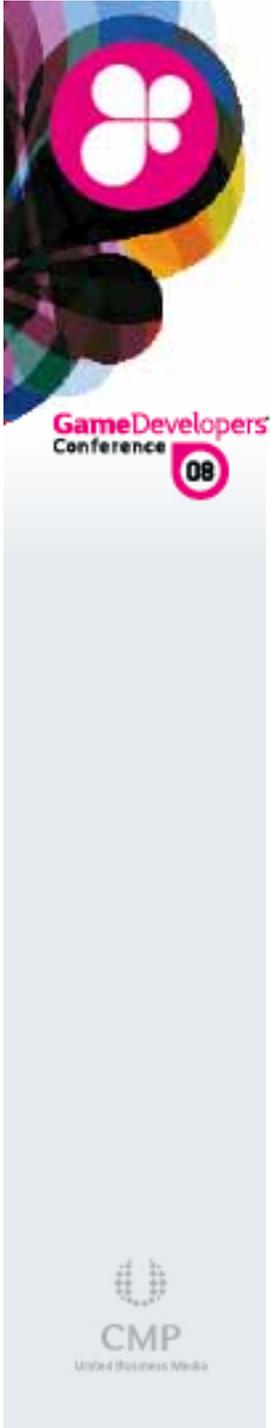




Beyond Printf

Debugging Graphics Through Tools



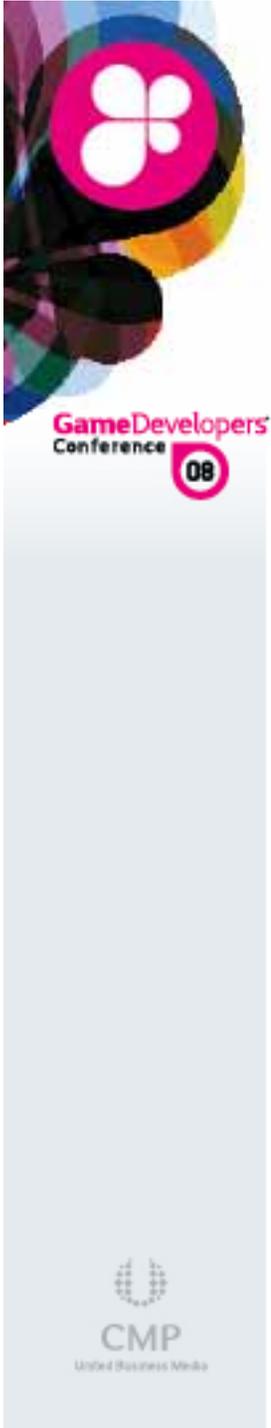
Presenters

④ Dave Aronson

NVIDIA – Technical Evangelist
daronson@nvidia.com

④ Karen Stevens

Microsoft – Software Design Engineer / Test
XNA Professional Game Platform
kstevens@microsoft.com



Purpose

- ④ To determine criteria for graphics tool selection
- ④ To demonstrate how tools can be used to identify and solve top game scenarios



Game Developers
Conference

08

Agenda

- ④ Tool Selection
- ④ Scenarios
- ④ Live Demos
- ④ Q&A
- ④ References



CMP

United Business Media

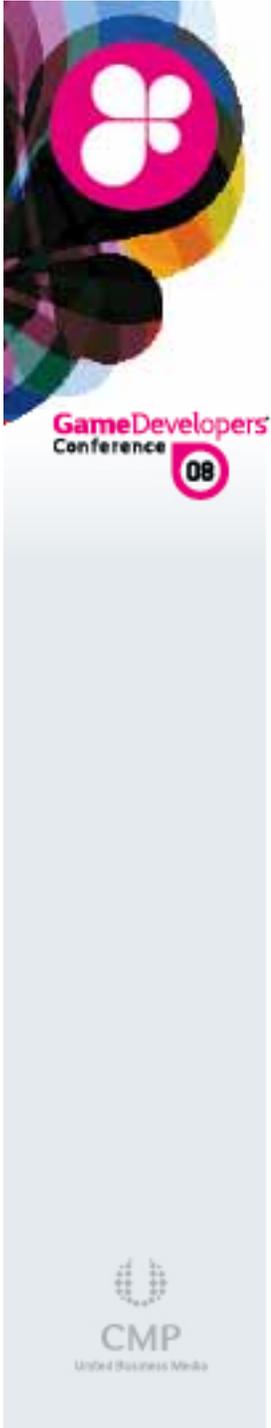
WWW.GDCONF.COM



Preliminary Criteria Points

When selecting a tool, consider:

- ③ Budget
- ③ General machine requirements
- ③ Hardware manufacturers
- ③ Additional required software
- ③ Code modification requirements
- ③ Product support
- ③ Features and general areas of interest



Popular Tool Areas of Interest

- ④ Game Assets

 - Textures, Shaders, Vertex Buffers, etc

- ④ API Usage

 - DirectX / OpenGL calls, state, debug spew

- ④ Driver

 - Driver versions, driver timing

- ④ Hardware

 - Timing, hardware usage



GameDevelopers
Conference

08

Tools Shown Today

AMD

GPU PerfStudio

GPU PerfStudio

Microsoft

PIX for Windows

PIX for Windows

NVIDIA

PerfHUD

PerfHUD

FX Composer

FX Composer



CMP

United Business Media

WWW.GDCONF.COM



Tool Categorization

🌀 Game Asset

PIX for Windows, GPU PerfStudio, FX Composer, PerfHUD

🌀 API

PIX for Windows, PerfHUD, GPU PerfStudio

🌀 Driver

PerfHUD, GPU PerfStudio

🌀 Hardware

PerfHUD, GPU PerfStudio



Example

Criteria:

- ⊕ Application uses DirectX 9 / HLSL
- ⊕ NVIDIA GeForce 7800 card is present
- ⊕ Do not want to change code to use tool
- ⊕ Preference towards free tools

Possible options from previous list:

- ⊕ FX Composer
- ⊕ PIX for Windows



How to Choose

- ④ Determine analysis levels of interest
 - ④ One strategy is to start at the game asset level and work down the list

- ④ Determine how tool fits criteria
 - ④ Prioritize your requirements

- ④ Experiment
 - ④ Most tools are free or have free trial periods, try a variety of scenarios



Scenarios

Glitches

 Incorrect behavior

Bottlenecks

 Poor performance



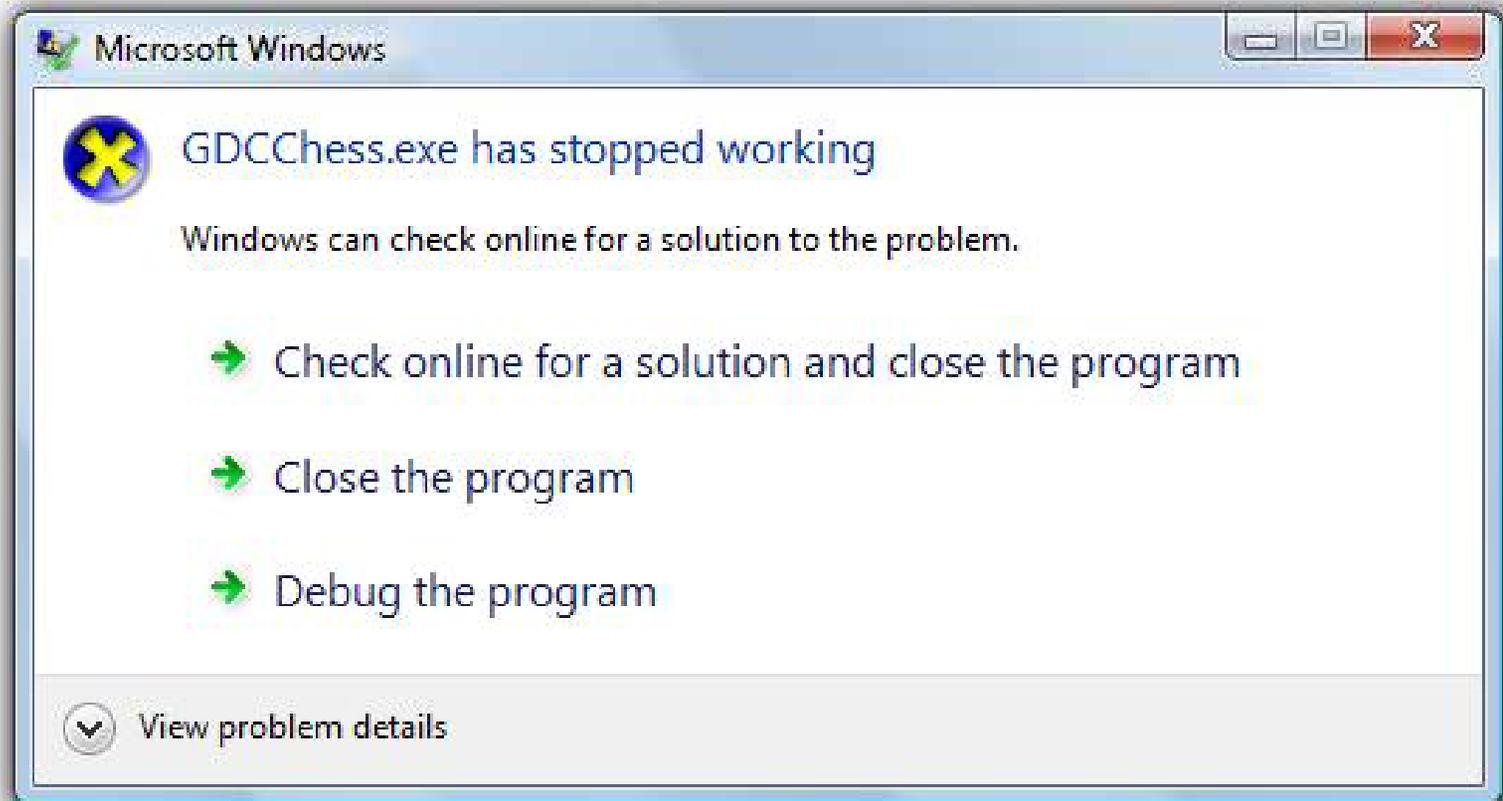
Glitches

The game is not behaving as expected:

- ⊗ Game Crash
- ⊗ Blank Screen
- ⊗ Missing Objects
- ⊗ Flickering



Game Crash





Game Crash

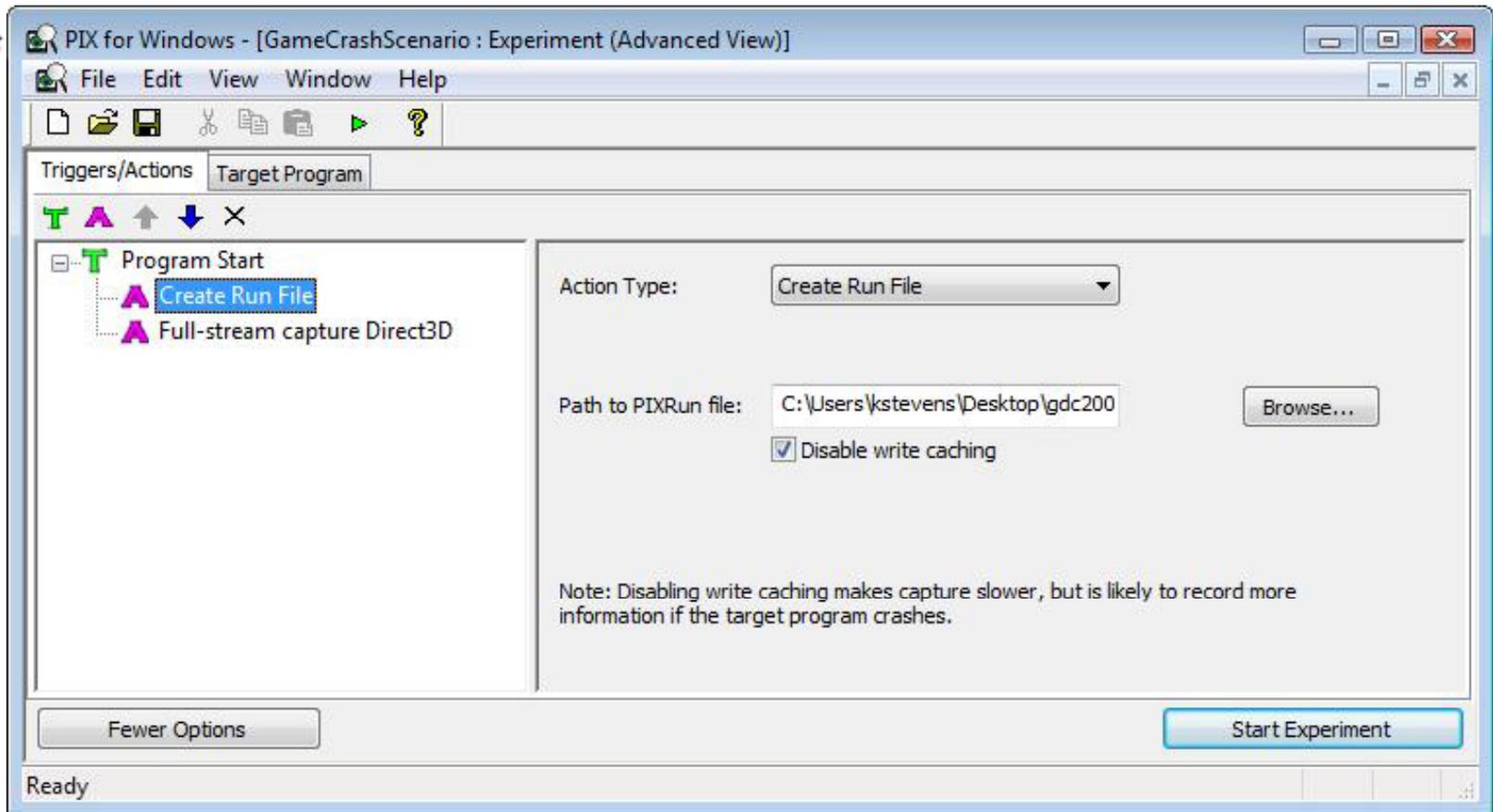
Scenario:

