在VR开发中使用Nsight Visual Studio Edition

阎安 – 图形工具QA经理

目录

Nvidia图形工具的最新更新

Nsight VSE图形调试入门

错误的Geometry显示

调试VR程序

针对GPU的Range Profiling

UNREAL ENGINE

Nvidia图形工具的最新更新

NSIGHT VISUAL STUDIO EDITION™

在Visual Studio中进行图形调试和剖析

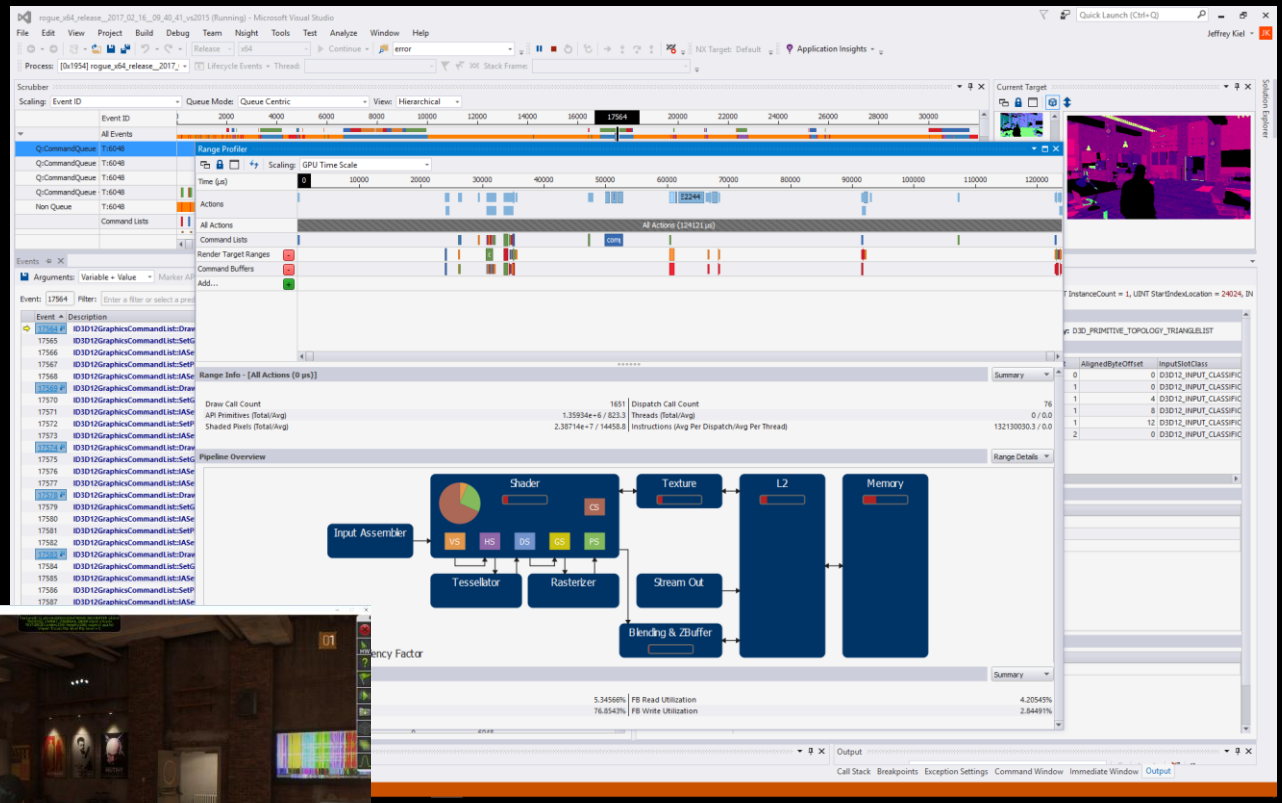
图形API调试

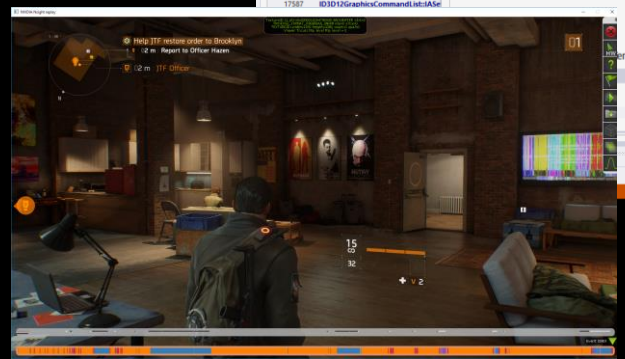
图形性能剖析

系统跟踪分析

DX9/DX11/DX12*/OpenGL/Vulkan**

支持Visual Studio 2012, 2013, 2015





NVIDIA

Nsight VSE 5.3

支持DX12和VR开发调试

完全支持DirectX 12的调试

支持VR APIs

支持在Win10 Hybrid笔记本上的调试体验

支持显示着色器的统计信息

对Range profiler的优化

更智能的memory viewer配置

支持对Vulkan程序进行帧截取并串行化为源代码

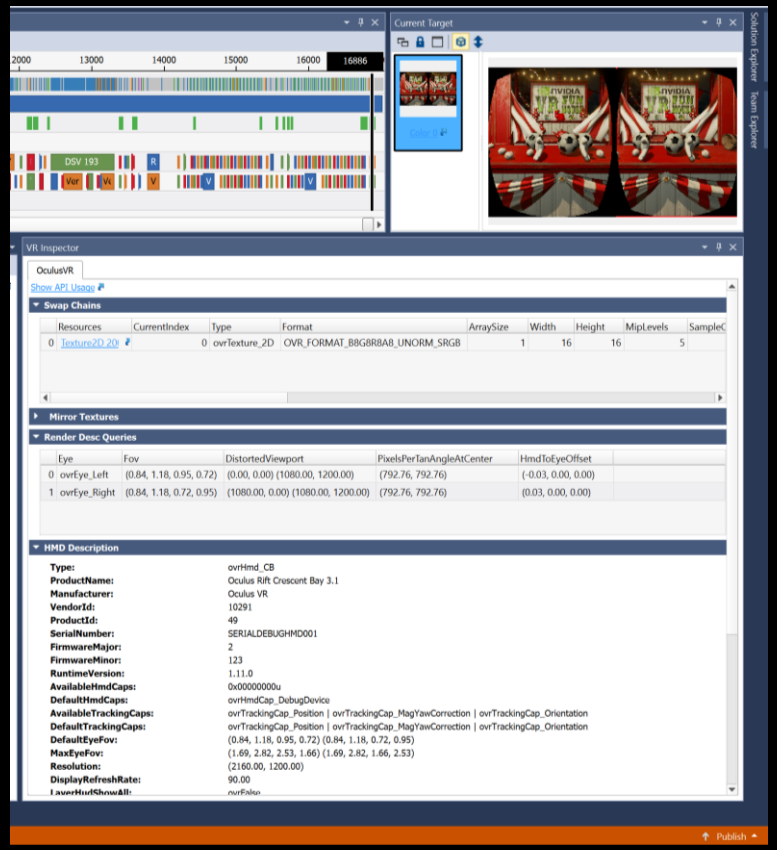
The New Razer Blade Pro
Powered by NVIDIA® GeForce® GTX 1080

