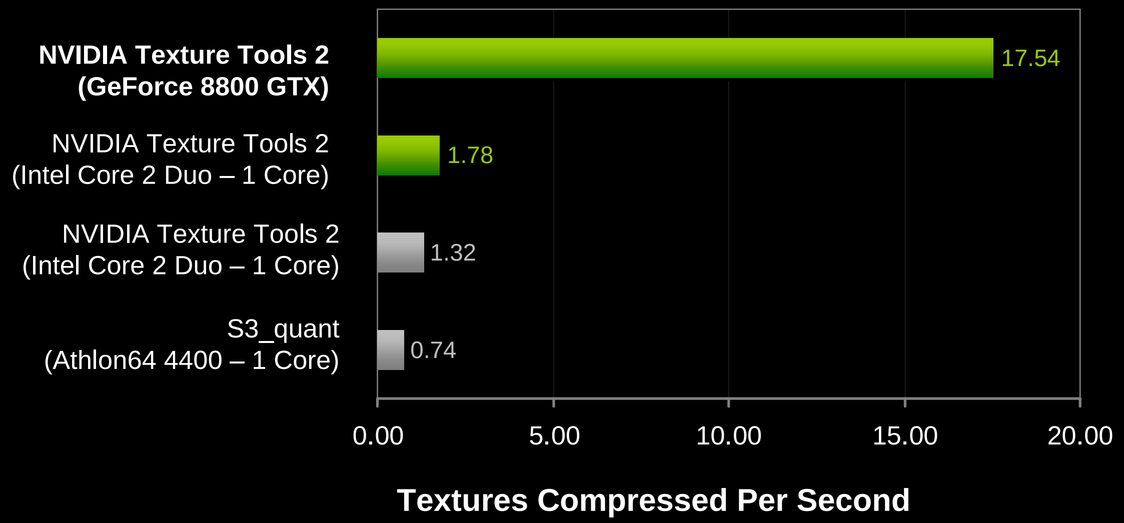
Name
The name of this graph item

Pixel: (X=174,Y=143)
Value: (R=0.82411, G=0.58284, B=0.43423, A=1)

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



Resource

- Online: downloads, videos, etc.

