



Beyond Printf

Debugging Graphics Through Tools



Presenters

➊ Dave Aronson

NVIDIA – Technical Evangelist

daronson@nvidia.com

➋ 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



Agenda

- ➊ Tool Selection
- ➋ Scenarios
- ➌ Live Demos
- ➍ Q&A
- ➎ References



Preliminary Criteria Points

When selecting a tool, consider:

- Ⓐ Budget
- Ⓐ General machine requirements
- Ⓐ Hardware manufacturers
- Ⓐ Additional required software
- Ⓐ Code modification requirements
- Ⓐ Product support
- Ⓐ 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



Tools Shown Today

AMD

GPU PerfStudio

[GPU PerfStudio](#)

Microsoft

PIX for Windows

[PIX for Windows](#)

NVIDIA

PerfHUD

[PerfHUD](#)

FX Composer

[FX Composer](#)



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

- ➊ Determine analysis levels of interest
 - ➋ One strategy is to start at the game asset level and work down the list
- ➋ Determine how tool fits criteria
 - ➌ Prioritize your requirements
- ➌ Experiment
 - ➍ Most tools are free or have free trial periods, try a variety of scenarios



Scenarios

- ➊ Glitches
 - ➋ Incorrect behavior

- ➌ Bottlenecks
 - ➋ Poor performance



Glitches

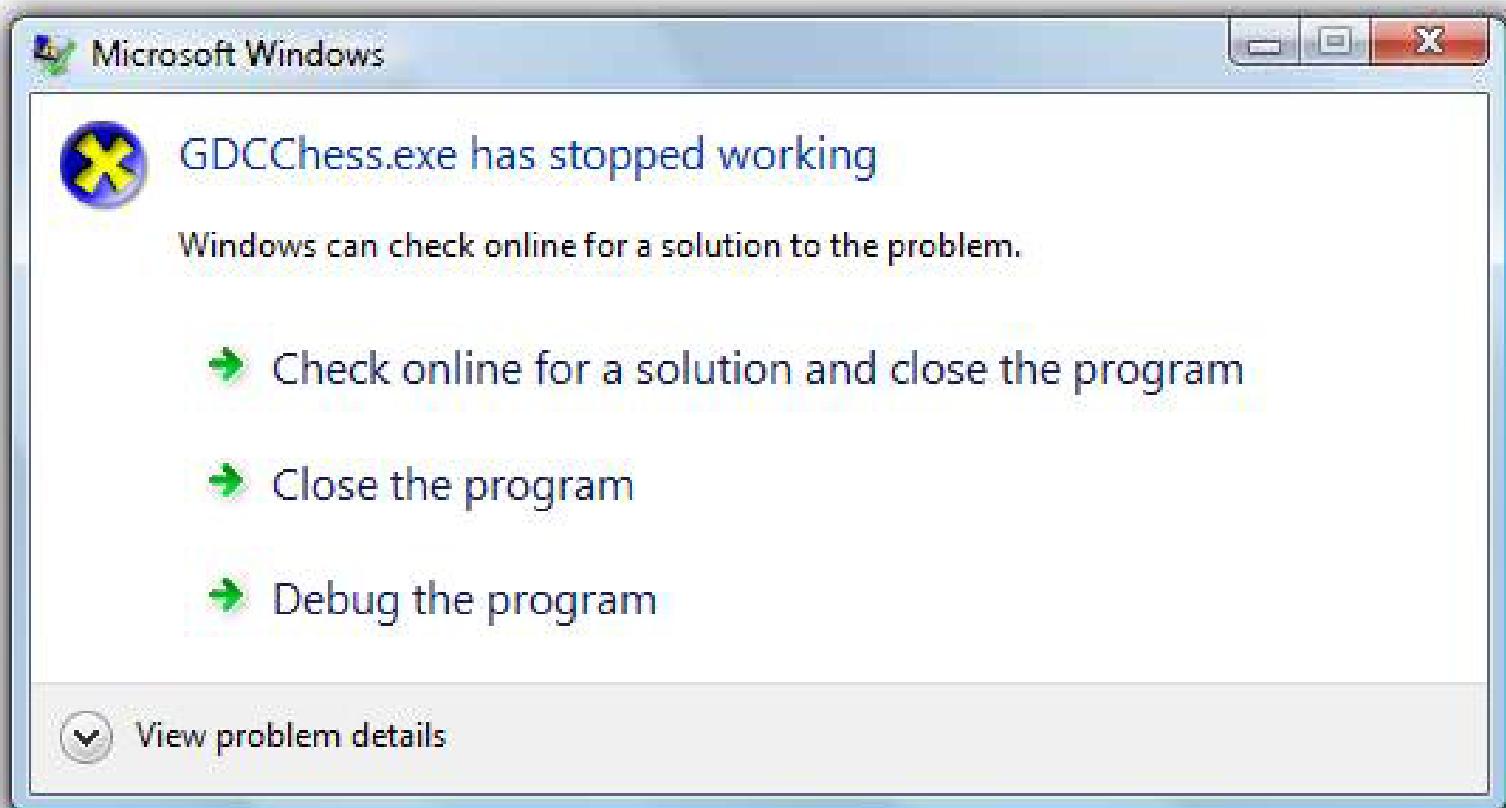
The game is not behaving as expected:

- ➊ Game Crash
- ➋ Blank Screen
- ➌ Missing Objects
- ➍ Flickering



GameDevelopers
Conference 08

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

The screenshot shows the PIX for Windows application window titled "PIX for Windows - [GameCrashScenario : Experiment (Advanced View)]". The window has a menu bar with File, Edit, View, Window, and Help. A toolbar with various icons is at the top. Below the toolbar are two tabs: "Triggers/Actions" (selected) and "Target Program". The "Triggers/Actions" tab displays a tree view of triggers: "Program Start" with "Create Run File" and "Full-stream capture Direct3D" selected. To the right of the tree view are settings for the selected trigger: "Action Type: Create Run File", "Path to PIXRun file: C:\Users\kstevens\Desktop\gdc200" with a "Browse..." button, and a checked checkbox for "Disable write caching". A note below states: "Note: Disabling write caching makes capture slower, but is likely to record more information if the target program crashes." At the bottom left is a "Fewer Options" button, and at the bottom right is a "Start Experiment" button. The status bar at the bottom says "Ready".



GameDevelopers
Conference 08

Game Crash

⌚ Setup diagnostic logging

PIX for Windows - [GameCrashScenario : Experiment (Advanced View)]

File Edit View Window Help

Triggers/Actions Target Program

Target startup options

Program path: C:\Users\kstevens\Documents\gdc2008\GDCChess\Debug\GDCChess.exe

Startup folder:

Command-line arguments:

Skip 0 processes before gathering

Record a diagnostic log

Include debug output messages in diagnostic log

Disable D3DX analysis

Fewer Options Start Experiment

Ready



GameDevelopers
Conference
08

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, 0xA934D60)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEEB00, 0xA934D60)
Frame 000003 .....POST: <0><this=0x03ceeb00> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: <this=0x03cef98>IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: RemoveObject(D3D9 State Block, 0x03CEFB98, 0xA934EC0)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEFB98, 0xA934EC0)
Frame 000003 .....POST: <0><this=0x03cef98> IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: <this=0x03cef610>IDirect3DStateBlock9::Release()
Frame 000003 .....PRE: RemoveObject(D3D9 State Block, 0x03CEF610, 0xA934E40)
Frame 000003 .....POST: <> RemoveObject(D3D9 State Block, 0x03CEF610, 0xA934E40)
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

WWW.GDCONF.COM



Game Crash

Analysis:

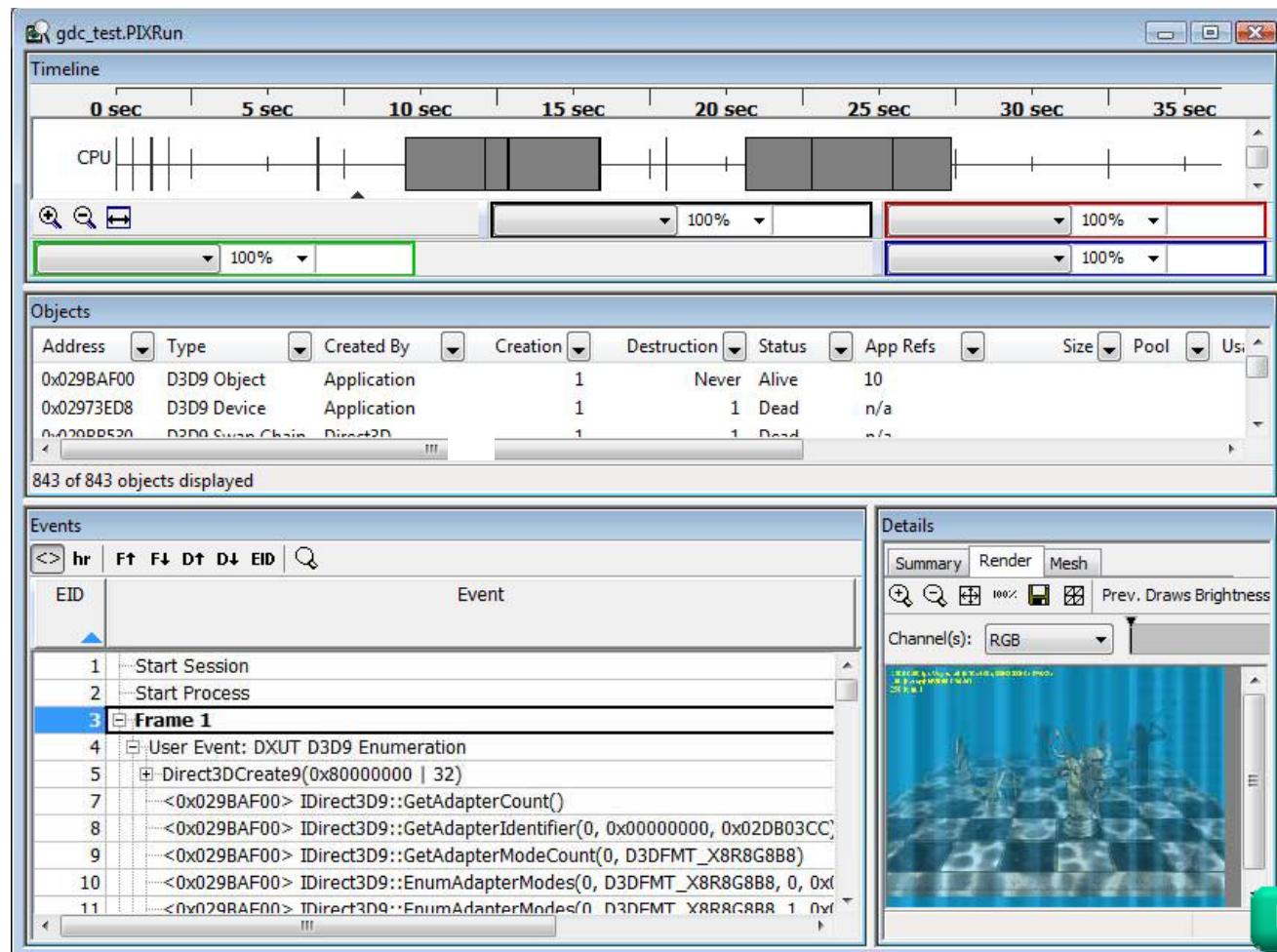
- ⌚ Error: Direct3D9: (ERROR) :All user created D3DPOOL_DEFAULT surfaces must be freed before ResetEx can succeed. ResetEx Fails. An unhandled exception occurred.