支持VR API

Oculus SDK

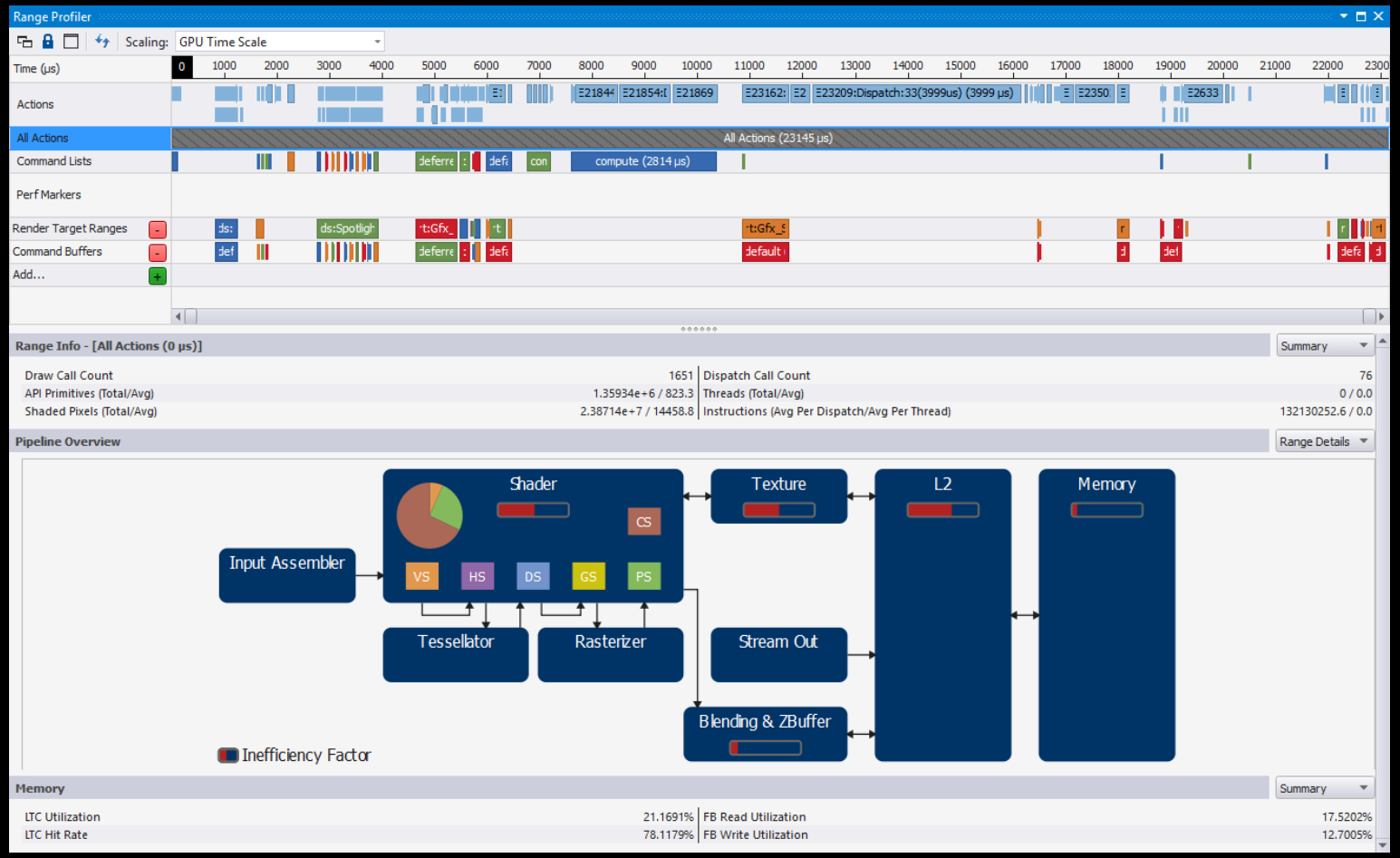
OpenVR & HTC Vive

NVAPI直接模式



gameworks.nvidia.com

Range Profiler的改进



Shader统计信息

ShaderPerf的回归

Linked Programs

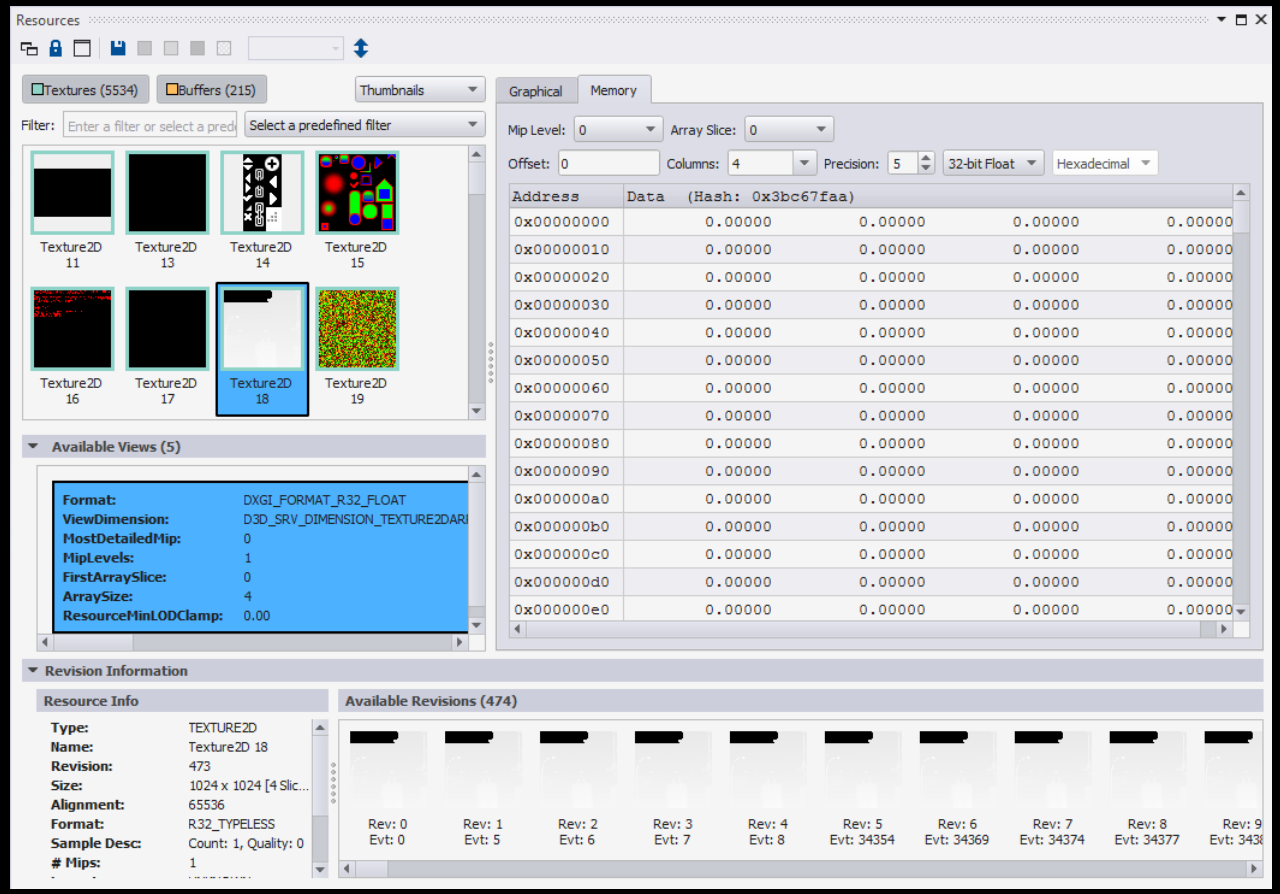
All Programs & Shaders

Name	Type	Binary	Debug Info	Cycles	Avg Cycles	ALU/TEX (Inst)	ALU/TEX (Cycles)	Regs	LMem (Bytes)
PipelineState 1	PSO		None	2935	170	9.35	3.69	27	0
PipelineState 10	PSO		None	3376	137	18.56	22.20	30	0
Unknown	CS		None	3376	137	18.56	22.20	30	0
PipelineState 100	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 101	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 102	PSO		None	672	45	17.75	7.82	18	0
PipelineState 103	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 104	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 105	PSO		None	1816	118	71.00	17.33	30	0
PipelineState 106	PSO		None	343	41	--	--	14	0
PipelineState 107	PSO		None	378	44	--	--	14	0
PipelineState 108	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 109	PSO		None	1174	97	16.70	4.60	30	0
PipelineState 11	PSO		None	3199	189	47.00	12.54	56	0
PipelineState 110	PSO		None	405	47	30.00	50.75	30	0
PipelineState 111	PSO		None	797	66	9.17	3.95	40	0
PipelineState 112	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 113	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 114	PSO		None	415	47	30.00	43.82	29	0
PipelineState 115	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A
PipelineState 116	PSO		None	N/A	N/A	N/A	N/A	N/A	N/A

Selected Program(s) & Shaders(s)

	Unknown
Float16	0
Float32	422
Slow Float32	47
Float64	0
Integer	28
Slow Integer	4
Interpolation	0
Tex	27
Read Generic	0
Read Thread Local	0
Read GPU Global	0
Read Attribute	0
Write Generic	0
Write Thread Local	0
Write GPU Global	1
Write Attribute	0
Flow Control	24
NOP	5

