



---

# Profiling and Debugging Your Game with PIX on Xbox 360

Bruce Dawson

[bdawson@microsoft.com](mailto:bdawson@microsoft.com)

Microsoft, Game Technology Group, Reading

# Tools Matter

- **Greater developer efficiency**
- **Better results: faster, better looking**
- **Educational**
- **Great tools make the job more fun**

**Performance Investigator for Xbox**

**Performance Investigator for Xbox, Windows, and Xbox 360**

**Performance and Debugging Investigator for Xbox, Windows, and Xbox 360**

- **Agenda:**
  - Why is PIX a vital tool?
  - What can it tell you?
  - Demos!

# Why PIX?

- **Optimizing**
  - Knowledge is power
  - Running tests is tedious
  - Analyzing the results is hard

# Why PIX?

- **Optimizing**
  - Knowledge is power
  - Running tests is tedious
  - Analyzing the results is hard
- **Debugging**
- **Understanding the software and hardware**

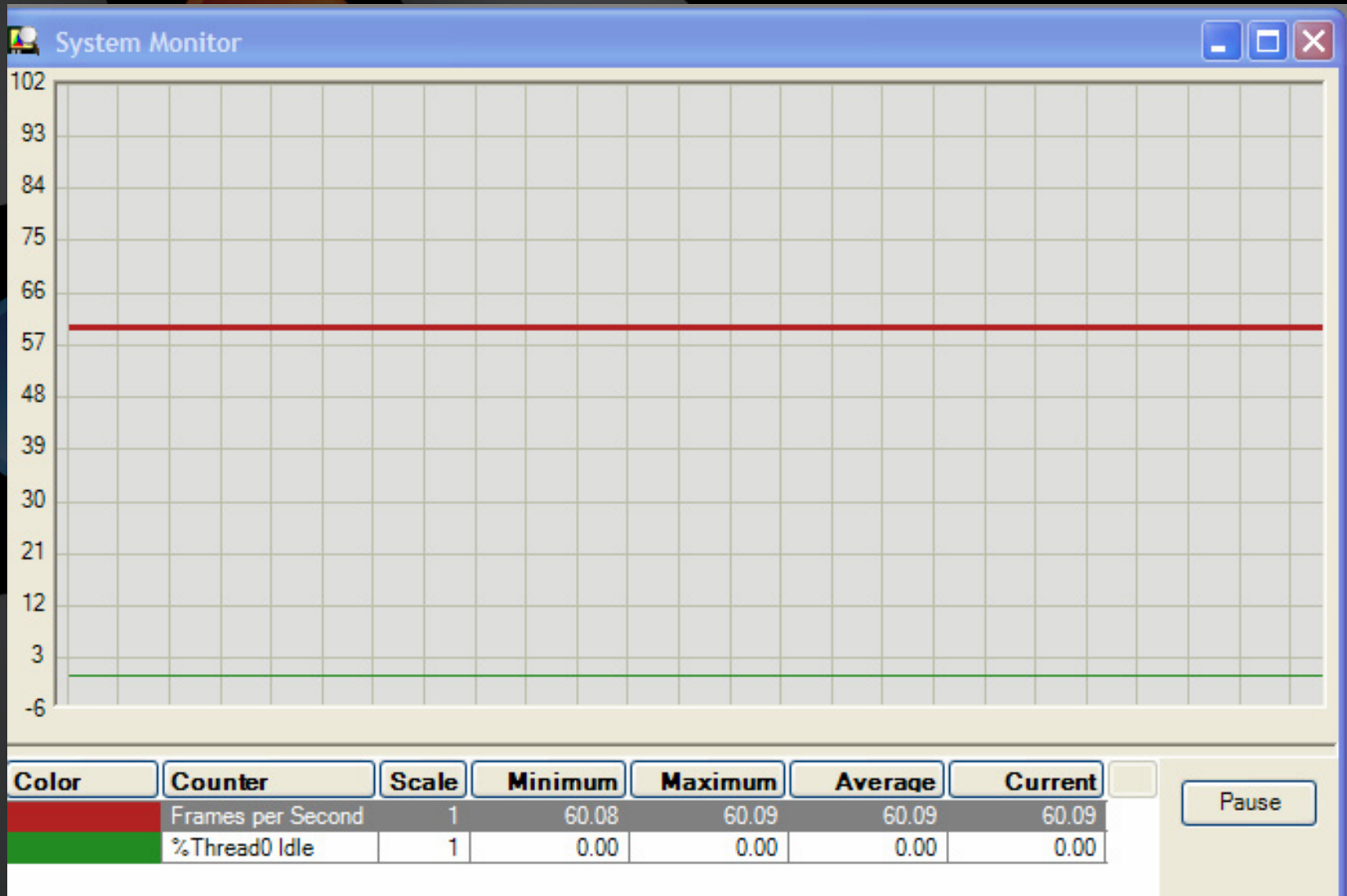
# Inside PIX for Xbox 360

- **Many profilers in one**
  - System Monitor
  - Timing Captures
  - GPU Captures
  - CPU Trace Captures
- **Timing and GPU captures use an instrumented version of D3D**
  - Only mildly intrusive
- **System Monitor and CPU Trace Captures work on full release builds**