GameDevelopers
Conference
08

Game Crash

⌚ Open run file for analysis



PIX for Windows

WWW.GDCONF.COM

CMP
United Business Media



Game Crash

⌚ Examine objects left after last valid call

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 rogue object creation point

GameDevelopers Conference 08

Ad	View Surface 0x02A7A588	App Refs	Pool	Usage	Format
0x0	Object Operations	0	Default	RenderTarget	D3DFMT
0x0	IDirect3DSurface9* 0x02A7A588	0	Default	Dynamic	D3DFMT
0x0	Creation: Frame 1, EID 10676	0	Default	Dynamic	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: D3DPPOOL_DEFAULT	n/a	Default	RenderTarget	D3DFMT
0x0	Dimensions: 256 x 256	n/a	Default	RenderTarget	D3DFMT
	App References: 0				



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> IDirect3DTexure9::LockRect(0, 0x0017F6DC, NU



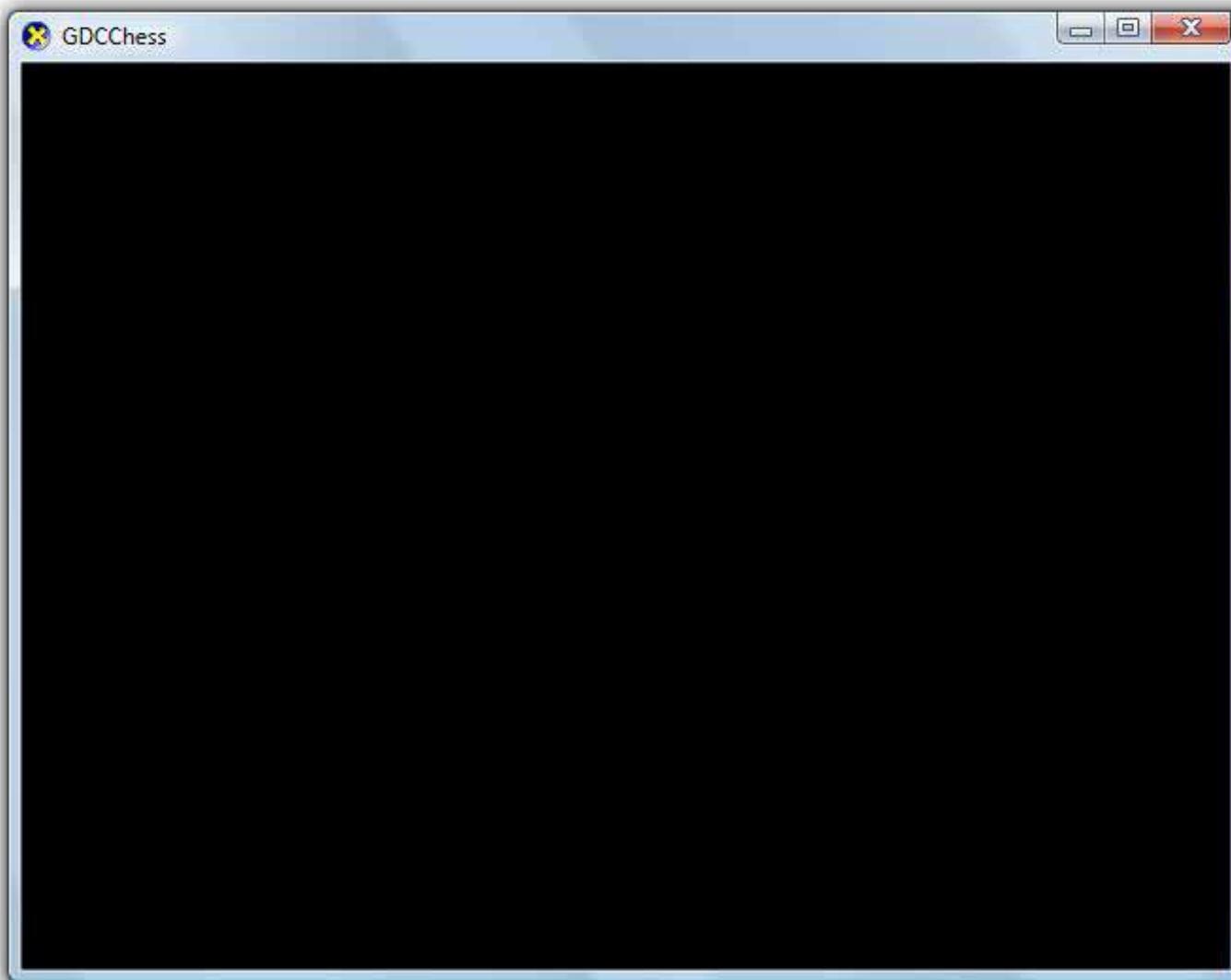
Game Crash

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)



Blank Screen



WWW.GDCONF.COM



Blank Screen

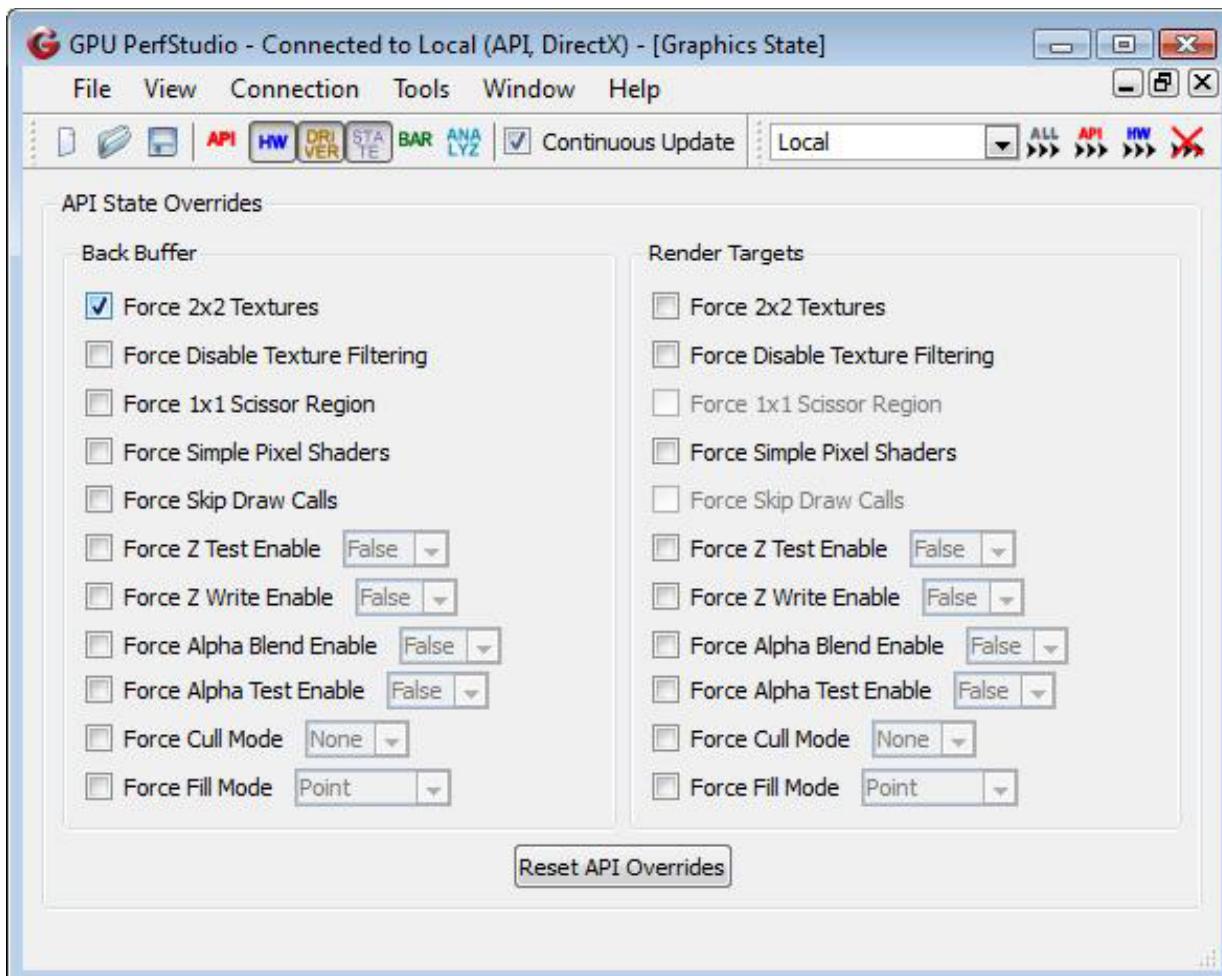
Scenario:

- Ⓐ 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

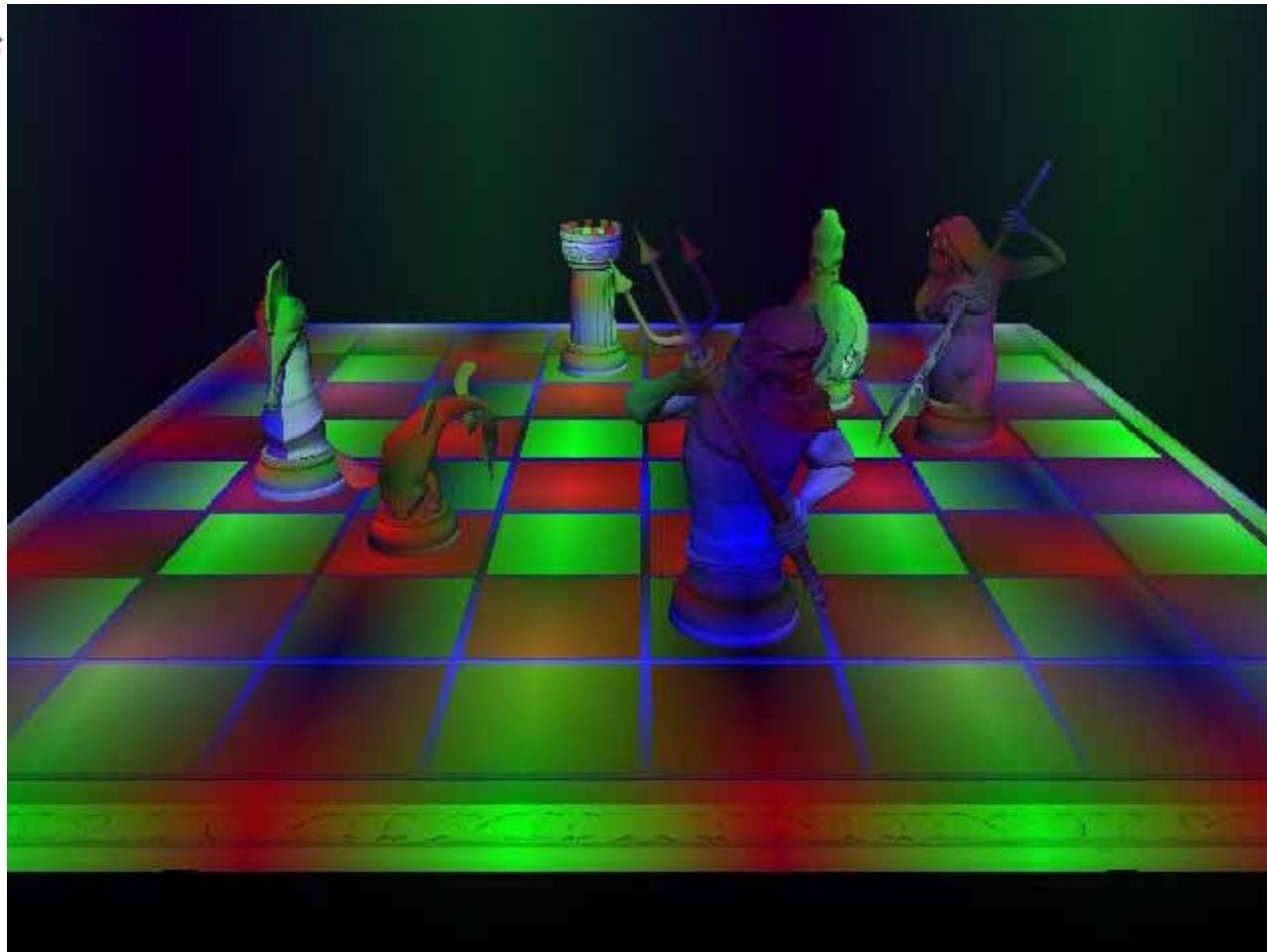




Game Developers
Conference 08

Blank Screen

- ⌚ Overriding texture renders scene viewable



GPU PerfStudio

CMP
United Business Media

WWW.GDCONF.COM



Game Developers
Conference 08

Blank Screen

⌚ Checking for sampler issues

The screenshot shows the PIX for Windows interface. The left pane, titled "Events", lists various GPU API calls in a tree view, including `ID3DXEffect::BeginPass()`, `ID3DXMesh::DrawSubset()`, and `IDirect3DDevice9::SetVertexData`. The right pane, titled "Details", is focused on "Device 0x03D2AD68". It has tabs for "Summary", "Render" (which is selected), "Mesh", "Device Info", "Input State", "Tessellation State", "Vertex State", "Pixel State", and "Output State". The "Render" tab shows a table for samplers:

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

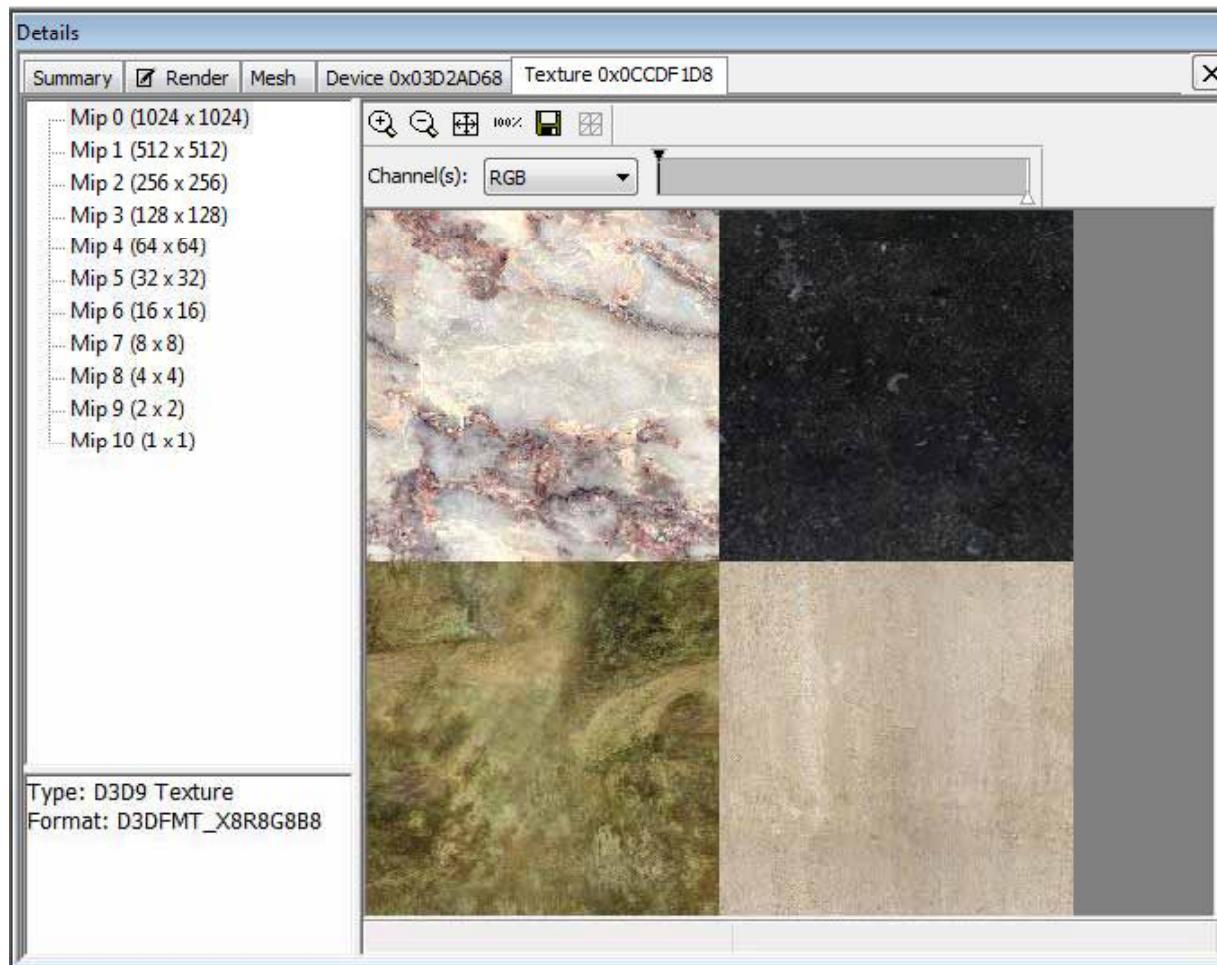
⌚ Samplers exist, values look ok



GameDevelopers
Conference 08

Blank Screen

⌚ Check texture sampler 0 - OK



CMP
United Business Media

PIX for Windows

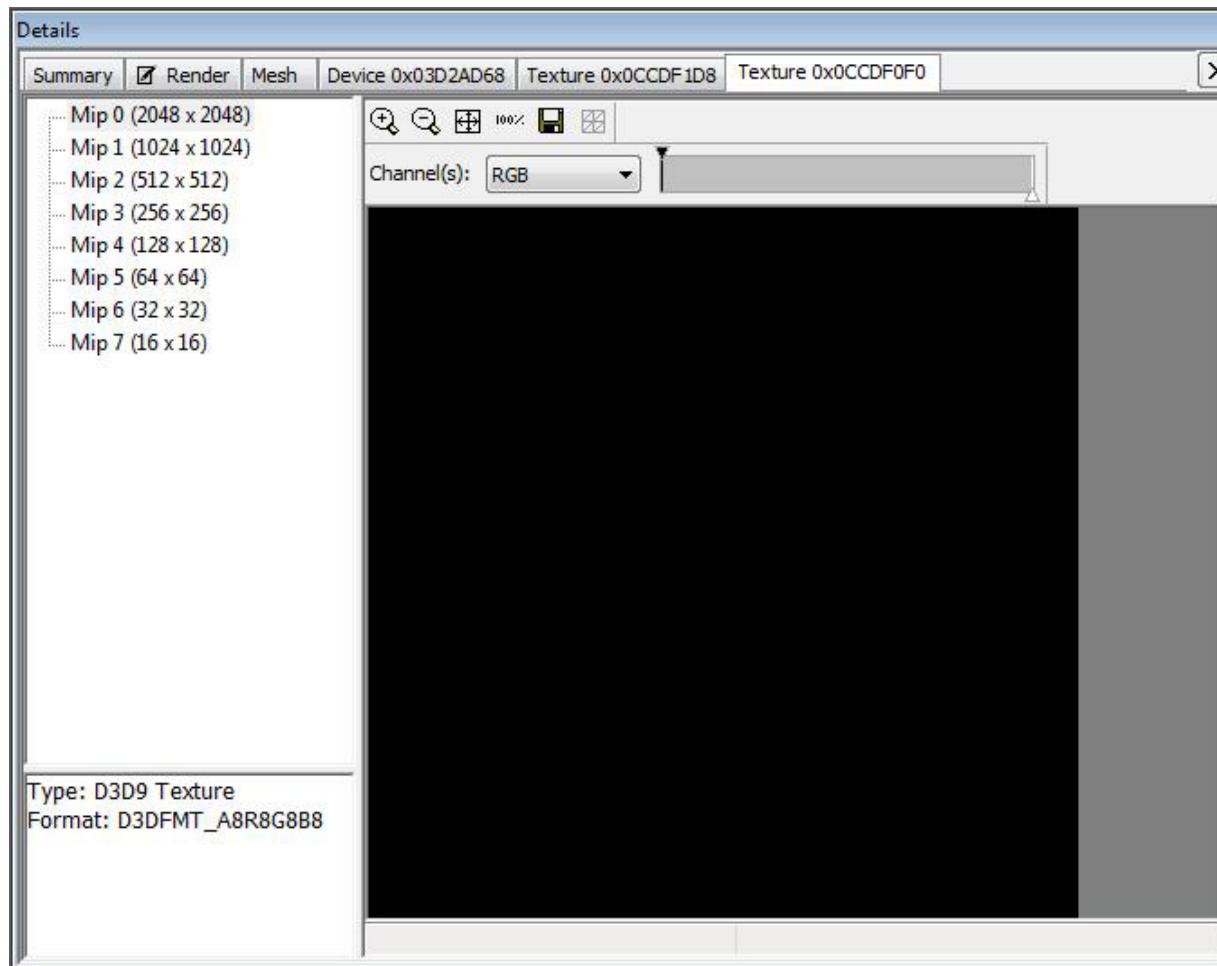
WWW.GDCONF.COM



GameDevelopers
Conference 08

Blank Screen

⌚ Sampler texture 1 should not be black



CMP
United Business Media

PIX for Windows

WWW.GDCONF.COM



GameDevelopers
Conference 08

Blank Screen

⌚ Render frame and select inaccurate pixel

The screenshot shows the PIX for Windows interface. On the left, there's a tree view of events under 'Frame 353'. The right side shows a black render frame with a context menu open over it. The menu includes 'Save Picture As...' and 'Debug This Pixel...'. The PIX interface has tabs for 'Summary', 'Render' (which is checked), and 'Mesh'. There are also various toolbars and status bars at the top.

- F↑ F↓ D↑ D↓ EID | 🔎
- Event
- Start Session
- Start Process
- Frame 353
 - User Event: Method: OnFrameMove
 - User Event: Method: OnD3D9FrameRen
 - <0x015D4F50> IDirect3DDevice9::Cle
 - <0x015D4F50> IDirect3DDevice9::Beg
 - <0x015D4360> ID3DXEffect::SetFloat
 - <0x015D4360> ID3DXEffect::SetTech
 - <0x015D4360> ID3DXEffect::SetVecto
 - <0x015D4360> ID3DXEffect::SetVecto
 - <0x015D4F50> IDirect3DDevice9::Set
 - <0x015D4360> ID3DXEffect::SetVecto
 - <0x015D4F50> IDirect3DDevice9::Set
 - <0x015D4360> ID3DXEffect::SetTextu
 - + User Event: Drawing Scene
 - + User Event: Method: DrawLightShafts
 - <0x015D4F50> IDirect3DDevice9::Enc
 - Save Picture As...
 - Debug This Pixel...

PIX for Windows

CMP
United Business Media

WWW.GDCONF.COM



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)

PIX for Windows



GameDevelopers
Conference

08

Blank Screen

- ⌚ Shader debugging proves black texture obliterates computed color

The screenshot shows the PIX for Windows debugger interface. At the top, there are tabs for Summary, Render, Mesh, and Debugger, with Debugger selected. Below the tabs are navigation icons for stepping through code. The main area displays the Disassembly tab for the file PIXGameDebugging.fx. The assembly code is as follows:

```
//-----  
  
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);  
}
```

Below the assembly code is a table showing variable values:

Name	Type
color	float3
movement	float2

The 'Registers' tab is selected in the table header. The variable 'color' has a value of (0.000, 0.000, 0.000). The variable 'movement' has a value of (0.845, 4.548).

In the bottom right corner, there is a green button labeled "PIX for Windows".

At the bottom left, there is a logo for CMP United Business Media.

At the bottom right, the website address is listed as WWW.GDCONF.COM.



