



NVIDIA Nsight™ VSE 3.0 Catzilla Engine Development in DirectX® 11 and OpenGL 4.2

Jeff Kiel - NVIDIA Corporation

Michal Szymczyk - CTO, Plastic

Michal Staniszewski - Creative Director, Plastic



Agenda

- Catzilla Demo/Benchmark
 - Developed by Plastic, Platige Image and ALLPlayer
 - OpenGL 4.2 and DirectX®
- NVIDIA Nsight Visual Studio Edition Overview
- Problems During Development & Nsight Helped
- Q&A

Motivation behind the project

- There was a chance to do it - European Union Competitiveness & Innovation Programme (CIP)
- Old engine was completely tailored for PS3(Datura® development)
- We wanted to prepare new engine for next gen machines



Development timeline

- Engine development before preproduction started - 2 months
- Preproduction - 1 month
- Production - 2 months
- Post Production - 2 months, 6 months total

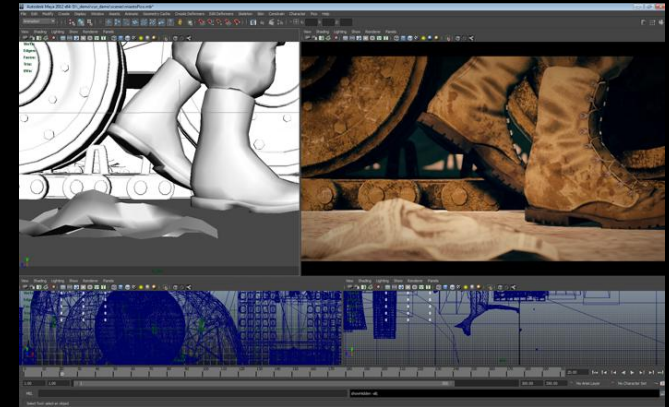
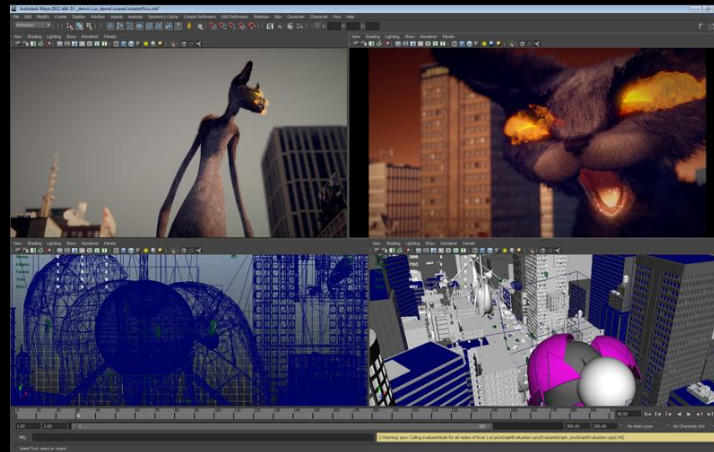


Demo



Features of the engine

- Post processing (HDR, adaptive luminance, DOF with Bokeh)
- Fur based on geometry shader generated fins
- Physically based lighting
- Engine completely integrated with Autodesk®Maya



OpenGL/DirectX® 11

- OpenGL needed because of WindowsXP support, possibility to port to MacOSX and Linux
- Still using OpenGL inside of Autodesk®Maya
- Possibility to compare performance between two APIs
- DirectX still faster?



Benchmark modes

- Physics - CPU based using PhysX® - test
- Fur (Geometry Shaders) test
- Fluid (GPU Memory Bandwidth) test
- Raymarching (GPU ALU) test



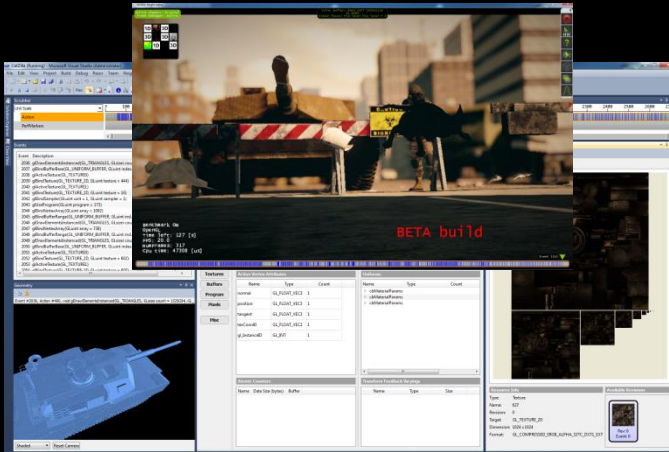
NVIDIA Nsight Visual Studio Edition

Visual Studio integrated development for GPU and CPU



NVIDIA Nsight Visual Studio Edition

Supports Direct3D 9/11 and OpenGL 4.2

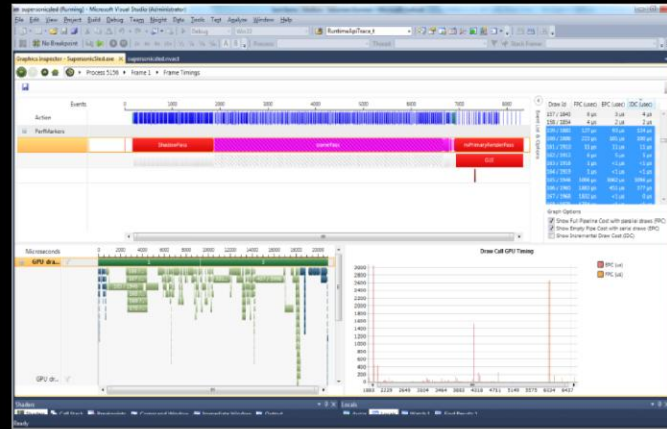


Frame Debugger

- HUD for scene scrubbing
- State inspection at each draw call
- Real time frame capture and replay
- Source code serialization for D3D9/11

HLSL and GLSL Shader Debugger

- Native GPU shader debugging and GPU memory views
- Complex conditional breakpoints and Pixel History
- Local, single GPU shader debugging

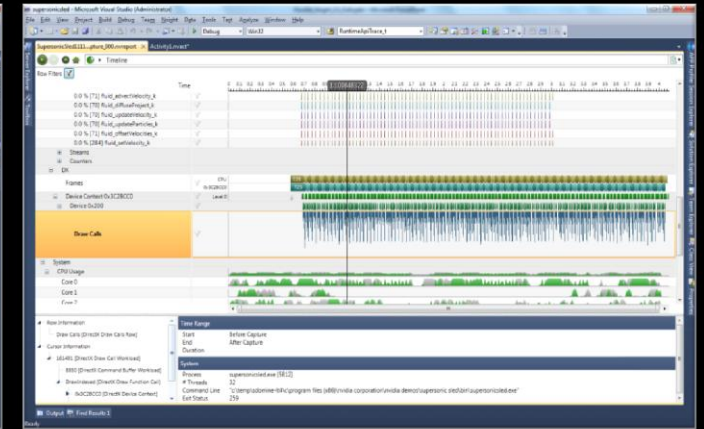


Frame Profiler

- Automatic GPU bottleneck determination
- Draw call and frame timings
- Direct3D Perf Markers and render state grouping/sorting

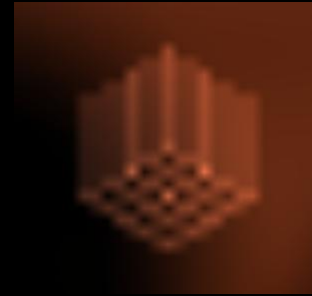
Application and System Trace

- Inspect Direct3D & OpenGL, CPU & GPU workloads
- Correlate threads, call stack, API calls, WDDM kernel queues and resulting GPU workloads
- Concurrent draw call execution and memory transfer trace