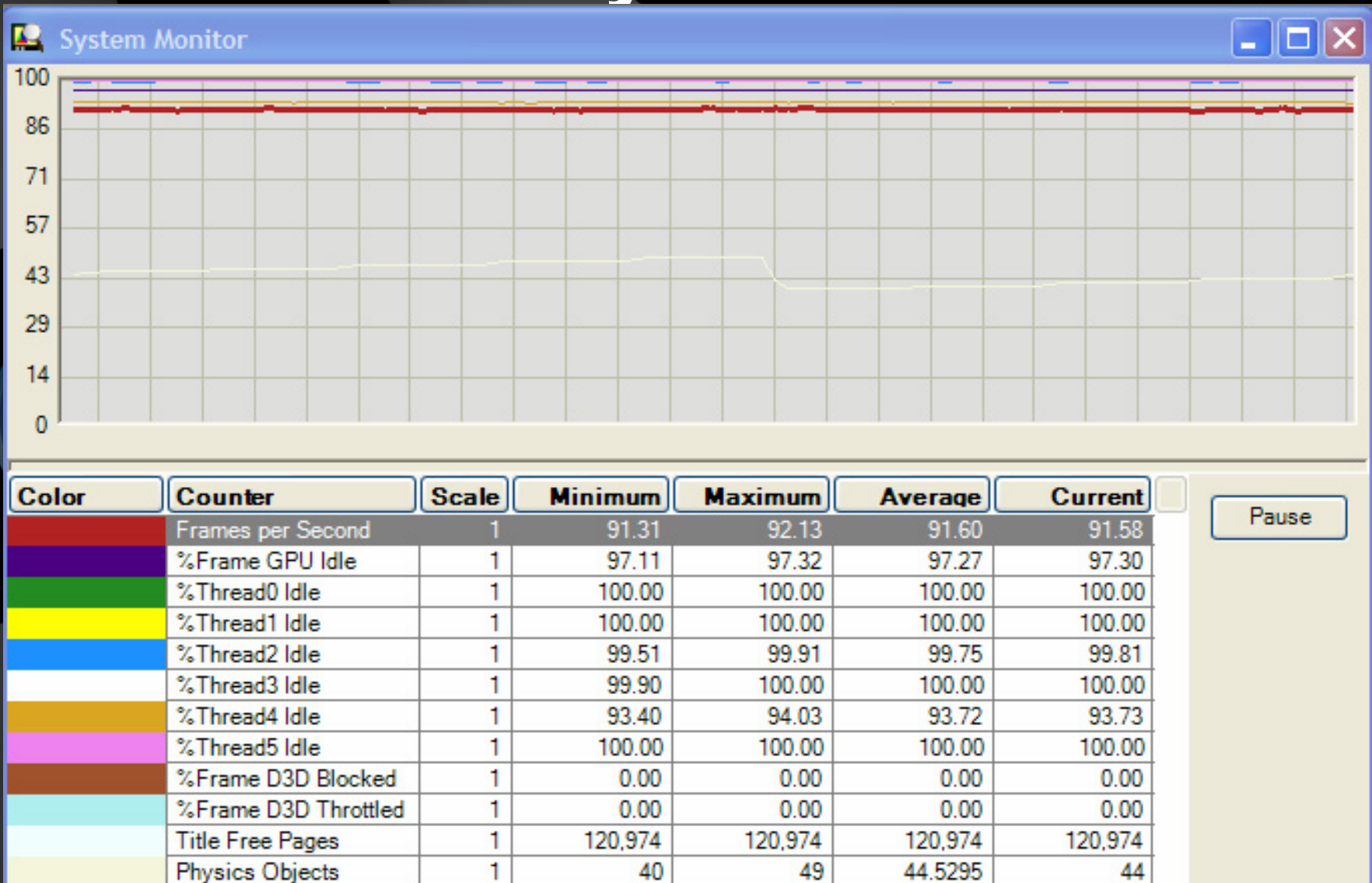
# Inside PIX for Windows

- **Many profilers in one**
  - Timing information
  - Full-frame capture
  - Direct3D call log
  - Direct3D call stream recording
- **Works on full release versions of your games**