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

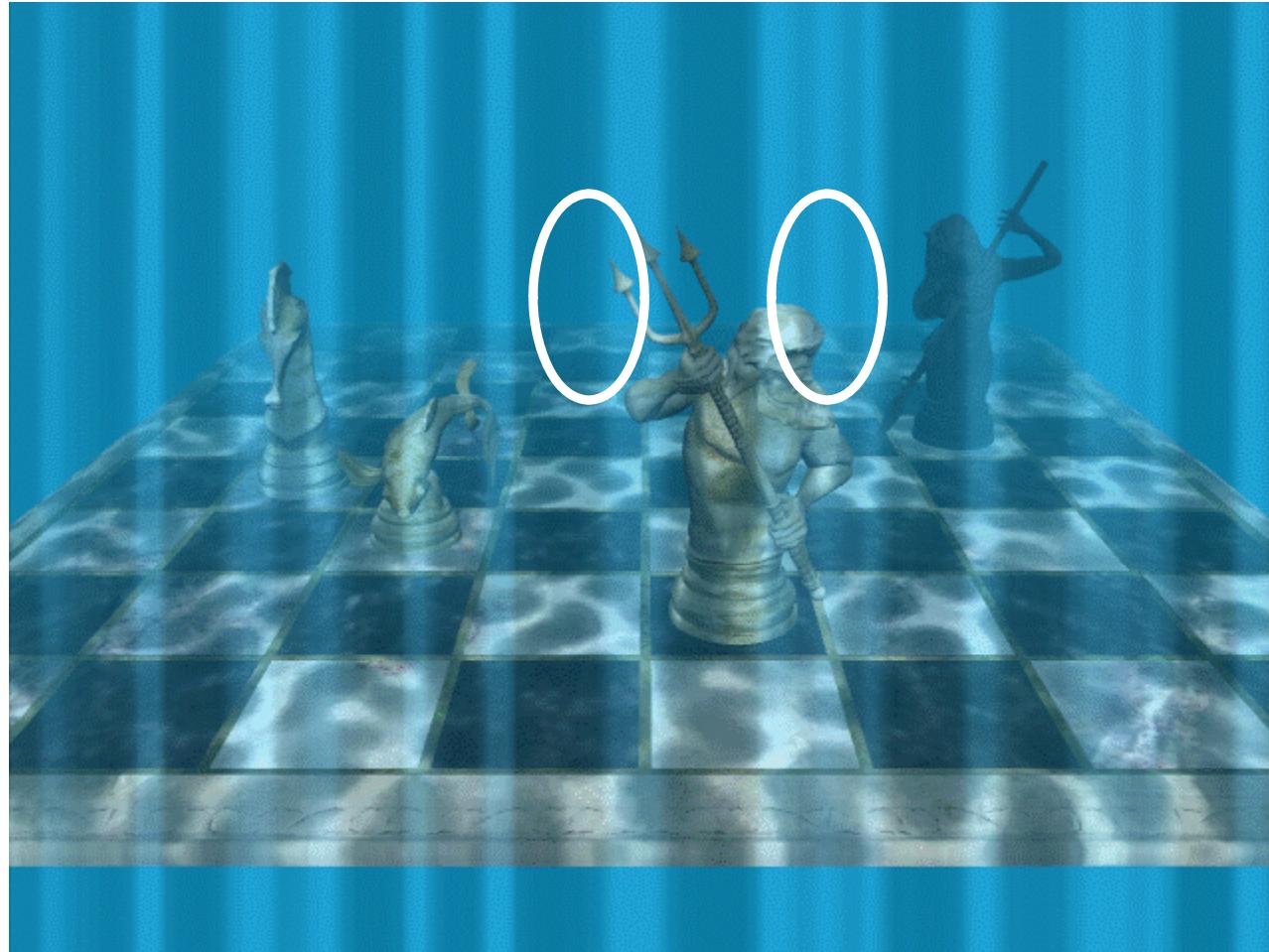
Conclusion:

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



GameDevelopers
Conference 08

Missing Objects



CMP
United Business Media

WWW.GDCONF.COM



Missing Objects

Scenario:

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



Missing Objects

- ⌚ Rendered scene has missing objects

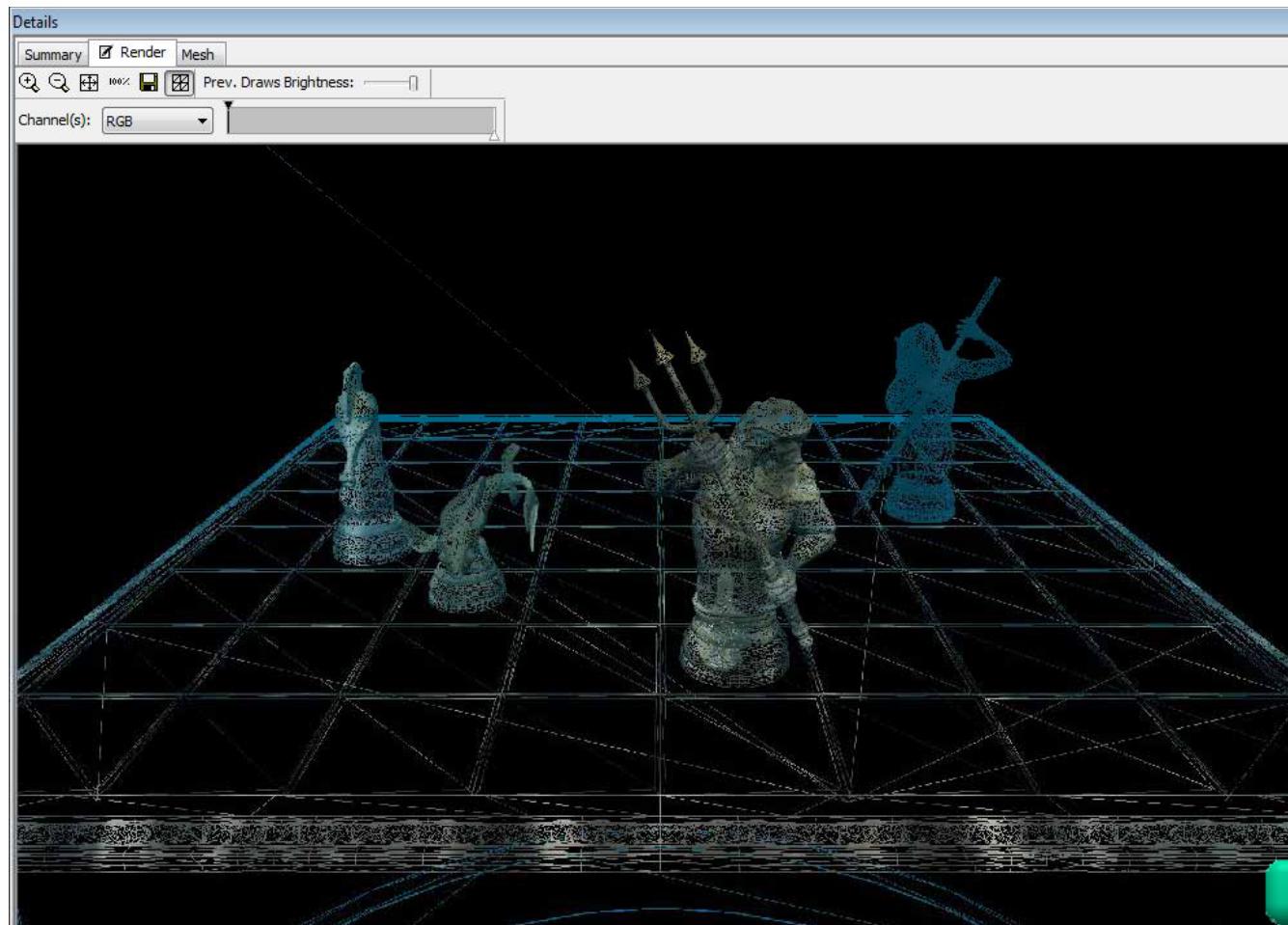




GameDevelopers
Conference 08

Missing Objects

- ⌚ Check wireframe geometry of scene



PIX for Windows

CMP
United Business Media

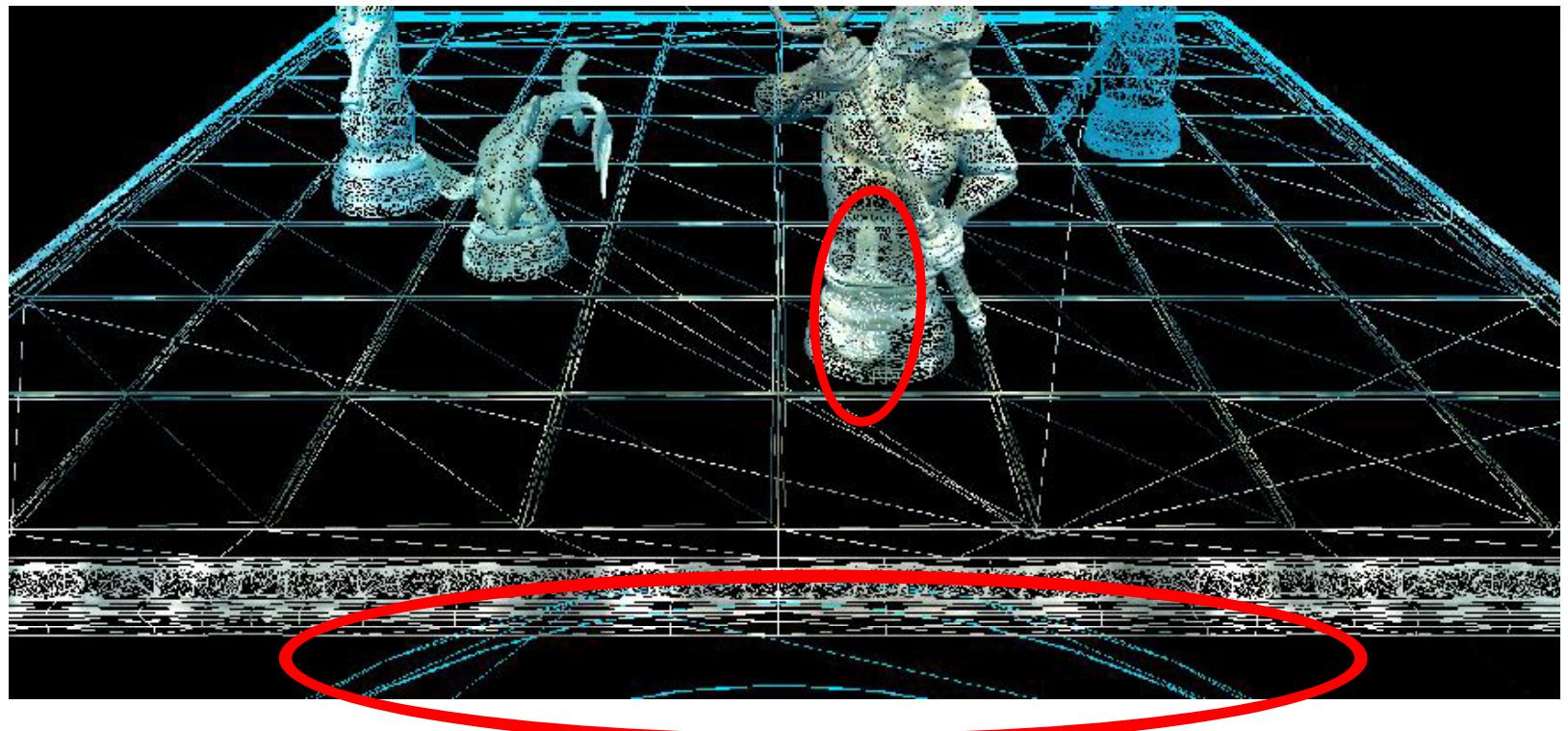
WWW.GDCONF.COM



GameDevelopers
Conference 08

Missing Objects

- ⌚ Suspicious artifacts present





GameDevelopers
Conference 08

Missing Objects

Incorrect vertex shader input

The screenshot shows the PIX for Windows interface. At the top, there are tabs for 'Summary', 'Render' (which is checked), and 'Mesh'. Below these are three render windows: 'Pre-Vertex Shader' showing a blue silhouette of a character, 'Post-Vertex Shader' showing the same blue silhouette but with some artifacts, and 'Viewport' showing a single small blue dot. Below the render windows is a table with two tabs: 'PreVS' and 'PostVS' (which is selected). The table has columns for VTX, IDX, Position, Normal, and TexCoord0. The data is as follows:

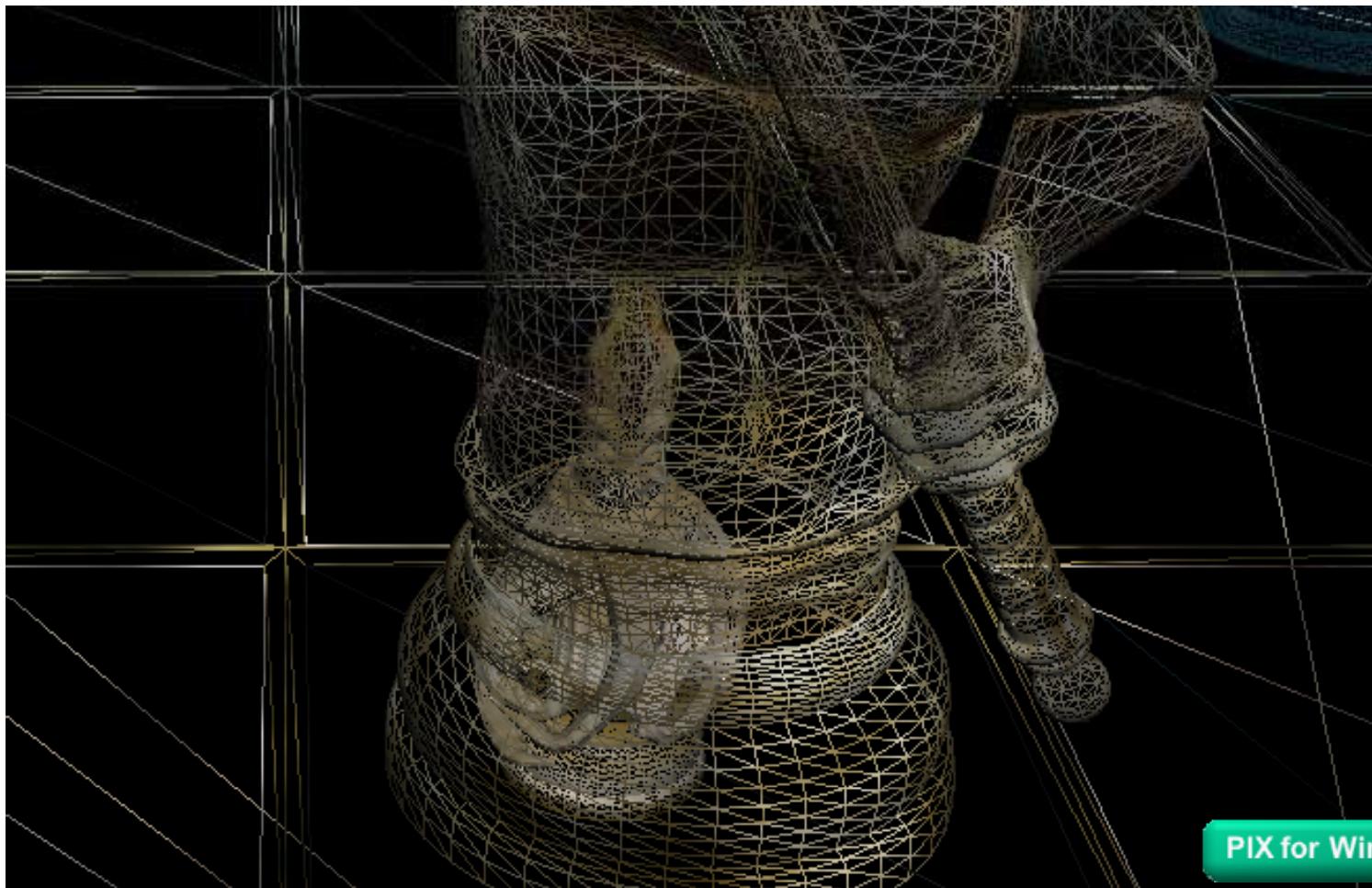
VTX	IDX	Position			Normal			TexCoord0	
0	0	0.000	0.050	0.000	1.000	0.015	-1.000	0.003	0.099 0.797
1	1	0.009	0.051	0.000	1.000	-0.008	-0.854	-0.521	-0.526 1.168
2	2	0.006	0.050	0.001	1.000	-0.034	-0.601	-0.799	-0.418 1.096
3	1	0.009	0.051	0.000	1.000	-0.008	-0.854	-0.521	-0.526 1.168

A green button labeled 'PIX for Windows' is located at the bottom right of the table.



Missing Objects

- ⌚ Yields unexpected output





Missing Objects

- Incorrect input & fogged out

GameDevelopers
Conference 08



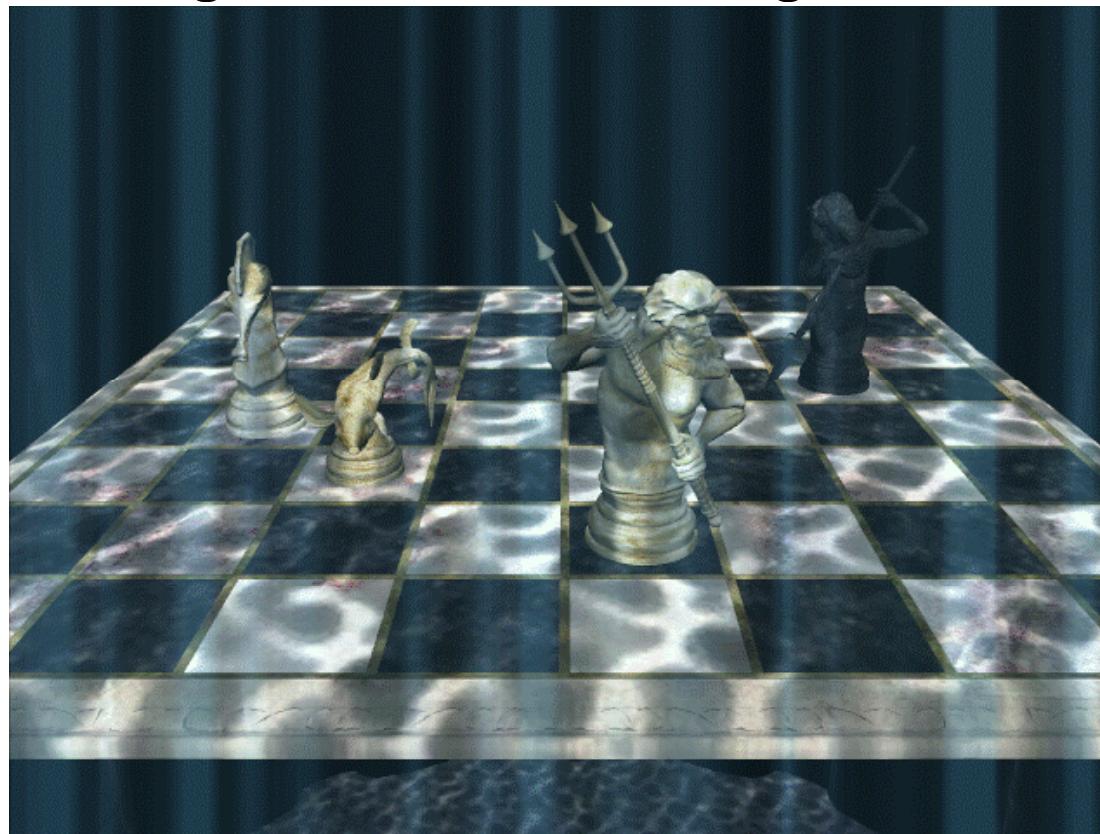
PreVS PostVS

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
	1	5159	-13.357	-34.861	9.552	9.651	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.090	0.006	0.422
	2	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.218	0.006	0.578
P1	3	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.218	0.006	0.578
	4	5159	-13.357	-34.861	9.552	9.651	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.090	0.006	0.422
	5	5161	-12.049	-29.956	14.632	14.731	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.336	0.054	0.727
P2	6	5160	-13.357	-32.347	12.156	12.255	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.218	0.006	0.578
	7	5161	-12.049	-29.956	14.632	14.731	D3DCOLOR_ARGB(0x00,0x00,0x00,0x00)		-2.336	0.054	0.727



Missing Objects

- ⌚ 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



Flickering



WWW.GDCONF.COM



Flickering

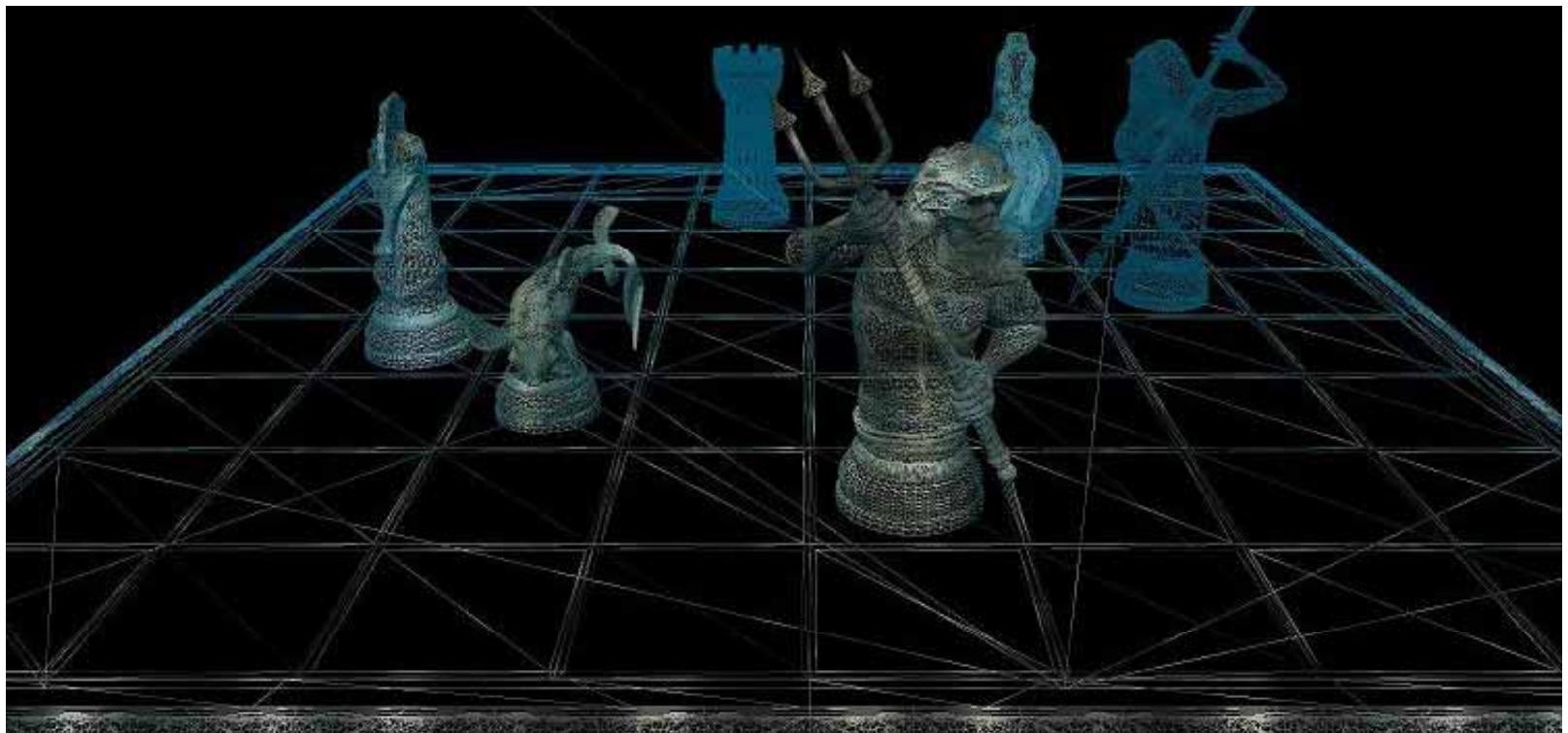
Scenario:

- Ⓐ 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

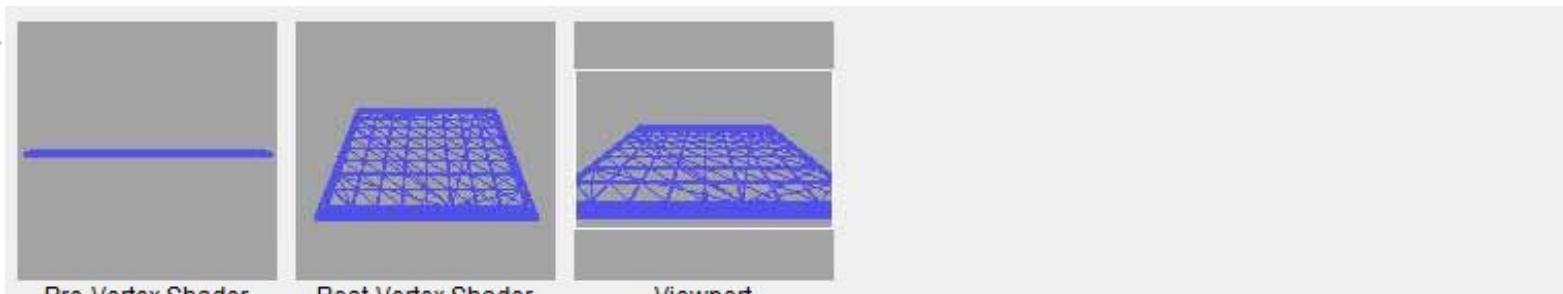
- ⌚ Examine wireframe for obvious z-fighting





Flickering

- Examine mesh view for hidden artifacts



Pre-Vertex Shader Post-Vertex Shader Viewport

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	5	-3.647	0.745	5.810	5.910	D3DCOLOR_ARGB(0x00,0x33,0x33,0x33)		0.292	0.701	0.968
	2	6	-3.647	1.285	6.369	6.468	D3DCOLOR_ARGB(0x00,0x33,0x33,0x33)		0.195	0.645	0.968
P1	3	7	-3.647	0.206	5.252	5.351	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)		0.387	0.756	0.968
	4	8	-3.647	1.285	6.369	6.468	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)		0.195	0.645	0.968
	5	9	-3.647	-0.872	4.135	4.235	D3DCOLOR_ARGB(0xa3,0xd6,0xd6,0xd6)		0.566	0.866	0.968
P2	6	10	-3.647	-0.872	4.135	4.235	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)		0.566	0.866	0.968
	7	11	-3.647	4.005	6.000	6.100	D3DCOLOR_ARGB(0xb9,0xec,0xec,0xec)		0.405	0.845	0.968



Flickering

- ⌚ Hidden mesh subset uncovered

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		



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:

- ➊ Culling & Render Order
- ➋ Buffer Sizes
- ➌ Ineffective Code
- ➍ Inefficient Shaders
- ➎ Batch Sizes



GameDevelopers
Conference 08

Culling & Render Order

- Look at the overdraw in the tool



PerfHUD

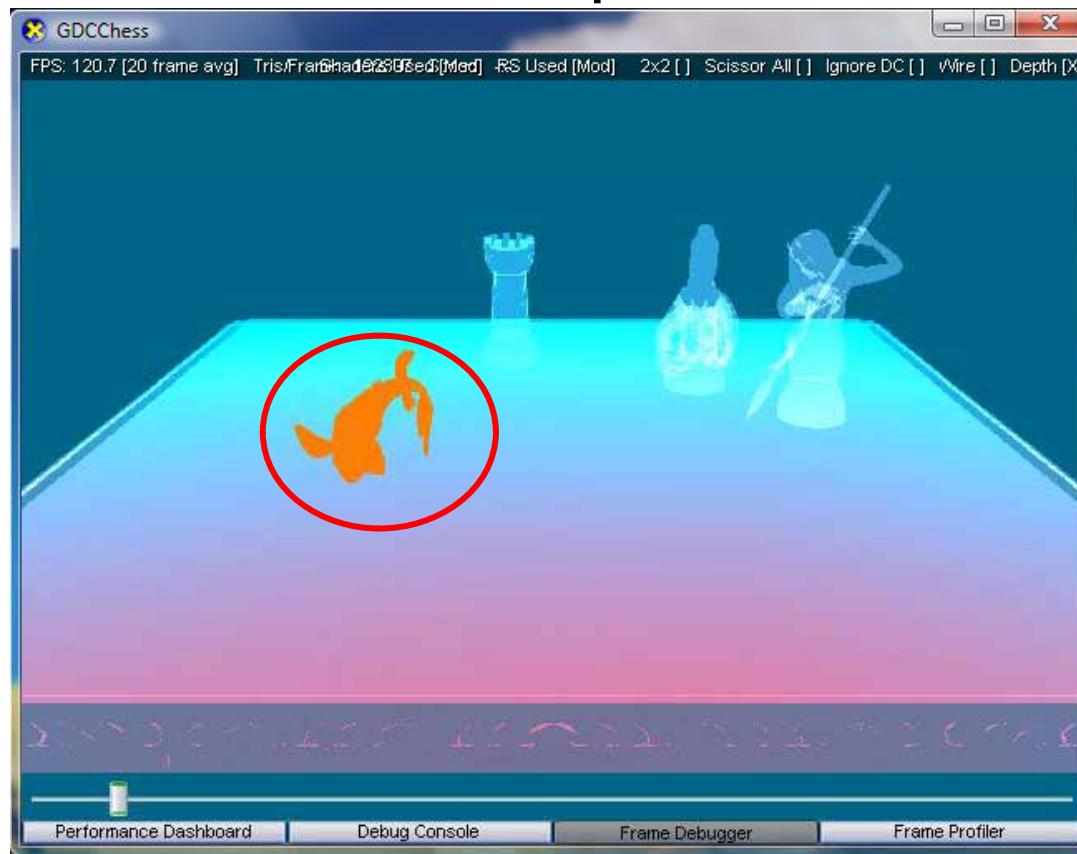


GameDevelopers
Conference

08

Culling & Render Order

- ⌚ Scroll through the draw calls to see how the frame is composed



PerfHUD

WWW.GDCONF.COM



Culling & Render Order

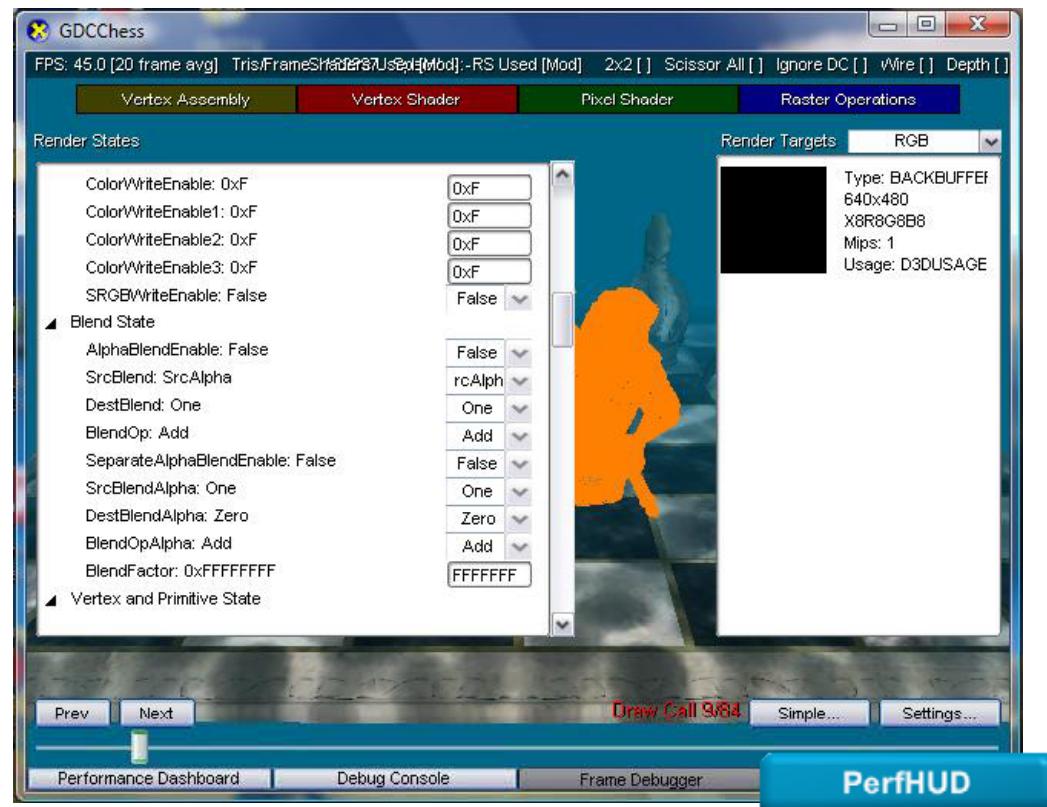
- ⌚ Notice how the draws are just stacking and nothing is culled
- ⌚ Are objects being rendered multiple times?





Culling & Render Order

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



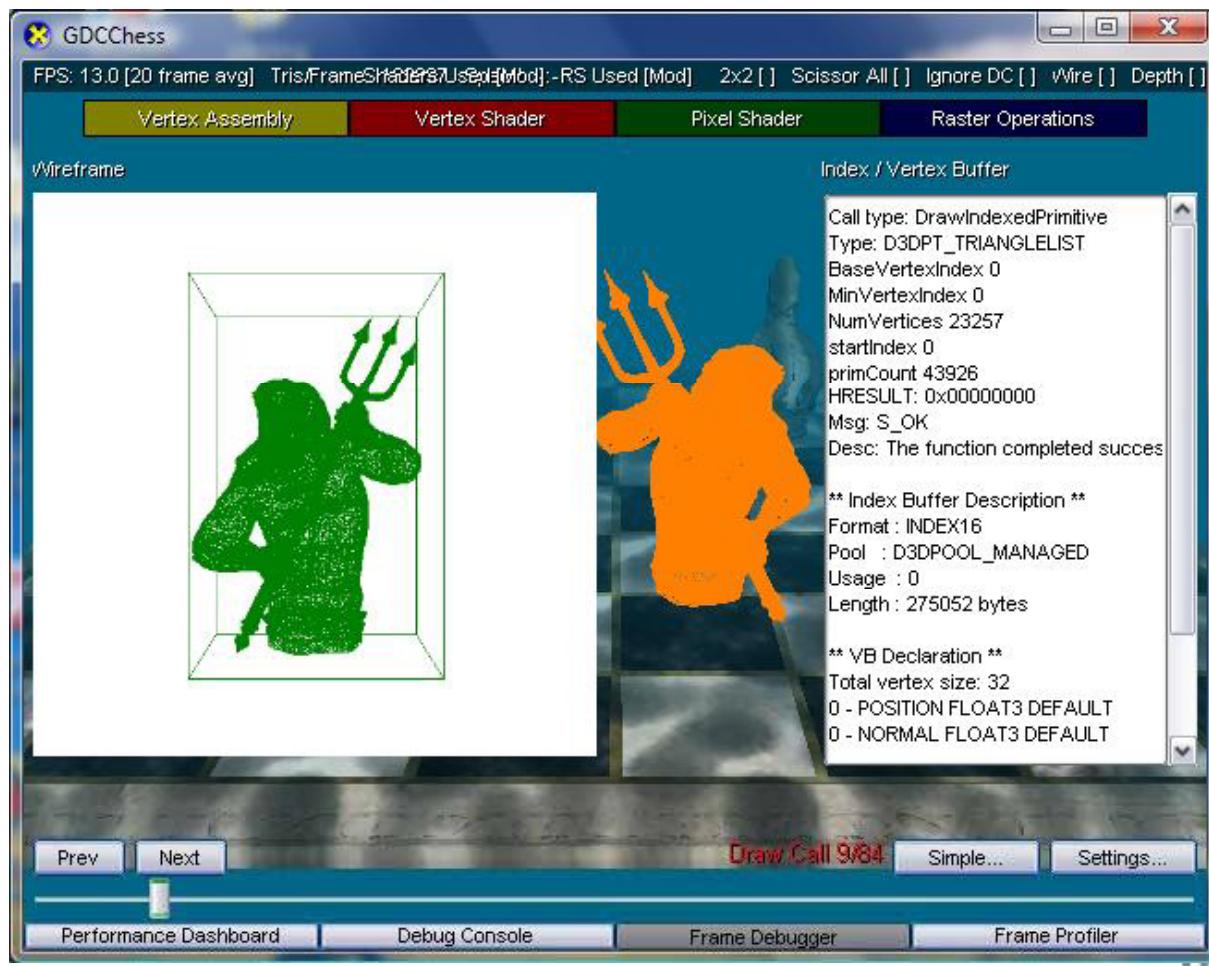


GameDevelopers
Conference

08

Culling & Render Order

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





Culling & Render Order

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



Culling & Render Order

Guidelines:

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



Buffer Sizes

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



Game Developers
Conference 08

Buffer Sizes

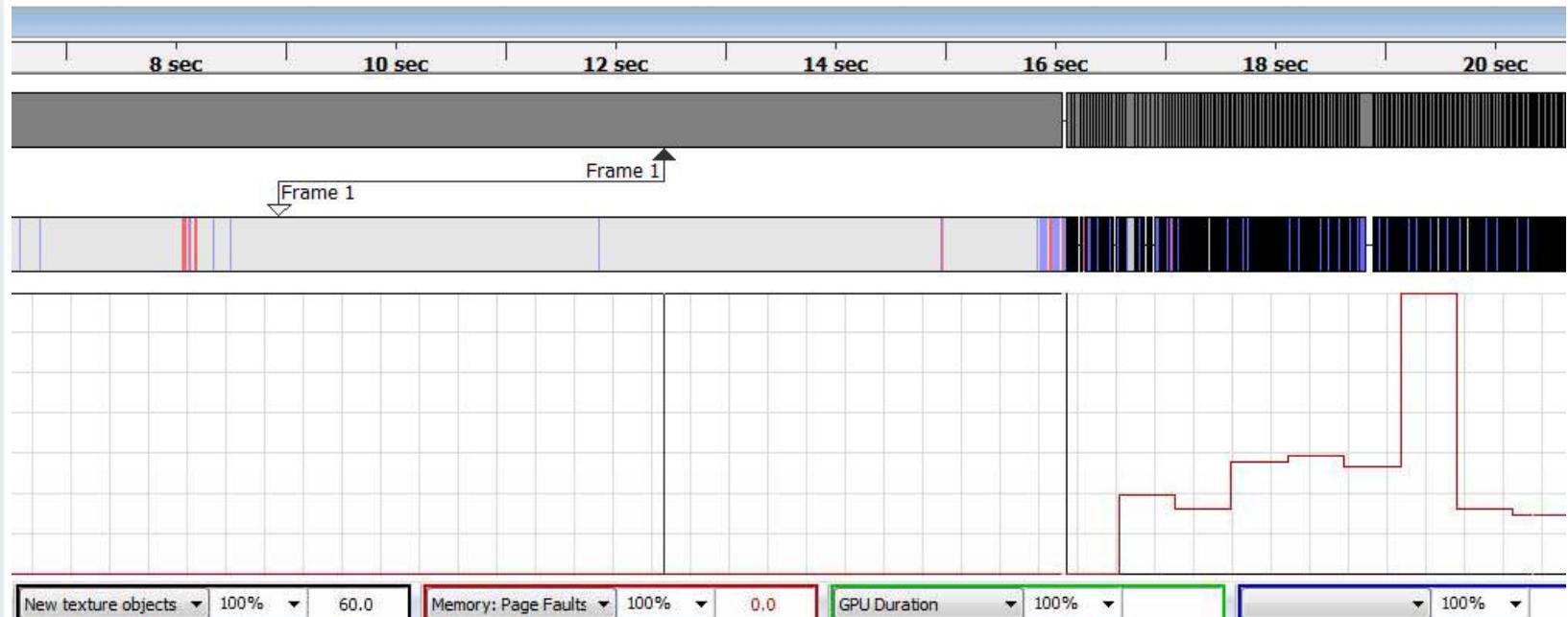
⌚ There could be lots of swapping occurring





Buffer Sizes

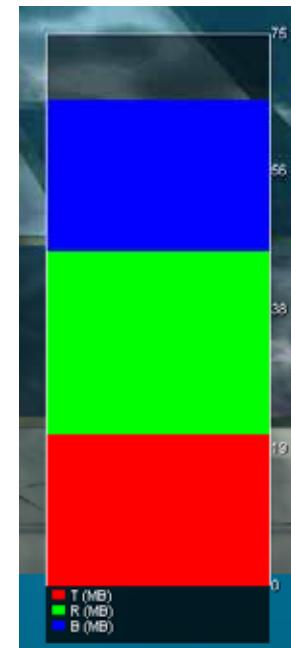
- ⌚ Look at the perfmon counter for memory page faults
is it too high?





Buffer Sizes

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



PerfHUD

WWW.GDCONF.COM



GameDevelopers
Conference

08

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
0x029B88680	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			
0x0AA0A0598	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



Buffer Sizes

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





Buffer Sizes

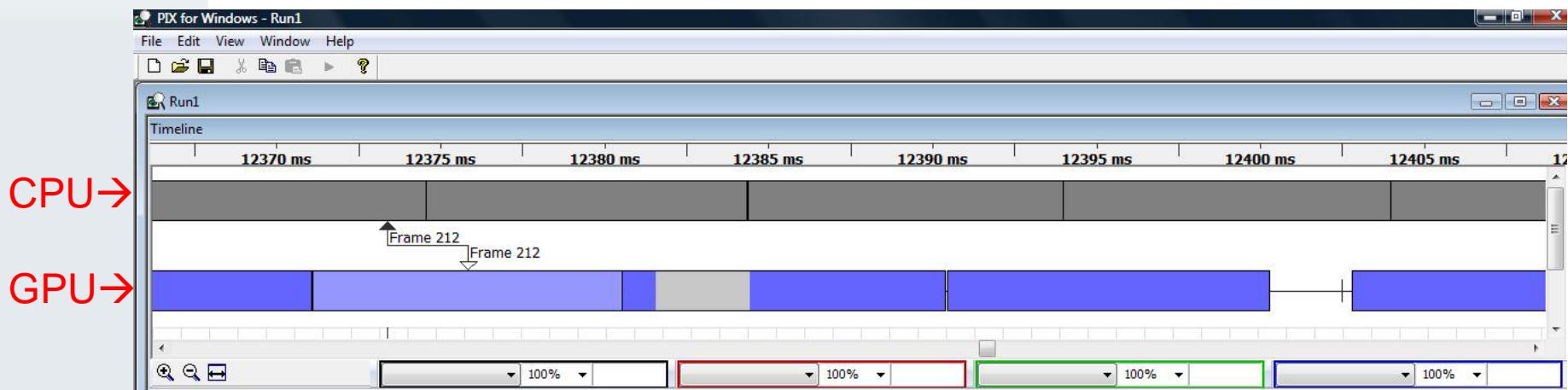
- ➊ Only use data where necessary
- ➋ Pack data buffers with a smaller vdecl
- ➌ Use LOD techniques to reduce the amount of data needed

- ➍ Use a paging algorithm for loading data
- ➎ Reuse Render targets when possible



Inefficient Code

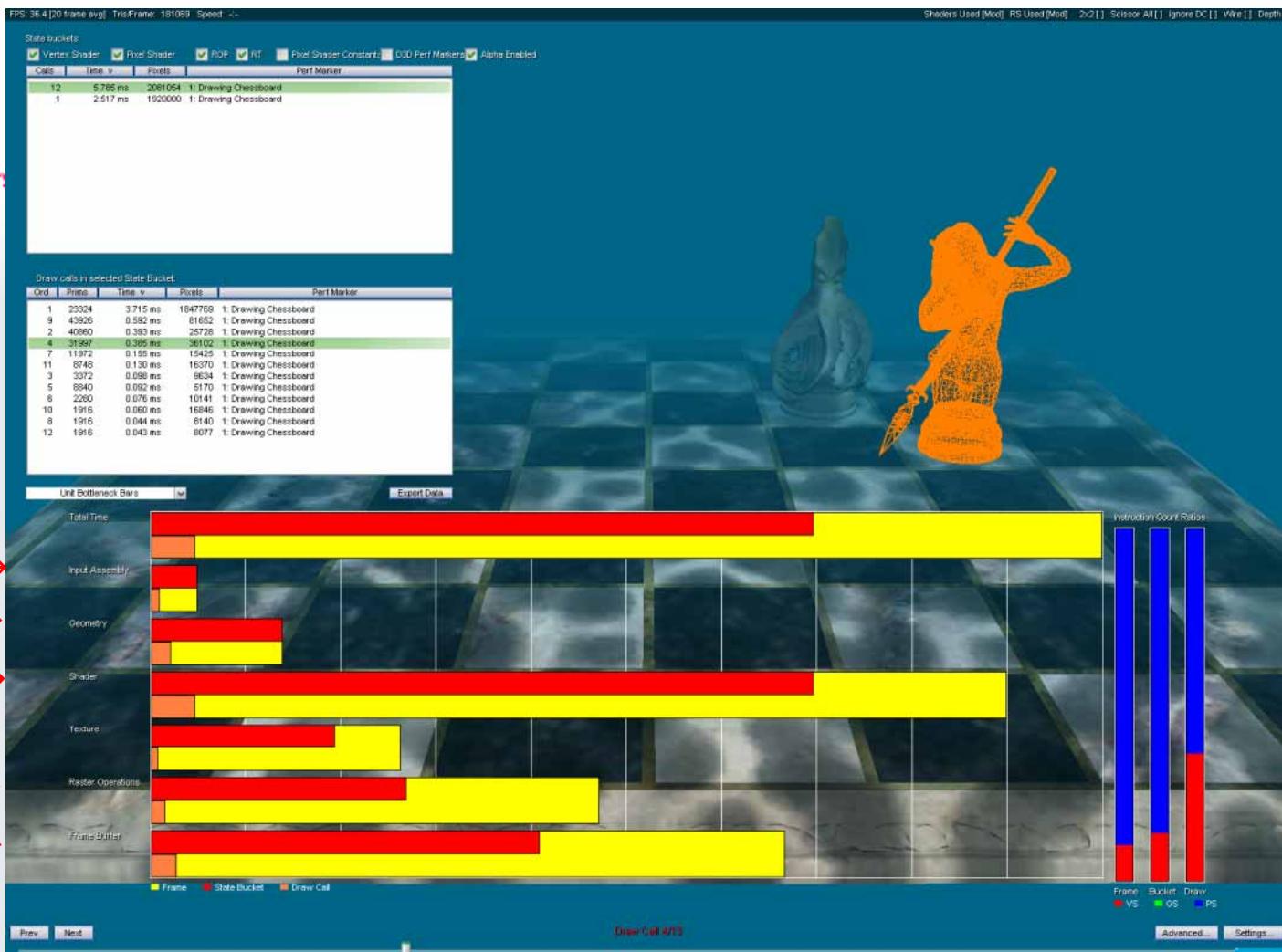
- ⌚ Are you sure you are GPU bound?
- ⌚ Look at the timing in PIX, PerfHUD