Problems During Development

#1 Broken Bokeh Filter

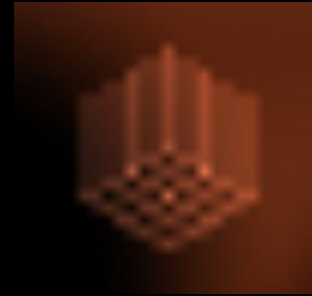


Use similar algorithm to what DICE showed at SIGGRAPH 2011

Bokeh pass has extra bright lines & chess board pattern

Problems During Development

#1 Broken Bokeh Filter



```
Frame0Part00.cpp X Main.cpp
(Global Scope) RunFra

// Event 49 of 13776
pID3D11DeviceContext_uidof_43->IASetPrimitiveTopology(D3D_PRIMITIVE_TOPOLOGY_POINTLIST);

// Event 50 of 13776
pID3D11DeviceContext_uidof_43->IASetInputLayout(pID3D11InputLayout_uidof_12484);

// Event 51 of 13776
static ID3D11Buffer* pID3D11Buffer_temp_6[3] = { pID3D11Buffer_uidof_6390, pID3D11Buffer_uidof_6416, pID3D11Buffer_uidof_6416 };
static UINT UINT_temp_2[3] = { 36, 16, 8 };
pID3D11DeviceContext_uidof_43->IASetVertexBuffers(0, 3, pID3D11Buffer_temp_6, UINT_temp_2, ((UINT*)array_of_0s));

// Event 52 of 13776
pID3D11DeviceContext_uidof_43->IASetIndexBuffer(pID3D11Buffer_uidof_6442, DXGI_FORMAT_R16_UINT, 0);

// Event 53 of 13776
static ID3D11RenderTargetView* pID3D11RenderTargetView_temp_1[4] = { pID3D11RenderTargetView_uidof_12484, pID3D11RenderTargetView_uidof_12484, pID3D11RenderTargetView_uidof_12484, pID3D11RenderTargetView_uidof_12484 };
pID3D11DeviceContext_uidof_43->OMSetRenderTargets(4, pID3D11RenderTargetView_temp_1, pID3D11DepthStencilState_uidof_126);

// Event 54 of 13776, Draw 1 of 2223
pID3D11DeviceContext_uidof_43->DrawInstanced(7706, 1, 0, 0);

// Event 55 of 13776
static ID3D11Buffer* pID3D11Buffer_temp_7[1] = { NULL };
pID3D11DeviceContext_uidof_43->SOSetTargets(1, pID3D11Buffer_temp_7, ((UINT*)array_of_0s));

// Event 56 of 13776
pID3D11DeviceContext_uidof_43->OMSetBlendState(pID3D11BlendState_uidof_46, ((FLOAT*)array_of_0s), 0xFFFFFFFF);

// Event 57 of 13776
pID3D11DeviceContext_uidof_43->OMSetDepthStencilState(pID3D11DepthStencilState_uidof_126, 0);

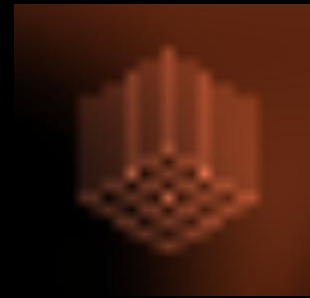
// Event 58 of 13776
pID3D11DeviceContext_uidof_43->RSSetState(pID3D11RasterizerState_uidof_86);

// Event 59 of 13776
pID3D11DeviceContext_uidof_43->VSSetShader(pID3D11VertexShader_uidof_117, NULL, 0);
```

Bug repro using D3D source generation for captured frames...edit generated code to quickly try debugging ideas

Problems During Development

#1 Broken Bokeh Filter



Pixel History helps narrow down the draw calls

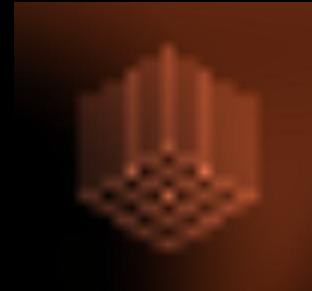
The screenshot shows the Visual Studio Graphics Inspector interface. The Pixel History window is open, displaying a list of draw calls. The table below represents the data shown in the Pixel History window:

Event	RT Before	Source	RT After	Z
void ID3D11DeviceContext::Event 7546	R: 0.82 G: 0.45 B: 0.28 A: 1.00	R: 0.00 G: 0.00 B: 0.00 A: 0.00	R: 0.00 G: 0.00 B: 0.00 A: 0.00	
void ID3D11DeviceContext::Event 13160 Event in 13160 / Action 2163	R: 0.00 G: 0.00 B: 0.00 A: 0.00	N/A Recess e shader , data not available	R: 0.36 G: 0.14 B: 0.05 A: 1.00	1
void ID3D11DeviceContext::Event 13237 Event in 13237 / Action 2169	R: 0.36 G: 0.14 B: 0.05 A: 1.00	N/A Recess e shader , data not available	R: 0.42 G: 0.18 B: 0.08 A: 0.90	1
void ID3D11DeviceContext::Event 13641 Event in 13641 / Action 2215 Debug Pixel	R: 0.42 G: 0.18 B: 0.08 A: 0.90	R: 0.00 G: 0.00 B: 0.00 A: 0.00	R: 5.19 G: 4.70 B: 3.26 A: 1.00	0
void ID3D11DeviceContext::Event 13672 Event in 13672 / Action 2217	R: 9.19 G: 4.70 B: 3.26 A: 1.00	N/A Recess e shader , data not available	R: 0.92 G: 0.45 B: 0.28 A: 1.00	1

The interface also includes a main viewport showing a 3D scene with a bokeh effect, a Color Transform dropdown, and a Shaders window at the bottom.