# Default Xbox 360 System Monitor



# Tricked Out System Monitor





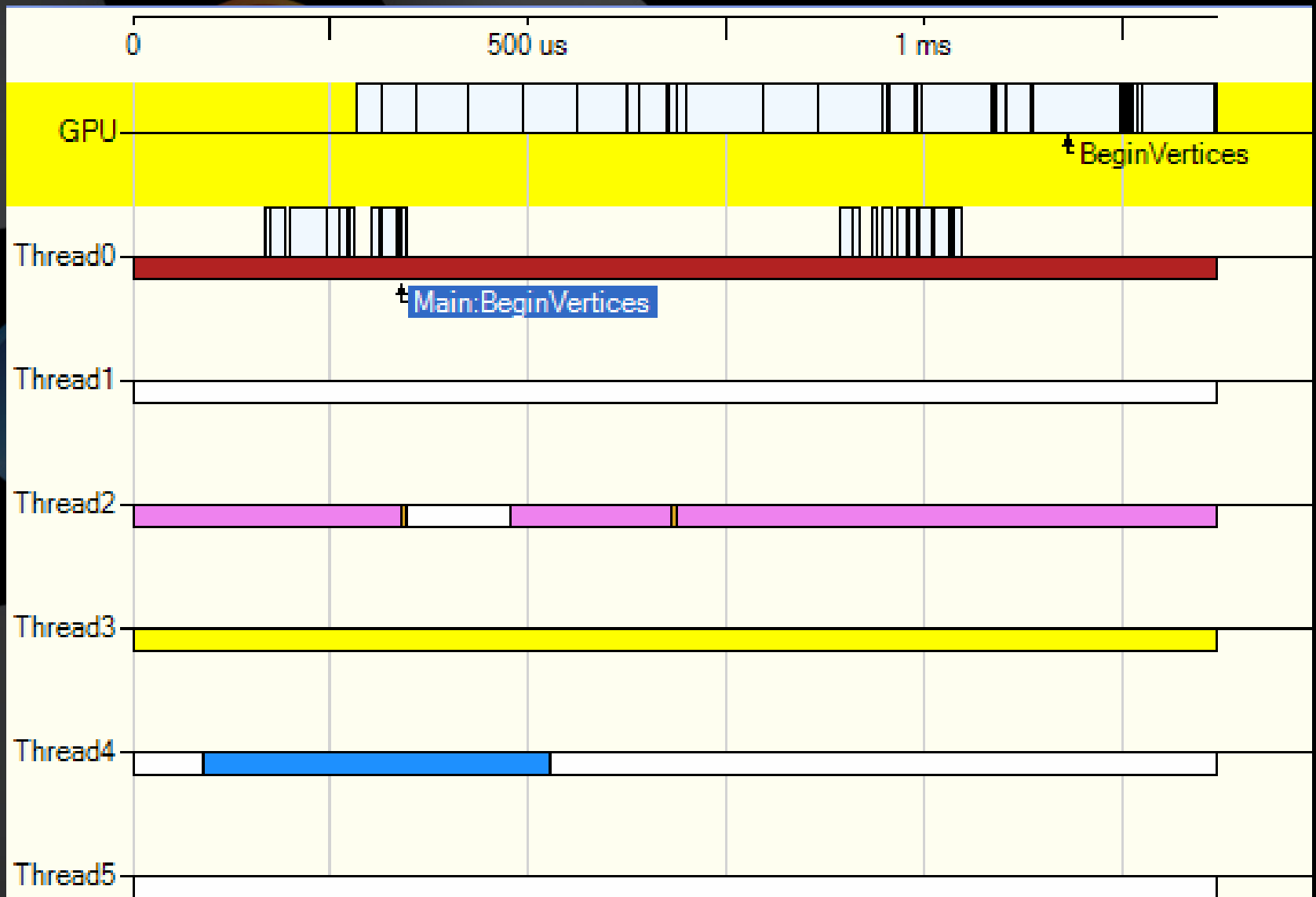
# Adding Custom Counters: Xbox 360

- Call PIXAddNamedCounter once per frame

```
PIXAddNamedCounter( m_nPhysObj, "Physics Objects" );
```

- Available in all build configurations, including release

# Timing Capture: Timelines



<+> Zoom In To Cursor, <-> Zoom Out, <Esc> - Reset

# Timing Capture: Events

## Events in OS Thread TitleThread0

	Item	CPU Duration (ns)	GPU Duration (ns)	CPU Begin (ns)	CPU End (ns)
+ Clear	1	9,120	267,792	61,360	70,480
... DrawIndexedPrimitive	3	18,500	138,008	75,220	93,720
... DrawIndexedPrimitive	4	1,560	2,698	95,960	97,520
+ DrawText: Dolphin	5	14,240	2,942	106,780	121,020
+ DrawText: 30.04 fps	7	8,680	2,242	122,140	130,820
+ DrawText: 00d00h00m05s	9	9,780	3,502	136,440	146,220
... SyncToVBlank	11	520	32,582,090	150,380	150,900
... Resolve	12	2,780	267,896	151,940	154,720
+ Swap	13	32,346,980	624	158,320	32,505,300
... Totals		32,443,940	33,267,790	61,360	32,505,300

# Adding Event Names: Xbox 360

- Wrap code to be measured in PIXBeginNamedEvent and PIXEndNamedEvent.

```
PIXBeginNamedEvent( 0xFFFFFFFF, "Draw Text: %s",  
                  message );
```

```
...
```

```
PIXEndNamedEvent();
```

- Requires instrumented version of D3D

# Adding Event Names: Windows

- Wrap code to be measured in PIXBeginNamedEvent and PIXEndNamedEvent.