- ⌚ Game crashes when moving from windowed to full screen
- ⌚ Only occurs on specific video cards
- ⌚ The game does not have a debug build due to performance/game play reasons



Game Crash

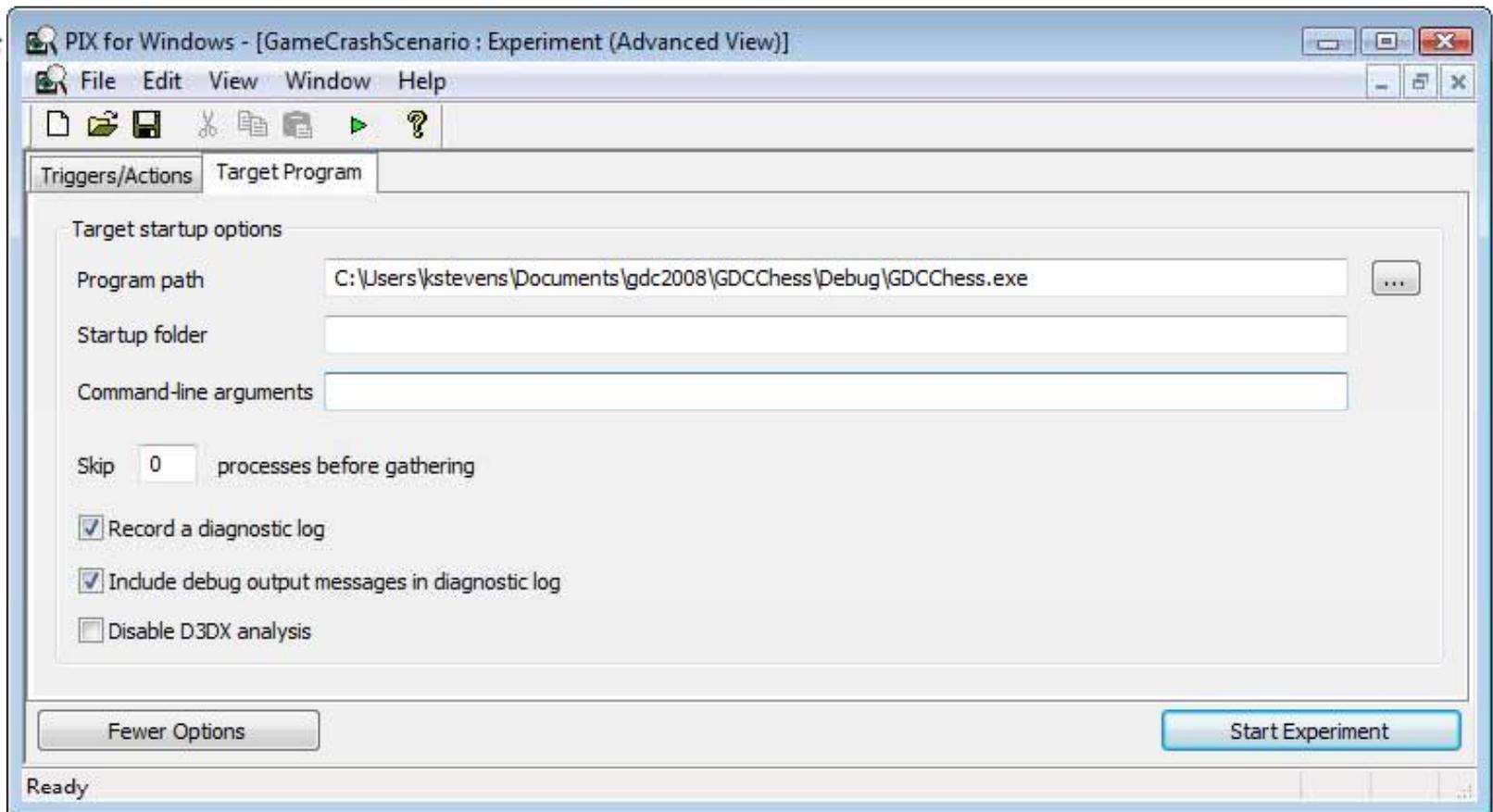
- Select settings to handle crash analysis





Game Crash

Setup diagnostic logging





Game Crash

PIX Diagnostic Log

A diagnostic log (3883 KB) was created while PIX was analyzing 'GDCChess.exe'.
Look for debug output messages regarding incorrect Direct3D usage, or invalid parameters in calls to Direct3D.

Diagnostic log file excerpt (click Save As to save the full log):

```
Frame 000003 .....PRE: RemoveObject(D3D9 State Block, 0x03CEEB00, 0x0A934D60)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEEB00, 0x0A934D60)
Frame 000003 .....POST: <0><this=0x03ceeb00> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: <this=0x03cefb98> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: RemoveObject(D3D9 State Block, 0x03CEFB98, 0x0A934EC0)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEFB98, 0x0A934EC0)
Frame 000003 .....POST: <0><this=0x03cefb98> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: <this=0x03cef610> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: RemoveObject(D3D9 State Block, 0x03CEF610, 0x0A934E40)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEF610, 0x0A934E40)
Frame 000003 .....POST: <0><this=0x03cef610> IDirect3DStateBlock9::Release()
Frame 000003 .....POST: <0><this=0x02389fe8> ID3DXSprite::Release()
Frame 000003 .....PRE: <this=0x03c53ed8> IDirect3DDevice9::Reset(0x04000F84)
Direct3D9: (ERROR) :All user created D3DPOOL_DEFAULT surfaces must be freed before ResetEx can succeed. Re
An unhandled exception occurred.
Closing Run File
```

Do you want to discard or save the log file?

PIX for Windows



Game Crash

Analysis:

- ⊙ Error: Direct3D9: (ERROR) :All user created D3DPOOL_DEFAULT surfaces must be freed before ResetEx can succeed. ResetEx Fails. An unhandled exception occurred.



Game Developers
Conference 08

Game Crash

🔗 Open run file for analysis

The screenshot displays the PIX for Windows interface for analyzing a game crash. The main window is titled "gdc_test.PIXRun".

Timeline: Shows a horizontal axis from 0 to 35 seconds. CPU usage is indicated by vertical bars and shaded blocks. A red box highlights the "100%" zoom level for the CPU usage view.

Objects: A table listing D3D9 objects. The first three rows are visible:

Address	Type	Created By	Creation	Destruction	Status	App Refs	Size	Pool	Us
0x029BAF00	D3D9 Object	Application	1	Never	Alive	10			
0x02973ED8	D3D9 Device	Application	1	1	Dead	n/a			
0x0298530	D3D9 Swap Chain	Direct3D	1	1	Dead	n/a			

843 of 843 objects displayed

Events: A list of events with columns for EID and Event. The event "Frame 1" is selected and expanded, showing a sequence of Direct3D calls:

- 1 Start Session
- 2 Start Process
- 3 **Frame 1**
- 4 User Event: DXUT D3D9 Enumeration
- 5 Direct3DCreate9(0x80000000 | 32)
- 6 <0x029BAF00> IDirect3D9::GetAdapterCount()
- 7 <0x029BAF00> IDirect3D9::GetAdapterIdentifier(0, 0x00000000, 0x02DB03CC,
- 8 <0x029BAF00> IDirect3D9::GetAdapterModeCount(0, D3DFMT_X8R8G8B8)
- 9 <0x029BAF00> IDirect3D9::EnumAdapterModes(0, D3DFMT_X8R8G8B8, 0, 0x
- 10 <0x029BAF00> IDirect3D9::EnumAdapterModes(0, D3DFMT_X8R8G8B8, 1, 0x
- 11 <0x029BAF00> IDirect3D9::EnumAdapterModes(0, D3DFMT_X8R8G8B8, 1, 0x

Details: Shows a 3D scene with a blue sky and a checkered floor. The "Channel(s)" dropdown is set to "RGB".

PIX for Windows

WWW.GDCONF.COM



Game Crash

🔍 Examine objects left after last valid call

Game Developers
Conference 08

Objects								
Address	Type	Destruction	Status	App Refs	Pool	Usage	Format	
0x02A7A390	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A438	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A4B9B8	D3D9 Surface	Never	Alive	0	Default	DepthStencil	D3DFMT	
0x02A4B910	D3D9 Surface	Never	Alive	0	Default	RenderTarget	D3DFMT	
0x02A7A4E0	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A588	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A630	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A6D8	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A780	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A7A828	D3D9 Surface	Never	Alive	0	Default	Dynamic	D3DFMT	
0x02A4B6B0	D3D9 Surface	1	Dead	n/a	Default	RenderTarget	D3DFMT	
0x02A4B7E0	D3D9 Surface	1	Dead	n/a	Default	RenderTarget	D3DFMT	



Game Crash

🕒 Located rouge object creation point

Game Developers Conference 08

Objects

Ad	View Surface 0x02A7A588	App Refs	Pool	Usage	Format
0x0	Object Operations				D3DFMT
0x0	IDirect3DSurface9* 0x02A7A588				D3DFMT
0x0	Creation: Frame 1, EID 10676	0	Default	RenderTarget	D3DFMT
0x0	Destruction: Never	0	Default	Dynamic	D3DFMT
0x0	Format: D3DFMT_A8R8G8B8	0	Default	Dynamic	D3DFMT
0x0	Usage: D3DUSAGE_DYNAMIC	0	Default	Dynamic	D3DFMT
0x0	Pool: D3DPOOL_DEFAULT	0	Default	Dynamic	D3DFMT
0x0	Dimensions: 256 x 256	0	Default	Dynamic	D3DFMT
0x0	App References: 0	0	Default	Dynamic	D3DFMT
		n/a	Default	RenderTarget	D3DFMT
		n/a	Default	RenderTarget	D3DFMT



Game Crash

Trace calls for objects requiring release

6926	+ User Event: Water Texture setup
10671	- <0x02A00B10> IDirect3DDevice9::CreateTexture(2048, 2048, 8,
10672	...CreateObject(D3D9 Texture, 0x02A62E38)
10673	...CreateObject(D3D9 Surface, 0x02A7A390)
10674	...CreateObject(D3D9 Surface, 0x02A7A438)
10675	...CreateObject(D3D9 Surface, 0x02A7A4E0)
10676	...CreateObject(D3D9 Surface, 0x02A7A588)
10677	...CreateObject(D3D9 Surface, 0x02A7A630)
10678	...CreateObject(D3D9 Surface, 0x02A7A6D8)
10679	...CreateObject(D3D9 Surface, 0x02A7A780)
10680	...CreateObject(D3D9 Surface, 0x02A7A828)
10681	<0x02A62E38> IDirect3DTexture9::LockRect(0, 0x0017F6DC, NU



Game Crash

Conclusion:

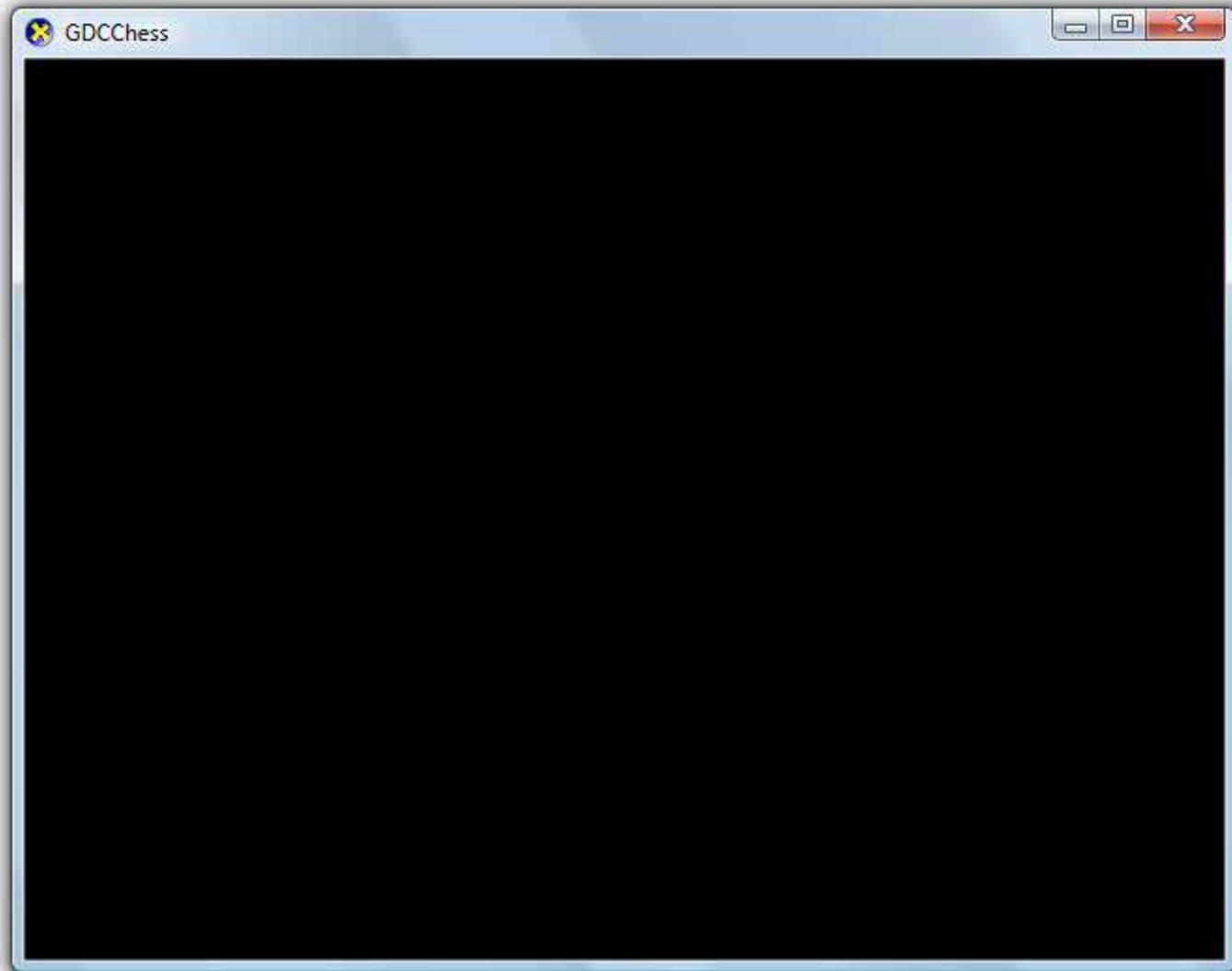
- ⊕ Some D3DPOOL_DEFAULT textures were not released before ResetEx occurred
- ⊕ Tools can examine remaining objects/textures to help ID items that require rework
- ⊕ Remaining objects are easily cleaned up once identified
- ⊕ Allows debugging of both retail and debug builds (assuming no copy write protection)