Problems During Development

#1 Broken Bokeh Filter



Dynamic Shader Editing to
test the bug fix

```
catzillaApp_2013_03_22_10_25_09_vc10 (Running) - Microsoft Visual Studio (Administrator) - Experimental Instance
File Edit View Project Build Debug Team Noight Data Tools Test Analyze Window Help
Debug Win32 GfxDebuggerIgnoreUnsup Hex
lastStagePostProcessing.hlsl Graphics Inspector...013_03_22_10_25_09

// Normalize the color sum
finalColor = colorSum / totalContribution;
out_colorRGBAUp = float4( finalColor.rgb, centerColor.a );

// left down pass
//
colorSum = centerColor;
//colorSum = 0;
totalContribution = 1;

// Run through all filter taps
for(int k = 0; k < numDiscSamples; k++)
{
    // Compute sample coordinates
    float2 kernelValue = dofKernelDice2[k].xy;
    float2 offset = kernelValue * cocSize;
    float2 tapCoord = centerCoord + offset;

    // Fetch the tap sample
    float4 tapColor = gColor0Tex.SampleLevel( gColor0TexSamp, tapCoord, 0 );

    float tapContribution = tapColor.a;

    // Accumulate color and sample contribution
    colorSum += tapColor * tapContribution;
    totalContribution += tapContribution;
}

//if ( totalContribution > 1 )
//{
//    colorSum -= centerColor;
//    totalContribution -= 1;
//}

// Normalize the color sum
finalColor = colorSum / totalContribution;
out_colorRGBALeftDown = float4( finalColor.rgb + out_colorRGBAUp.rgb, centerColor.a );
}

void ps_BokehDiceOptPass1_low(
    in vs_output_ScreenQuad IN ,
    out float4 out_colorRGBAUp : SV_Target0 ,
    out float4 out_colorRGBALeftDown : SV_Target1 )
{
    BokehDiceOptPass1Function( out_colorRGBAUp, out_colorRGBALeftDown, IN.texCoord01, 8 );
}

void ps_BokehDiceOptPass1_normal(
    in vs_output_ScreenQuad IN ,
    out float4 out_colorRGBAUp : SV_Target0 ,
    out float4 out_colorRGBALeftDown : SV_Target1 )
{
    BokehDiceOptPass1Function( out_colorRGBAUp, out_colorRGBALeftDown, IN.texCoord01, 16 );
}

// Normalize the color sum
finalColor = colorSum / totalContribution;
out_colorRGBAUp = float4( finalColor.rgb, centerColor.a );

// left down pass
//
colorSum = centerColor;
//colorSum = 0;
totalContribution = 1;

// Run through all filter taps
for(int k = 0; k < numDiscSamples; k++)
{
    // Compute sample coordinates
    float2 kernelValue = dofKernelDice2[k].xy;
    float2 offset = kernelValue * cocSize;
    float2 tapCoord = centerCoord + offset;

    // Fetch the tap sample
    float4 tapColor = gColor0Tex.SampleLevel( gColor0TexSamp, tapCoord, 0 );

    float tapContribution = tapColor.a;

    // Accumulate color and sample contribution
    colorSum += tapColor * tapContribution;
    totalContribution += tapContribution;
}

if ( totalContribution > 1 )
{
    colorSum -= centerColor;
    totalContribution -= 1;
}

// Normalize the color sum
finalColor = colorSum / totalContribution;
out_colorRGBALeftDown = float4( finalColor.rgb + out_colorRGBAUp.rgb, centerColor.a );
}

void ps_BokehDiceOptPass1_low(
    in vs_output_ScreenQuad IN ,
    out float4 out_colorRGBAUp : SV_Target0 ,
    out float4 out_colorRGBALeftDown : SV_Target1 )
{
    BokehDiceOptPass1Function( out_colorRGBAUp, out_colorRGBALeftDown, IN.texCoord01, 8 );
}

void ps_BokehDiceOptPass1_normal(
    in vs_output_ScreenQuad IN ,
    out float4 out_colorRGBAUp : SV_Target0 ,
    out float4 out_colorRGBALeftDown : SV_Target1 )
{
    BokehDiceOptPass1Function( out_colorRGBAUp, out_colorRGBALeftDown, IN.texCoord01, 16 );
}

Memory 1
Address: Columns:
Shaders
Name Shader Type Symbolics Status Language Technique Pass
Shaders Breakpoints Call Stack Autos Locals Watch 1 Find Results 1
Ln 1049 Col 14 Ch 5 INS
```


Problems During Development

#2 Redundant State Changes

catzillaApp130408_...apture_000.nvreport X Activity1.nvact*

Summary Report

Session Overview

Summary of session information related to the captured data.

- [Session Summary](#)
- [Timeline](#)
- [Activity](#)

catzillaApp.exe (Trace Application)
Captured 4.21 seconds of data on 4/8/2013 4:02:49 PM

Arguments: data/ -fake -opengl -profileNormal -exit0 -music0 -secDisp1 -fullscreen0 -nodemo -notest1 -notest2 -notest3 -notest4
Working Dir: c:\temp\catzilla-red
Connection: DTPNCWin764-01

OpenGL Overview

Summary of captured OpenGL activity.

- [API Calls | Summary](#)
- [Draw Calls | Transfers](#)
- [Frames](#)

CPU All Cores			
API Time (%)	Min	Avg	Max
35.0	API/F	8,615.6	

GPU 0 GeForce GT 640			
Utilization (%)	Min	Avg	Max
91.6	ms/F	110	173
	Cmd%	91.9	96.8
	#Draw	1,012	1,042

⚠ Device context 0xFFFFFFF1010E9C on render context 0x40000 has 39 CPU frames, but only 38 GPU frames.

Analysis Summary shows high API Call (8615.6) to Draw Call (1012) ratio

Problems During Development

#2 Redundant State Changes

Event	Description	CPU Duration (ns)	GPU Duration (ns)
3330	glBindBuffer(GL_ARRAY_BUFFER, GLuint buffer = 811)	6813	0
3331	glVertexAttribPointer(GLuint index = 1, GLint size = 4, GL_SHORT, GL_TRUE, GLsizei stride ...)	9880	0
3332	glVertexAttribPointer(GLuint index = 2, GLint size = 2, GL_FLOAT, GL_FALSE, GLsizei stride ...)	9539	0
3333	glBindBufferRange(GL_UNIFORM_BUFFER, GLuint index = 1, GLuint buffer = 5, GLintptr o...)	12265	0
3334	glDrawElementsInstanced(GL_TRIANGLES, GLsizei count = 6, GL_UNSIGNED_SHORT, GLv...)	97780	10432
3335	glBindBufferBase(GL_UNIFORM_BUFFER, GLuint index = 2, GLuint buffer = 25)	13287	0
3336	glActiveTexture(GL_TEXTURE0)	340	0
3337	glBindTexture(GL_TEXTURE_2D, GLuint texture = 224)	8517	0
3338	glActiveTexture(GL_TEXTURE1)	340	0
3339	glBindTexture(GL_TEXTURE_2D, GLuint texture = 19)	6132	0
3340	glUseProgram(GLuint program = 137)	8858	0
3341	glBindBuffer(GL_ARRAY_BUFFER, GLuint buffer = 759)	35092	0
3342	glVertexAttribPointer(GLuint index = 0, GLint size = 3, GL_FLOAT, GL_FALSE, GLsizei stride ...)	11583	0
3343	glBindBuffer(GL_ARRAY_BUFFER, GLuint buffer = 760)	7495	0
3344	glVertexAttribPointer(GLuint index = 1, GLint size = 4, GL_SHORT, GL_TRUE, GLsizei stride ...)	10220	0
3345	glVertexAttribPointer(GLuint index = 2, GLint size = 2, GL_UNSIGNED_SHORT, GL_TRUE, G...)	9539	0
3346	glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, GLuint buffer = 761)	22486	0
3347	glBindBufferRange(GL_UNIFORM_BUFFER, GLuint index = 1, GLuint buffer = 5, GLintptr o...)	12265	0
3348	glDrawElementsInstanced(GL_TRIANGLES, GLsizei count = 36, GL_UNSIGNED_SHORT, GLv...)	97780	22912
3349	glBindBufferBase(GL_UNIFORM_BUFFER, GLuint index = 2, GLuint buffer = 121)	14990	0
3350	glActiveTexture(GL_TEXTURE0)	340	0
3351	glBindTexture(GL_TEXTURE_2D, GLuint texture = 277)	8517	0
3352	glActiveTexture(GL_TEXTURE1)	340	0
3353	glBindTexture(GL_TEXTURE_2D, GLuint texture = 19)	6132	0
3354	glUseProgram(GLuint program = 425)	11583	0
3355	glBindBuffer(GL_ARRAY_BUFFER, GLuint buffer = 216)	8517	0
3356	glVertexAttribPointer(GLuint index = 0, GLint size = 3, GL_FLOAT, GL_FALSE, GLsizei stride ...)	11583	0
3357	glBindBuffer(GL_ARRAY_BUFFER, GLuint buffer = 217)	7495	0
3358	glVertexAttribPointer(GLuint index = 1, GLint size = 4, GL_SHORT, GL_TRUE, GLsizei stride ...)	9880	0
3359	glVertexAttribPointer(GLuint index = 2, GLint size = 2, GL_FLOAT, GL_FALSE, GLsizei stride ...)	9880	0
3360	glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, GLuint buffer = 218)	22145	0

