



DirectX 11 Debugging with Parallel Nsight™ 2.1 Update

March 2012

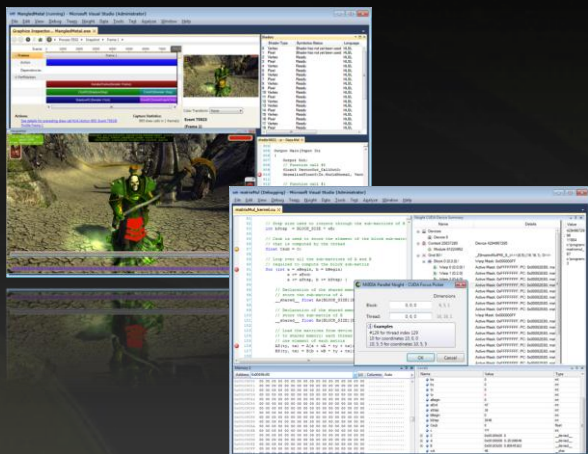
NVIDIA Parallel Nsight™



Visual Studio integrated development for GPU and CPU

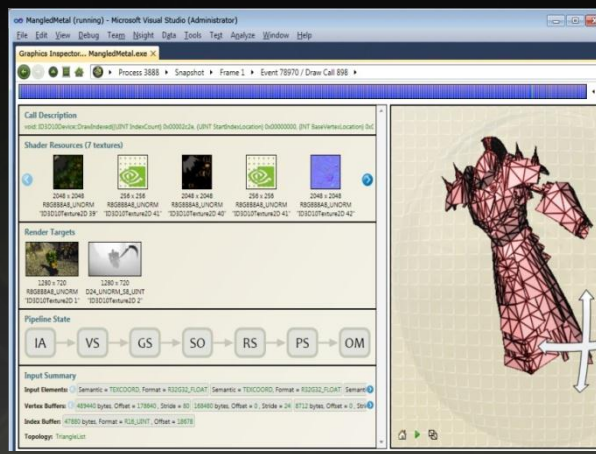


NVIDIA Parallel Nsight™



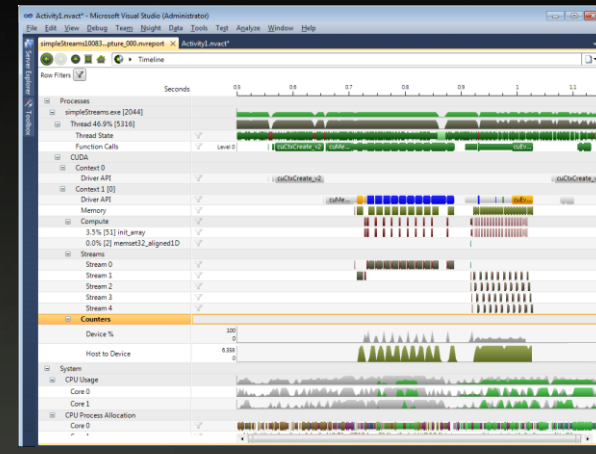
GPU Debugger

GPU native Compute and Graphics debugging
GPU breakpoints including complex conditionals
GPU memory views and exception reporting



Graphics Inspector

Real-time inspection of Direct3D API calls
Investigate GPU pipeline states
See contributing fragments with Pixel History
Profile frames to find GPU bottlenecks



System Analysis

View CPU & GPU events on a single timeline
Examine workload dependencies, memory transfers
CPU/OS, Compute, Direct3D and OpenGL Trace

Free License!

One computer, one NVIDIA GPU



Host + Target (32/64 bit)



- ✓ Frame Debugger
- ✓ Frame Profiler
- ✓ Frame Timings
- ✓ System Analysis

- Install appropriate NVIDIA driver
- Install Parallel Nsight Host and Monitor

Two computers, one with NVIDIA GPUs



Host (32/64-bit)



Target (32/64-bit)



- ✓ Frame Debugger
- ✓ Frame Profiler
- ✓ Frame Timings
- ✓ System Analysis
- ✓ Shader Debugger
- ✓ Pixel History

- Install appropriate NVIDIA driver on the Target System
- Install Parallel Nsight Monitor on the Target System
- Install Parallel Nsight Host on the Development System

One computer, two NVIDIA GPUs



Host + Target (32/64-bit)



- ✓ Frame Debugger
- ✓ Frame Profiler
- ✓ Frame Timings
- ✓ System Analysis
- ✓ Shader Debugger
- ✓ Pixel History