更智能的MEMORY VIEW



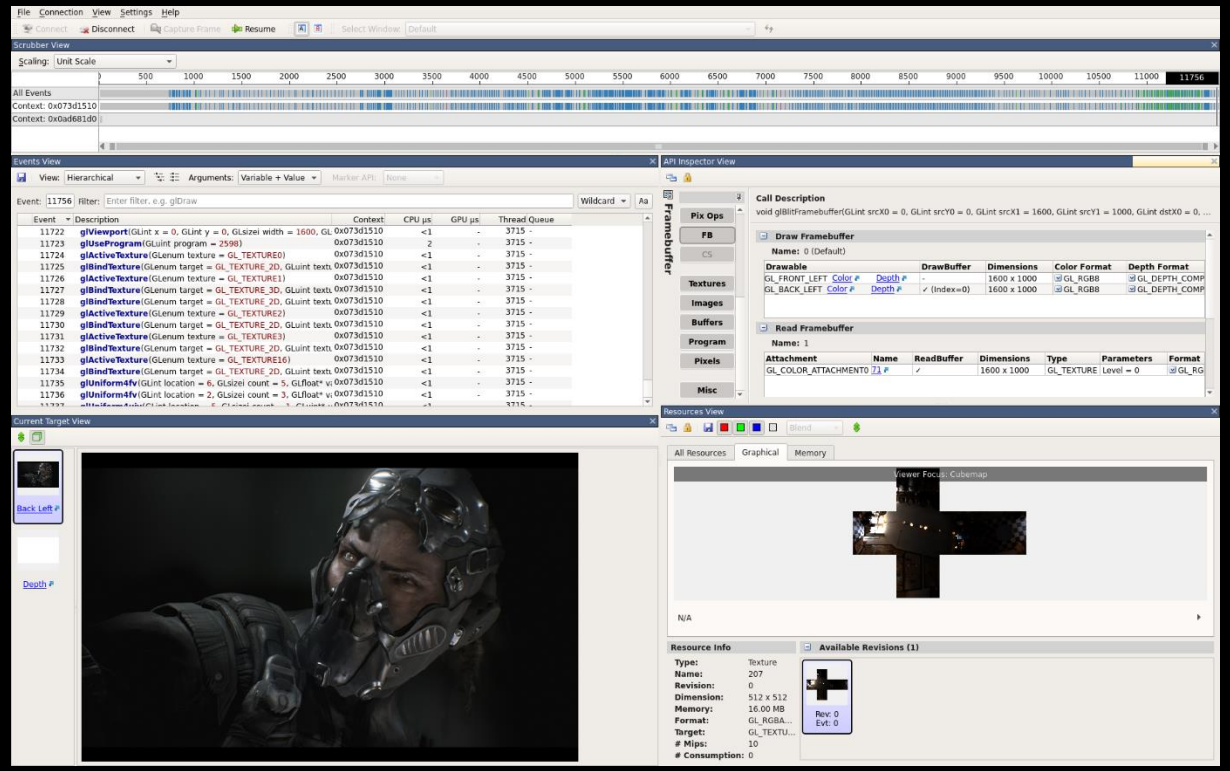
LINUX GRAPHICS DEBUGGER

针对Linux平台的图形调试器

帧截取

帧串行化为源代码工程

Range Profiler支持



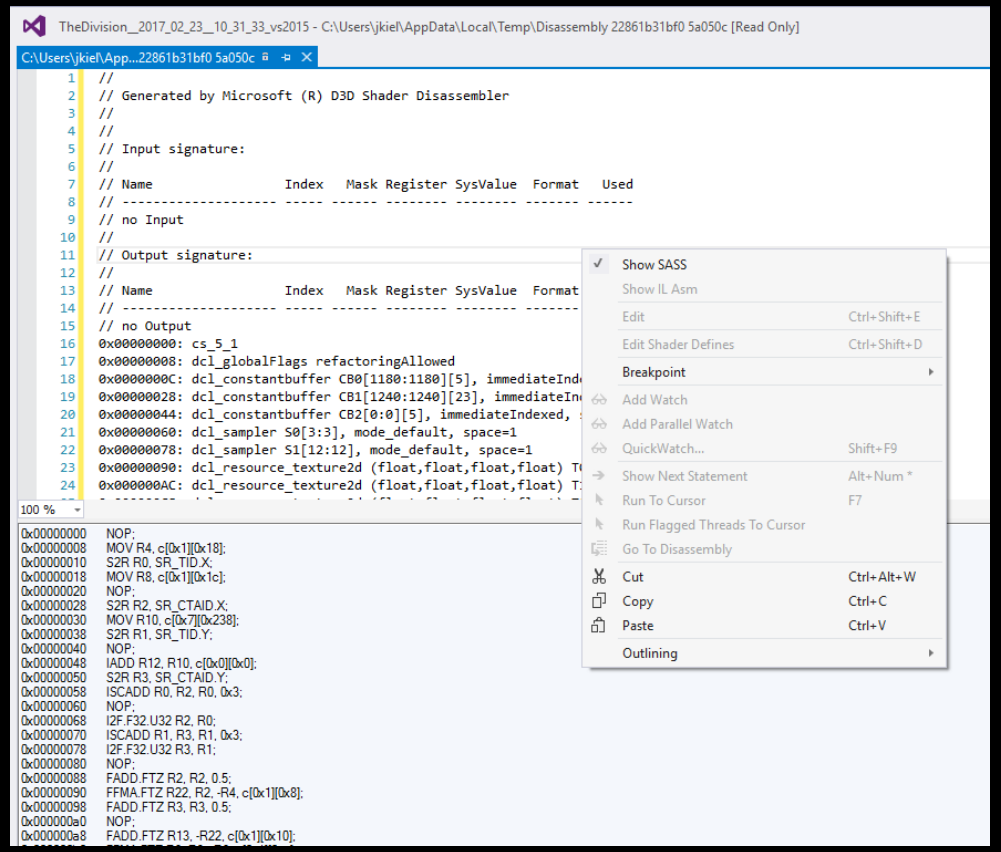
gameworks.nvidia.com

Nsight VSE PRO™

显示SASS指令

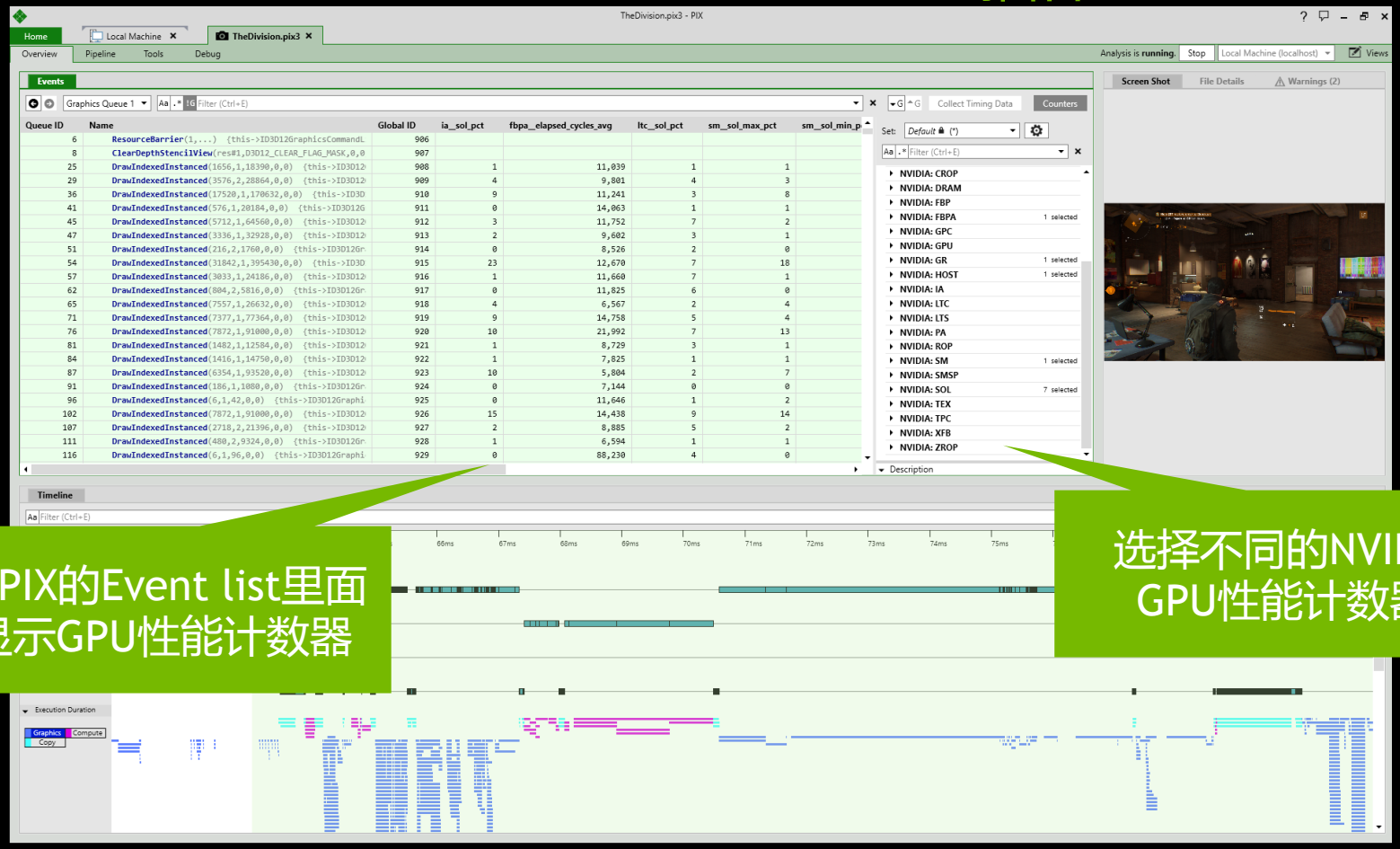
高级Performance Metrics (future)

着色器剖析 (future)



对微软PIX工具的支持

GPU Performance Metrics 插件



在PIX的Event list里面显示GPU性能计数器

选择不同的NVIDIA GPU性能计数器

CodeWorks

为游戏开发者提供Android开发支持

简化Android开发环境设置

支持Visual Studio上的交叉编译和远程调试

系统跟踪和多核CPU剖析

图形调试和剖析

和Unreal Engine 4 Mobile一起发布

可以在Tegra和非Tegra设备上工作*

在SHIELD设备上体验最好 – 不需要rooted OS Image

CODEWORKS 1r6

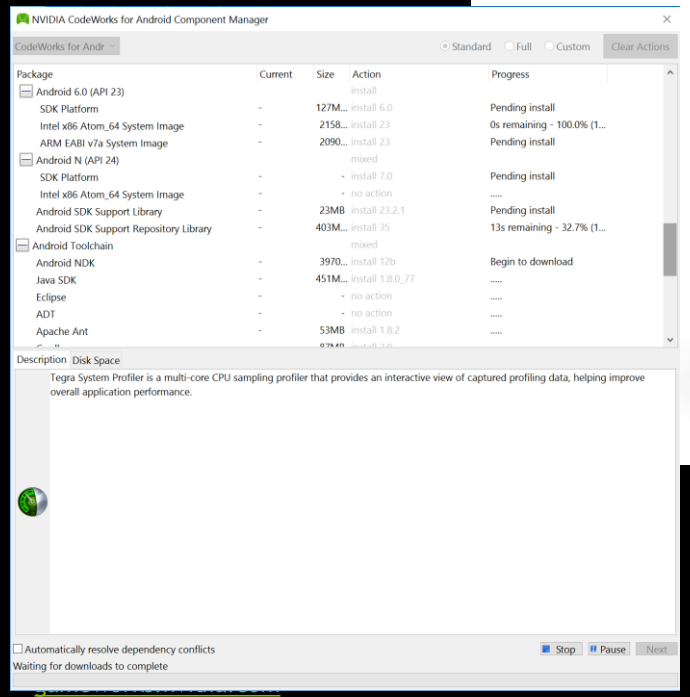
Android N/7.x, Android SDK 24.4, NDK r12b