Look at Event List to help confirm redundant state calls

Problems During Development

#2 Redundant State Changes

API Inspector

Vtx Spec

VS

TCS

TES

GS

XFB

Raster

FS

Pix Ops

FB

Textures

Buffers

Program

Pixels

Misc

Call Description

void glDrawElementsInstanced(GL_TRIANGLES, GLsizei count = 36, GL_UNSIGNED_SHORT, GLvoid* indices = 0x00000000, GLsizei primcount = 1)

Vertex Attributes for Vertex Array Object: 1, VAO Element Array Buffer: 761

Index	Enabled	Type	Size	Buffer	Stride	Pointer	Normalized	Integer	Divisor
0	GL_TRUE	GL_FLOAT	3	759	12	0	GL_FALSE	GL_FALSE	0
1	GL_TRUE	GL_SHORT	4	760	12	0	GL_TRUE	GL_FALSE	0
2	GL_TRUE	GL_UNSIGNED_SHORT	2	760	12	8	GL_TRUE	GL_FALSE	0
3	GL_FALSE	GL_UNSIGNED_BYTE	4	3704	16	12	GL_FALSE	GL_FALSE	0
4	GL_FALSE	GL_UNSIGNED_BYTE	4	2776	16	12	GL_FALSE	GL_FALSE	0

Generic Attributes

Index	Type	Size	Normalized	Integer	X	Y	Z	W
0	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
1	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
2	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
3	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
4	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
5	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
6	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
7	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
8	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
9	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
10	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0
11	GL_FLOAT	16	GL_FALSE	GL_FALSE	0.0	0.0	0.0	1.0

Viewports And Depth Ranges

Index	X	Y	Width	Height	Near	Far
0	0.0	0.0	1280.0	720.0	0.0	1.0
1	0.0	0.0	1280.0	720.0	0.0	1.0
2	0.0	0.0	1280.0	720.0	0.0	1.0
3	0.0	0.0	1280.0	720.0	0.0	1.0

Element Array Buffer

Name [761](#)

Type GL_UNSIGNED_SHORT

Size 2952

Array Buffer

Name [760](#)

Draw Indirect Buffer

Name 0

Coloring State

Clamp Read Color GL_TRUE

Provoking Vertex GL_LAST_VERTEX_CONVENTION

Transformation State

Depth Clamp GL_FALSE

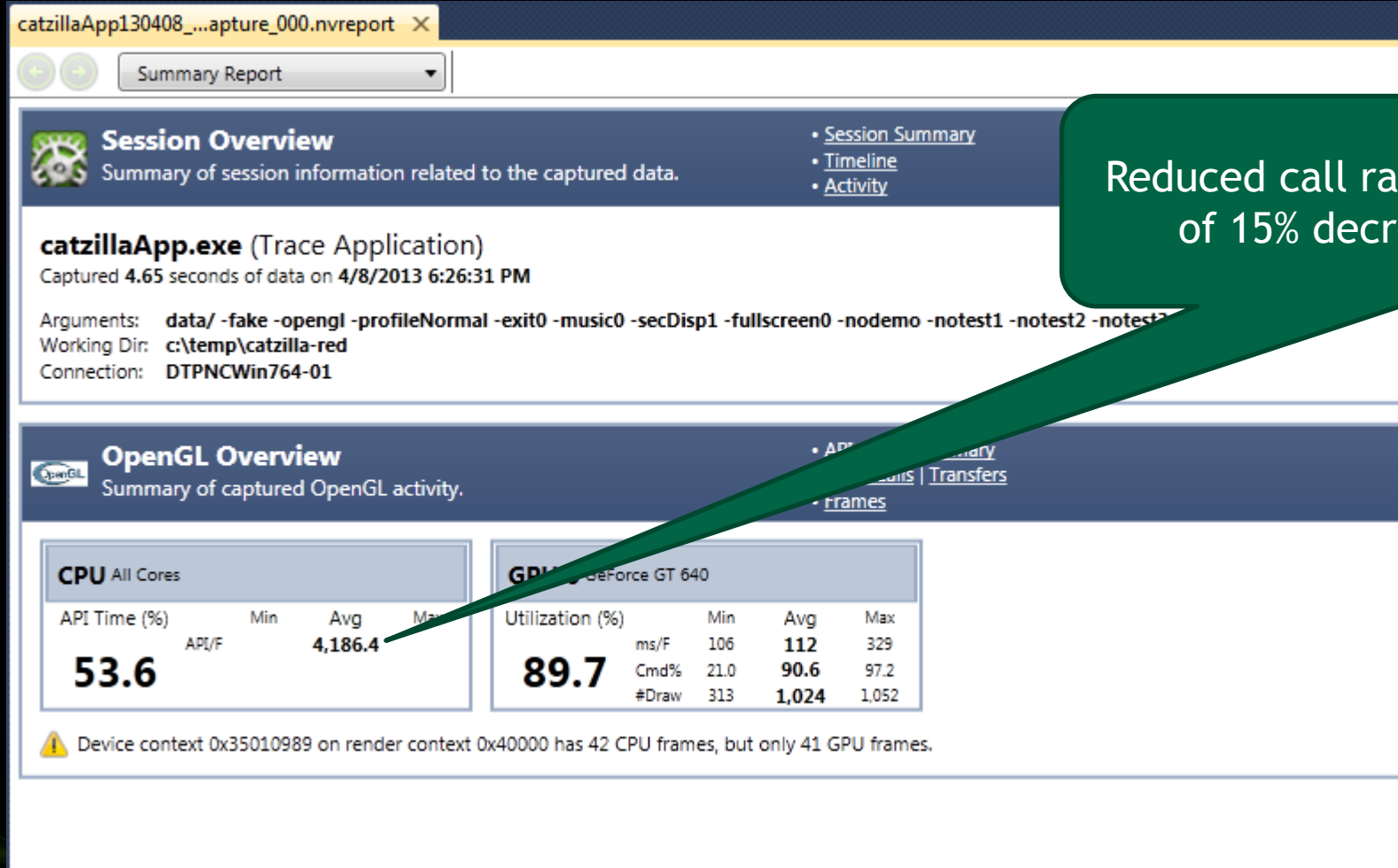
Primitive Restart

Enabled GL_FALSE

API Inspector allows you to make sure that code edits to remove redundant calls didn't break anything