- Install appropriate NVIDIA driver
- Install Parallel Nsight Host and Monitor
- Configure Local Headless Debugging (see User's Manual)

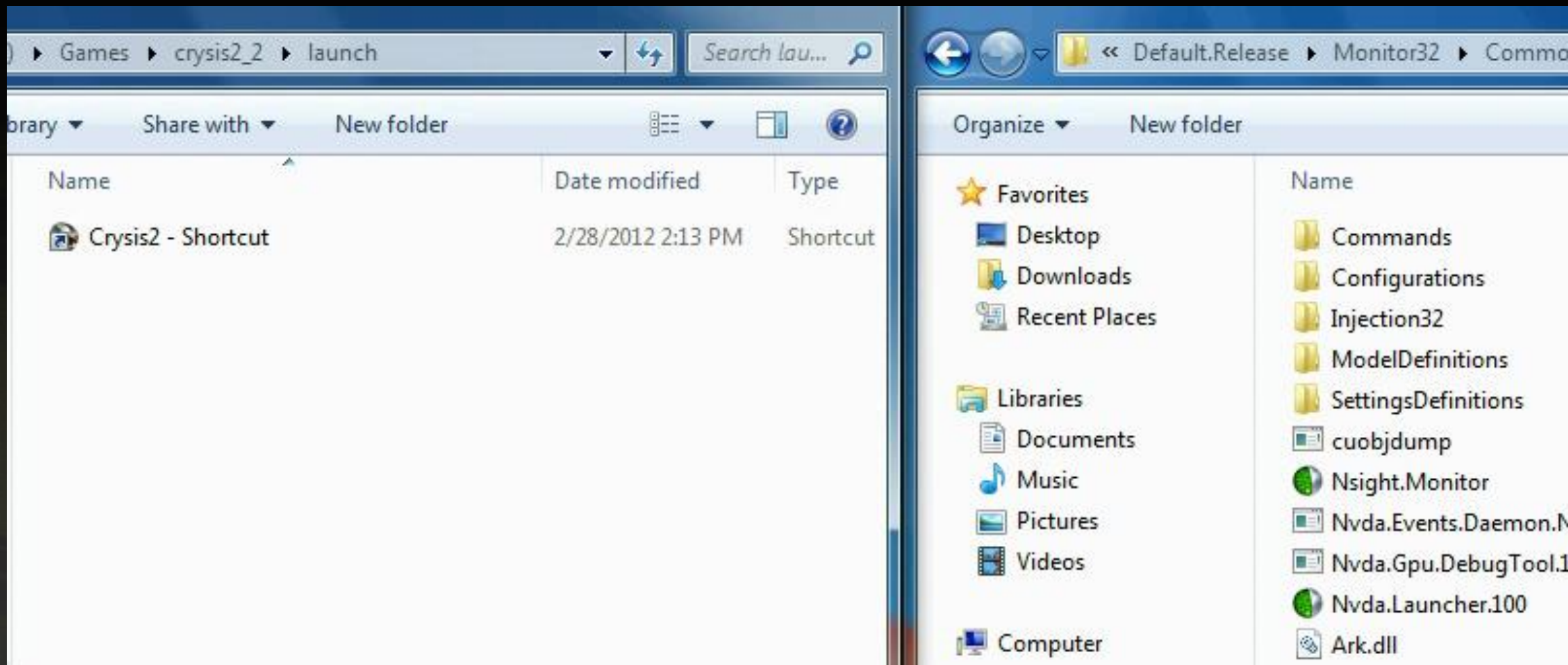


New in Parallel Nsight 2.1



- **Dynamic shader editing**
- **Show constant buffer with HLSL variable names**
- **Frame timings page**
- **Save and load profiler sessions**
- **Call stack trace capture**
- **Trace workload correlation**
- **5X trace overhead improvement**
- **New system information page**