GameDevelopers
Conference

08

Blank Screen




CMP
United Business Media

WWW.GDCONF.COM



Blank Screen

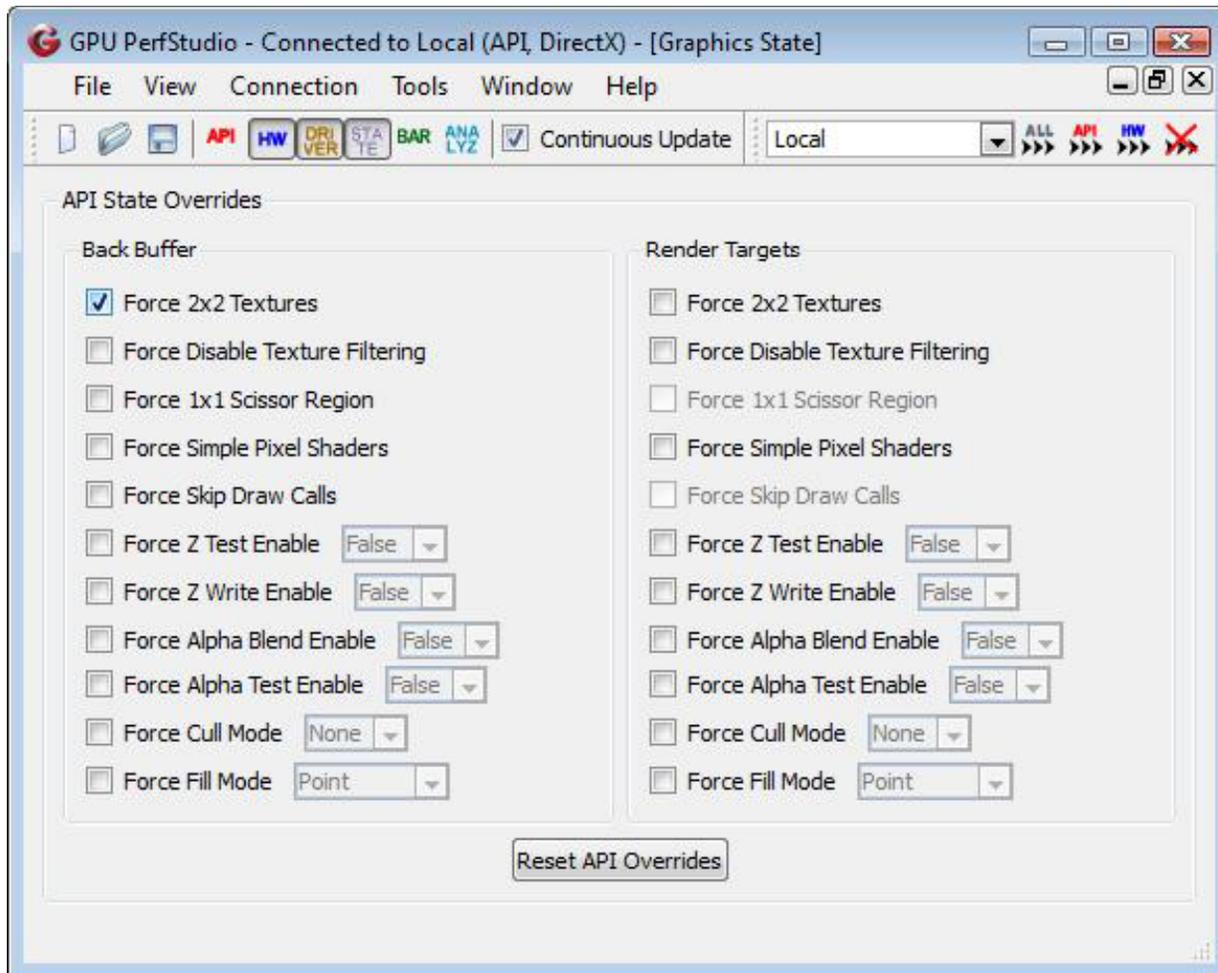
Scenario:

- ⊕ Many machines render a black screen
- ⊕ The program works fine on some machines
- ⊕ Video card is the same on all machines
- ⊕ Video driver is the same on all machines



Blank Screen

⚙️ Overriding states can rule out issues early

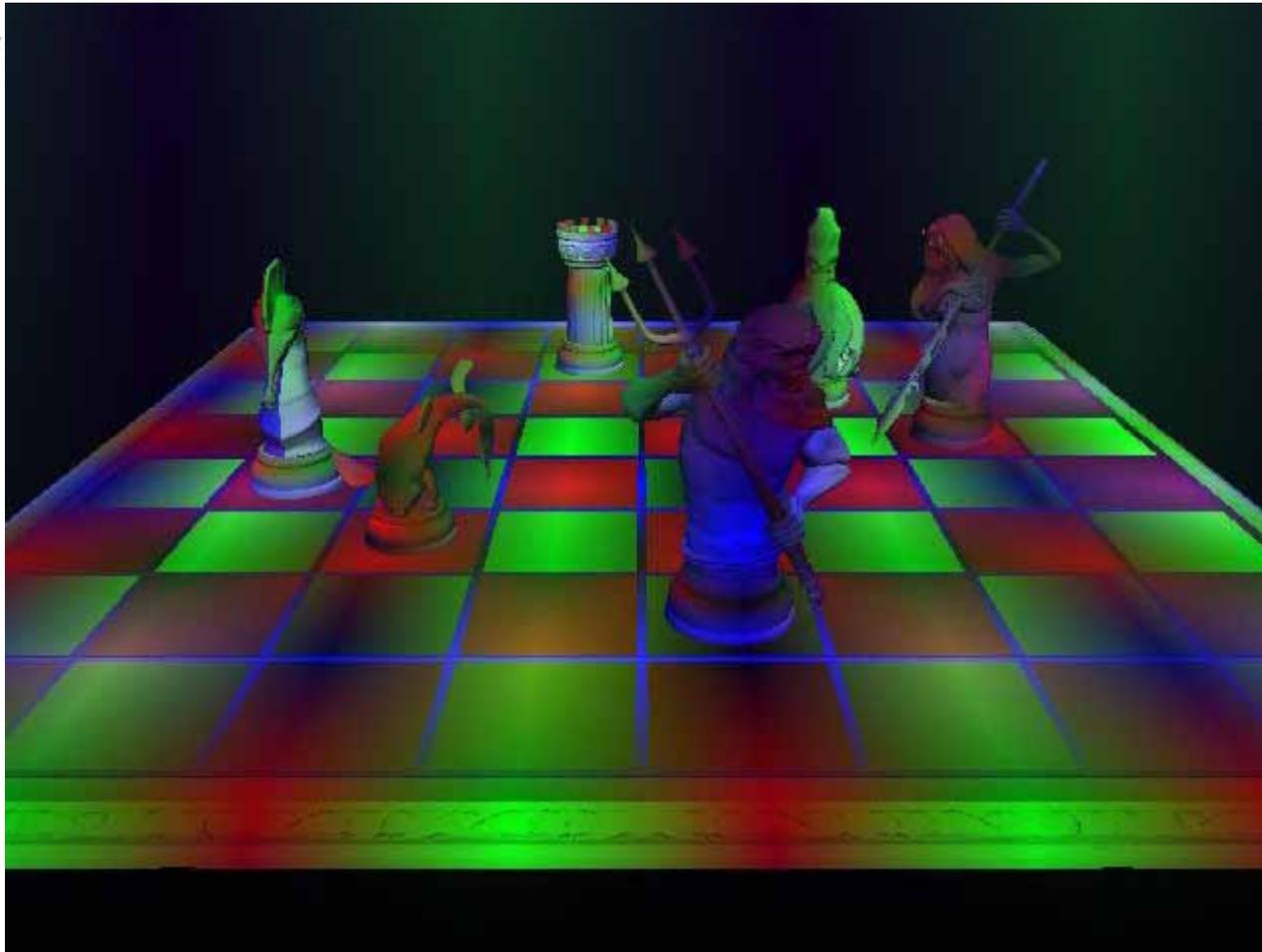




Blank Screen

- ⌚ Overriding texture renders scene viewable

Game Developers
Conference 08



CMP
United Business Media

GPU PerfStudio

WWW.GDCONF.COM



Blank Screen

🕒 Checking for sampler issues

The screenshot shows the Visual Studio Performance Profiler. The left pane displays a list of events for a Direct3D device (0x03D2AD68). The right pane shows the 'Details' view for the selected device, with the 'Render' state selected. A table below the state tabs shows the configuration for two samplers.

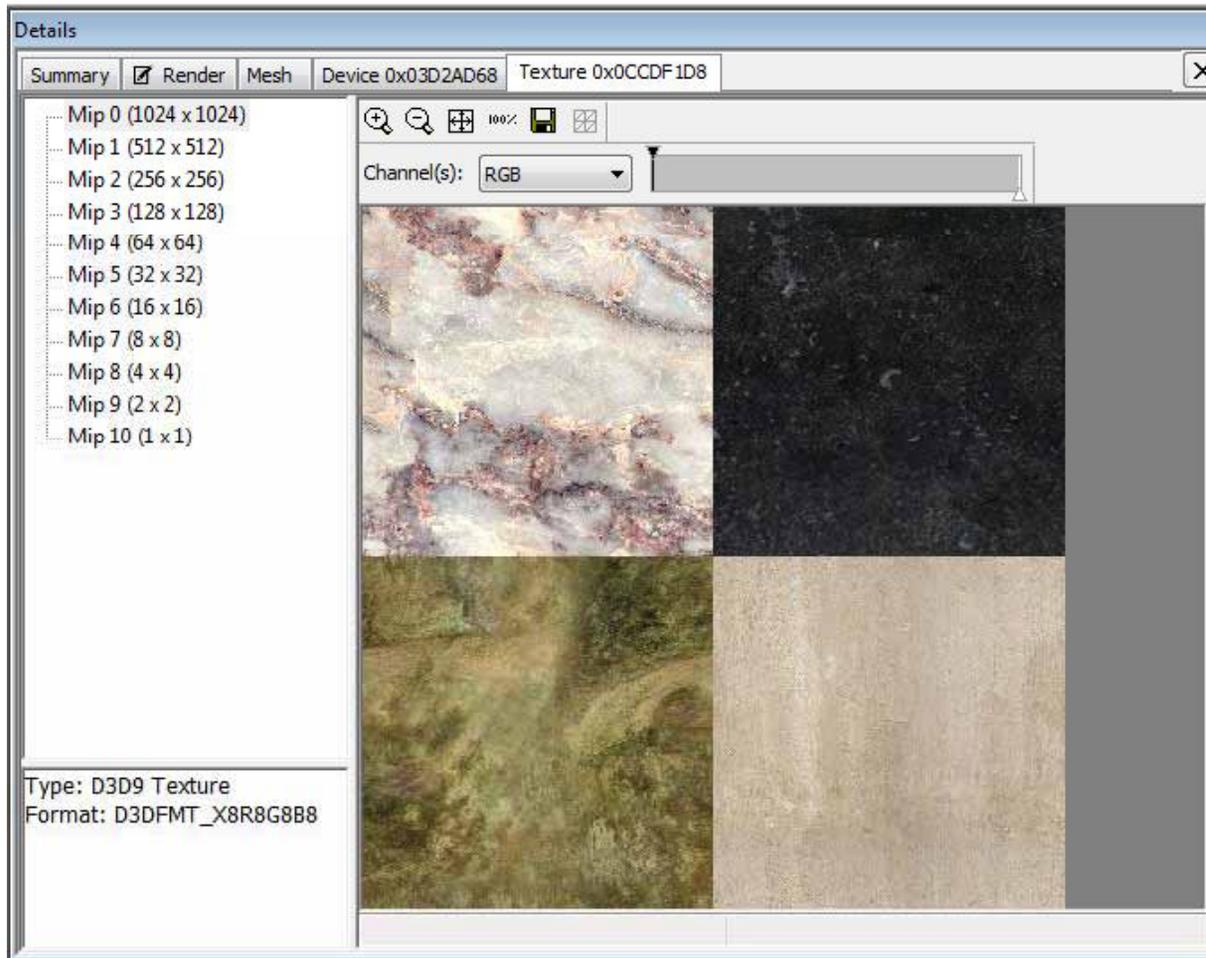
Sampler	Texture	Min Filter	Mag Filter	Mip Filter
0	0x0CCDF1D8	D3DTEXF_LINEAR	D3DTEXF_LINEAR	D3DTEXF_LIN
1	0x0CCDF0F0	D3DTEXF_LINEAR	D3DTEXF_LINEAR	D3DTEXF_L