Problems During Development

#2 Redundant State Changes



Reduced call ratio resulted in average of 15% decrease in frame time

Problems During Development

#3 glMapBufferRange too expensive

CatZilla - Microsoft Visual Studio (Administrator) - Experimental Instance

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

catzillaApp130311_...apture_000.nvreport

OpenGL API Call Summary

Filter

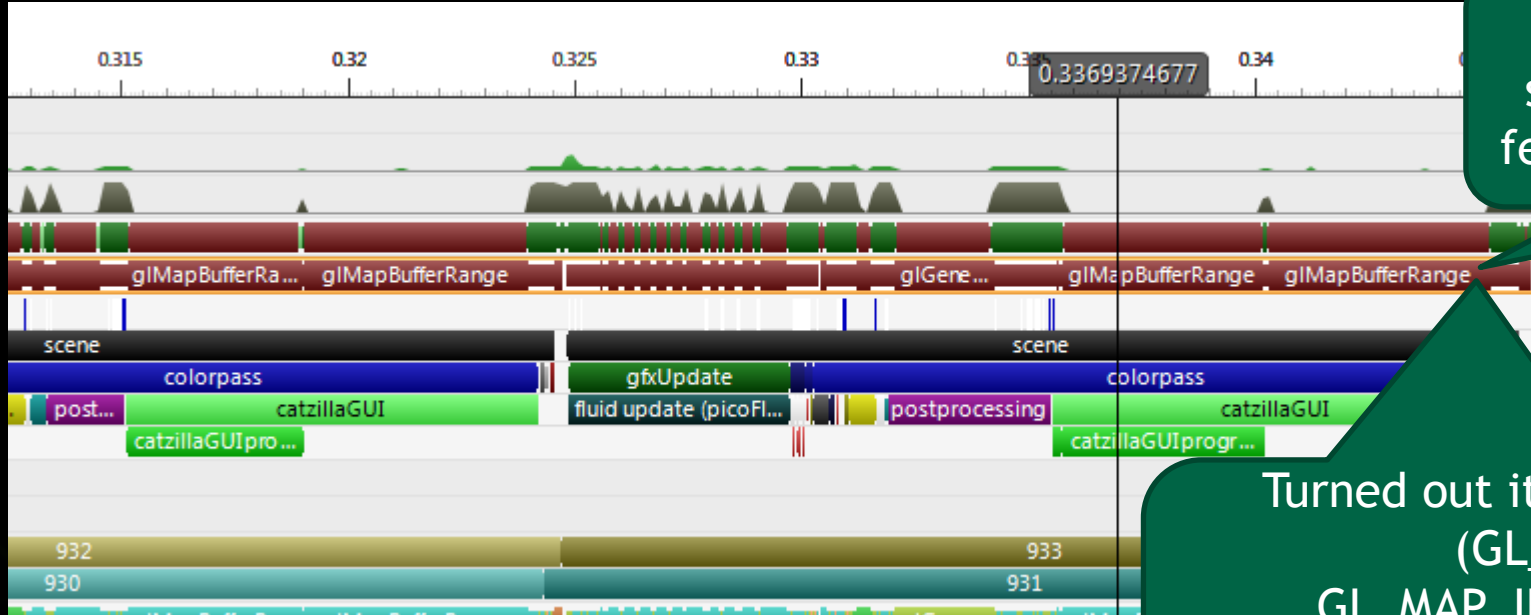
Drag a column header and drop it here to group by that column

	Name	Count	Capture Time %	Total Time (μs)	Min (μs)	Avg (μs)	Max (μs)
1	glMapBufferRange	2977	61.97	884,963.808	1.621	297.266	8,911.756
2	glFramebufferTexture	11595	6.98	99,700.528	0.222	8.598	7,771.283
3	glBlitFramebuffer	124	6.06	86,607.355	23.955	698.446	3,065.610
4	glDrawArraysInstanced	5085	3.11	44,469.250	0.552	8.745	1,229.671
5	SwapBuffers	62	0.99	14,069.494	90.972	226.927	4,406.661
6	glUseProgram	3783	0.54	7,732.866	0.796	2.044	11.042
7	glGenerateMipmap	62	0.38	5,386.277	33.718	86.875	2,235.139
8	glDrawElementsInstanced	1550	0.38	5,365.692	1.035	3.461	19.988
9	glCheckFramebufferStatusEXT	4526	0.35	4,956.068	0.270	1.095	13.104
10	glBindTexture	6574	0.32	4,596.462	0.232	0.699	23.584
11	glClear	434	0.32	4,535.782	3.877	10.451	47.901

3000 calls to glMapBufferRange taking 62% of the frame time

Problems During Development

#3 glMapBufferRange too expensive



Looked at other possible solutions like buffer pool with fences but hurt SLI performance

Turned out it was bad flags...was passing
(GL_MAP_WRITE_BIT |
GL_MAP_INVALIDATE_BUFFER_BIT |
GL_MAP_UNSYNCHRONIZED_BIT)
but the invalidate caused the driver to make
too many temporary buffers

Problems During Development

#3 glMapBufferRange too expensive

CatZilla - Microsoft Visual Studio (Administrator) - Experimental Instance

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

catzillaApp130311_...apture_000.nvreport

OpenGL API Call Summary

Filter

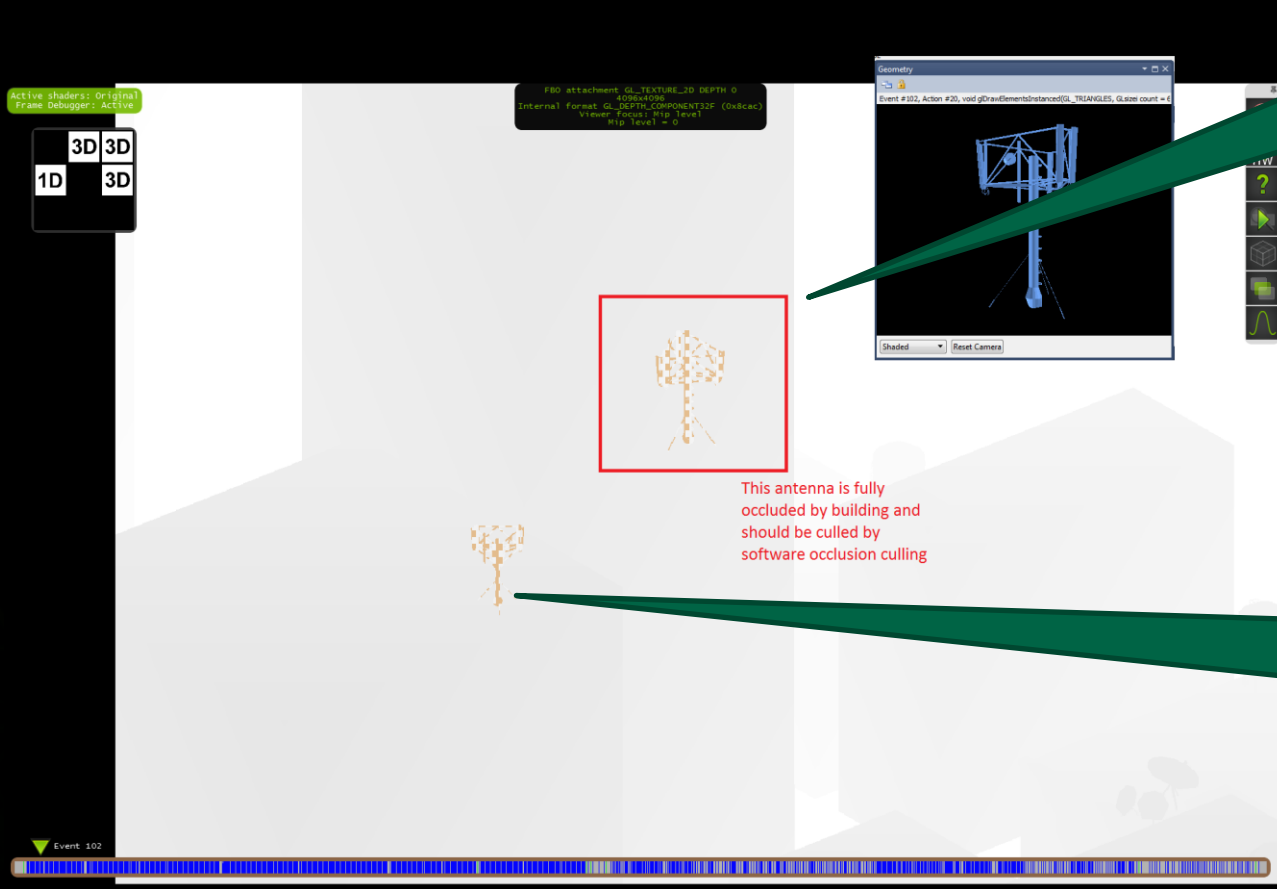
Drag a column header and drop it here to group by that column