SHIELD TV OTA 5.x 支持

Nsight Tegra VSE 3.4

Tegra Graphics Debugger 2.4

Tegra System Profiler 3.7



Nsight Tegra VSE 3.4

支持非Tegra的Android设备

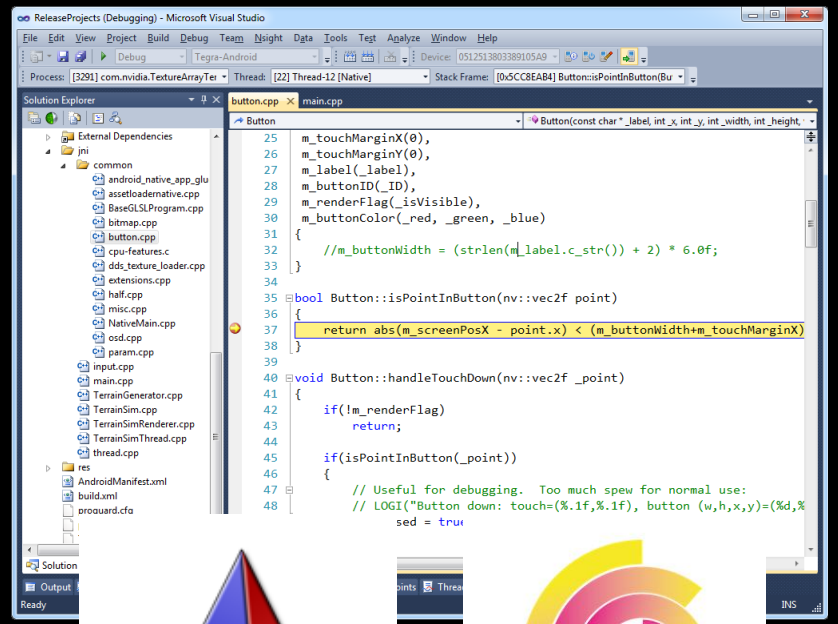
支持GDB Native C/C++的远程调试

单独的C++工程

支持交叉编译的配置

支持Incredibuild和CMake

支持Unreal Engine 4 native game project



TEGRA SYSTEM PROFILER 3.7

多核CPU的性能采样剖析

系统范围的内核，进程，线程的跟踪

多进程的API级的CPU/GPU跟踪

支持OpenGL/ES 和 NVTX

多窗口支持

Tegra GRAPHICS DEBUGGER 2.4

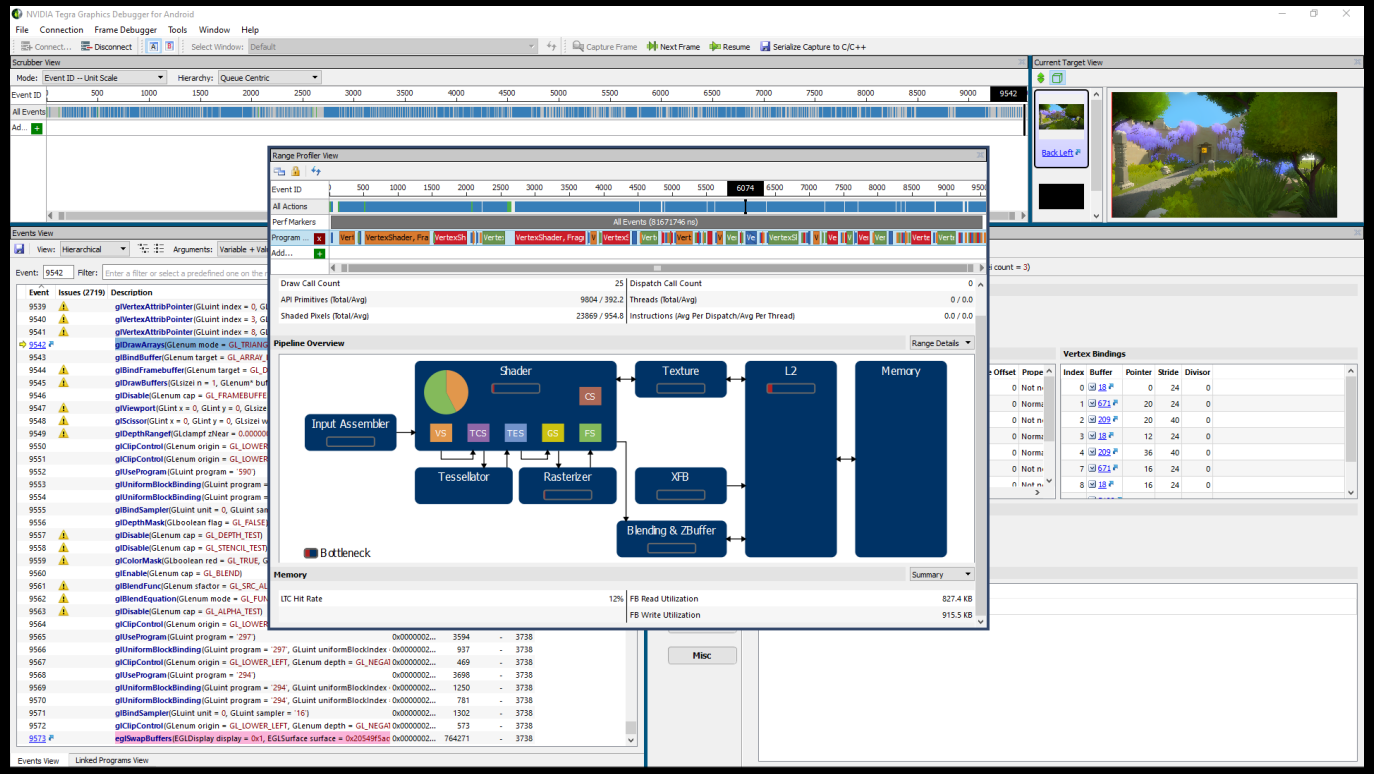
帧截取

同时串行化为源代码

Range Profiler

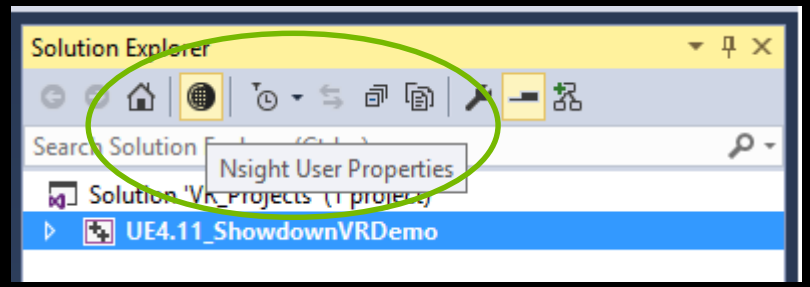
支持非root系统

支持非Tegra的Android设备



入门

入门

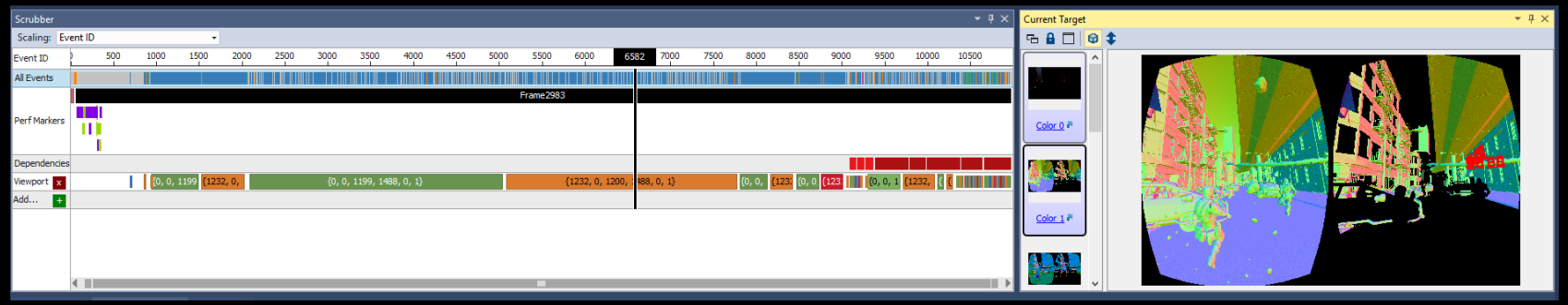


- 配置你工程的Nsight User Settings
- 点击VS中的Nsight菜单，选择Start Graphics Debugging
- 程序启动后，可以看到Nsight的HUD
  - 状态，FPS
  - GPU性能图示
- 使用快捷键Ctrl-Z
  - 可以移动，缩放HUD内的元素

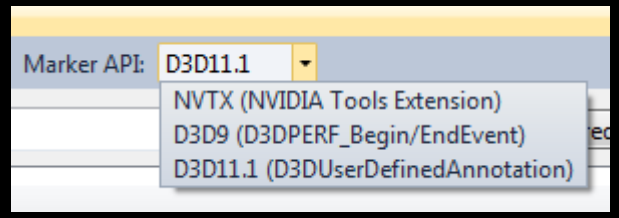
入门

- 按下空格键，捕捉当前帧
- 实时捕捉，程序依旧运行，只是被暂停了
- HUD允许你:
  - 查看当前场景的Drawcall
  - 查看贴图和Render Target
  - 以上功能可以脱离Visual Studio使用
  - 可以序列化当前的一帧，并生成C++的源代码和工程，支持VR API!

入门



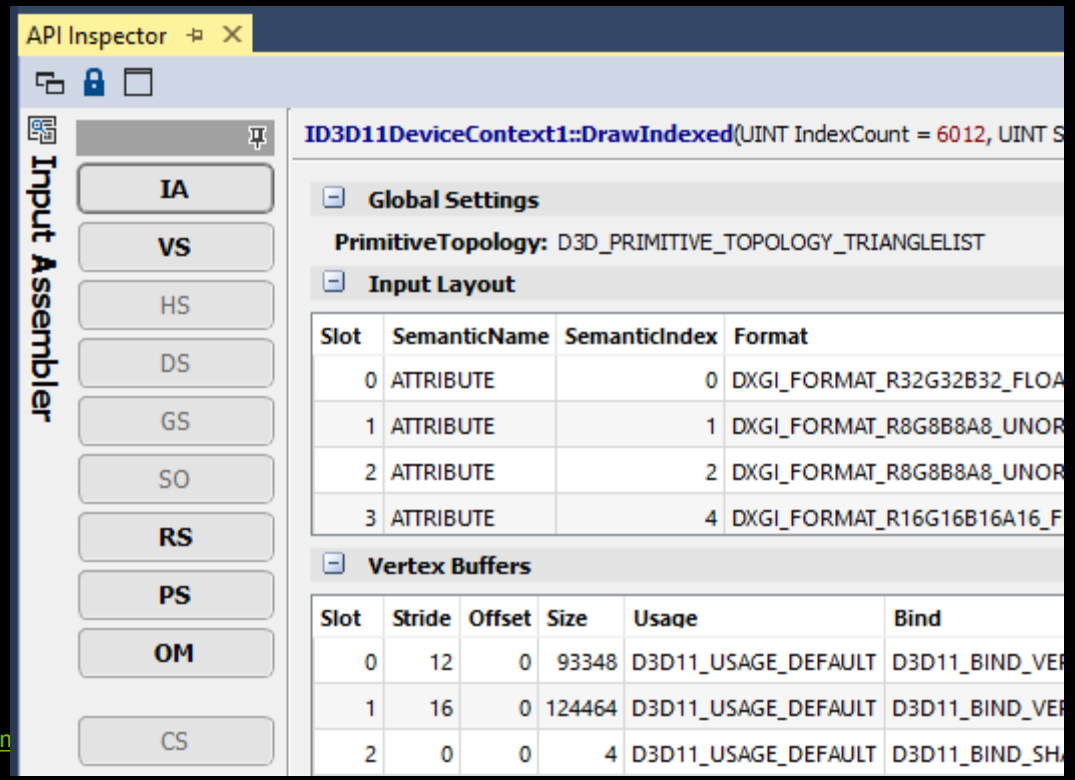
- 转到Visual Studio
- Scrubber视图
  - 允许你查看当前的所有Drawcall，从Perf Markers上获得当前Drawcall的上下文信息
- Event视图
  - 显示所有的API调用，附带参数信息
  - 和Scrubber视图同步
  - 根据关键字过滤 – ‘draw|pssetshader\(‘
  - 如果使用了现成的中间件或者引擎，开发者可能没有办法选择需要的Perfmarker API. Nsight允许显示指定的Perfmarker，开发者可以混用不同的API



- Current Targets视图 – 显示当前Drawcall绑定的所有RTV和DSV

入门

- 选中某一个Drawcall, 打开API Inspector视图
- 视图左边显示了基本的GPU管线, 分为一个个Stage
- 每个Stage表示当前Drawcall在对应管线的Stage上的状态
  - IA: 渲染管线的输入状态设置
  - VS, HS, DS, PS, CS: shader状态
  - SO
  - Raster
  - OM



入门

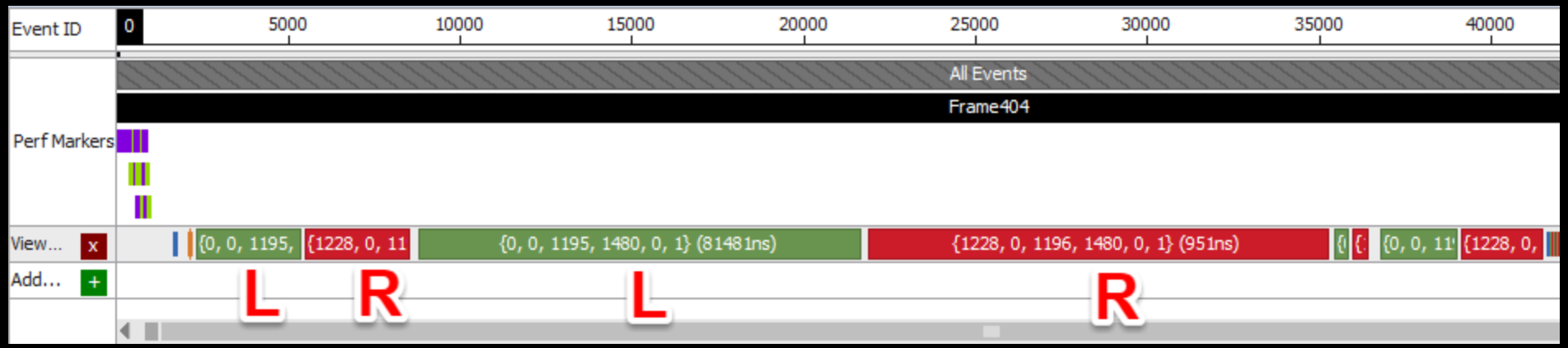
- 打开API Statistics视图，有一千左右的Draws，但有三万九千多的其他类型的API调用
- 其中有AddRef/Release调用大概一万三千左右
- 如果你过滤GetDesc，会发现有大约六百个，但是Desc是不可更改的，可以缓存下来
- 根据GetDesc的Object排序

Summary									
Draws:	1185	Clears:	44	Presents:	1	Misc. Data Update:	0	Other:	39205
Dispatches:	2	Blits:	1147	Command List Executes	0	Non-API:	1	Total:	41585

- 可以发现有一些Texture Object被调用了GetDesc大约8-12次
- 但是缓存一个D3D11_TEXTURE2D_DESC只需要44字节！

入门

- 其他Nsight的功能也对VR程序的调试有所帮助
- Scrubber视图新功能: 根据Drawcall不同的特性添加不同的Scrubber的Range
  - 选择”+”按钮, 选择”viewport”, 提供针对双眼的各自Drawcall的Range
- 当你没有配置自定义的Perfmarker的时候, 特别有用



入门

欢迎使用Nsight

HUD显示的performance图表和Drawcall scrubbing

Visual Studio集成的视图

- Scrubber, Events & Current Target
- API Inspector & API Statistics
- Geometry & Resources

使用基本视图功能来定位潜在的CPU性能问题

错误的Geometry

错误的Geometry

- 通过Nsight来启动对应的程序
  - 仅仅用来模拟Nsight用户曾遇到过的一类问题：我的渲染结果为什么变成这样了？

gameworks.nvidia.com

错误的Geometry

- 暂停并截取一帧，拖动HUD上的scrubber到*错误的Drawcall*
- 问题的根源可能有很多，使用Nsight来逐一检查
  - API Inspector – VS stage Vertex Shader 出问题了?
  - API Inspector – RS stage 光栅化设置出问题了?
  - Etc.

错误的Geometry

- API Inspector – VS stage，展开所有的项，检查各个buffer，有地方不对劲么?