🕒 Samplers exist, values look ok



Blank Screen

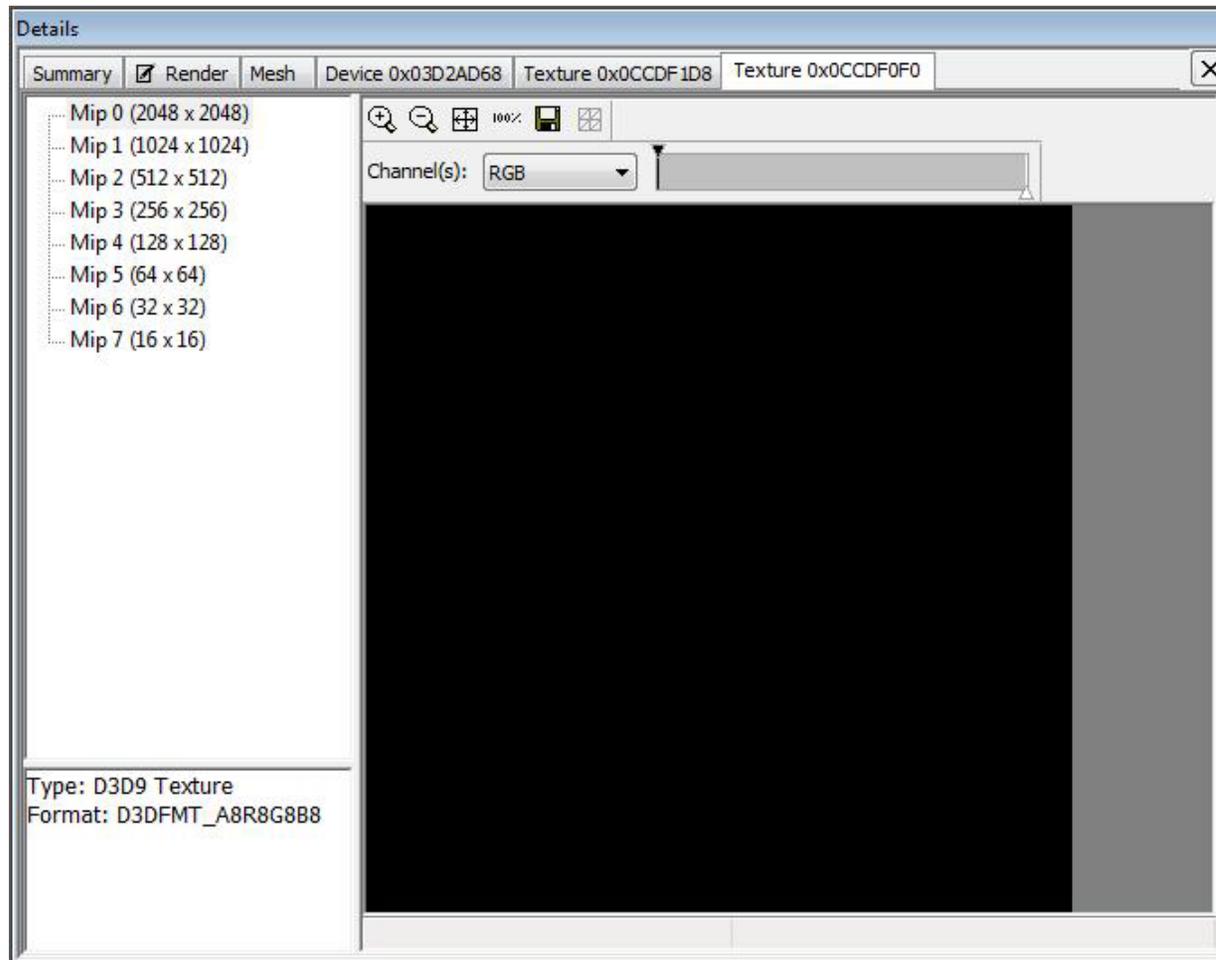
🔍 Check texture sampler 0 - OK





Blank Screen

- ⊕ Sampler texture 1 should not be black





Blank Screen

Render frame and select inaccurate pixel

The screenshot shows the PIX for Windows interface. On the left, the Event Log is expanded to 'Frame 353', showing a list of GPU events including 'User Event: Method: OnD3D9FrameRen', 'IDirect3DDevice9::Cle', 'IDirect3DDevice9::Beg', 'ID3DXEffect::SetFloat', 'ID3DXEffect::SetTech', 'ID3DXEffect::SetVecto', 'IDirect3DDevice9::Set', 'ID3DXEffect::SetVecto', 'IDirect3DDevice9::Set', 'ID3DXEffect::SetText', 'User Event: Drawing Scene', 'User Event: Method: DrawLightShafts', and 'IDirect3DDevice9::Enc'. The main window shows a black screen with a context menu open, containing 'Save Picture As...' and 'Debug This Pixel...'. The interface includes a toolbar with zoom and render controls, and a 'Channel(s): RGB' dropdown.

PIX for Windows



Blank Screen

Pixel history shows all calls output black

Details

Summary Render Mesh Debugger

Event 1254: IDirect3DDevice9::DrawIndexedPrimitive(D3DPT_TRIANGLELIST, 0, 4, 13608, 6, 23324)

Primitive 3 of 23324

Vertex Shader: [0x016EC8F0](#)
[Debug Vertex 0](#)
[Debug Vertex 1](#)
[Debug Vertex 2](#)

Pixel Shader: [0x016EC880](#)
[Debug Pixel \(369, 368\)](#)

Pixel shader output:

Alpha:	1.000
Red:	0.000
Green:	0.000
Blue:	0.000

Final framebuffer color:

Alpha:	0.000
Red:	0.000
Green:	0.000
Blue:	0.000

Event 1254: IDirect3DDevice9::DrawIndexedPrimitive(D3DPT_TRIANGLELIST, 0, 4, 13608, 6, 23324)



Game Developers Conference

08

Blank Screen

- Shader debugging proves black texture obliterates computed color

```
//-----  
  
float4 CausticPS(VS_OUT IN) : COLOR  
{  
    float2 movement = IN.TexCoord1.xy;  
  
    movement.x = movement.x + cos(Time * 0.2f) * 0.3f;  
    movement.y = movement.y + sin(Time * 0.3f) * 0.2f;  
  
    float3 color = IN.Color.rgb * tex2D(CausticTextureSampler, movement.xy * 0.9f);  
    color = color * tex2D(MeshTextureSampler, IN.TexCoord0.xy);  
}
```

Name	Value	Type
color	(0.000, 0.000, 0.000)	float3
movement	(0.845, 4.548)	float2



Blank Screen

Analysis:

- ⌚ Incorrect texture is used
- ⌚ The texture is involved in all lighting operations, therefore everything is black
- ⌚ Black is a common fallback for textures which were unable to be loaded at runtime



Blank Screen

Conclusion:

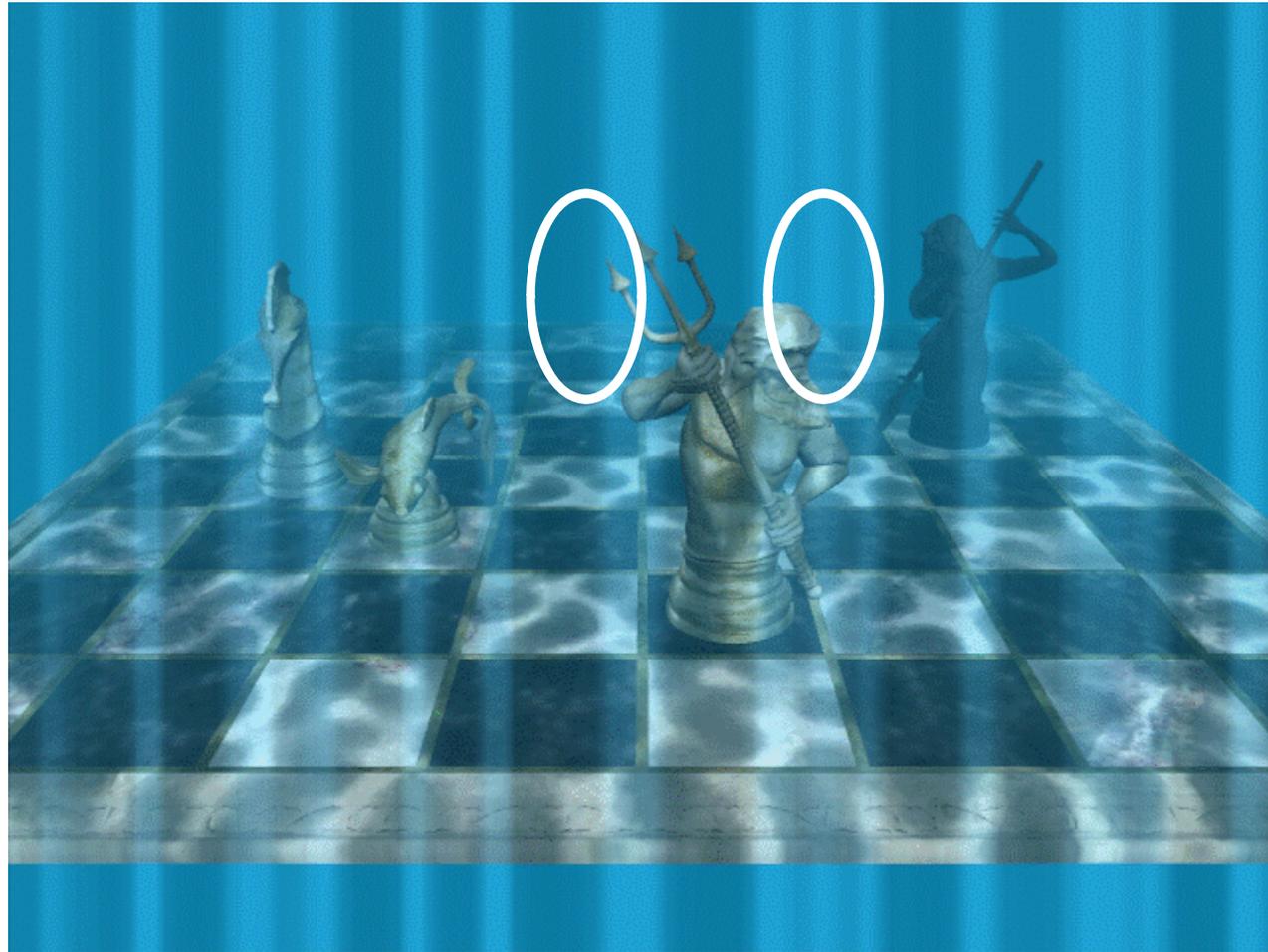
- ④ The texture failed to load
- ④ Texture loading is based on a file path
- ④ Machines with an incorrect path didn't load the texture
- ④ Correcting path in setup restored lighting to all machines



Game Developers
Conference

08

Missing Objects



CMP
United Business Media

WWW.GDCONF.COM



Missing Objects

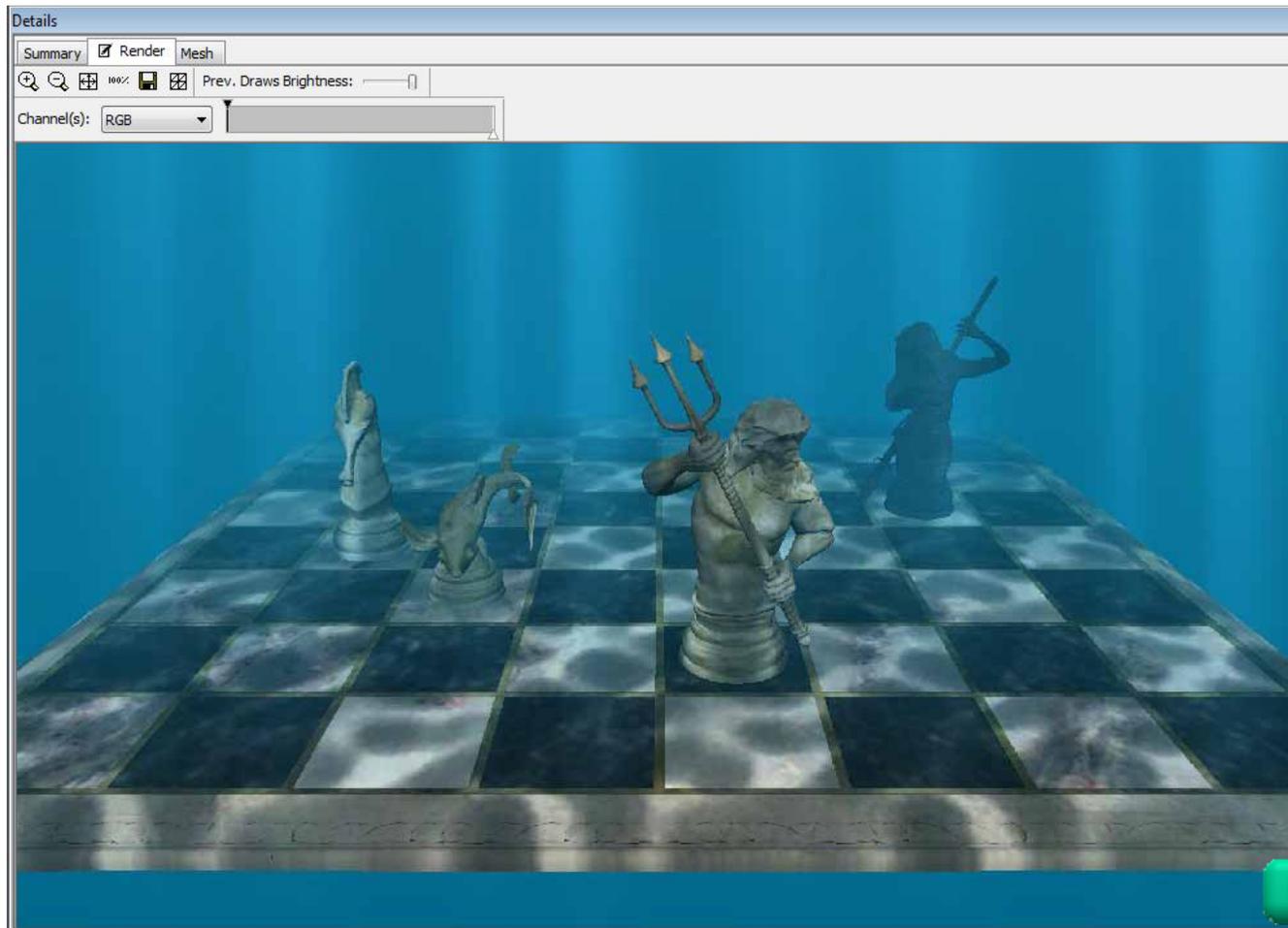
Scenario:

- ④ Code traces prove all draw calls are executed
- ④ A few of the objects drawn are not displaying on the screen



Missing Objects

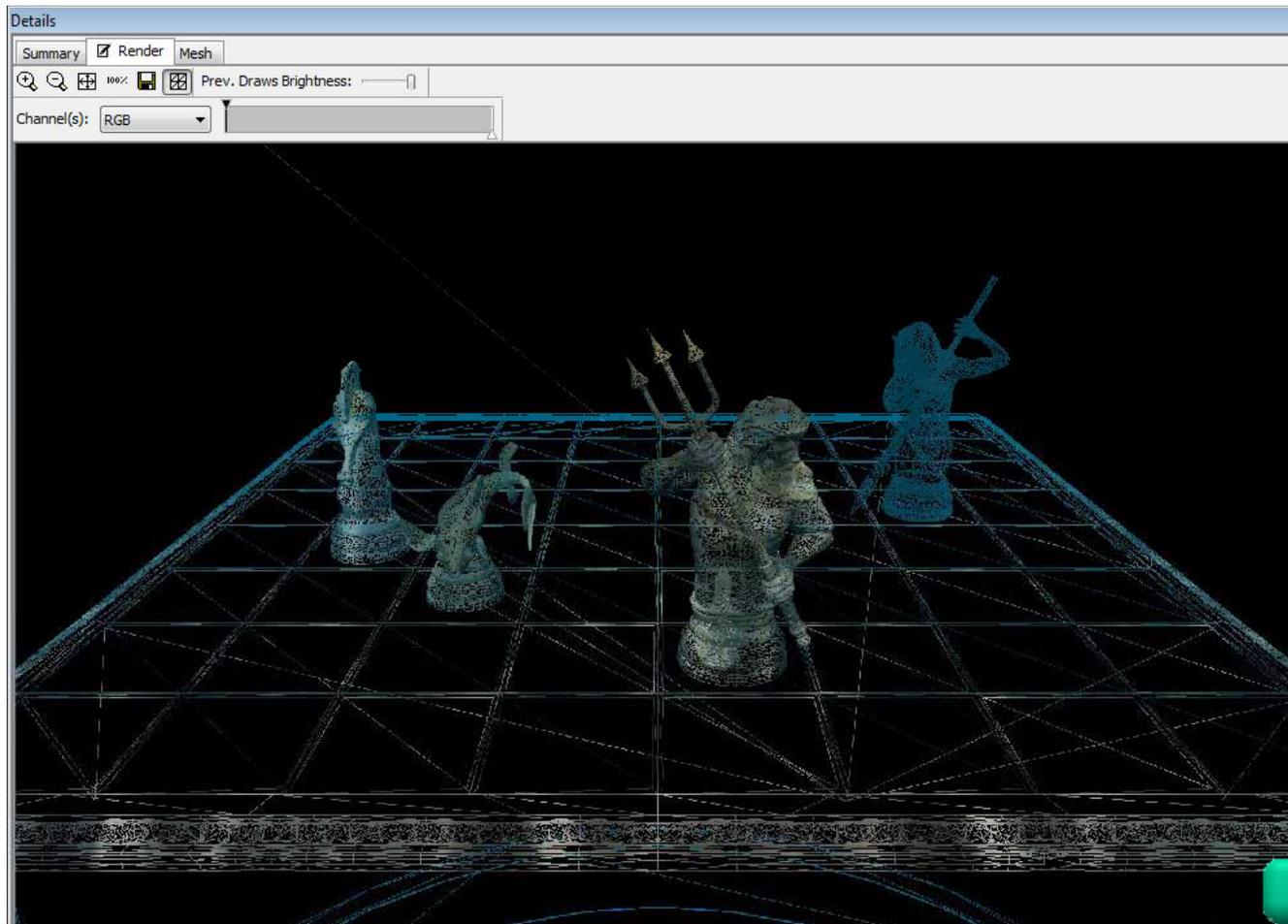
🕸 Rendered scene has missing objects





Missing Objects

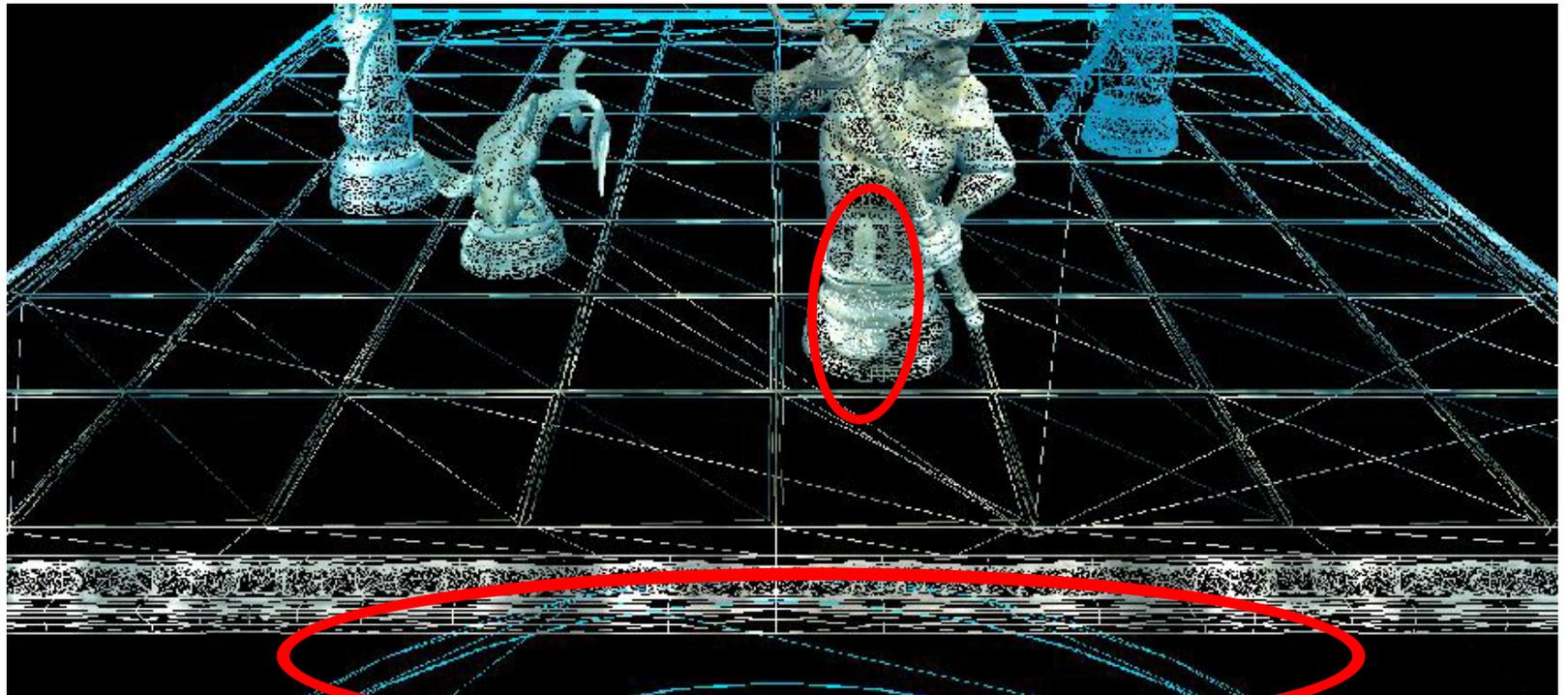
- 🔍 Check wireframe geometry of scene





Missing Objects

- ⊕ Suspicious artifacts present





Missing Objects

⊕ Incorrect vertex shader input

Summary Render Mesh

Pre-Vertex Shader Post-Vertex Shader Viewport

PreVS PostVS

VTX	IDX	Position				Normal			TexCoord0	
0	0	0.000	0.050	0.000	1.000	0.015	-1.000	0.003	0.099	0.797
1	1	0.009	0.051	0.000	1.000	-0.008	-0.854	-0.521	-0.526	1.168
2	2	0.006	0.050	0.001	1.000	-0.034	-0.601	-0.799	-0.418	1.096
3	1	0.009	0.051	0.000	1.000	-0.008	-0.854	-0.521	-0.526	1.168

PIX for Windows

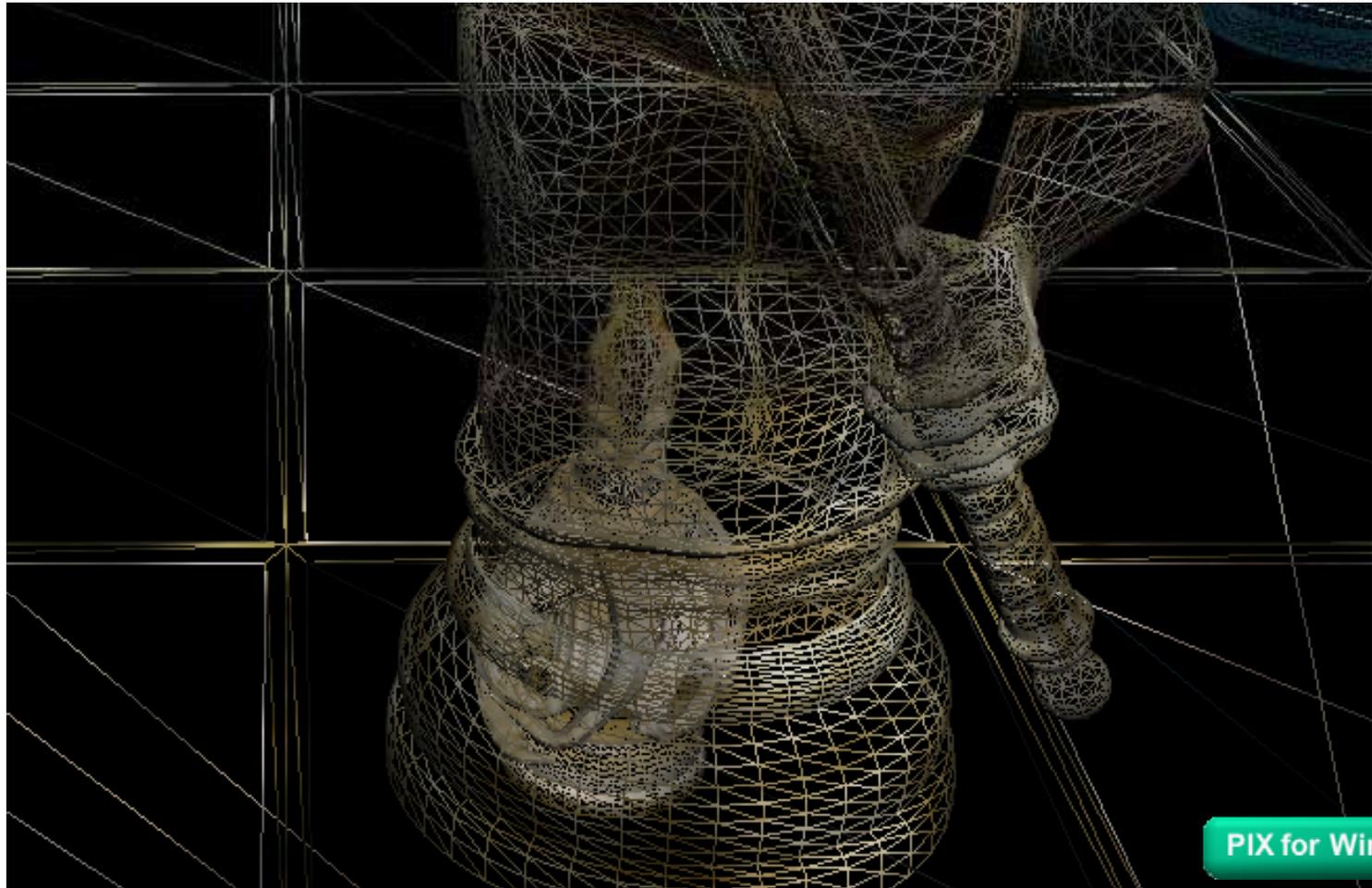


GameDevelopers
Conference

08

Missing Objects

- ⊕ Yields unexpected output



PIX for Windows

WWW.GDCONF.COM



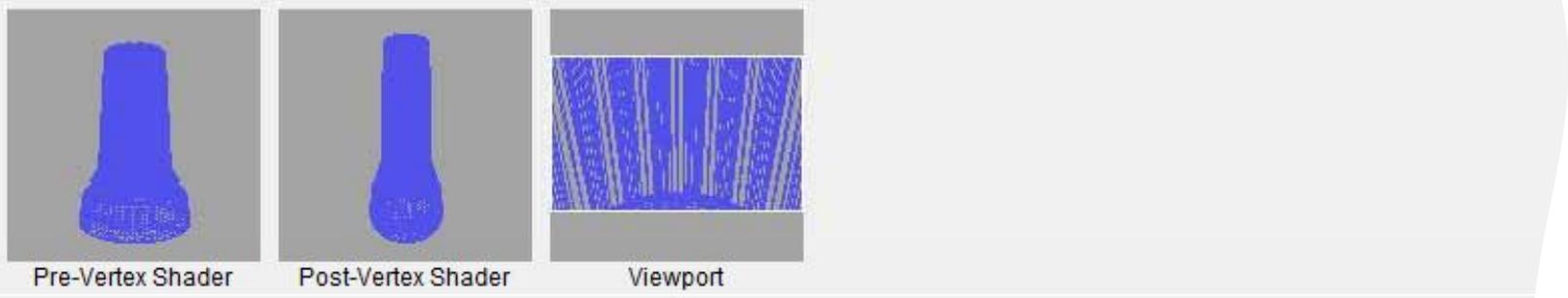
CMP

United Business Media



Missing Objects

Incorrect input & fogged out



PreVS PostVS

Prim	VTX	IDX	Position				Diffuse	Fog	TexCoord0		TexCoord1	
P0	0	5158	-13.523	-33.604	10.854	10.953	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.155	0.000	0.500	-6.722	8.2
	1	5159	-13.357	-34.861	9.552	9.651	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.090	0.006	0.422	-6.639	7.2
	2	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.218	0.006	0.578	-6.639	9.2
P1	3	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.218	0.006	0.578	-6.639	9.2
	4	5159	-13.357	-34.861	9.552	9.651	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.090	0.006	0.422	-6.639	7.2
	5	5161	-12.049	-29.956	14.632	14.731	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.336	0.054	0.727	-5.989	1.2
P2	6	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.218	0.006	0.578	-6.639	9.2
	7	5161	-12.049	-29.956	14.632	14.731	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)	-2.336	0.054	0.727	-5.989	1.2