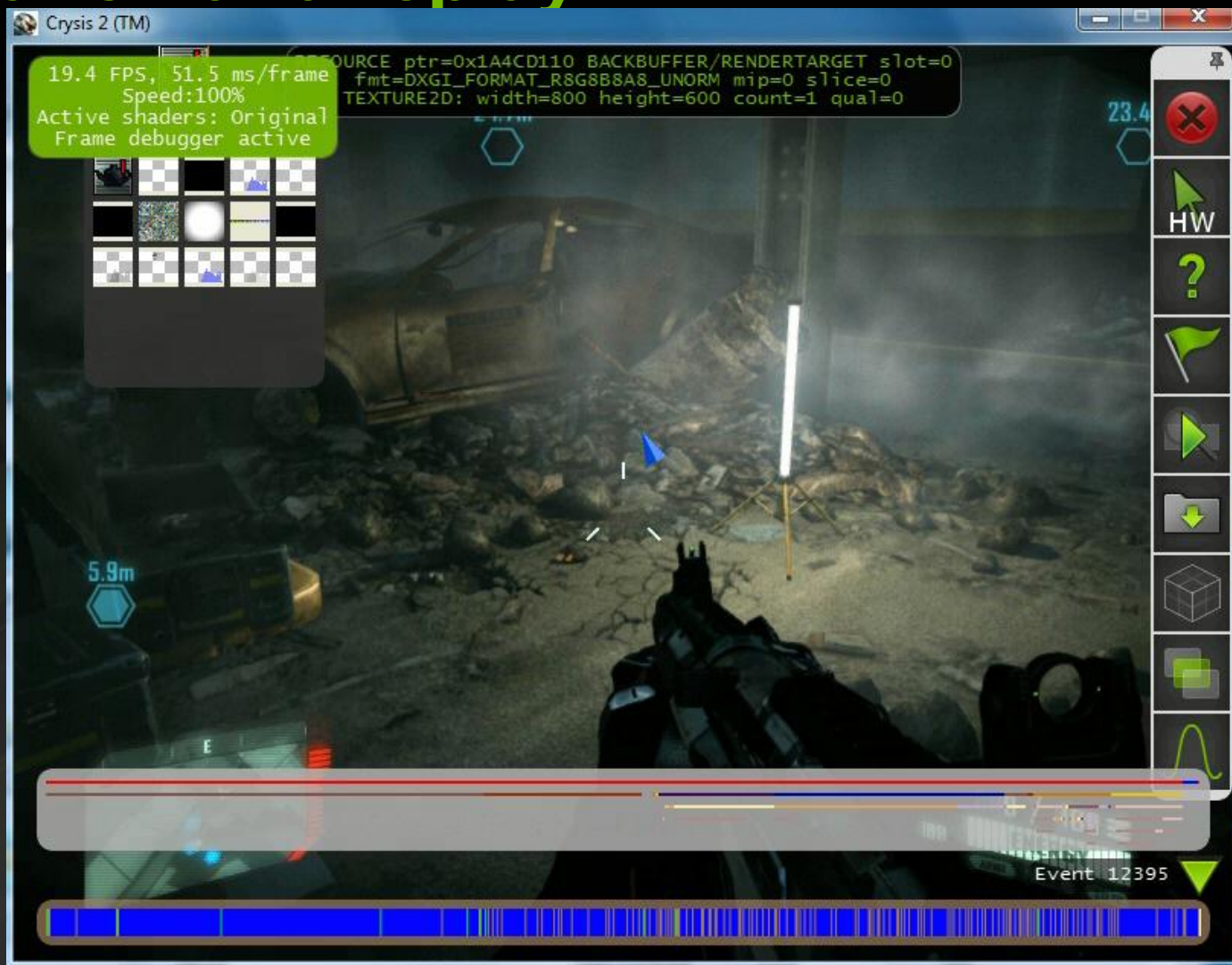
Target side HUD: Launch and running mode