API Inspector

Vertex Shader: IA | VS | HS | DS | GS | SO | RS | PS | OM | CS

ID3D11DeviceContext1::DrawIndexed(UINT IndexCount = 93144, UINT StartIndexLocation = 194592, INT BaseVertexLocation = 0x00000000)

VertexShader uid:5422

Source code (Main)

Shader Resource Views

SRV uid:1254 (Slot 0)

View

Name:	SRV uid:1254
Format:	DXGI_FORMAT_R32G32B32A32_FLOAT
ViewDimension:	D3D_SRV_DIMENSION_BUFFER
ElementOffset:	0
ElementWidth:	192

Resource Description

Name:	Buffer uid:1253
ByteWidth:	3072
Usage:	D3D11_USAGE_DYNAMIC
BindFlags:	D3D11_BIND_SHADER_RESOURCE | D3D11_BIND_VERTEX_BUFFER
CpuAccessFlags:	D3D11_CPU_ACCESS_WRITE
MiscFlags:	0x00000000u
StructureByteStride:	0

Constant Buffers

Slot	HLSL Buffer Name	HLSL Variables	Width	Stride	Object Name	FirstConstant	NumConstants	Usage	BindFlags	AccessFlags	MiscFlags
0	$Globals	11	512	0	Buffer uid:43	0	4096	D3D11_USAGE_DEFAULT	D3D11_BIND_CONSTANT_BUFFER	0x00000000u	0x00000000u
1	View	114	4096	0	Buffer uid:5384	0	4096	D3D11_USAGE_DYNAMIC	D3D11_BIND_CONSTANT_BUFFER	D3D11_CPU_ACCESS_WRITE	0x00000000u
2	Primitive	18	512	0	Buffer uid:3756	0	4096	D3D11_USAGE_DYNAMIC	D3D11_BIND_CONSTANT_BUFFER	D3D11_CPU_ACCESS_WRITE	0x00000000u

Samplers

Slot	Filter	Address U	Address V	Address W	MipLODBias	MaxAnisotropy	ComparisonFunc	RGBA
0	D3D11_FILTER_MIN_MAG_MIP_POINT	D3D11_TEXTURE_ADDRESS_WRAP	D3D11_TEXTURE_ADDRESS_WRAP	D3D11_TEXTURE_ADDRESS_WRAP	0.00	0	D3D11_COMPARISON_NEVER	
1	D3D11_FILTER_MIN_MAG_MIP_POINT	D3D11_TEXTURE_ADDRESS_CLAMP	D3D11_TEXTURE_ADDRESS_CLAMP	D3D11_TEXTURE_ADDRESS_CLAMP	0.00	0	D3D11_COMPARISON_NEVER	
2	D3D11_FILTER_MIN_MAG_MIP_POINT	D3D11_TEXTURE_ADDRESS_CLAMP	D3D11_TEXTURE_ADDRESS_CLAMP	D3D11_TEXTURE_ADDRESS_CLAMP	0.00	0	D3D11_COMPARISON_NEVER	

错误的Geometry

- API Inspector – RS stage, 光栅化里面的cull mode设置, fill mode设置, depth bias设置, 等等, 有地方不对劲么?

API Inspector

Rasterizer

IA
VS
HS
DS
GS
SO
RS
PS
OM
CS

ID3D11DeviceContext1::DrawIndexed(UINT IndexCount = 93144, UINT StartIndexLocation = 194592, INT BaseVertexLocation = 0x00000000)

Rasterizer State

FillMode:	D3D11_FILL_SOLID	**DepthBias:**	0	**DepthClipEnable:**	TRUE	**AntialiasedLineEnable:**	FALSE
CullMode:	D3D11_CULL_BACK	**DepthBiasClamp:**	0.00	**ScissorEnable:**	TRUE		
FrontCounterClockwise:	TRUE	**SlopeScaledDepthBias:**	0.00	**MultisampleEnable:**	TRUE		

Viewports

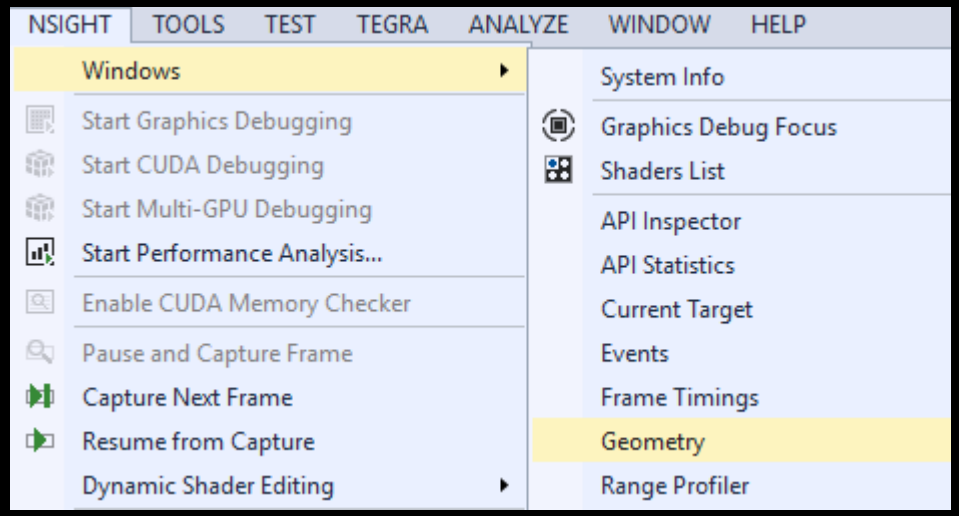
	TopLeftX	TopLeftY	Width	Height	MinDepth	MaxDepth
0	0.00	0.00	1195.00	1480.00	0.00	1.00

Scissors

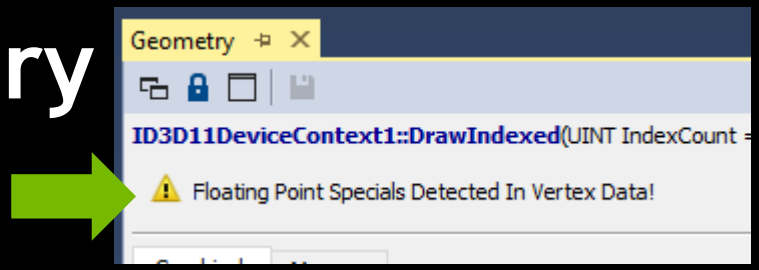
	Left	Top	Right	Bottom
0	0	0	16384	16384

错误的Geometry

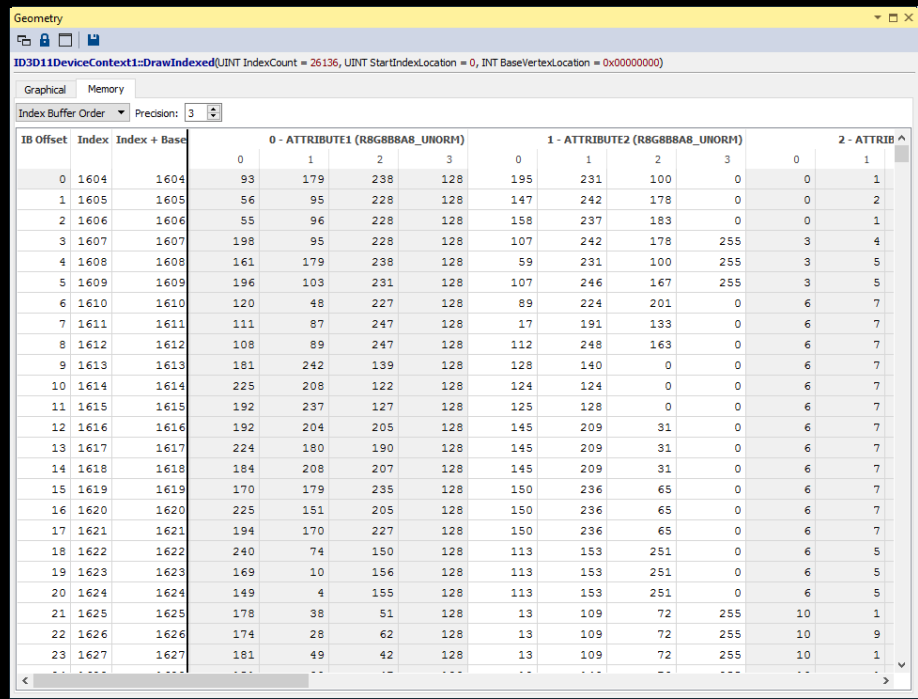
- 看上去渲染管线的设置都没问题，可能是数据的问题?
- 打开Geometry视图



错误的Geometry



- 打开Geometry View [Nsight 5.1新功能]
  - Nsight已经检测到一些浮点数异常: “Floating Point Specials Detected In Vertex Data!”
- Graphical子视图
  - 新的Geometry View可以映射任何顶点属性为渲染需要的位置/颜色/法线信息
  - 显示法线或者UV信息为颜色
  - 一直维持顶点属性的设置，直到vertex layout改变
- Memory子视图
  - 每一个顶点为一行，列为顶点对应的各个属性
  - 可以以index buffer order或者vertex buffer order显示
  - 数据显示已经考虑到了对应的offset



IB Offset	Index	Index + Base	0 - ATTRIBUTE1 (R8G8B8A8_UNORM)				1 - ATTRIBUTE2 (R8G8B8A8_UNORM)				2 - ATTRIB	
			0	1	2	3	0	1	2	3	0	1
0	1604	1604	93	179	238	128	195	231	100	0	0	1
1	1605	1605	56	95	228	128	147	242	178	0	0	2
2	1606	1606	55	96	228	128	158	237	183	0	0	1
3	1607	1607	198	95	228	128	107	242	178	255	3	4
4	1608	1608	161	179	238	128	59	231	100	255	3	5
5	1609	1609	196	103	231	128	107	246	167	255	3	5
6	1610	1610	120	48	227	128	89	224	201	0	6	7
7	1611	1611	111	87	247	128	17	191	133	0	6	7
8	1612	1612	108	89	247	128	112	248	163	0	6	7
9	1613	1613	181	242	139	128	128	140	0	0	6	7
10	1614	1614	225	208	122	128	124	124	0	0	6	7
11	1615	1615	192	237	127	128	125	128	0	0	6	7
12	1616	1616	192	204	205	128	145	209	31	0	6	7
13	1617	1617	224	180	190	128	145	209	31	0	6	7
14	1618	1618	184	208	207	128	145	209	31	0	6	7
15	1619	1619	170	179	235	128	150	236	65	0	6	7
16	1620	1620	225	151	205	128	150	236	65	0	6	7
17	1621	1621	194	170	227	128	150	236	65	0	6	7
18	1622	1622	240	74	150	128	113	153	251	0	6	5
19	1623	1623	169	10	156	128	113	153	251	0	6	5
20	1624	1624	149	4	155	128	113	153	251	0	6	5
21	1625	1625	178	38	51	128	13	109	72	255	10	1
22	1626	1626	174	28	62	128	13	109	72	255	10	9
23	1627	1627	181	49	42	128	13	109	72	255	10	1

错误的Geometry

- 进一步查看”Float Point Special Detected In Vertex Data!”
- 还可以使用Resource视图中的tag功能来显示场景中所有使用了这个buffer的Drawcall. 不仅仅标记vertex buffer, 还支持任意资源, 例如texture资源

Geometry

ID3D11DeviceContext1::DrawIndexed(UINT IndexCount = 25056, UINT StartIndexLocation = 96240, INT BaseVertexLocation = 0x00000000)

Floating Point Specials Detected In Vertex Data!