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

Feedback and Support: http://developer.nvidia.com/forums

VRD for PhysX

VRD

- VRD: Visual Remoter Debugger
  - A very useful tool for problem-solving

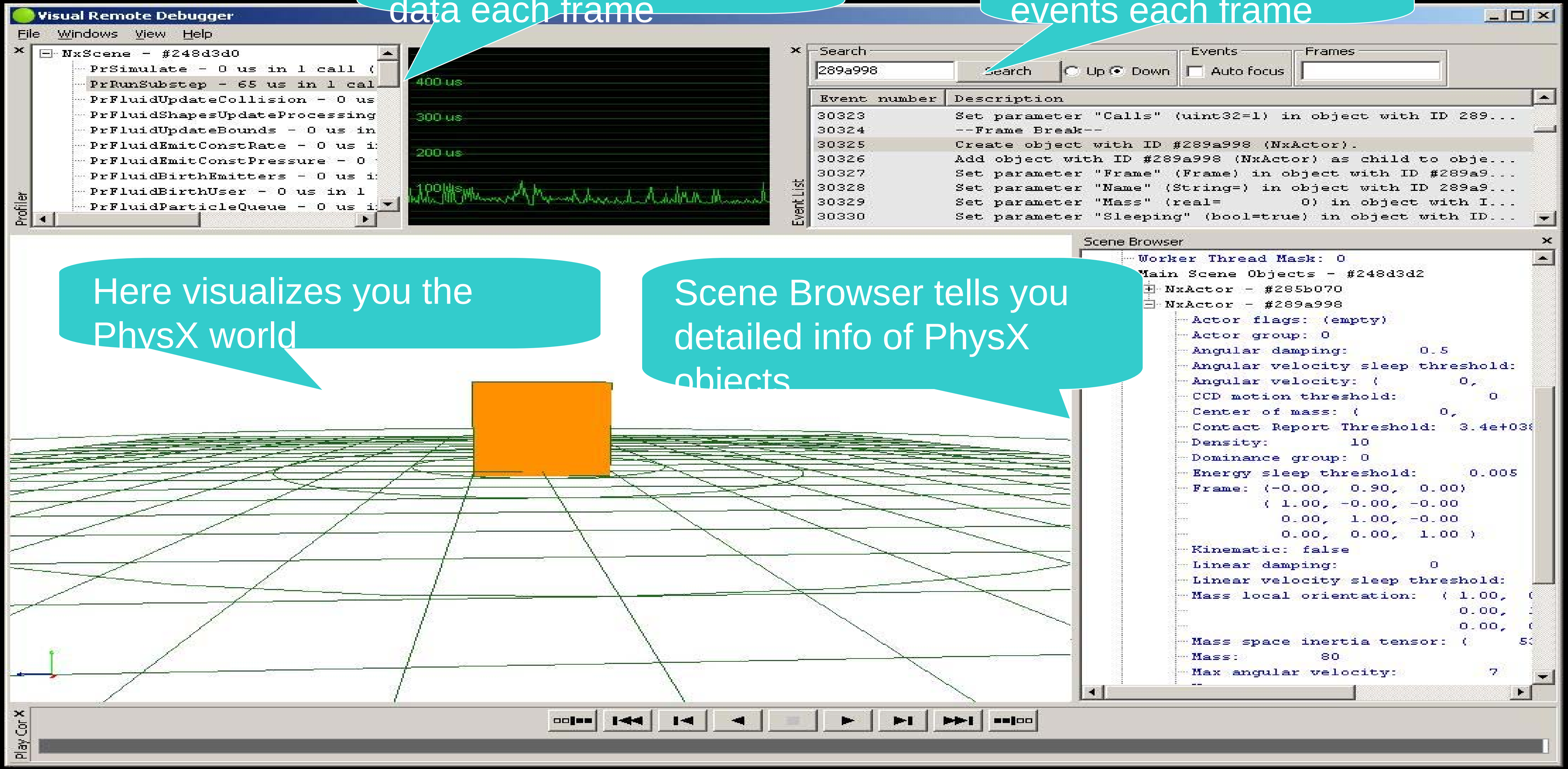
- Functionalities
  - Visualize PhysX objects to give you a pure physics world
  - Show detailed information of PhysX objects
  - Record detailed simulation status and can playback
  - Interact with game at real time when recording

Profiler shows Profiling data each frame

Event List shows events each frame

Here visualizes you the PhysX world

Scene Browser tells you detailed info of PhysX objects



```
NxScene - #248d3d0
    PrSimulate - 0 us in 1 call (
    PrRunSubstep - 65 us in 1 cal
    PrFluidUpdateCollision - 0 us
    PrFluidShapesUpdateProcessing
    PrFluidUpdateBounds - 0 us in
    PrFluidEmitConstRate - 0 us i
    PrFluidEmitConstPressure - 0
    PrFluidBirthEmitters - 0 us i
    PrFluidBirthUser - 0 us in 1
    PrFluidParticleQueue - 0 us i
```

Search: 289a998 Search ○ Up ● Down

□ Auto focus

```
Event number   Description
30323          Set parameter "Calls" (uint32=1) in object with ID 289...
30324          --Frame Break--
30325          Create object with ID #289a998 (NxActor).
30326          Add object with ID #289a998 (NxActor) as child to obje...
30327          Set parameter "Frame" (Frame) in object with ID #289a9...
30328          Set parameter "Name" (String=) in object with ID 289a9...
30329          Set parameter "Mass" (real=          0) in object with I...
30330          Set parameter "Sleeping" (bool=true) in object with ID...
```

```
Worker Thread Mask: 0
Main Scene Objects - #248d3d2
  NxActor - #285b070
  NxActor - #289a998
    Actor flags: (empty)
    Actor group: 0
    Angular damping:          0.5
    Angular velocity sleep threshold:
    Angular velocity: (          0,
    CCD motion threshold:              0
    Center of mass: (          0,
    Contact Report Threshold:  3.4e+03
    Density:          10
    Dominance group: 0
    Energy sleep threshold:      0.005
    Frame: (-0.00,  0.90,  0.00)
           ( 1.00, -0.00, -0.00
             0.00,  1.00, -0.00
             0.00,  0.00,  1.00 )
    Kinematic: false
    Linear damping:           0
    Linear velocity sleep threshold:
    Mass local orientation:  ( 1.00,
                               0.00,
                               0.00,
    Mass space inertia tensor: (     5
    Mass:          80
    Max angular velocity:          7
```