Missing Objects

- ④ Defect demonstration, modifying application: no fog, no cull, zooming out





Missing Objects

Conclusion:

- ⊕ Incorrect values were sent to vertex shaders in both cases
- ⊕ Culling reduced odds of detecting the scene was inside the rook, fogging hid few remaining visible faces



Game Developers
Conference

08

Flickering




CMP
United Business Media

WWW.GDCONF.COM



Flickering

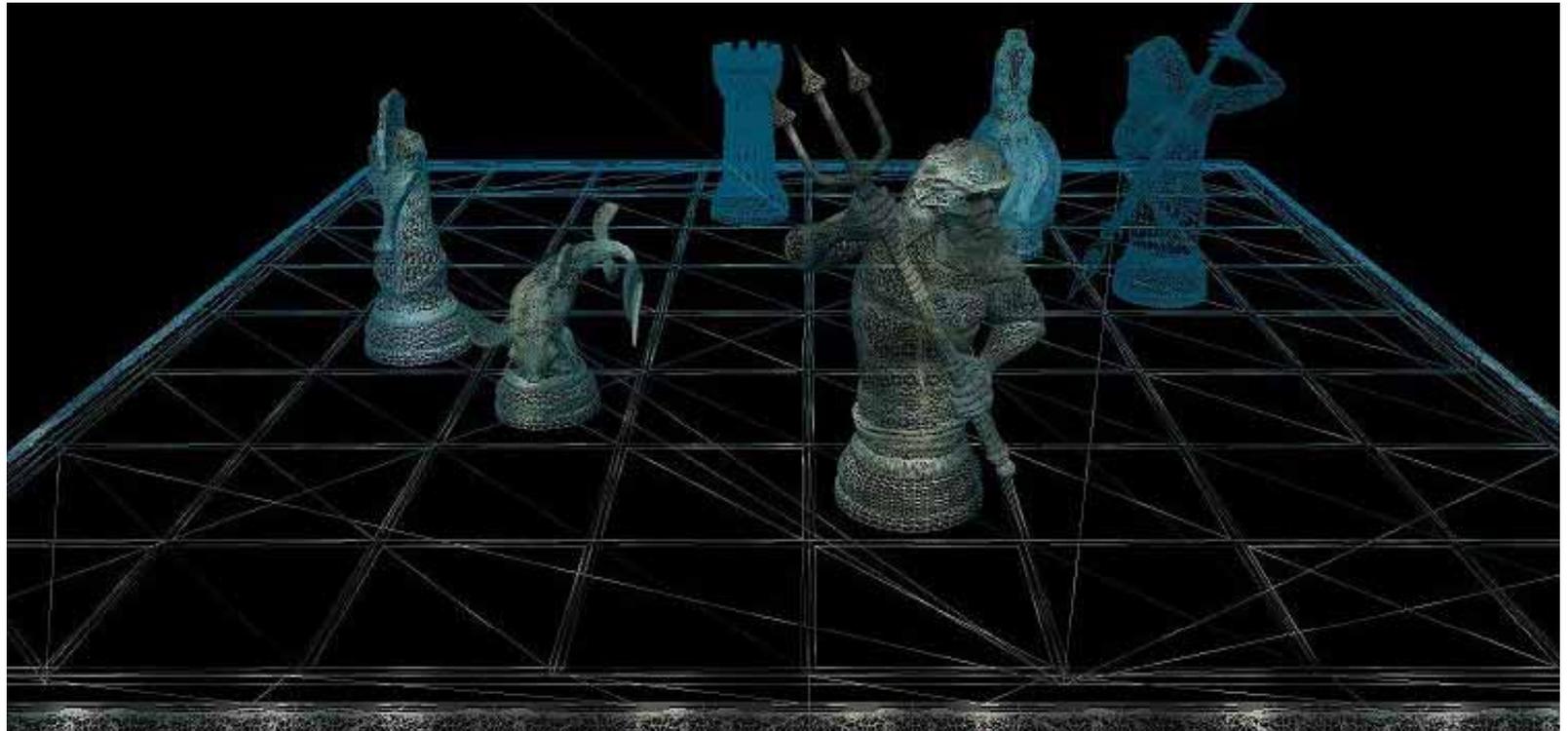
Scenario:

- ⌚ Texture shifts between two images every time mouse is moved or scene position changes
- ⌚ There is only one known mesh object used for the chess board



Flickering

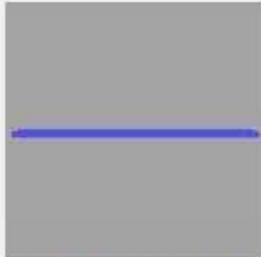
- ⊕ Examine wireframe for obvious z-fighting



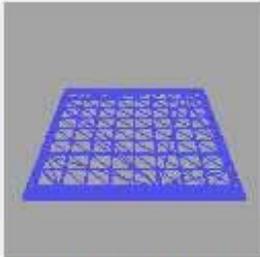


Flickering

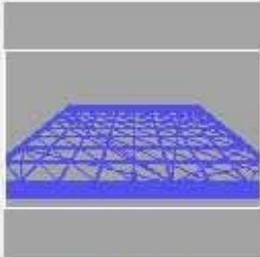
🔍 Examine mesh view for hidden artifacts



Pre-Vertex Shader



Post-Vertex Shader



Viewport

PreVS | **PostVS**

Prim	VTX	IDX	Position				Diffuse	Fog	TexCoord0		TexCoord1	
P0	0	4	-3.647	0.206	5.252	5.351	D3DCOLOR_ARGB(0x00,0x33,0x33,0x33)	0.387	0.756	0.968	-1.813	4.014
	1	5	-3.647	0.745	5.810	5.910	D3DCOLOR_ARGB(0x00,0x33,0x33,0x33)	0.292	0.701	0.968	-1.813	4.432
	2	6	-3.647	1.285	6.369	6.468	D3DCOLOR_ARGB(0x00,0x33,0x33,0x33)	0.195	0.645	0.968	-1.813	4.851
P1	3	7	-3.647	0.206	5.252	5.351	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)	0.387	0.756	0.968	-1.813	4.014
	4	8	-3.647	1.285	6.369	6.468	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)	0.195	0.645	0.968	-1.813	4.851
	5	9	-3.647	-0.872	4.135	4.235	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)	0.566	0.866	0.968	-1.813	3.176
P2	6	10	-3.647	-0.872	4.135	4.235	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.566	0.866	0.968	-1.813	3.176
	7	11	-3.647	4.005	0.000	0.000	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.405	0.045	0.000	4.013	4.054



Flickering

🕒 Hidden mesh subset uncovered

Pre-Vertex Shader

Post-Vertex Shader

Viewport

PreVS | PostVS

Prim	VTX	IDX	Position				Diffuse	Fog	TexCoord0		TexCoord1	
P0	0	0	3.602	-1.883	3.026	3.126	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.726	1.000	1.000	1.790	2.344
	1	1	-3.602	-1.883	3.026	3.126	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.726	0.000	1.000	-1.790	2.344
	2	2	3.602	2.397	7.459	7.558	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.002	1.000	0.000	1.790	5.669
P1	3	3	-3.602	2.397	7.459	7.558	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.002	0.000	0.000	-1.790	5.669
	4	2	3.602	2.397	7.459	7.558	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.002	1.000	0.000	1.790	5.669
	5	1	-3.602	-1.883	3.026	3.126	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)	0.726	0.000	1.000	-1.790	2.344

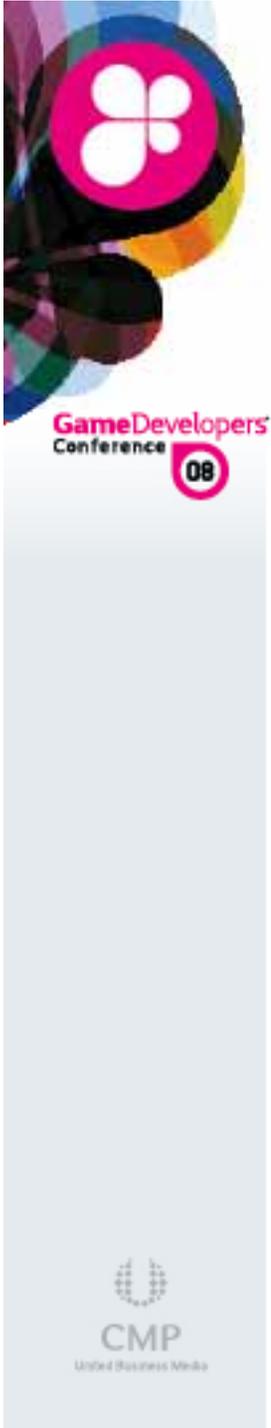


Game Developers
Conference 08

Flickering

Conclusion:

- ⊕ The checkerboard mesh had 2 subsets
- ⊕ 1 subset was coplanar with the board top
- ⊕ Removal of subset fixed unanticipated z-fighting



Bottleneck Analysis

Overall behavior is correct, but rendering takes longer than expected:

- ⊗ Culling & Render Order
- ⊗ Buffer Sizes
- ⊗ Ineffective Code
- ⊗ Inefficient Shaders
- ⊗ Batch Sizes



Culling & Render Order

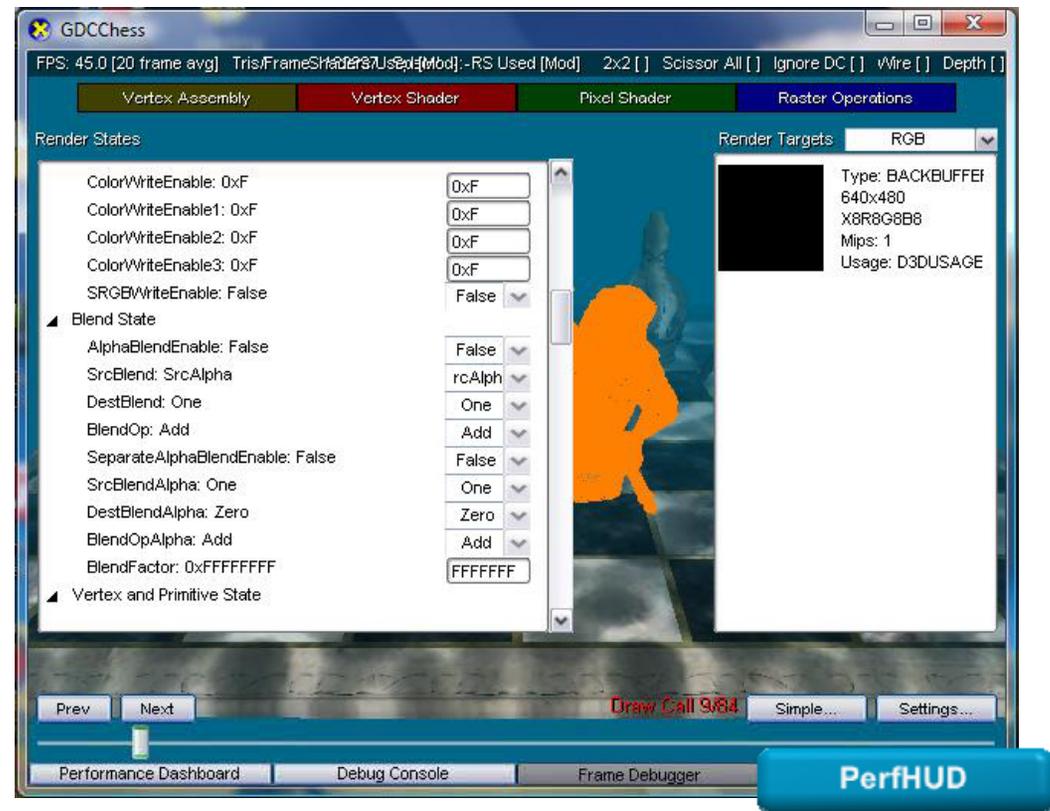
- ⊕ Notice how the draws are just stacking and nothing is culled
- ⊕ Are objects being rendered multiple times?





Culling & Render Order

- ⊕ Check the render states
- ⊕ Render state changes can happen in multiple places





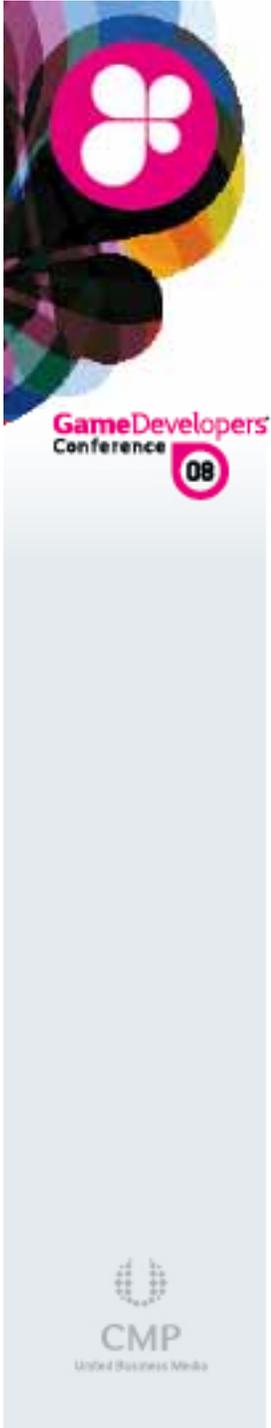
Culling & Render Order

- 🕒 You want to draw where the culling behavior will have the most effect.