Graphical | Memory

Index Buffer Order | Precision: 3 | Filter To Specials

IB Offset	Index	Index + Base	2 - ATTRIBUTE3 (R8G8B8A8_UINT)				3 - ATTRIBUTE4 (R8G8B8A8_UNORM)				4 - ATTRIBUTE0 (R32G32B32_FLOAT)			5 - ATTRIBUTE8 (R16G.	
			0	1	2	3	0	1	2	3	0	1	2	0	1
0	220...	22082	4	0	0	0	255	0	0	0	116.805	nan	204.358	1.525	0.49
1	220...	22088	4	0	0	0	255	0	0	0	nan	55.767	197.995	1.576	0.47
2	220...	22089	4	0	0	0	255	0	0	0	116.143	56.535	197.428	1.576	0.49
3	220...	22081	4	0	0	0	255	0	0	0	113.202	57.592	204.868	1.525	0.47
4	220...	22090	4	0	0	0	255	0	0	0	111.499	52.312	198.907	1.576	0.44
5	220...	22088	4	0	0	0	255	0	0	0	nan	55.767	197.995	1.576	0.47
6	220...	22081	4	0	0	0	255	0	0	0	113.202	57.592	204.868	1.525	0.47
7	220...	22088	4	0	0	0	255	0	0	0	nan	55.767	197.995	1.576	0.47
8	220...	22082	4	0	0	0	255	0	0	0	116.805	nan	204.358	1.525	0.49
9	220...	22075	4	0	0	0	255	0	0	0	117.024	60.949	210.377	1.479	0.49
10	220...	22081	4	0	0	0	255	0	0	0	113.202	57.592	204.868	1.525	0.47
11	220...	22082	4	0	0	0	255	0	0	0	116.805	nan	204.358	1.525	0.49
12	220...	22083	4	0	0	0	255	0	0	0	112.184	54.229	206.098	1.525	0.44
13	220...	22090	4	0	0	0	255	0	0	0	111.499	52.312	198.907	1.576	0.44
14	220...	22081	4	0	0	0	255	0	0	0	113.202	57.592	204.868	1.525	0.47
15	220...	22081	4	0	0	0	255	0	0	0	113.202	57.592	204.868	1.525	0.47
16	220...	22074	4	0	0	0	255	0	0	0	113.417	60.127	210.799	1.479	0.47
17	220...	22083	4	0	0	0	255	0	0	0	112.184	54.229	206.098	1.525	0.44

错误的GEOMETRY

顶点数据错误

API Inspector视图帮助检查渲染管线设置

Geometry视图帮助检查变形前的数据

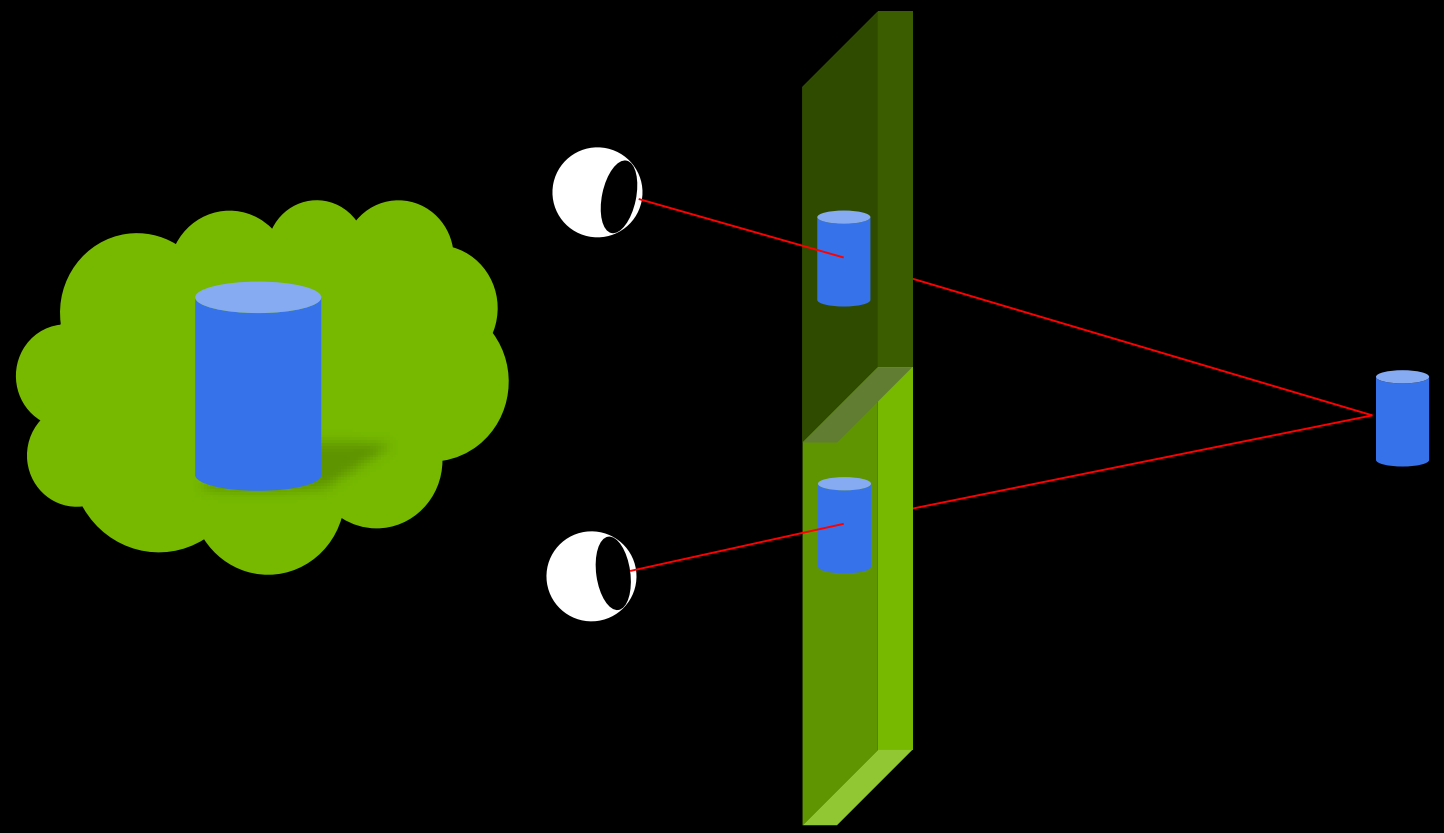
Resources视图 + ‘tag’来显示何处使用了资源

VR...但不那么立体

视差和双目视觉

- 看向前方，选择一个物体作为前景，例如你的手指，或者某个人的头部
- 交替的遮住你的左右眼睛
- 当你切换遮住的眼睛的时候，注意到你选择的物体相对于背景是移动的
- 这个叫做视差，是人类双目视觉系统本能的一部分
- 赋予我们感知深度的能力，同时VR利用其来实现更加沉浸式的体验

视差和双目视觉



VR...但不那么立体

- 有些时候，VR渲染的立体感被破坏，进而影响体验
- 看到过立体场景里面的物体像是若干个不同深度的billboard?
- 注意下面场景截图中飞在半空的水泥块
- 如果是在实际的VR头戴显示器上观察，就会发现这些物体不正常的贴在半空，就像一个UI或者狙击枪的十字标线一样

16.2 FPS, 61.78 ms/frame
Active shaders: Original
HUD activated
Geom Busy
Shader Busy
Texture Busy
25.0
12.5
0.0
GPU Busy
50.0
25.0
0.0
Histogram: prims per draw
750
375
0
HW

VR...但不那么立体

- 通过Nsight来启动对应的程序
- 截取当前帧(Ctrl-Z + Space作为快捷键)
- 打开Resources视图, 来看看水泥块对应的深度是否会导致这个问题
- 首先, 找到导致”不那么立体”效果的水泥块的Drawcall

VR...但不那么立体

- 找到渲染左眼Render Target的Drawcall
- 在Event视图中按下Ctrl+F2来收藏这个event
- 找到渲染右眼Render Target的Drawcall
- 在Event视图中按下Ctrl+F2来收藏这个event
- 如果你没有自定义PerfMarker，可以使用Scrubber视图中自定义的Viewport项添加新的一行Range集合，进而确认这些Drawcall对应不同的眼睛
- 打开Current Target视图，选择并点击当前的深度图，来打开Resources视图，进而检查对应深度贴图的详细信息

Scrubber

Scaling: Event ID

Event ID 500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000 7780 8500 9000 9500 10000 10500

All Events

Frame2983

Perf Markers

Dependencies

Viewport {0, 0, 1199 {1232, 0, {0, 0, 1199, 1488, 0, 1} {1232, 0, 1200, 1488, 0, 1}

Add...

Current Target

Color 5

Depth

Geometry API Inspector Events

View: Hierarchical Arguments: Variable + Value Marker API: NVTX

Event: 7780 Filter: Enter a filter or select a predefined one on the right Select a predefined filter

Event	Description	Object	CPU µs	GPU µs	Thread
7769	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 0, UINT NumBuf	0x000001F...	0	-	8524
7770	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 2, UINT NumBuf	0x000001F...	<1	-	8524
7771	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 3, UINT NumBuf	0x000001F...	0	-	8524
7772	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 4, UINT NumBuf	0x000001F...	0	-	8524
7773	ID3D11DeviceContext1::IASetIndexBuffer(ID3D11Buffer* pIndexBuffer = 'Bu	0x000001F...	0	-	8524
7774	ID3D11DeviceContext1::DrawIndexedInstanced(UINT IndexCountPerInstance	0x000001F...	0	10	8524
7775	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 0, UINT NumBuf	0x000001F...	0	-	8524
7776	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 2, UINT NumBuf	0x000001F...	0	-	8524
7777	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 3, UINT NumBuf	0x000001F...	<1	-	8524
7778	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 4, UINT NumBuf	0x000001F...	0	-	8524
7779	ID3D11DeviceContext1::IASetIndexBuffer(ID3D11Buffer* pIndexBuffer = 'Bu	0x000001F...	0	-	8524
7780	ID3D11DeviceContext1::DrawIndexedInstanced(UINT IndexCountPerInstance	0x000001F...	0	13	8524
7781	ID3D11DeviceContext1::IASetInputLayout(ID3D11InputLayout* pInputLayou	0x000001F...	<1	-	8524
7782	ID3D11DeviceContext1::VSSetShader(ID3D11VertexShader* pVertexShader =	0x000001F...	0	-	8524
7783	ID3D11DeviceContext1::PSSetShader(ID3D11PixelShader* pPixelShader = 'Pi	0x000001F...	0	-	8524
7784	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 0, UINT NumBuf	0x000001F...	<1	-	8524
7785	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 1, UINT NumBuf	0x000001F...	<1	-	8524
7786	ID3D11DeviceContext1::IASetVertexBuffers(UINT StartSlot = 2, UINT NumBuf	0x000001F...	0	-	8524
7787	ID3D11DeviceContext1::VSSetConstantBuffers(UINT StartSlot = 1, UINT Numl	0x000001F...	0	-	8524
7788	ID3D11DeviceContext1::PSSetConstantBuffers(UINT StartSlot = 3, UINT Num	0x000001F...	0	-	8524
7789	ID3D11DeviceContext1::VSSetConstantBuffers(UINT StartSlot = 4, UINT Numl	0x000001F...	0	-	8524
7790	ID3D11DeviceContext1::VSSetConstantBuffers(UINT StartSlot = 3, UINT Numl	0x000001F...	<1	-	8524
7791	ID3D11DeviceContext1::PSSetShaderResources(UINT StartSlot = 0, UINT Num	0x000001F...	<1	-	8524