Game Developers
Conference 08

Inefficient Code



Total time →

Input Assembly →

Geometry →

Shader →

Texture →

Raster Ops →

Frame Buffer →

PerfHUD

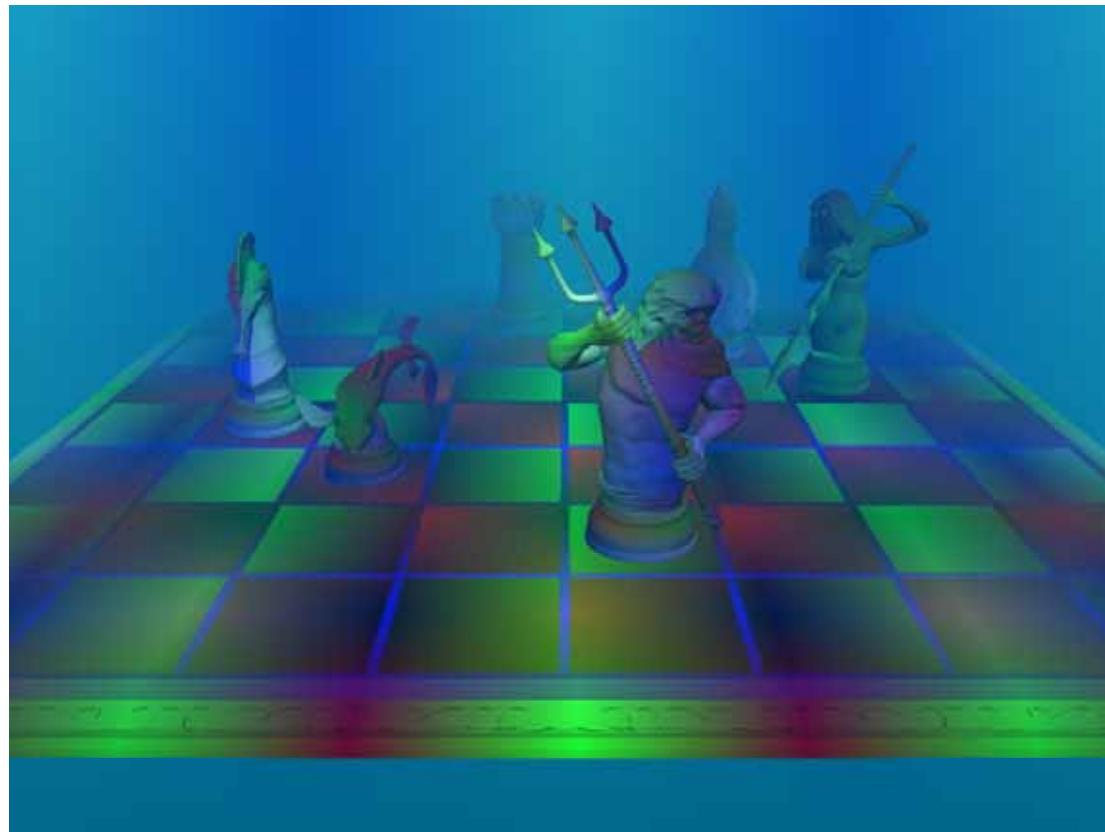
WWW.GDCONF.COM



Inefficient Code

Adjust

render size, texture sizes, cull objects



PerfHUD



Inefficient Code

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

Events					
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_CW)	19801493504			
928	+<0x02991B20> ID3DXEffect::SetTexture(0x005F5EF0, 0x090CBA70)	19801499648			
1010	User Event: Drawing Chessboard	19801667584		90865664	
1011	<0x02901B20> ID3DXEffect::SetMatrix(0x005F5F88, 0x0012F700)	19801675776			



GameDevelopers
Conference

08

Inefficient Shaders

Use a tool to analyze your shader

The screenshot shows the FX Composer interface for analyzing shaders. The main window displays a table of performance metrics for various NVIDIA GPUs running different shader versions. The table includes columns for Registers (Regs.) and Cycles, as well as MPix/s performance. Most entries show 'n/a' for registers and cycles, indicating inefficiencies. The GPU list on the left includes GeForceFX 5800 Ultra through 6200, GeForce 6800 Ultra through 7800 GTX, and GeForce 6600 GT.

Techniques:	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	⚠️ n/a																																																																																																																																																																																																																								
TexturedVS																																																																																																																																																																																																																									
p0	⚠️ n/a																																																																																																																																																																																																																								
UntexturedPS																																																																																																																																																																																																																									
p0	⚠️ n/a																																																																																																																																																																																																																								
TexturedPS																																																																																																																																																																																																																									
p0	⚠️ n/a																																																																																																																																																																																																																								
Show: Fragment Shader																																																																																																																																																																																																																									
Drivers:	GPUs:																																																																																																																																																																																																																								
163.20	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 6600 GT (NV43-GT) GeForce 6200 (NV44)	<table border="1"><thead><tr><th></th><th>1</th><th>1</th><th>1</th><th>1</th><th>1</th><th>1</th><th>1</th><th>1</th><th>1</th><th>2105</th><th>2105</th><th>2105</th><th>2105</th><th>2105</th></tr><tr><th></th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th><th>0%</th></tr></thead><tbody><tr><td>NV30 (GeForceFX 5800 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV35 (GeForceFX 5900 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV34 (GeForceFX 5200 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV31 (GeForceFX 5600 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV36 (GeForceFX 5700 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV38 (GeForceFX 5950 Ultra)</td><td>⚠️</td><td colspan="4">n/a</td><td colspan="4">n/a</td><td colspan="4">n/a</td></tr><tr><td>NV40 (GeForce 6800 Ultra)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>2105</td><td>2105</td><td>2105</td><td>2105</td><td>2105</td></tr><tr><td>NV40-GT (GeForce 6800 GT)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1305</td><td>1305</td><td>1305</td><td>1305</td><td>1305</td></tr><tr><td>NV40-12 (GeForce 6800)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>3900</td><td>3900</td><td>3900</td><td>3900</td><td>3900</td></tr><tr><td>NV43-GT (GeForce 6600 GT)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>4000</td><td>4000</td><td>4000</td><td>4000</td><td>4000</td></tr><tr><td>NV44 (GeForce 6200)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1400</td><td>1400</td><td>1400</td><td>1400</td><td>1400</td></tr><tr><td>G70-GT (GeForce 7800 GTX)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1730</td><td>1730</td><td>1730</td><td>1730</td><td>1730</td></tr></tbody></table>													1	1	1	1	1	1	1	1	1	2105	2105	2105	2105	2105		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	NV30 (GeForceFX 5800 Ultra)	⚠️	n/a				n/a				n/a				NV35 (GeForceFX 5900 Ultra)	⚠️	n/a				n/a				n/a				NV34 (GeForceFX 5200 Ultra)	⚠️	n/a				n/a				n/a				NV31 (GeForceFX 5600 Ultra)	⚠️	n/a				n/a				n/a				NV36 (GeForceFX 5700 Ultra)	⚠️	n/a				n/a				n/a				NV38 (GeForceFX 5950 Ultra)	⚠️	n/a				n/a				n/a				NV40 (GeForce 6800 Ultra)	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	1305	1305	1305	1305	1305	NV40-12 (GeForce 6800)	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	4000	4000	4000	4000	4000	NV44 (GeForce 6200)	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	1730	1730	1730	1730	1730
	1	1	1	1	1	1	1	1	1	2105	2105	2105	2105	2105																																																																																																																																																																																																											
	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%																																																																																																																																																																																																											
NV30 (GeForceFX 5800 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV35 (GeForceFX 5900 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV34 (GeForceFX 5200 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV31 (GeForceFX 5600 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV36 (GeForceFX 5700 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV38 (GeForceFX 5950 Ultra)	⚠️	n/a				n/a				n/a																																																																																																																																																																																																															
NV40 (GeForce 6800 Ultra)	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	1305	1305	1305	1305	1305																																																																																																																																																																																																											
NV40-12 (GeForce 6800)	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	4000	4000	4000	4000	4000																																																																																																																																																																																																											
NV44 (GeForce 6200)	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	1730	1730	1730	1730	1730																																																																																																																																																																																																											

FX Composer

WWW.GDCONF.COM



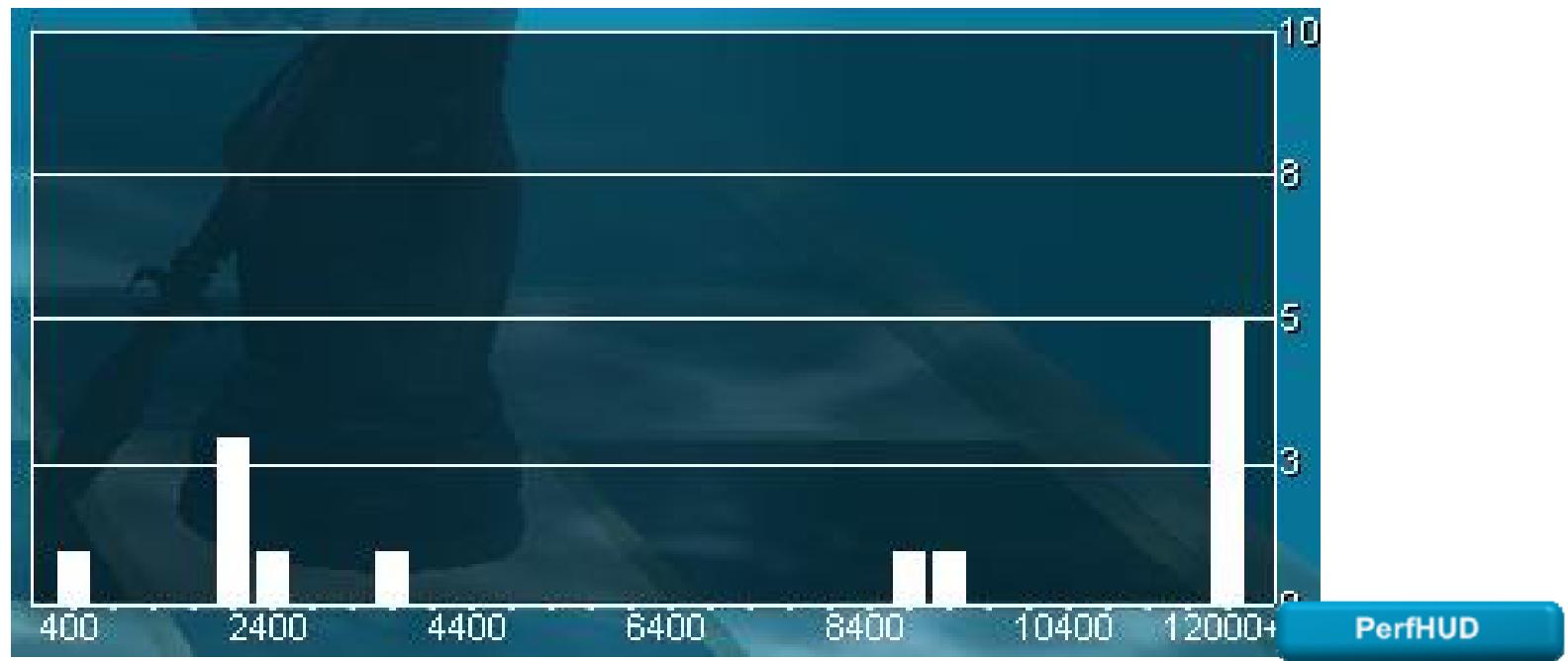
Inefficient Shaders

- ⌚ Are you sure it is the shader?
- ⌚ Swap the shader for a simpler shader, did that make a difference?
- ⌚ 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

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



GameDevelopers
Conference 08

Live Demos

- ④ Microsoft - PIX for Windows
- ④ NVIDIA - PerfHUD



Game Developers
Conference 08

Q&A

Questions, Comments, Concerns?

CMP
United Business Media

WWW.GDCONF.COM



Resources

- Ⓐ Tools shown today can be downloaded at:
 - Ⓐ AMD
 - Ⓐ <http://developer.amd.com>
 - Ⓐ Microsoft
 - Ⓐ <http://msdn.microsoft.com/directx>
 - Ⓐ NVIDIA
 - Ⓐ <http://developer.NVIDIA.com/>
- Ⓐ 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

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