The screenshot shows the GDCChess application window. At the top, it displays performance metrics: FPS: 13.0 [20 frame avg], Tris/Frame: 23257, Used [Mod]: -RS Used [Mod], 2x2 [], Scissor: All [], Ignore DC [], Wire [], Depth []. Below this is a navigation bar with four tabs: Vertex Assembly (selected), Vertex Shader, Pixel Shader, and Raster Operations. The main view is split into two panels. The left panel, labeled 'Wireframe', shows a green character model with a trident, enclosed in a wireframe bounding box. The right panel shows the same character model in a solid orange color. On the right side of the window, there is a 'PerfHUD' overlay displaying technical data for the current draw call:

```
Index / Vertex Buffer
Call type: DrawIndexedPrimitive
Type: D3DPT_TRIANGLELIST
BaseVertexIndex 0
MinVertexIndex 0
NumVertices 23257
startIndex 0
primCount 43926
HRESULT: 0x00000000
Msg: S_OK
Desc: The function completed succes
** Index Buffer Description **
Format : INDEX16
Pool : D3DPOOL_MANAGED
Usage : 0
Length : 275052 bytes
** VB Declaration **
Total vertex size: 32
0 - POSITION FLOAT3 DEFAULT
0 - NORMAL FLOAT3 DEFAULT
```

At the bottom of the window, there are buttons for 'Prev', 'Next', 'Draw/Call 9/84', 'Simple...', and 'Settings...'. Below these buttons is a 'Performance Dashboard' bar with tabs for 'Performance Dashboard', 'Debug Console', 'Frame Debugger', and 'Frame Profiler'. A blue 'PerfHUD' button is located in the bottom right corner of the application window.



Culling & Render Order

- ④ Remember that transparent objects must be drawn after opaque objects. They also need to be drawn via the painters algorithm.
 - ④ Render back to front



Culling & Render Order

Guidelines:

- ④ Order of culling methods used:
 - ④ Software (portal/scene)
 - ④ View Frustum
 - ④ Z-test
 - ④ Bounding box – hw queries
(did any pixels render or potentially render?)



Buffer Sizes

- ⌚ Performance is slow
- ⌚ But everything looks correct
- ⌚ Thrashing of system resources



Buffer Sizes

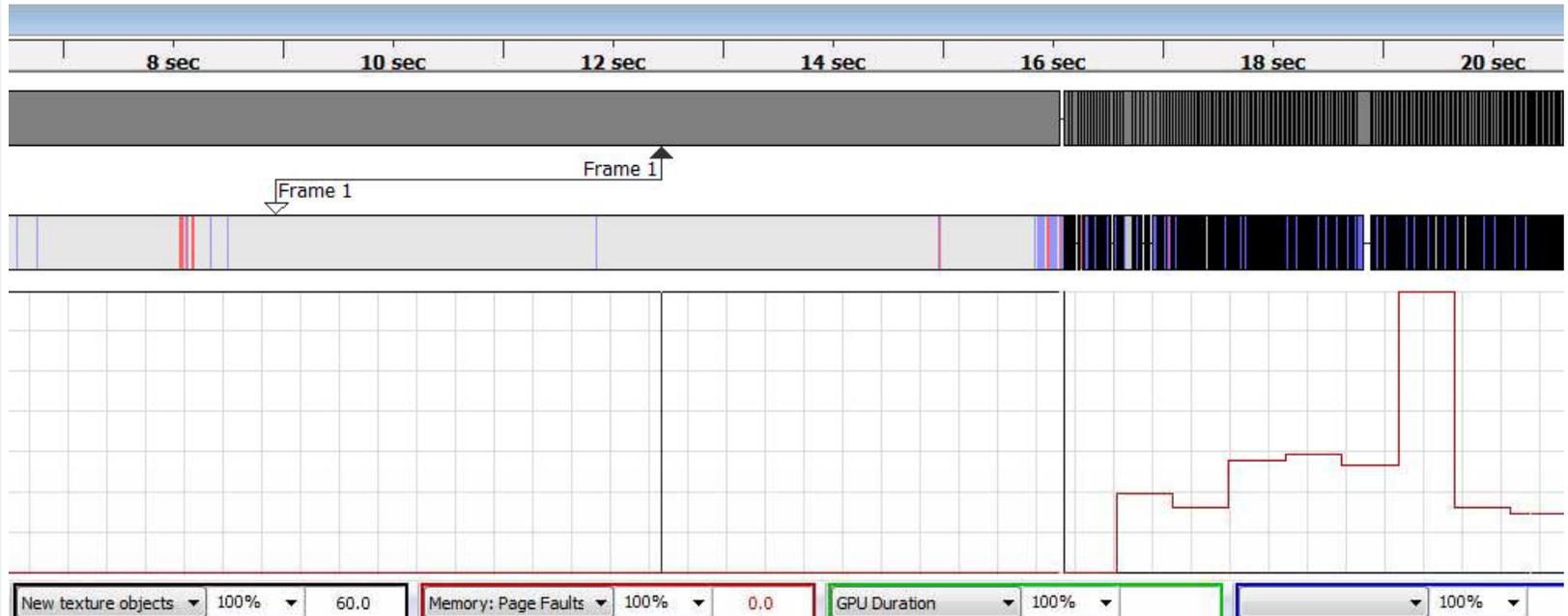
There could be lots of swapping occurring





Buffer Sizes

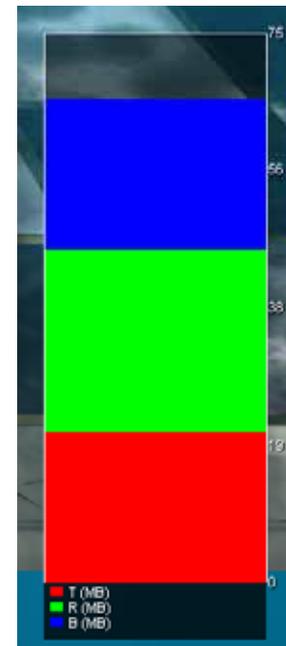
- Look at the perfmon counter for memory page faults
is it too high?





Buffer Sizes

- ⌚ Is the swapping due to textures or other buffers
- ⌚ Look at the signals in PerfHUD





Buffer Sizes

Sort the object table textures in PIX by size

Address	Type	Created By	Creation	Destruction	Status	App Refs	Size	Pool	Usage	Format	Width	Height	Depth	Mips
0x02988680	D3D9 Vertex Buffer	Application	1	649	Alive	1	435,584 bytes	Managed		D3DFMT_VERTEXDATA				
0x090CDD10	D3D9 Surface	Direct3D	1	649	Alive	0	8 bytes	Managed		D3DFMT_DXT1	4	4		
0x0AA0D598	D3D9 Surface	Direct3D	1	649	Alive	0	16 bytes	Managed		D3DFMT_X8R8G8B8	2	2		
0x02904AF0	D3D9 Surface	Direct3D	1	649	Alive	0	1,228,800 bytes	Default	DepthStencil	D3DFMT_D24X8	640	480		
0x0AB4DC28	D3D9 Surface	Direct3D	1	649	Alive	0	262,144 bytes	Managed		D3DFMT_X8R8G8B8	256	256		
0x090CBE30	D3D9 Texture	Application	1	649	Alive	1	2,744 bytes	Managed		D3DFMT_DXT1	64	64		7
0x0AB4EBE8	D3D9 Surface	Direct3D	1	649	Alive	0	1,024 bytes	Managed		D3DFMT_X8R8G8B8	16	16		
0x090CDD88	D3D9 Surface	Direct3D	1	649	Alive	0	8 bytes	Managed		D3DFMT_DXT1	2	2		
0x0AA0E5E0	D3D9 Index Buffer	Application	1	649	Alive	1	49,152 bytes	Default	WriteOnly	D3DFMT_INDEX16				
0x0AA0D640	D3D9 Surface	Direct3D	1	649	Alive	0	4 bytes	Managed		D3DFMT_X8R8G8B8	1	1		
0x0AB4DCD0	D3D9 Surface	Direct3D	1	649	Alive	0	65,536 bytes	Managed		D3DFMT_X8R8G8B8	128	128		
0x0AB4EC90	D3D9 Surface	Direct3D	1	649	Alive	0	256 bytes	Managed		D3DFMT_X8R8G8B8	8	8		
0x0AD17F00	D3D9 State Block	Application	1	649	Alive	1								

169 of 169 objects displayed



Game Developers
Conference

08

Buffer Sizes

- ④ Use mip-mapped textures
- ④ Use smaller textures
- ④ Use a compact texture format
- ④ Don't become infatuated with new features
E.g. Selectively use aniso on textures





Buffer Sizes

- ④ Only use data where necessary
- ④ Pack data buffers with a smaller vdecl
- ④ Use LOD techniques to reduce the amount of data needed
- ④ Use a paging algorithm for loading data
- ④ Reuse Render targets when possible

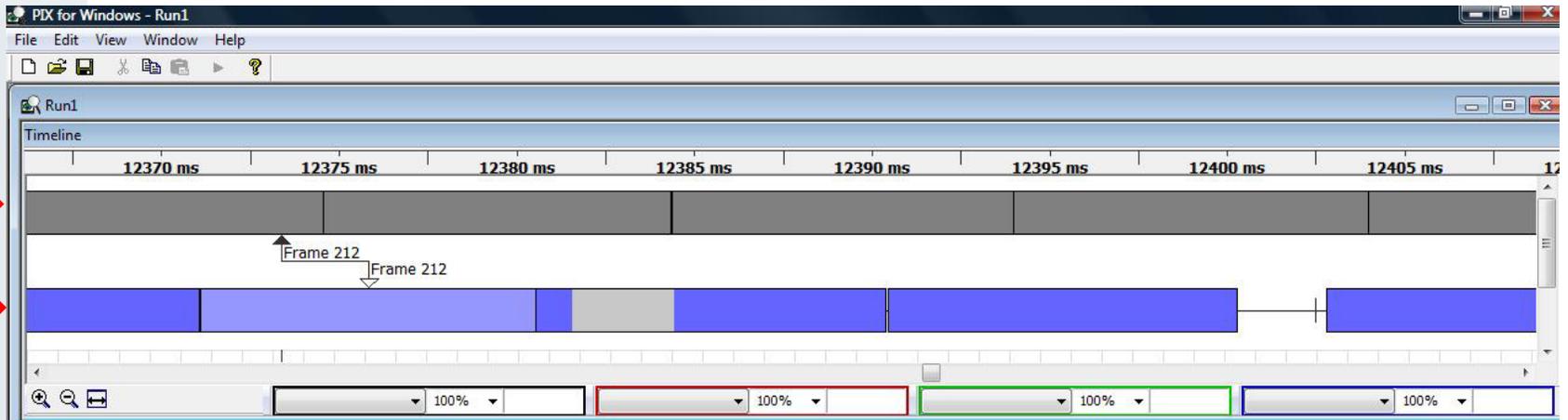


Inefficient Code

- ⌚ Are you sure you are GPU bound?
- ⌚ Look at the timing in PIX, PerfHUD

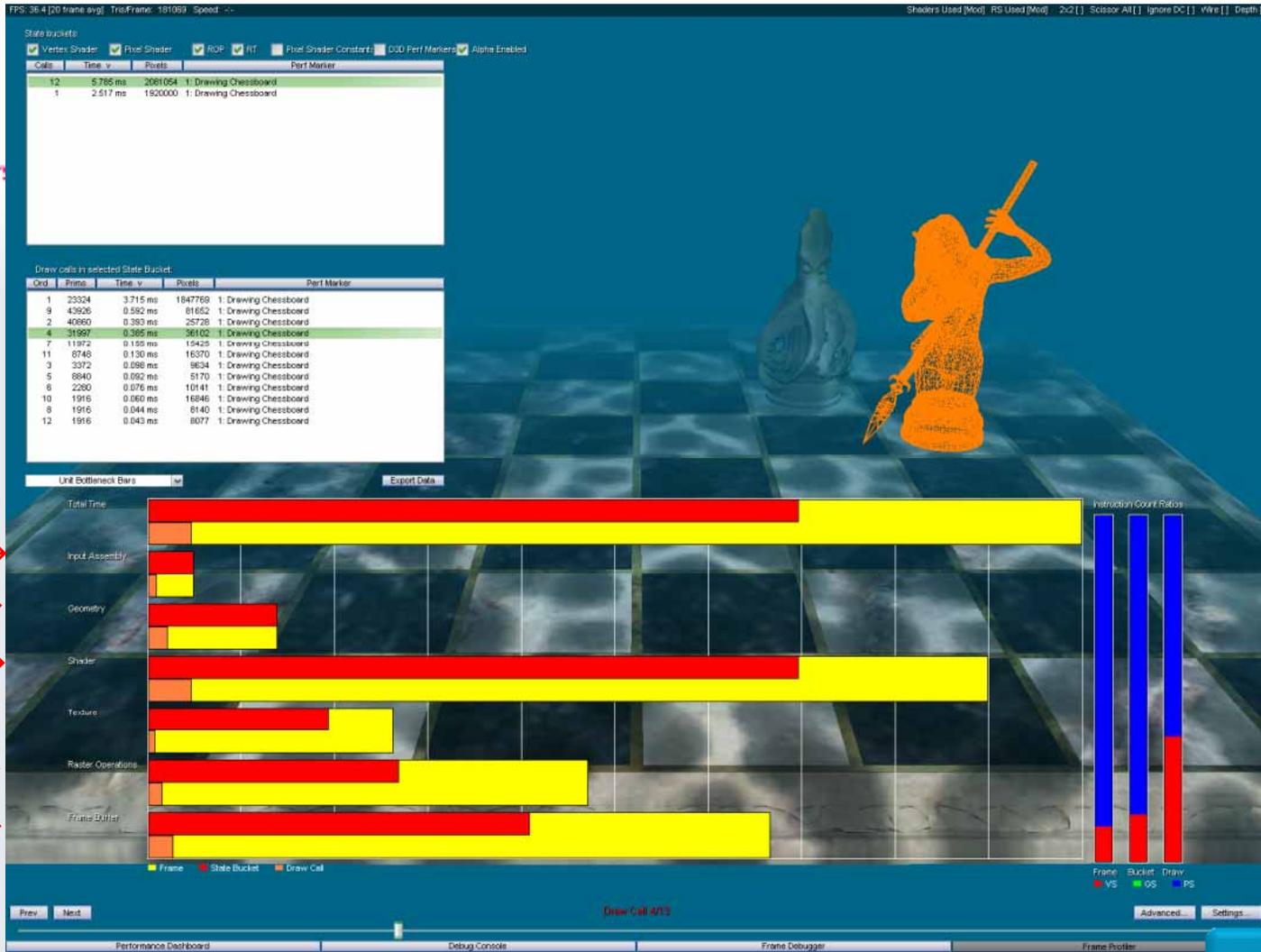
CPU →

GPU →





Inefficient Code



- Total time →
- Input Assembly →
- Geometry →
- Shader →
- Texture →
- Raster Ops →
- Frame Buffer →