VR...但不那么立体

- 在Resources视图中检查当前的DSV
- 注意，你可能只能看到全黑或者全白的图像
- 打开histogram and renormalize (texture图片右下角的小箭头)，拖动max箭头来调整当前显示值的范围，这样就可以看到深度贴图的细节信息
- 注意水泥块在每个眼睛下的深度信息

Resources

Textures (365) | Buffers (1146) | Thumbnails

Filter: Enter a filter or select a pred | Select a predefined filter

Texture2D 775 | Texture2D 776 | Texture2D 777 | Texture2D 778
Texture2D 779 | Texture2D 780 | Texture2D 781 | Texture2D 785
Texture2D 790 | Texture2D 798 | Texture2D 800 | Texture2D 802
Texture2D 830 | Texture2D 830 | Texture2D 830 | Texture2D 850

Available Views (2)

Name:	DSV 81
Format:	DXGI_FORMAT_D24_UNORM_S8_UINT
ViewDimension:	D3D11_DSV_DIMENSION_TEXTURE2D
MipSlice:	0

Graphical | Memory

Viewer Focus: Mip Level
Mip Level = 0

0.000000 Min | Max 0.014795

Pos: (1837, 850) Raw Value: 0x9626e Normalize: By Histogram View Stencil

Revision Information

Resource Info

Type:	2D Texture
Name:	Texture2D 830
Revision:	843
Size:	2432 x 1488
Memory:	13.80 MB
Format:	DXGI_FORMAT_R24G8_TYPELESS
# Mips:	1
# Consumptions:	76

Available Revisions (891)

Rev: 832 Evt: 7692 | Rev: 833 Evt: 7698 | Rev: 834 Evt: 7704 | Rev: 835 Evt: 7710 | Rev: 836 Evt: 7716 | Rev: 837 Evt: 7722 | Rev: 838 Evt: 7728 | Rev: 839 Evt: 7746 | Rev: 840 Evt: 7762 | Rev: 841 Evt: 7768 | Rev: 842 Evt: 7774 | Rev: 843 Evt: 7780 | Rev: 844 Evt: 7795 | Rev: 845 Evt: 9034 | Rev: 846 Evt: 9282 | Rev: 847 Evt: 9435

VR…但不那么立体

- 使用API Inspector视图来确认渲染管线的设置，看上去都没问题
- 还有其他因素可能影响么？我们需要查看VR的设置！
- 打开VR Inspector视图
- 可以查看Swap Chains, Mirror Textures, Render Desc Queries, HMD信息

VR Inspector

Show API Usage

Swap Chains

	Resources	CurrentIndex	Type	Format	ArraySize	Width	Height	MipLevels	SampleCount	StaticImage	MiscFlags
0	Texture2D 342	0	ovrTexture_2D	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1	16	16	5	1	ovrTrue	ovrTextureMisc_AllowGenerateMip
1	Texture2D 343 Texture2D 344 Texture2D 345	0	ovrTexture_2D	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1	2432	1488	1	1	ovrFalse	ovrTextureMisc_DX_Typeless

Mirror Textures

	Resource	Format	Width	Height	MiscFlags
0	Texture2D 346	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1920	1080	ovrTextureMisc_DX_Typeless

Render Desc Queries

	Eye	Fov	DistortedViewport	PixelsPerTanAngleAtCenter	HmdToEyeOffset
0	ovrEye_Left	(1.33, 1.33, 1.06, 1.09)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)
1	ovrEye_Left	(1.33, 1.33, 1.09, 1.06)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)

HMD Description

Type:	ovrHmd_DK2
ProductName:	Oculus Rift DK2
Manufacturer:	Oculus VR
VendorId:	10291
ProductId:	33
SerialNumber:	209QKS04EVHM
FirmwareMajor:	2
FirmwareMinor:	12
RuntimeVersion:	1.6.0
AvailableHmdCaps:	0x00000000u
DefaultHmdCaps:	0x00000000u
AvailableTrackingCaps:	ovrTrackingCap_Position | ovrTrackingCap_MagYawCorrection | ovrTrackingCap_Orientation
DefaultTrackingCaps:	ovrTrackingCap_Position | ovrTrackingCap_MagYawCorrection | ovrTrackingCap_Orientation
DefaultEyeFov:	(1.33, 1.33, 1.06, 1.09) (1.33, 1.33, 1.09, 1.06)
MaxEyeFov:	(1.87, 1.87, 1.62, 1.68) (1.87, 1.87, 1.68, 1.62)
Resolution:	(1920.00, 1080.00)
DisplayRefreshRate:	75.00
LayerHudShowAll:	ovrFalse
DebugHudStereoGuideInfoEnable:	ovrTrue
PlayerHeight:	1.78
EyeHeight:	1.67
NeckEyeDistance:	{ 0.08, 0.08 }
EyeToNoseDist:	{ 0.03, 0.03 }
DebugHudStereoGuideSize2f:	{ 1.00, 1.00 }
DebugHudStereoGuidePosition3f:	{ 0.00, 0.00, -1.50 }
DebugHudStereoGuideYawPitchRoll3f:	{ 0.00, 0.00, 0.00 }
DebugHudStereoGuideColor4f:	{ 1.00, 0.50, 0.10, 0.80 }
PerfHudMode:	0
LayerHudMode:	0
LayerHudCurrentLayer:	0
DebugHudStereoMode:	0
User:	Unknown
Name:	Unknown
Gender:	Male

VR...但不那么立体

- 点击其中的‘Show API Usage’可以在Events视图中显示所有的VR API调用

Event ID | All Events | Perf Markers | Dependencies | Add...

Geometry | API Inspector | Events

View: Hierarchical | Arguments: Variable + Value | Marker API: NVTX

Event: 7780 | Filter: ovr_

Event	Description	Object
2	ovr_GetTimeInSeconds() = 569.551015	-
4	ovr_GetTimeInSeconds() = 569.551023	-
59	ovr_GetTimeInSeconds() = 569.577314	-
61	ovr_GetTimeInSeconds() = 569.577321	-
63	ovr_GetTimeInSeconds() = 569.577323	-
65	ovr_GetTimeInSeconds() = 569.577323	-
67	ovr_GetPredictedDisplayTime(ovrSession session = 0xb5cace38, int64_t frameIndex = 0x000009ac) = 0.000000	-
68	ovr_GetTrackingState(ovrSession session = 0xb5cace38, double absTime = 436.565595, ovrBool latencyMarker = ovrFalse) = { 'Tracking State' }	-
690	ovr_GetPredictedDisplayTime(ovrSession session = 0xb5cace38, int64_t frameIndex = 0x000009ab) = 0.000000	-
691	ovr_GetPredictedDisplayTime(ovrSession session = 0xb5cace38, int64_t frameIndex = 0x000009ab) = 0.000000	-
692	ovr_GetTrackingState(ovrSession session = 0xb5cace38, double absTime = 436.552261, ovrBool latencyMarker = ovrTrue) = { 'Tracking State' }	-
10932	ovr_CommitTextureSwapChain(ovrSession session = 0xb5cace38, ovrTextureSwapChain chain = 0xb59d8fc0) = 0x00000000	-
10933	ovr_GetTextureSwapChainCurrentIndex(ovrSession session = 0xb5cace38, ovrTextureSwapChain chain = 0xb59d8fc0, int32_t* out_Index = 0x000000A17D0FDAB4) = 0x00	-
10984	ovr_SubmitFrame(ovrSession session = 0xb5cace38, int64_t frameIndex = 0x00000000, ovrViewScaleDesc* viewScaleDesc = 0x00007FF66F613C9C, ovrLayerHeaderPtr* laye	-
10985	ovr_GetTimeInSeconds() = 569.726791	-
10986	ovr_GetTimeInSeconds() = 569.726792	-
10987	ovr_GetTimeInSeconds() = 569.730898	-
10988	ovr_GetTimeInSeconds() = 569.730898	-
10989	ovr_GetTextureSwapChainCurrentIndex(ovrSession session = 0xb5cace38, ovrTextureSwapChain chain = 0xb59d8fc0, int32_t* out_Index = 0x00007FF66F5E8498) = 0x000	-

VR Inspector

Show API Usage

Swap Chains

	Resources	CurrentIndex	Type	Format	ArraySize	Width
0	Texture2D 342	0	ovrTexture_2D	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1	16
1	Texture2D 343 Texture2D 344 Texture2D 345	0	ovrTexture_2D	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1	2432

Mirror Textures

	Resource	Format	Width	Height	MiscFlags
0	Texture2D 346	OVR_FORMAT_B8G8R8A8_UNORM_SRGB	1920	1080	ovrTextureMisc_DX_Typeless

Render Desc Queries

	Eye	Fov	DistortedViewport	PixelsPerTanAngleAtCenter	HmdToEyeOffset
0	ovrEye_Left	(1.33, 1.33, 1.06, 1.09)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)
1	ovrEye_Left	(1.33, 1.33, 1.09, 1.06)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)

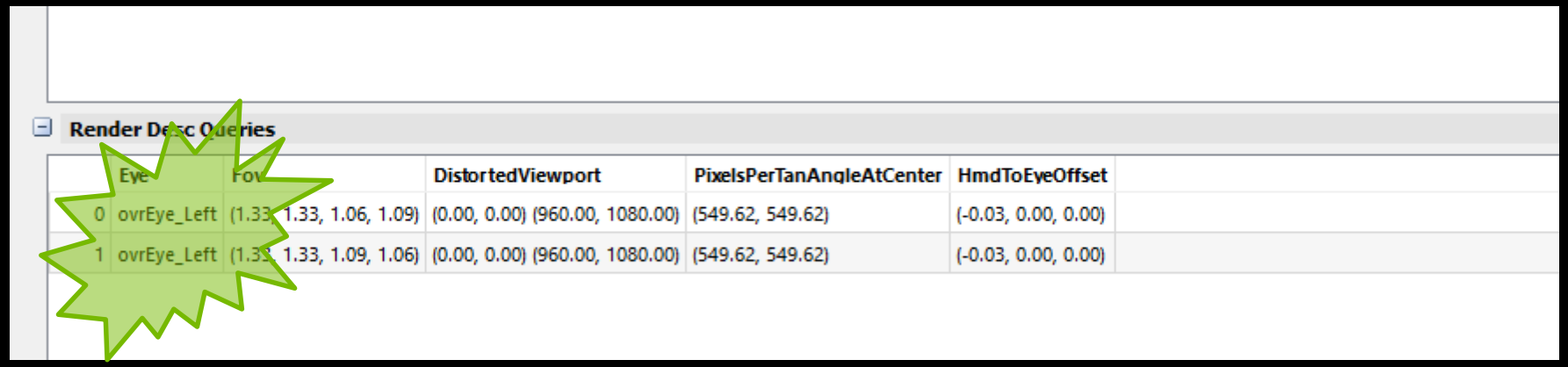
HMD Description

Type: ovrHmd_DK2
ProductName: Oculus Rift DK2
Manufacturer: Oculus VR
VendorId: 10291
ProductId: 33
SerialNumber: 209QKS04EVHM
FirmwareMajor: 2