```
D3DPERF_BeginEvent( 0xFFFFFFFF, "Draw Hero" );  
...  
D3DPERF_EndEvent();
```

- Requires instrumented version of D3D

# GPU Capture: Shader View

GPU Capture GPU View

450 us | 550 us | 650 us | 750 us | 850 us | 950 us

Draw mesh R\_eyeBall

Accounting for GPU only, this frame would take about 1.0 ms and run at 965.7 frames per second.

Name	Item	GPU Event	Vertices	Vertex Vectors	Quads	Pixel Vectors
Clear	0	G0	3	1	25,600	1,600
BeginTiling: 0	1	G1 - G11	67,688	510	40,581	971
BeginTiling: 1	17	G12 - G22	44,672	358	42,927	1,113
BeginTiling: 2	33					
Draw mesh body	34	G23 - G26	19,704	152	4,699	298
Draw mesh Head	39	G27	0	0	0	0
Draw mesh L_eyeBall	41	G28	0	0	0	0
Draw mesh R_eyeBall	43					
DrawIndexedPrimitive	44	G29	0	0	0	0
DrawPrimitiveUP	45	G30	164	3	627	40
Resolve	46	G31	0	0	0	0
Resolve	47	G32	0	0	0	0
Resolve	48	G33	3	1	25,600	0
BeginVertices	49	G34	3	1	76,800	4,800
DrawText: SkinnedCharacter	50	G35	64	1	2,241	141
DrawText: 514.93 fps	52	G36	40	1	956	60
DrawText: 433.280 ms	54	G37	40	1	858	54
DrawText: 0.1118 ms Update	56	G38	64	1	1,197	75
DrawText: 0.0703 ms Render	58	G39	64	1	1,229	77
DrawText: Skinning Method 1	60	G40	68	2	1,624	102
DrawText: Implementation: Verte	62	G41	116	2	2,889	181
DrawText: Bone Palette: Shadowe	64	G42	156	3	3,917	245
Resolve	66	G43	3	1	76,800	0
Totals	67	G43	132,852	1,039	308,545	9,757