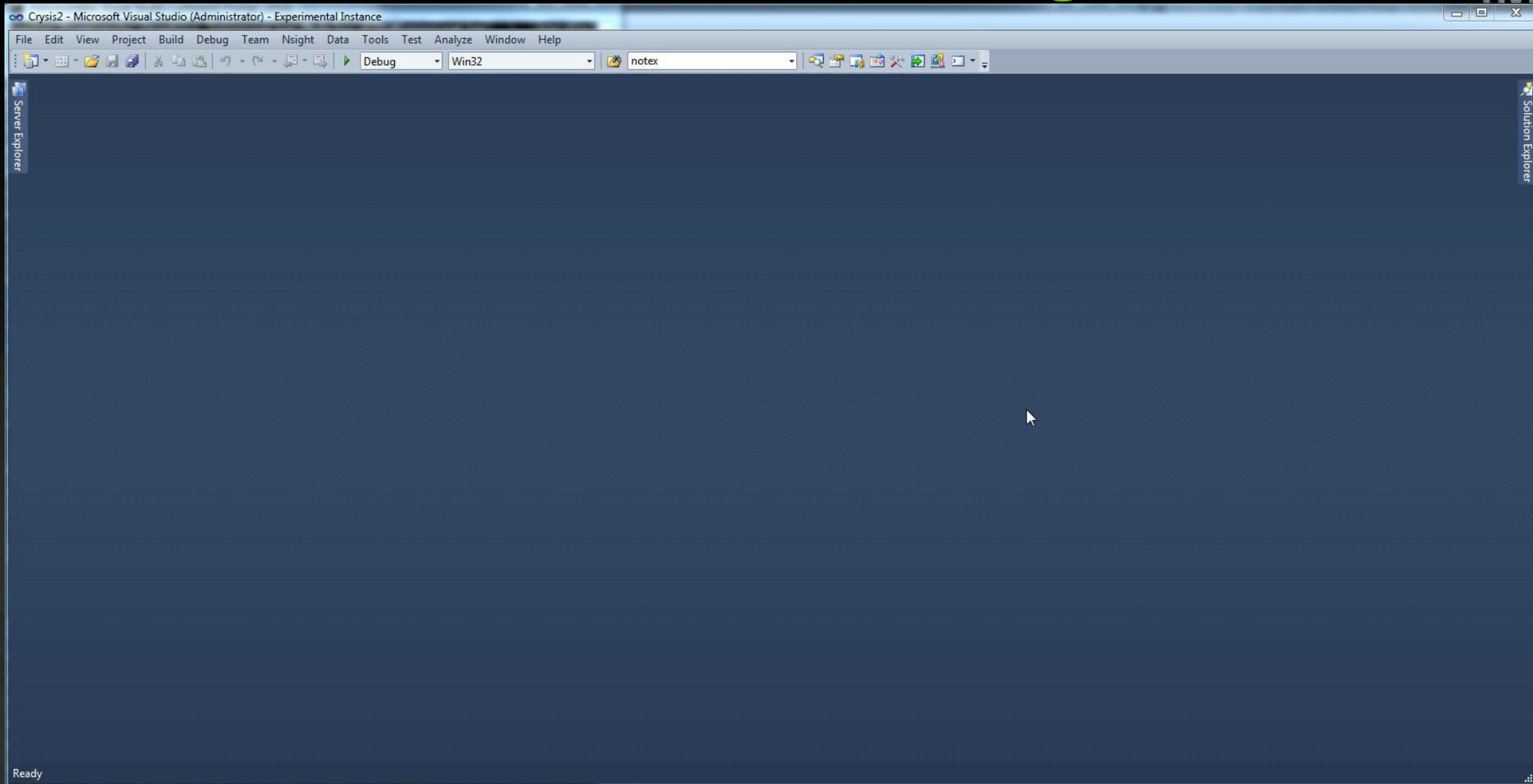
Target side HUD: Frame Debugger



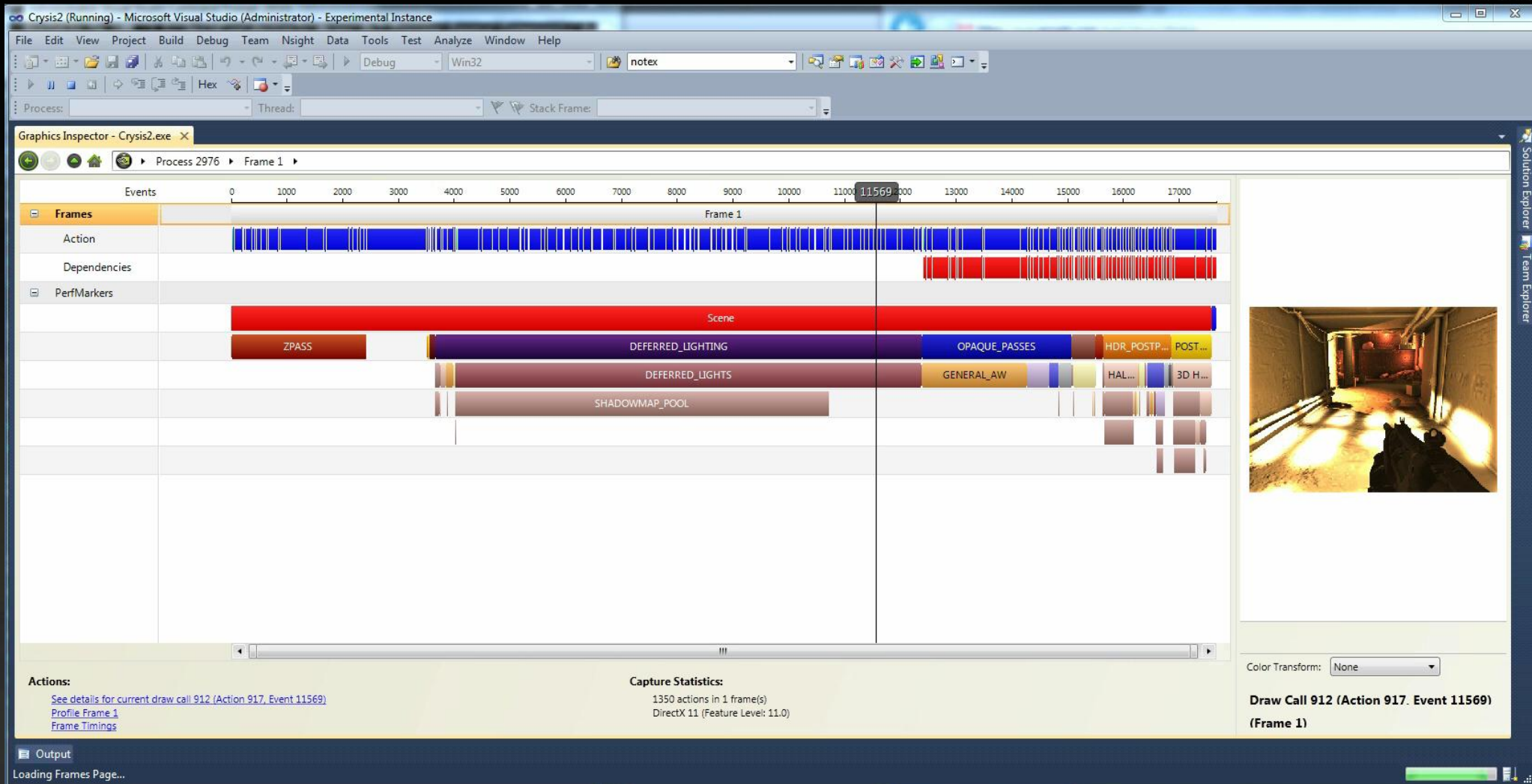
Serialization and replay



Host side: Launch and Frames Page



Draw call page



Pixel History and Shader Debugger



Crysis2 (Running) - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Nsight Data Tools Test ReShaper Analyze Window Help

Process: Thread: Stack Frame: Shader: Illum@IlluminationPS(80090000)()

Graphics Inspector - Crysis2.exe

Process 4444 Frame 1 Event 29355 / Action 2430

Call Description
void ID3D11DeviceContext::Draw(UINT VertexCount = 0x00000005, UINT StartVertexLocation = 0x00000000)

Shader Resources (16 resources)

VS:0 400 x 300	PS:0 256 x 256	PS:1 32 x 32	PS:2 800 x 600	PS:3 1024 x 32	PS:4 1024 x 32	PS:5 1024 x 32
R16G16B16A16_FLOAT	BC5_UNORM_SRGB	R8G8B8A8_UNORM	R32_FLOAT	R16G16B16A16_FLOAT	R16G16B16A16_FLOAT	R16G16B16A16_FLOAT
"ID3D11Texture2D 79"	"ID3D11Texture2D 3602"	"ID3D11Texture2D 2796"	"ID3D11Texture2D 128"	"ID3D11Texture2D 3844"	"ID3D11Texture2D 3845"	"ID3D11Texture2D 3846"

Render Targets

RT:0 800 x 600	DS 800 x 600
R16G16B16A16_FLOAT	D24_UNORM_S8_UINT
"ID3D11Texture2D 76"	"ID3D11Texture2D 66"

Pipeline State

IA → VS → HS → DS → GS → SO → RS → PS → OM → CS

Input Summary

Input Elements: Semantic = POSITION, Format = R32G32B32_FLOAT; Semantic = TEXCOORD, Format = R32G32B32A32_FLOAT; Semantic = COLOR, Format = R8G8B8A8_FLOAT

Vertex Buffers: 1048576 bytes, Offset = 123264, Stride = 64; 524288 bytes, Offset = 85984, Stride = 16