LayerHudShowAll:
DebugHudStereoGuideInfoE
PlayerHeight:
EyeHeight:
NeckEyeDistance:
EyeToNoseDist:
DebugHudStereoGuideSize2

VR...但不那么立体

- 仔细查看VR Inspector视图里面的Render Desc Queries
- VR程序query了左眼的状态两次
- 这可能是在copy-paste的时候，忘记更改右眼的状态了 ☹
- 改回左右眼query各一次，重新启动，问题解决了！



Render Desc Queries

	Eye	Fov	DistortedViewport	PixelsPerTanAngleAtCenter	HmdToEyeOffset
0	ovrEye_Left	(1.33, 1.33, 1.06, 1.09)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)
1	ovrEye_Left	(1.33, 1.33, 1.09, 1.06)	(0.00, 0.00) (960.00, 1080.00)	(549.62, 549.62)	(-0.03, 0.00, 0.00)

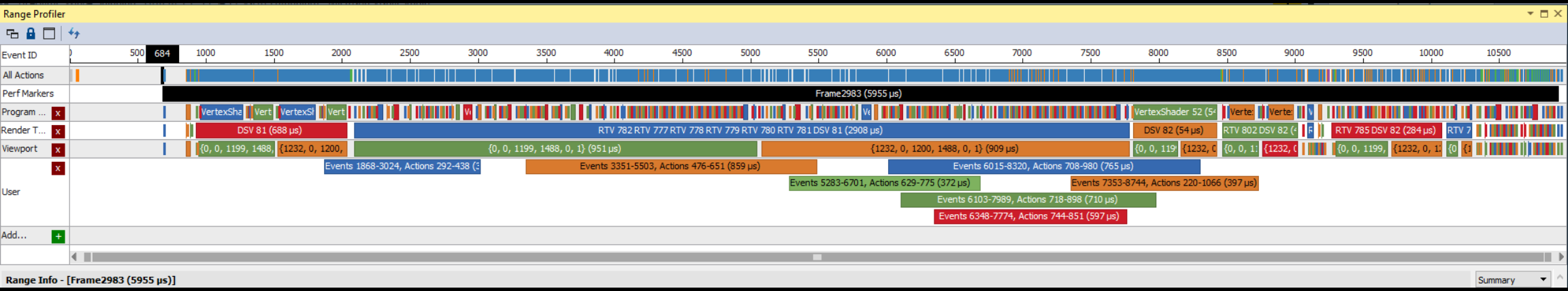
VR…但不那么立体

在Events视图中收藏特定的Drawcall
在Resources视图中查看贴图
图像的柱状图查看器及重映射
在API Inspector视图中检查渲染管线的设置
在VR Inspector视图中检查VR的设置

瓶颈在哪儿?

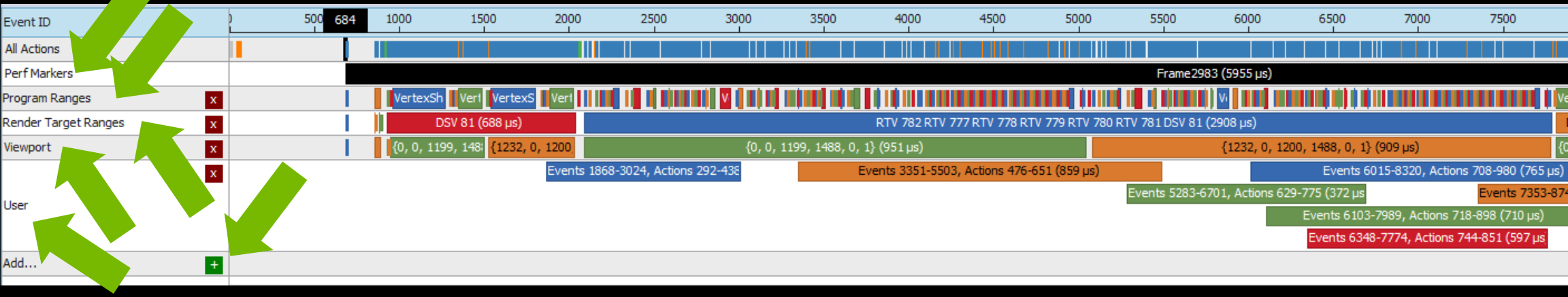
瓶颈在哪儿?

- 通过Nsight来启动对应的程序
- 如果是GPU bound的话，你希望使用profiler来定位渲染管线的那一部分是瓶颈，会导致程序性能下降
- 在Scrubber视图上右键单击，选择”Profile Frame”来打开新的Range Profiler视图
- 传统的Profiler是以Drawcall为中心，允许按照管线状态来分类Drawcall
- 新的Profiler是以“Range”为中心：一系列的Drawcall定义了一个Range.



瓶颈在哪儿?

- 新的范式是“Range”，基于渲染管线的状态定义
  - 通过Perf Markers来定义Range
  - 通过Shader Program来定义Range (支持所有6种Shader)
  - 通过Viewport来定义Range (适用于VR程序)
  - 通过Render Target来定义Range (显示那一组RTV，DSV使用时间最长)
  - 用户自定义的Range - 通过拖动来创建一个Range



瓶颈在哪儿?

- Range Profiler在Range Scrubber下面提供了3个面板以提供更加详细的信息
  - Range Info
    - Summary显示当前Range的基本信息 (图元, 像素, compute shader, 等等)
    - 右上方的组合框里还提供‘Action Details’, 允许你查看每个Drawdcall或者Dispatch call的信息
    - Range Profiler相较于以前的一项关键性改进是可以有选择的获取性能剖析所需要的数据. 这对于现代的图形程序十分有用, 因为Drawcall的个数已经从几千个增长到以万为单位!

Range Info - [Frame2983 (7442 µs)]

Action Details

Event	GPU Time	Draw Call Count	GPU Primitive Count	Shaded Pixels
684	96	1	2	49104
854	4640	1	16	32
857	128	1	16	16
861	0	1	16	80
864	0	1	16	80
868	0	1	16	16
885	3488	1	16	32

Event	GPU Time	Dispatch Call Count	Inst Executed
9232	240000	1	6.99606e+08
9243	240288	1	7.02002e+08

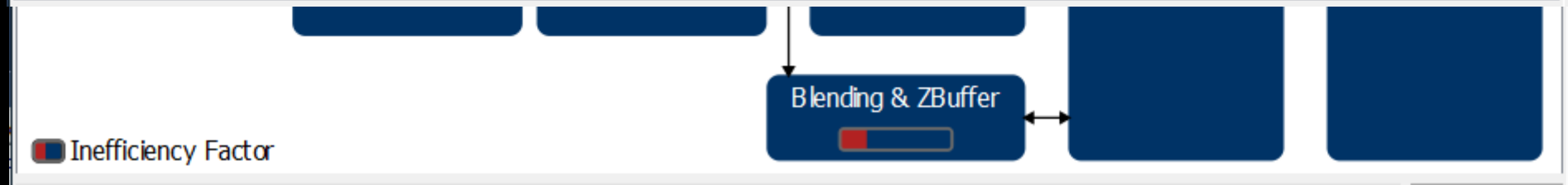
瓶颈在哪儿?

- Pipeline Overview
  - “Summary”, 显示效率最低的前4个stage
  - “Range Details”, 显示传统的管线示意图, 可以看到每个stage的瓶颈是多少
  - “Action Details”, 显示每个Drawcall对应管线单元的详细信息

Pipeline Overview

Action Details

Event	VS Cycle Ratio	TCS Cycle Ratio	TES Cycle Ratio	GS Cycle Ratio	PS Cycle Ratio	CS Cycle Ratio	Input Assembler Inefficiency	Shader Inefficiency	Texture Inefficiency	L
684	0	0	0	0	15	0	1	36.5	4	1
854	1	0	0	0	4	0	0	13	4	6
857	1	0	0	0	2	0	0	10	6	5
861	0	0	0	0	8	0	0	16	7	5
864	1	0	0	0	9	0	0	18	4	4
868	1	0	0	0	2	0	0	11	6	4
885	0	0	0	0	7	0	0	15	5	5



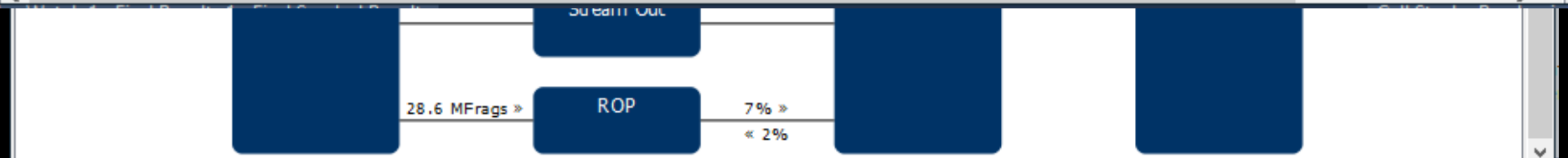
瓶颈在哪儿?

- Memory
  - “Summary”, 显示基本的memory使用统计信息, 例如L2, Framebuffer等等
  - “Range Details”, 显示管线里面每个使用memory的stage之间的request和bytes
  - “Action Details”, 显示对应每个Drawcall的更加详细的信息

Memory

Action Details

Event	L2->IA	Shader->Texture	Texture->Shader	Shader->ROP	Texture->L2	L2->Texture	Xfb->L2	ROP->L2	L2->ROP	L2->Memory	Memory->L2	L2 Hit Rate	L2 utilizat
684	0	0	0	0	0	0	0	3	0	3	1	91	1
854	0	0	0	32	0	0	0	0	0	0	0	49	0
857	0	0	0	16	0	0	0	0	0	0	1	56	0
861	0	0	0	80	0	0	0	0	0	0	0	64	0
864	0	0	0	80	0	0	0	0	0	0	0	32	0
868	0	0	0	16	0	0	0	0	0	0	0	23	0
885	0	0	0	32	0	0	0	0	0	0	0	48	0



瓶颈在哪儿?

- Range Profiler和Dynamic Shader Edit
  - 在Profiler视图中选取一个真正耗时的Drawcall, 创建一个User Range, 并选中
  - 转到API Inspector视图
  - 点击Shader的链接, 打开Shader源代码
  - 在Shader源码中点击右键, 选择Shader Edit, 对代码进行一些优化
  - 点击Profiler视图上的Lock按钮, 并点击Clone按钮, 打开一个新的Profiler视图
  - 在新的Profiler视图中重新运行profiler, 确认你的修改确实让性能得到提升

瓶颈在哪儿?

- Range Profiler
  - Scrubber和Ranges
  - 新的数据视图
- 可以一边优化一边重跑Profiler
  - 动态的编辑Shader, 对其进行优化
  - 重跑Profiler, 确认优化确实起作用了

Questions?

- Nsight支持论坛:
  - https://devtalk.nvidia.com/default/board/84/nsight-visual-studio-edition/
- 开发工具支持邮箱:
  - devtools-support@nvidia.com
- 欢迎进行询问，讨论和建议

谢谢！☺