Framebuffer | Textures | Mesh | Debugger | Warnings | Analysis | D3D State | GPU State

Command Disassembly | Shaders | D3D Constants | GPU Constants | Overview

Show microcode

### Vertex Shader

Cycles/64 vertex vector: ALU 17.3, vertex 32, texture 0, sequencer 12  
 5 GPRs, 31 threads, Performance (if enough threads): ~32 cycles per vector

Source File: "c:\devtools\xedk\Source\Samples\Media\Shaders\SkinnedCharacter.hlsl"  
 Entrypoint: TransformVS  
 Line: 329

### Pixel Shader

Cycles/64 pixel vector: ALU 41.3, vertex 0, texture 24, sequencer 16, interpolator 32  
 8 GPRs, 24 threads, Performance (if enough threads): ~32 cycles per vector

Source File: "c:\devtools\xedk\Source\Samples\Media\Shaders\SkinnedCharacter.hlsl"  
 Entrypoint: NormalMapPS  
 Line: 382

//-----

# PIX for Windows Demo

- A live demo occurs at this point. This demo will show many types of capture and analysis and will be narrated in a witty manner.
- Please wait...

# Shader Debugging

```
Disassembly

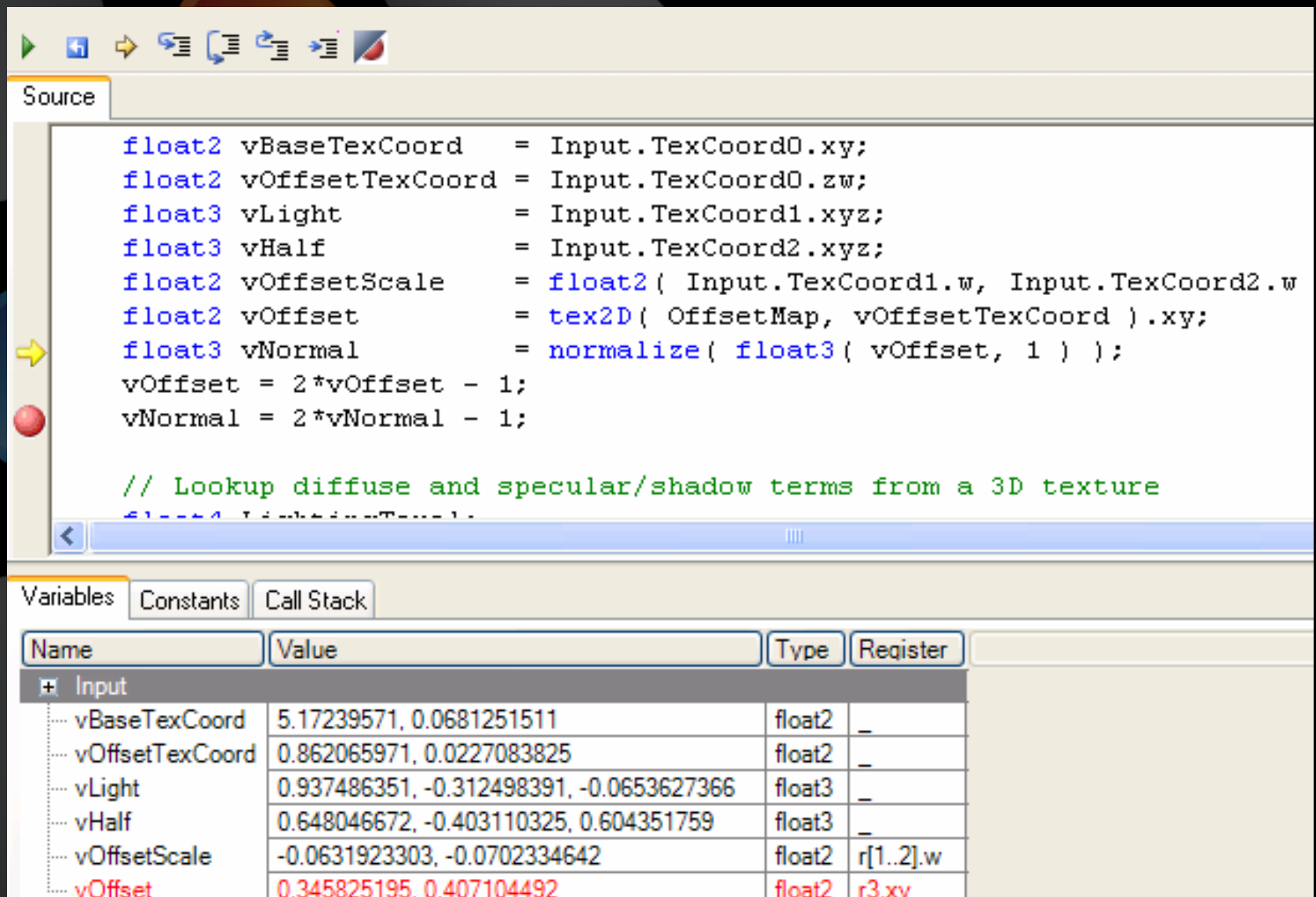
ps_1_1
def c0, 0.333339989, 0.333339989, 0.333339989, 0
def c1, 0.666660011, 0.666660011, 0.666660011, 1
tex t0
tex t1
tex t2
tex t3
→ mul_sat r0.w, t1.w, t0.w
  lrp_d2 r0.xyz, r0.w, t1, t0
+ mul_sat r1.w, t0.w, t2.w
  lrp_d2 r1.xyz, r1.w, t2, t0
  add r0.xyz, r0, r1
+ mul_sat r1.w, t0.w, t3.w
  lrp r1.xyz, r1.w, t3, t0
  mul r1.xyz, r1, c0
  mad r0.xyz, r0, c1, r1
+ mov r0.w, c1.w

// approximately 11 instruction slots used (4 texture, 7 arithmetic)
```

Name	Value	Type