Inefficient Code

⊕ Adjust

render size, texture sizes, cull objects



PerfHUD

WWW.GDCONF.COM



Inefficient Code

- Still slow? CPU bound
- Redundant state setting, set texture calls

EID	Event	StartTime	Frame	Duration	FPS
917	<0x029071B8> IDirect3DDevice9::BeginScene()	19801423920			
918	<0x02991B20> ID3DXEffect::SetFloat(0x005F6104, 13.949f)	19801436160			
919	<0x02991B20> ID3DXEffect::SetMatrix(0x005F60A0, 0x0012FA10)	19801448448			
920	<0x02991B20> ID3DXEffect::SetMatrix(0x005F602C, 0x0012F9D0)	19801452544			
921	<0x02991B20> ID3DXEffect::SetMatrix(0x005F5FBC, 0x0012F990)	19801456640			
922	<0x02991B20> ID3DXEffect::SetTechnique(0x005F5F38)	19801464832			
923	<0x02991B20> ID3DXEffect::SetVector(0x005F5F24, 0x0012F93C)	19801470976			
924	<0x02991B20> ID3DXEffect::SetVector(0x005F5F18, 0x0012F914)	19801477120			
925	<0x029071B8> IDirect3DDevice9::SetRenderState(D3DRS_FOGENABLE, TRUE)	19801481216			
926	<0x02991B20> ID3DXEffect::SetVector(0x005F5F00, 0x0012F904)	19801489408			
927	<0x029071B8> IDirect3DDevice9::SetRenderState(D3DRS_CULLMODE, D3DCULL_C	19801493504			
928	<0x02991B20> ID3DXEffect::SetTexture(0x005F5EF0, 0x090CBA70)	19801499648			
1010	User Event: Drawing Chessboard	19801667584		90865664	
1011	<0x029071B8> IDirect3DDevice9::SetRenderState(D3DRS_CULLMODE, D3DCULL_C	19801675776			



Inefficient Shaders

Use a tool to analyze your shader

Shader Performance

Startup Form `lambSkin.cgfx`

Analyze a Pass
Compare Passes

Techniques:

- UntexturedVS
 - p0
- TexturedVS
 - p0
- UntexturedPS
 - p0
- TexturedPS
 - p0

Show: Fragment Shader

Drivers: 163.20

GPUs:

- GeForceFX 5800 Ultra (NV30)
- GeForceFX 5900 Ultra (NV35)
- GeForceFX 5200 Ultra (NV34)
- GeForceFX 5600 Ultra (NV31)
- GeForceFX 5700 Ultra (NV36)
- GeForceFX 5950 Ultra (NV38)
- GeForce 6800 Ultra (NV40)
- GeForce 6800 GT (NV40-GT)
- GeForce 6800 (NV40-12)
- GeForce 6800 GT (NV43-GT)
- GeForce 6200 (NV44)

	Version 163.20														
	Regs.					Cycles					MPix/s				
	Normal	FP16	FP32	FP16 Min Branch	FP32 Min Branch	Normal	FP16	FP32	FP16 Min Branch	FP32 Min Branch	Normal	FP16	FP32	FP16 Min Branch	FP32 Min Branch
UntexturedVS:p0															
NV30 (GeForceFX 5800 Ultra)															
NV35 (GeForceFX 5900 Ultra)															
NV34 (GeForceFX 5200 Ultra)															
NV31 (GeForceFX 5600 Ultra)															
NV36 (GeForceFX 5700 Ultra)															
NV38 (GeForceFX 5950 Ultra)															
NV40 (GeForce 6800 Ultra)	1	1	1	1	1	1	1	1	1	1	2105	2105	2105	2105	2105
NV40-GT (GeForce 6800 GT)	1	1	1	1	1	1	1	1	1	1	1305	1305	1305	1305	1305
NV40-12 (GeForce 6800)	1	1	1	1	1	1	1	1	1	1	3900	3900	3900	3900	3900
NV43-GT (GeForce 6600 GT)	1	1	1	1	1	1	1	1	1	1	4000	4000	4000	4000	4000
NV44 (GeForce 6200)	1	1	1	1	1	1	1	1	1	1	1400	1400	1400	1400	1400
G70-GT (GeForce 7800 GTX)	1	1	1	1	1	1	1	1	1	1	1730	1730	1730	1730	1730



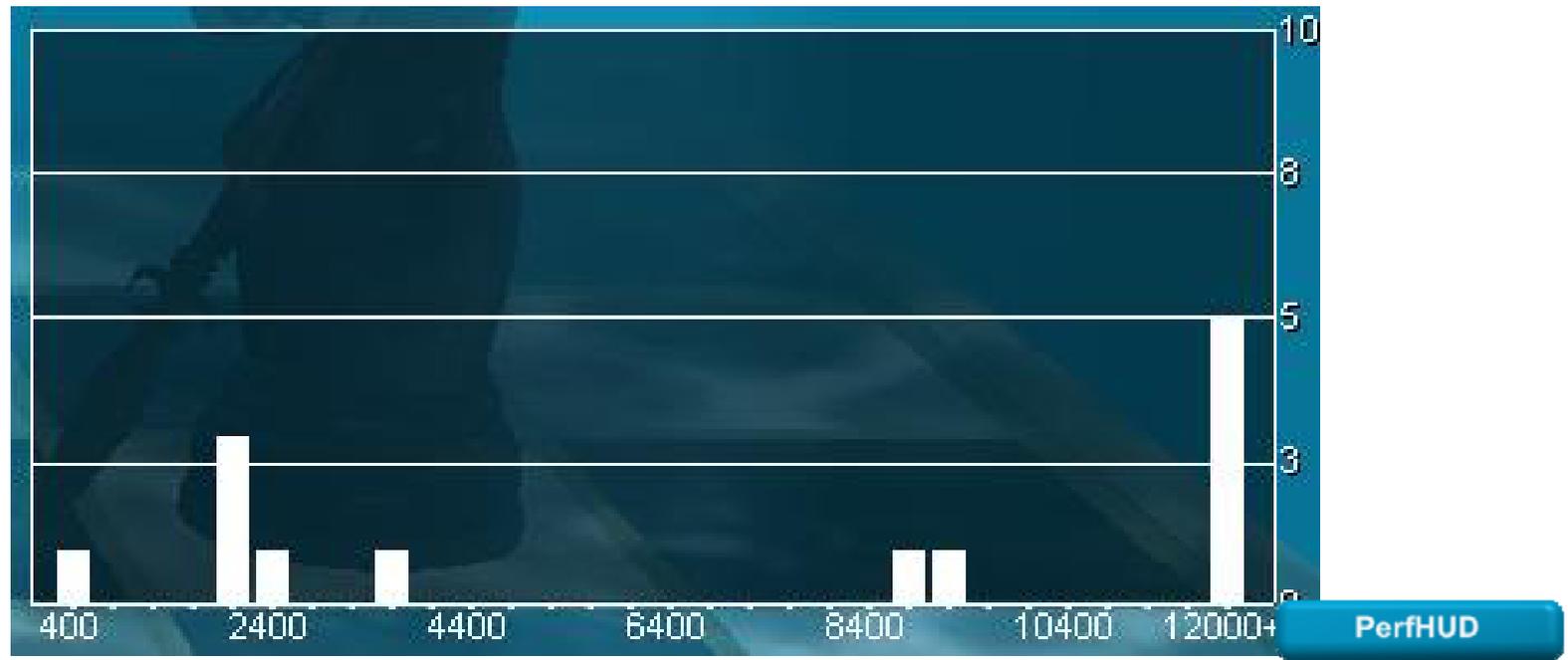
Inefficient Shaders

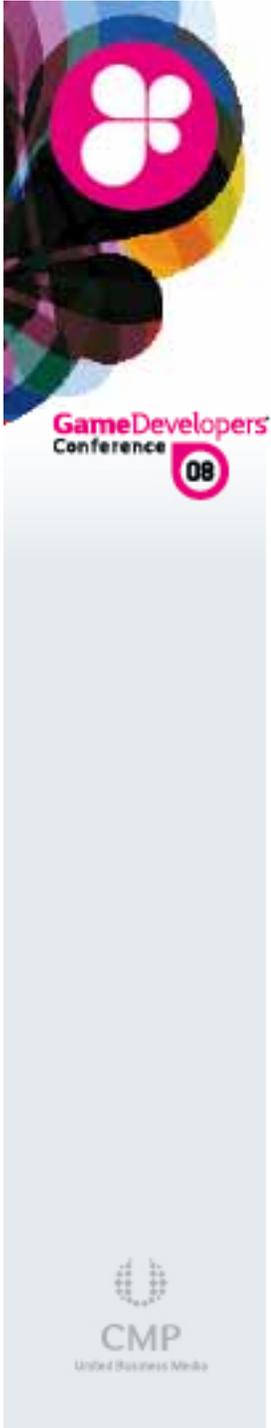
- ③ Are you sure it is the shader?
- ③ Swap the shader for a simpler shader, did that make a difference?
- ③ Suboptimal code in inner loop



Batch Sizes

- ⊕ Small batch sizes are inefficient and hard to detect
- ⊕ Just because the batches are big doesn't mean that it is good either





Summary

- ④ Tools can be a valuable aid to quickly determine root causes of a variety of graphics problems
- ④ Tools can cover a variety of debugging levels, from high-level API issues to low-level hardware issues



GameDevelopers
Conference

08

Live Demos

⌚ Microsoft - PIX for Windows

⌚ NVIDIA - PerfHUD



CMP

United Business Media

WWW.GDCONF.COM



Q&A

Questions, Comments, Concerns?

GameDevelopers
Conference

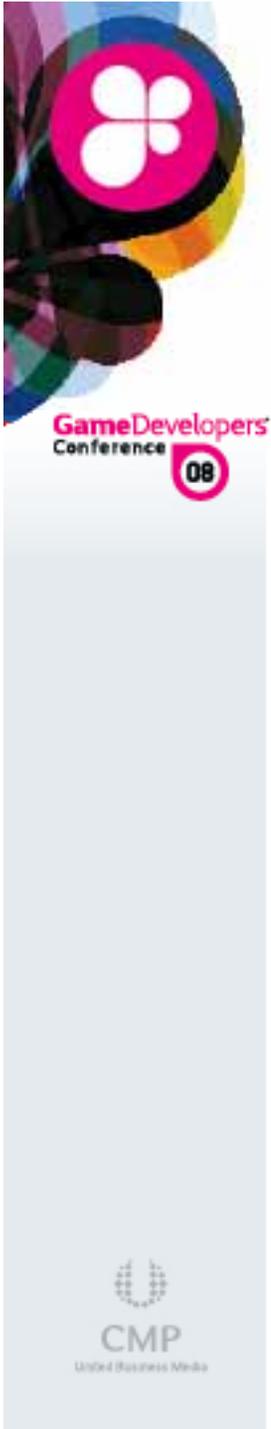
08



CMP

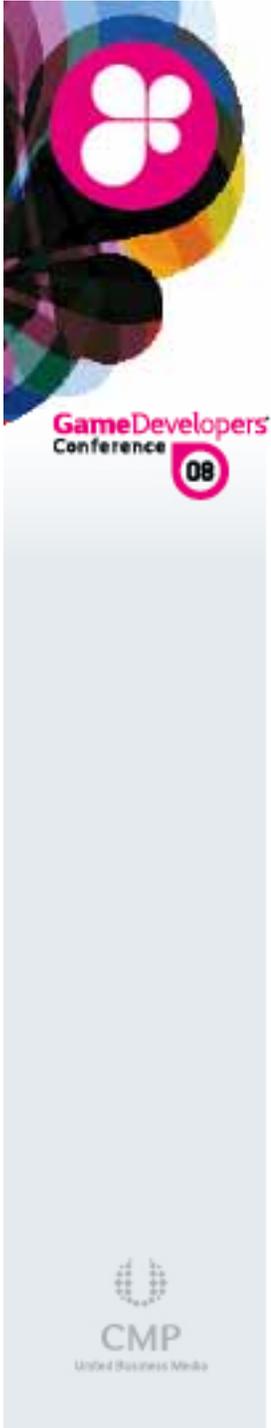
United Business Media

WWW.GDCONF.COM



Resources

- ④ Tools shown today can be downloaded at:
 - ④ AMD
 - ④ <http://developer.amd.com>
 - ④ Microsoft
 - ④ <http://msdn.microsoft.com/directx>
 - ④ NVIDIA
 - ④ <http://developer.NVIDIA.com/>
- ④ The “PIXGameDebugging” application used in this presentation is available as a d3d9 tutorial in the DirectX Software Development Kit, March 2008 release.



Resources

Recommended Newsgroups, sites, & Forums

- ③ <http://developer.NVIDIA.com/forums/>
- ③ <http://forums.xna.com/>
- ③ <http://www.gamedev.net/>
- ③ <http://developer.intel.com>
- ③ <http://www.opengl.org>
- ③ <http://www.gremedy.com/>
- ③ <http://www.acm.org>