	Name	Count	Capture Time %	Total Time (μs)	Min (μs)	Avg (μs)	Max (μs)
1	SwapBuffers	61	75.72	1,049,830.937	84.812	17,210.343	62,491.170
2	glFramebufferTexture	11412	2.84	39,393.592	0.258	3.451	242.000
3	glDrawArraysInstanced	5012	2.02	28,017.694	0.434	5.500	36.948
4	glUseProgram	3725	0.40	5,602.002	0.822	1.503	19.888
5	glMapBufferRange	2937	0.30	4,132.509	0.506	1.407	45.375
6	glDrawElementsInstanced	1525	0.29	4,016.098	1.032	2.633	39.207
7	glBlitFramebuffer	122	0.28	3,863.241	22.278	31.665	84.931
8	glCheckFramebufferStatusEXT	4454	0.25	3,469.317	0.269	0.778	8.109

Fix the flags, let the driver manage the memory, and perf improved!

Problems During Development

#4 Bug in SW culling



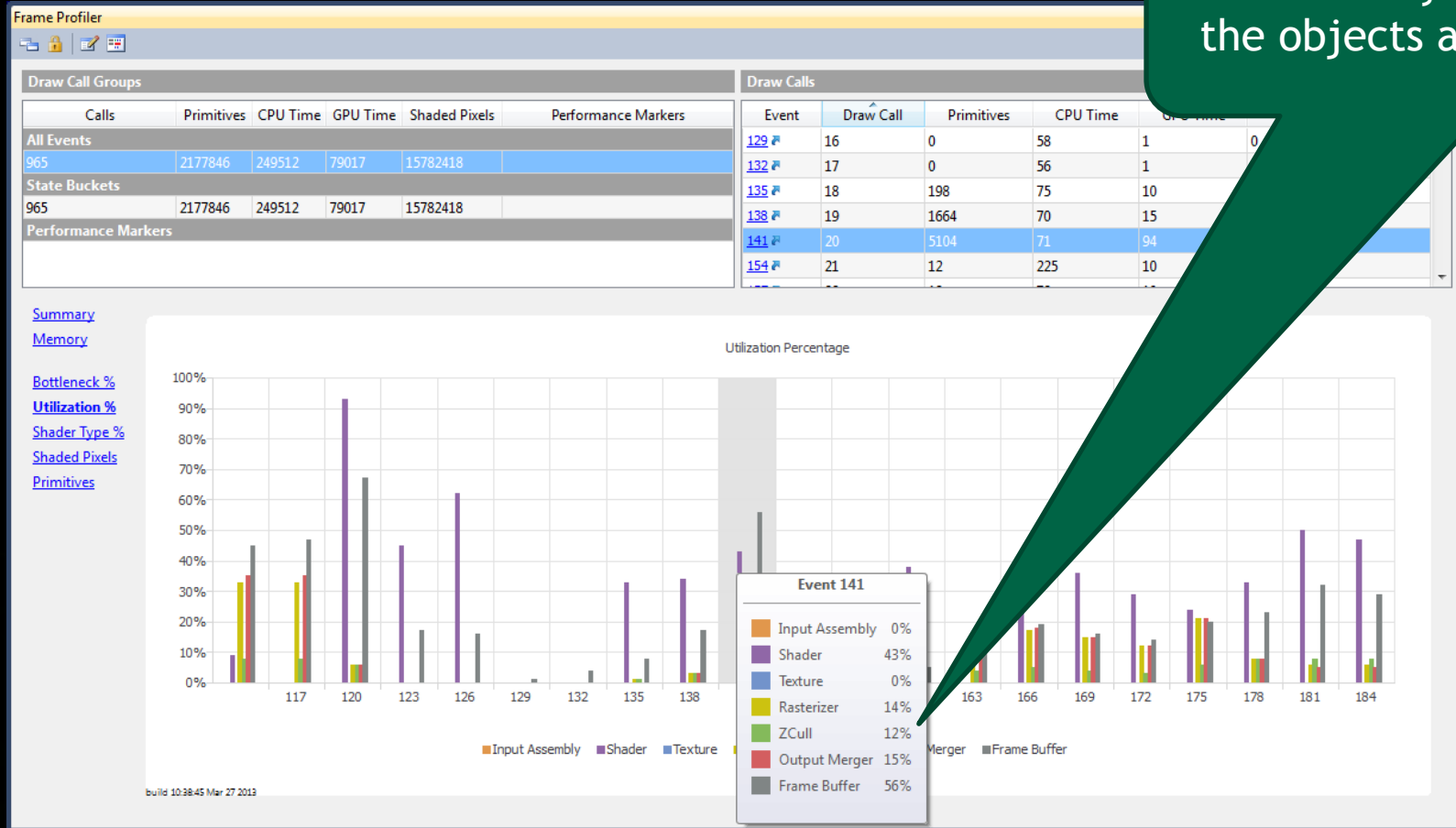
Scrubbing through the scene & saw items showing in depth buffer that never impacted the scene...

Using SW based method based on presentation by Daniel Conlin from DICE at GDC 2011

Problems During Development

#4 Bug in SW culling

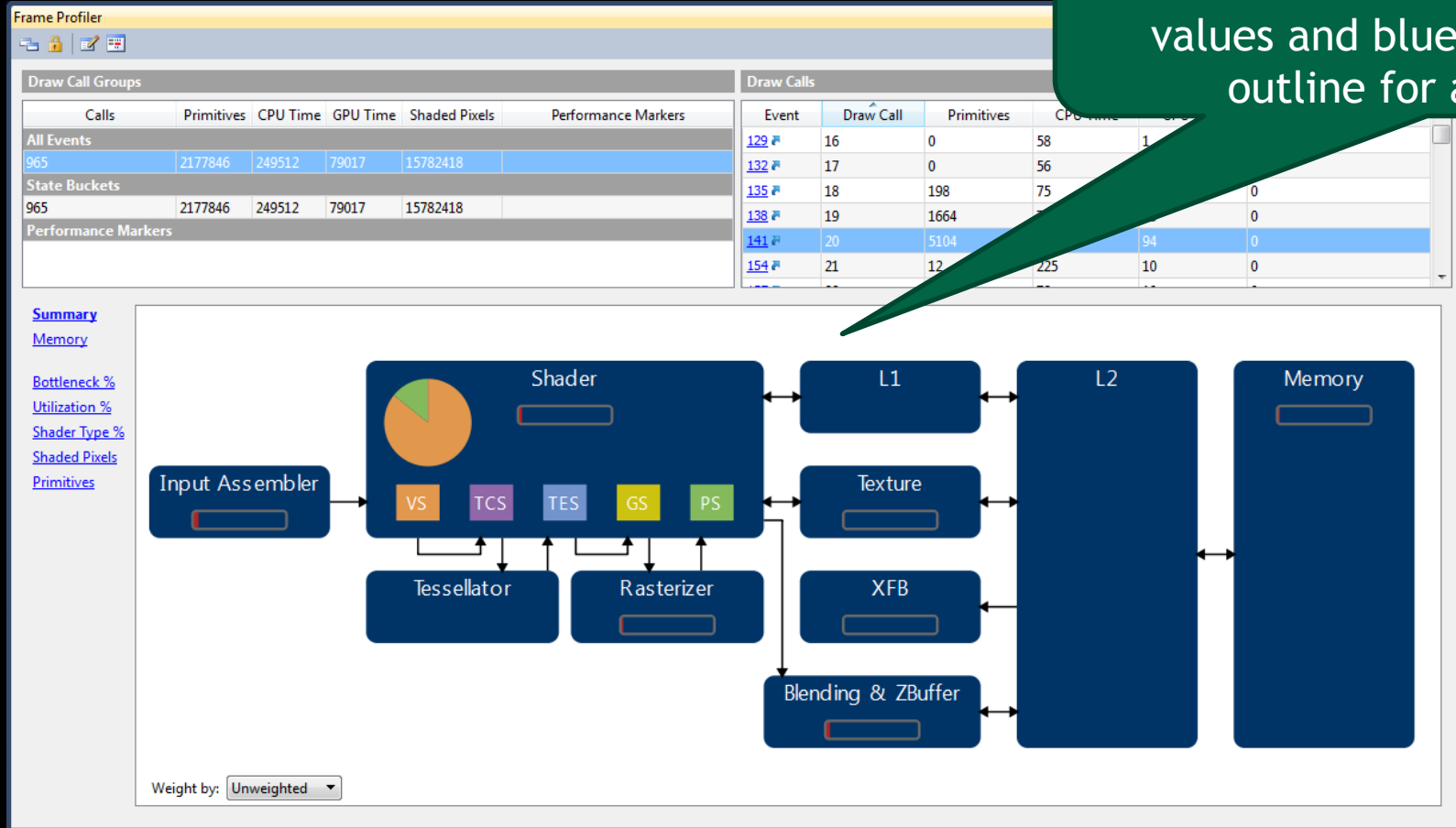
ZCull activity in the profiler confirms the objects are not going to show...