t0	(0.784, 0.545, 0.416, 0.749)	float4
t1	(0.537, 0.463, 0.447, 0.749)	float4
t2	(0.732, 0.505, 0.401, 0.749)	float4
t3	(0.575, 0.386, 0.320, 0.749)	float4



# HLSL Shader Debugging!



The screenshot displays a shader debugging interface. The top section shows the source code of an HLSL shader. A yellow arrow on the left points to the line `float3 vNormal = normalize( float3( vOffset, 1 ) );`. Below the code is a scroll bar. The bottom section shows a 'Variables' window with a table of current values.

```
float2 vBaseTexCoord = Input.TexCoord0.xy;
float2 vOffsetTexCoord = Input.TexCoord0.zw;
float3 vLight = Input.TexCoord1.xyz;
float3 vHalf = Input.TexCoord2.xyz;
float2 vOffsetScale = float2( Input.TexCoord1.w, Input.TexCoord2.w );
float2 vOffset = tex2D( OffsetMap, vOffsetTexCoord ).xy;
float3 vNormal = normalize( float3( vOffset, 1 ) );
vOffset = 2*vOffset - 1;
vNormal = 2*vNormal - 1;

// Lookup diffuse and specular/shadow terms from a 3D texture
float3 LightDir = ...
```

Name	Value	Type	Register
+	Input		
vBaseTexCoord	5.17239571, 0.0681251511	float2	_
vOffsetTexCoord	0.862065971, 0.0227083825	float2	_
vLight	0.937486351, -0.312498391, -0.0653627366	float3	_
vHalf	0.648046672, -0.403110325, 0.604351759	float3	_
vOffsetScale	-0.0631923303, -0.0702334642	float2	r[1..2].w
vOffset	0.345825195, 0.407104492	float2	r3.xy

# Shaders Tab: Xbox 360

- Pixel Shader
- These performance estimates are based upon a static analysis of the shader. Use the "Analysis" tab for a more accurate dynamic analysis.