AgPerfMon for PhysX

AgPerfMon

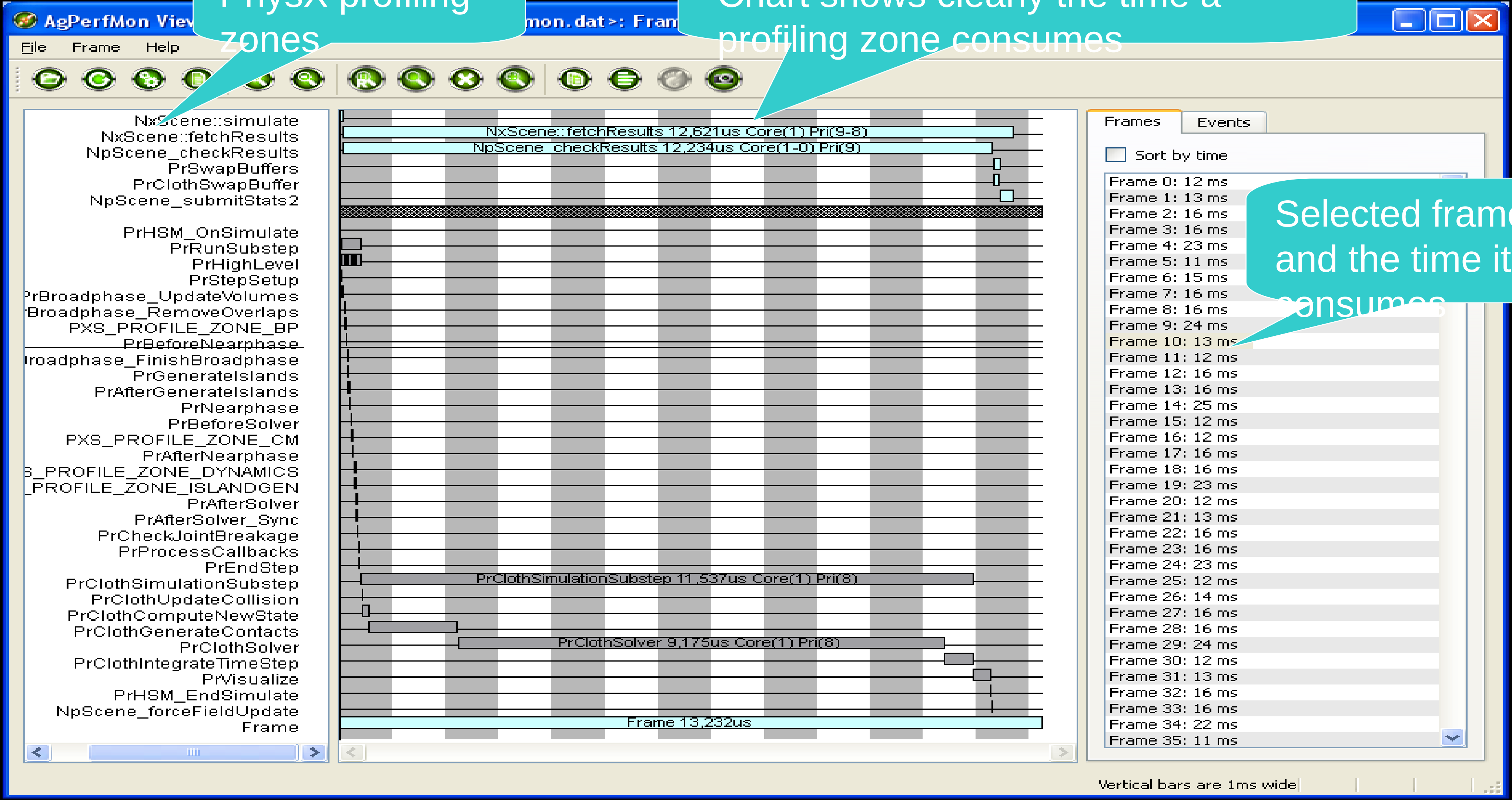
- AgPerfMon
  - a comprehensive profiling tool for physics applications