Problems During Development

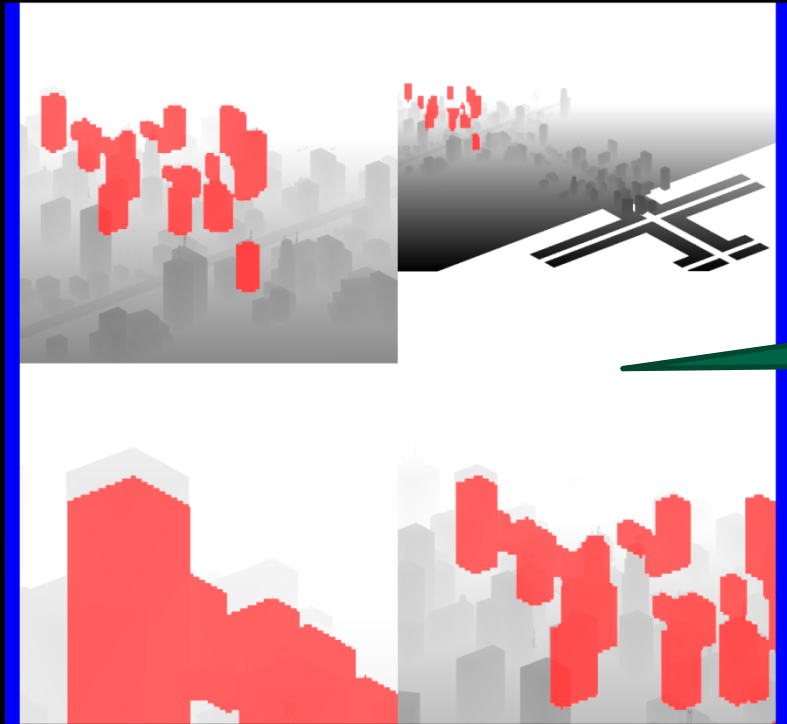
#4 Bug in SW culling

Note new summary screen for OpenGL Frame Profiler. Bars show unit bottleneck values and blue boxes will have red outline for areas of interest