Cycles/64 pixel vector: ALU 9, vertex  
0, texture 4, sequencer 8, interpolator  
32

# Analysis Tab: Xbox 360

## Bottleneck Summary

Simulation Time:	155370	*****
Shading:	151555	
ALU:	19244	
VS ALU:	4	
PS ALU:	19240	***
Control Flow:	86586	
VS CF:	6	
PS CF:	86580	*****
Texture Fetch:	57720	
Bilinear/Point:	57720	*****
...		
Vertex Fetch:	4	
Vertices:	4	
Vfetch_fulls:	2	
Shader Stalls:	64969	
Vertex Cache:	0	
Texture Cache:	64969	*****
...		
Rasterization and Fill:	124775	
Scan and Interpolate:	115440	
Quads:	115440	*****
Interpolation:	115440	*****
Tiles:	14620	**

# Analysis Tab: Xbox 360

## General

Measured Time:	296
Indices:	36
Primitives:	12
Transformed Vertices:	30
Vertex Vectors:	1
Quads:	0
Tiles:	6
Pixel Vectors:	0
Non-pipelined Busy:	1033
Primitive Type:	TRILIST
Ideal Fill:	2
Max VS Threads:	9
Max PS Threads:	30
VS Clauses:	2
PS Clauses:	1

Warning: low vertex thread count due to high GPR usage may hurt performance. Performance simulation may be inaccurate.

# Warnings Tab: Xbox 360

## Scene Warnings and Errors

Priority	Warnings and Errors
1	No rendering was done with a D3DFMT_D24FS8 floating point depth buffer with Hierarchical Z enabled and an inverted depth buffer.

## Event Warnings and Errors

Item	Priority	Warnings and Errors
366	2	Shader patch: the vertex shader's outputs had to be patched to match the pixel shader's inputs. The vertex shader's output: COLOR.
369	3	Call SetPixelShader(NULL) when doing depth/stencil-only rendering to double the fill-rate.
378	2	Shader patch: unused vertex shader output: COLOR
378	2	Shader patch: unused vertex shader output: TEXCOORD
378	2	Shader patch: the vertex shader's outputs had to be patched to match the pixel shader's inputs. The vertex shader's output: COLOR.
456	2	Shader patch: unused vertex shader output: TEXCOORD
456	2	Shader patch: unused vertex shader output: COLOR
456	2	Shader patch: the vertex shader's outputs had to be patched to match the pixel shader's inputs. The vertex shader's output: COLOR.
459	3	Call SetPixelShader(NULL) when doing depth/stencil-only rendering to double the fill-rate.
464	2	Use a D3DFMT_D24FS8 floating point depth buffer here with Hierarchical Z enabled and an inverted depth buffer.
592	2	Shader patch: the vertex shader's outputs had to be patched to match the pixel shader's inputs. The vertex shader's output: COLOR.
595	2	Use a D3DFMT_D24FS8 floating point depth buffer here with Hierarchical Z enabled and an inverted depth buffer.
1166	2	Use a D3DFMT_D24FS8 floating point depth buffer here with Hierarchical Z enabled and an inverted depth buffer.
1172	2	Use a D3DFMT_D24FS8 floating point depth buffer here with Hierarchical Z enabled and an inverted depth buffer.

# Summary

- **Tools matter**
- **Learn the available tools**
- **Modify your code and configurations to use PIX**
  - It's easy

# Resources

- **PIX for Windows**

<http://msdn.microsoft.com/DirectX/>

- **PIX for Xbox 360**

<https://xds.xbox.com/xbox360/nav.aspx?Page=xdkssoftware/xdkdownload.htm>

- **Live PIX for Windows (August 2005 version) demo:**

<http://channel9.msdn.com/ShowPost.aspx?PostID=109622>

# Questions?

- **Don't forget your evaluation forms**