- Functionalities
  - Powerful event-logging allows you to understand exactly how your application is using PhysX
  - Event Viewer displays a Gantt chart of all events
  - Handy utility allows you to export your data to a graph, or to a comma-delimited format (CSV)

PhysX profiling zones

Chart shows clearly the time a profiling zone consumes

Selected frame # and the time it consumes



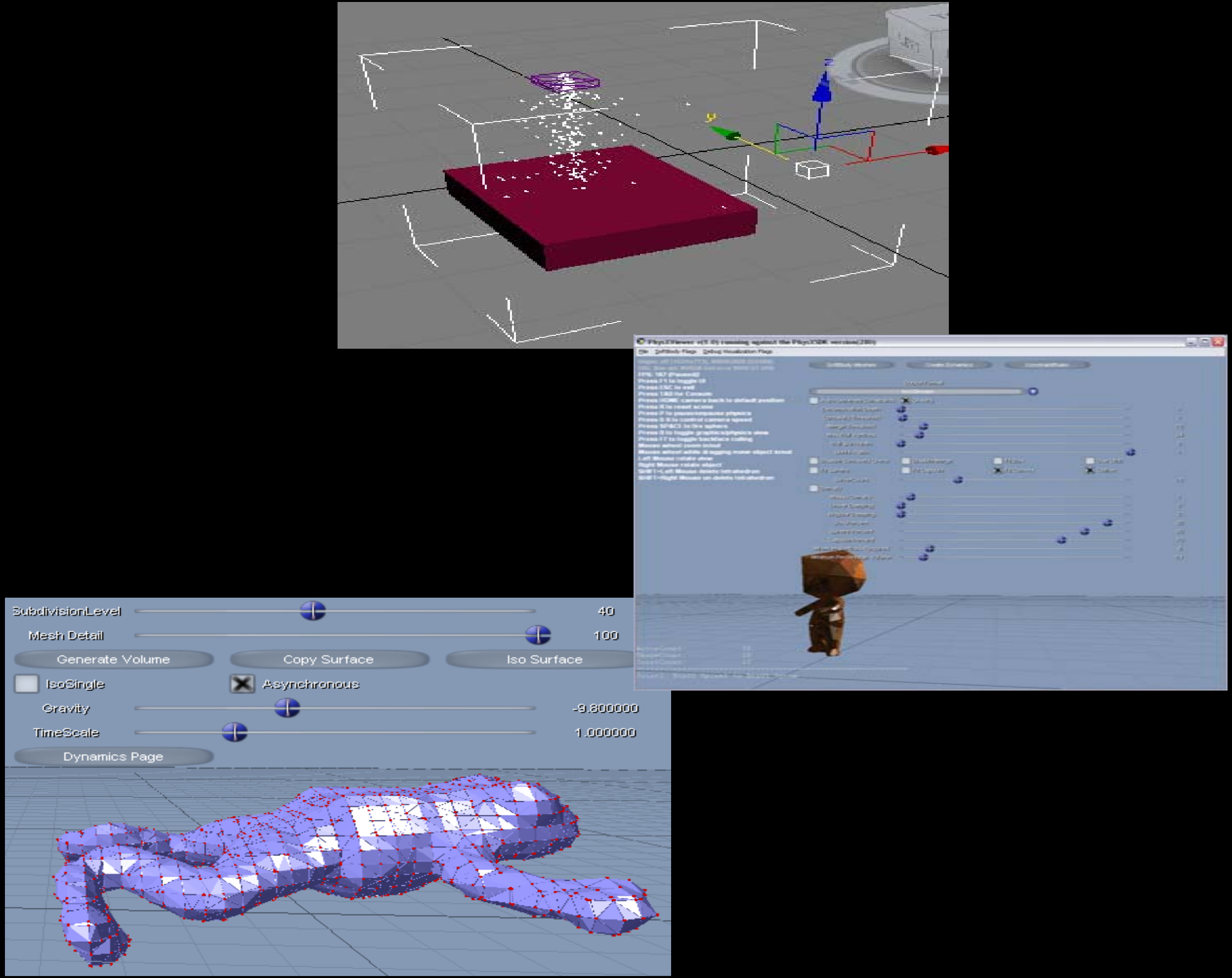
□ Sort by time

```
Frame 0: 12 ms
Frame 1: 13 ms
Frame 2: 16 ms
Frame 3: 16 ms
Frame 4: 23 ms
Frame 5: 11 ms
Frame 6: 15 ms
Frame 7: 16 ms
Frame 8: 16 ms
Frame 9: 24 ms
Frame 10: 13 ms
Frame 11: 12 ms
Frame 12: 16 ms
Frame 13: 16 ms
Frame 14: 25 ms
Frame 15: 12 ms
Frame 16: 12 ms
Frame 17: 16 ms
Frame 18: 16 ms
Frame 19: 23 ms
Frame 20: 12 ms
Frame 21: 13 ms
Frame 22: 16 ms
Frame 23: 16 ms
Frame 24: 23 ms
Frame 25: 12 ms
Frame 26: 14 ms
Frame 27: 16 ms
Frame 28: 16 ms
Frame 29: 24 ms
Frame 30: 12 ms
Frame 31: 13 ms
Frame 32: 16 ms
Frame 33: 16 ms
Frame 34: 22 ms
Frame 35: 11 ms
```