Problems During Development

#4 Bug in SW culling

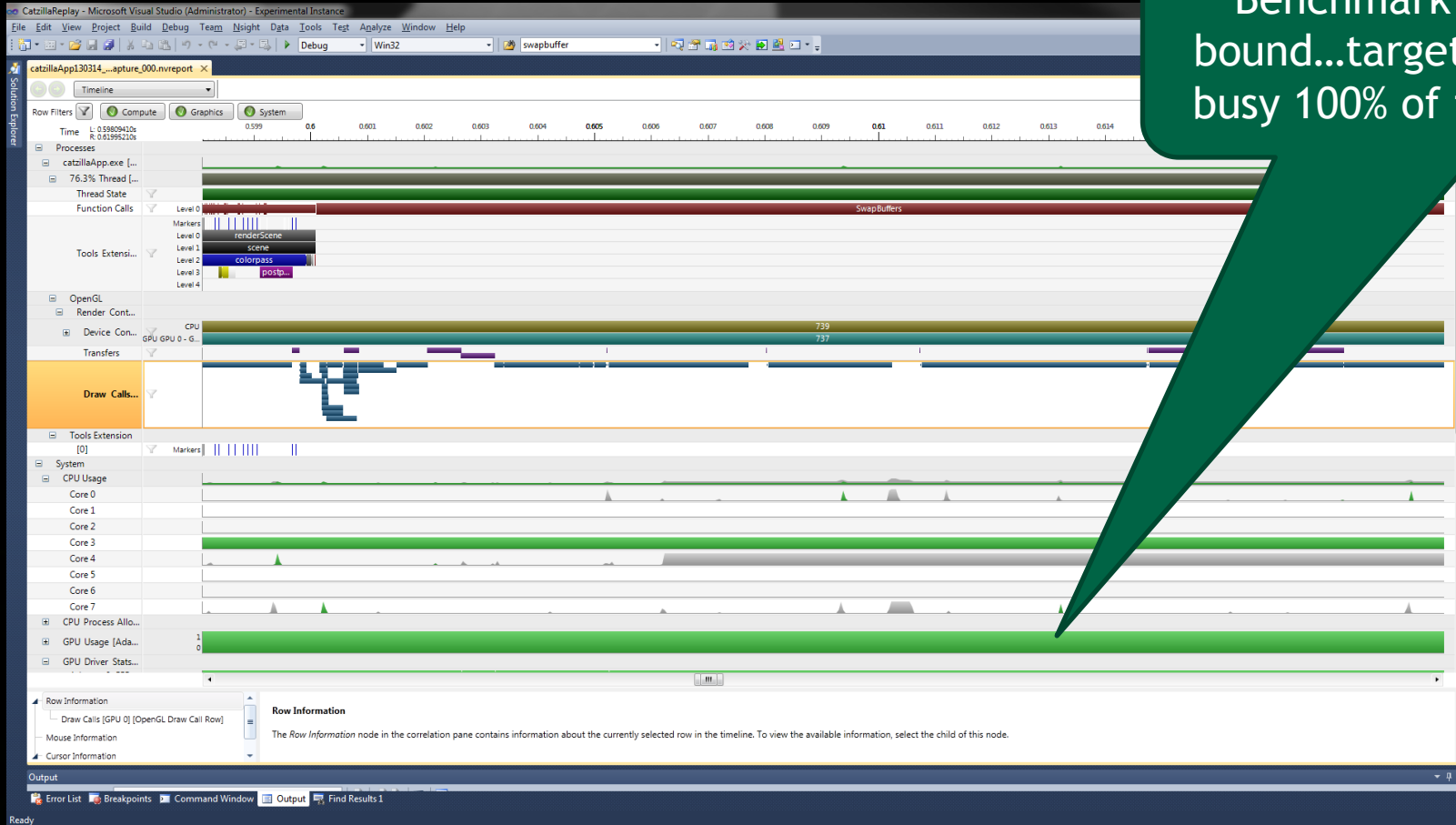


Found bug in the linear depth calculation, reduced scenes by 400-500 draw calls

Problems During Rendering

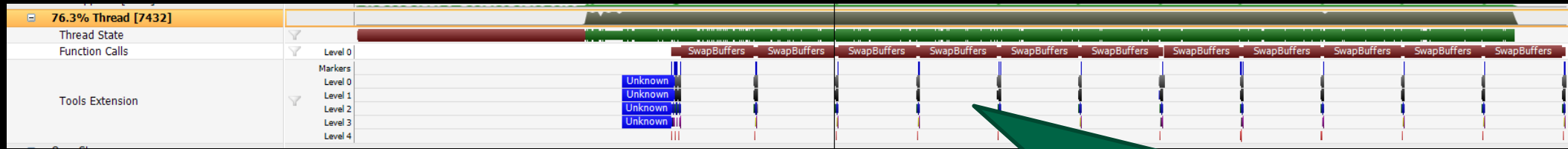
#5 Indepth look at Raymarch Test

Benchmark designed to be GPU bound...target process showing GPU busy 100% of the time...that's good!



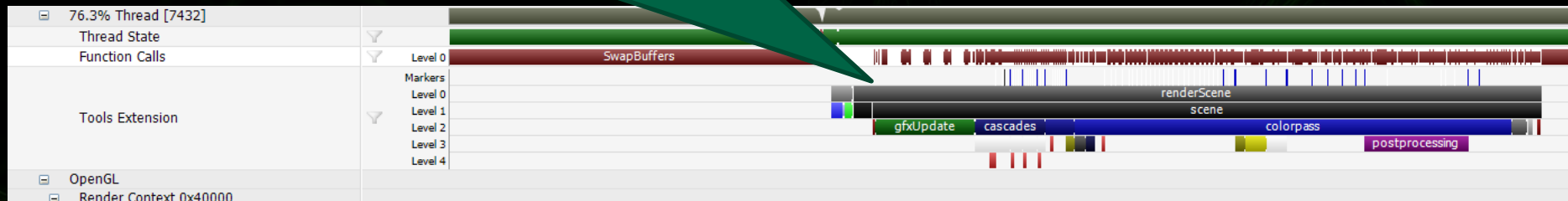
Problems During Rendering

#5 Indepth look at Raymarch Test



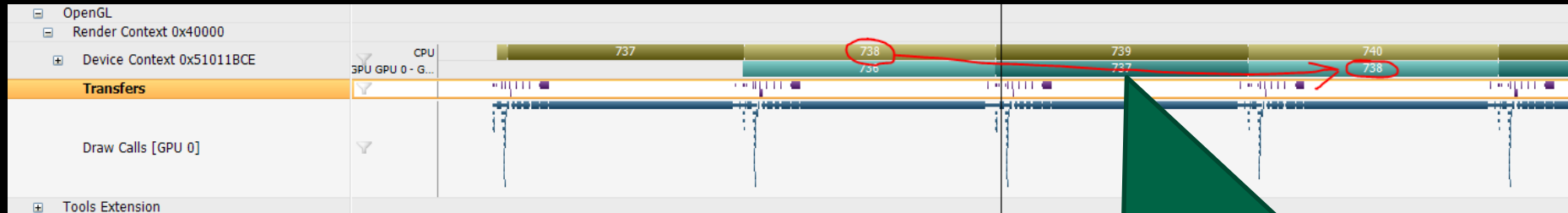
Main thread active all of the time...repeating frame pattern

NVTX used to annotate the frame to help see what was happening in the scene



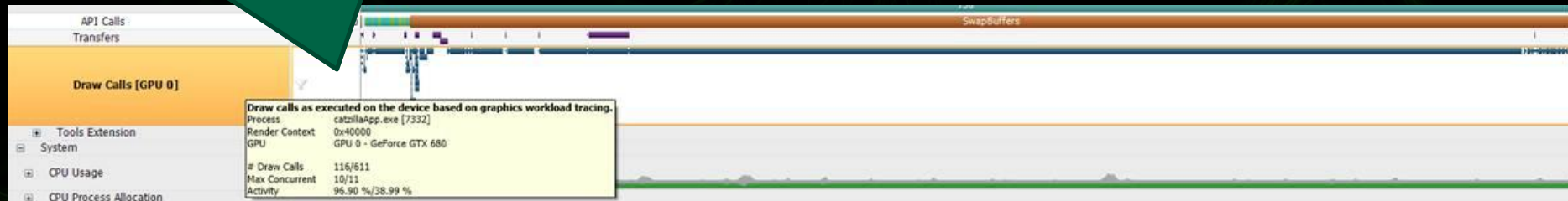
Problems During Rendering

#5 Indepth look at Raymarch Test



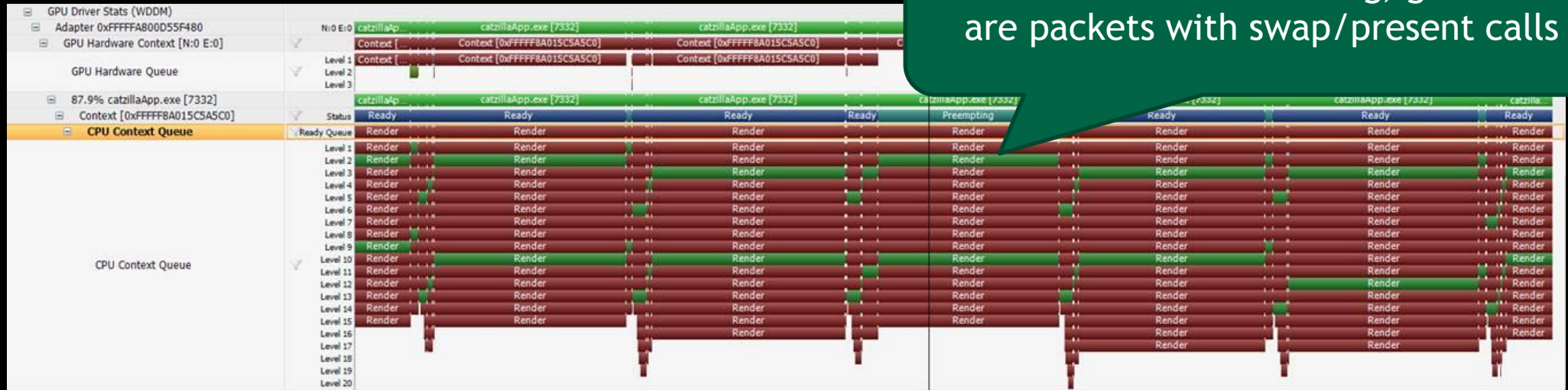
CPU is 2 frames behind the GPU, as expected for GPU bound app

Row tooltips show per frame stats, Draw Call row shows concurrent draw calls



Problems During Rendering

#5 Indepth look at Raymarch Test



Wrapping Up

- Questions/Comments?
- Resources
 - Info: <https://developer.nvidia.com>
 - Forums: <https://devtalk.nvidia.com>
- Downloads
 - <http://www.nvidia.com/nsight>
 - <http://www.allbenchmark.com/download>
- GDC Exhibit Booth: #1602