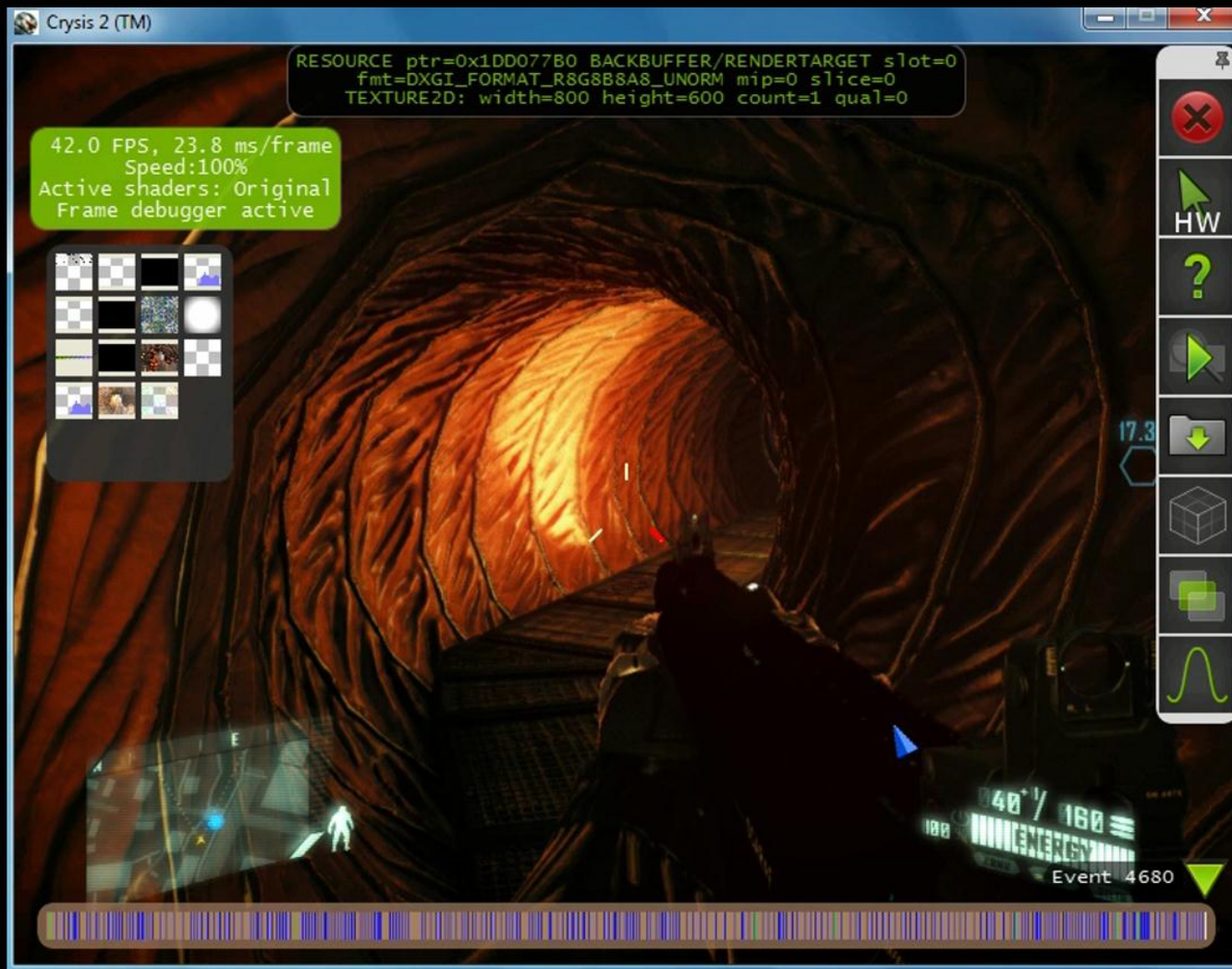
Index Buffer: 196608 bytes, Format = R16_UINT, Offset = 0

Topology: PointList

Shaders Watch 1 Locals Find and Replace Immediate Window Modules Threads Find Results 1 Call Stack Breakpoints Command Window Output Error List Find Symbol Results

Ready

Shader edit and continue – New in 2.1!



Wrap up



- **See Jeff Kiel's live demo of all Nsight features Thursday, 10-11**
- **My other talk: DirectX 11 Profiling with Parallel Nsight, Friday 1-2**
- **Questions?**