DCC Tools

Available PhysX Tools

- **Plugins for 3D Model tools**
  - **3DS Max Plugin**
  - **Maya Plugin**
- **PhysXViewer**
  - ***Softbody edit***
  - ***Ragdoll transfer***



SubdivisionLevel 40

Mesh Detail 100

☐ IsoSingle

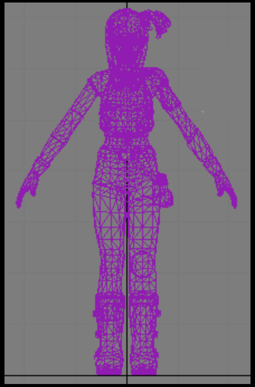
☒ Asynchronous

Gravity -9.800000

TimeScale 1.000000

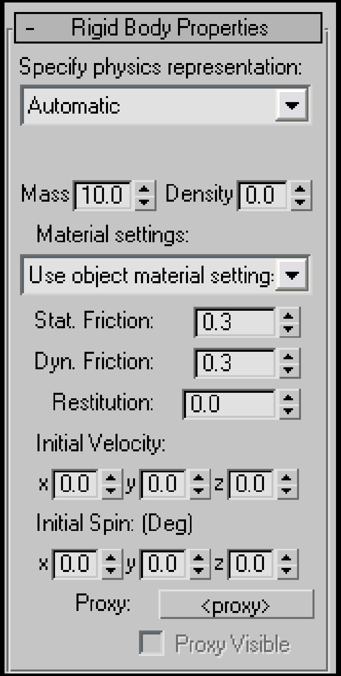
DCC Tools

PhysX Pipeline



1. Create Art Model
2. Model to PhysX Actors
   - PhysX Properties

     Shapes, Mass, Speed, Material
   - Use PhysX Tools
3. Export to NxuStream
4. Load it in games



- Rigid Body Properties

Specify physics representation: Automatic

Mass 10.0 Density 0.0

Material settings: Use object material setting:

Stat. Friction: 0.3

Dyn. Friction: 0.3

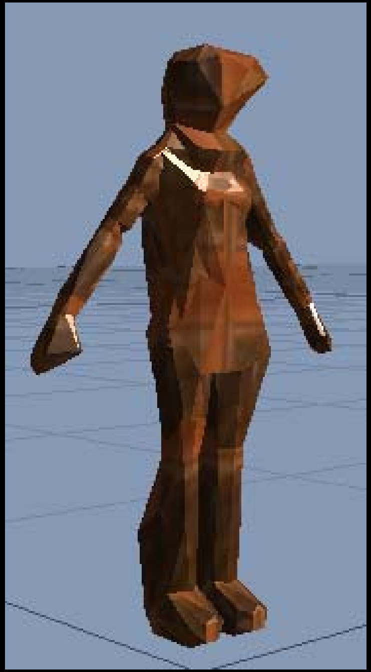
Restitution: 0.0

Initial Velocity: x 0.0 y 0.0 z 0.0

Initial Spin: (Deg) x 0.0 y 0.0 z 0.0

Proxy: <proxy>

□ Proxy Visible



Questions?