DEVELOPER TOOLS FOR MOBILE PLATFORMS

Jonas Yang, Manager of Mobile Graphics Tools

AGENDA

Introducing NVIDIA® Tegra Graphics Debugger 2.0

Profiling UE4 SumTemple Demo

Introducing Tegra System Profiler

NVIDIA DEVELOPER TOOLS

BUILD. DEBUG. PROFILE.

Microsoft DirectX | OpenGL | NVIDIA CUDA | OpenGL ES | Android NDK | GNU C/C++

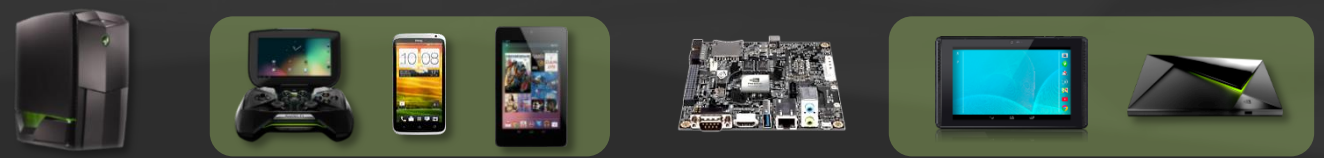
IDE INTEGRATION

Visual Studio

eclipse

STANDALONE TOOLS

HARDWARE SUPPORT
CPU AND GPU DEBUGGING & PROFILING

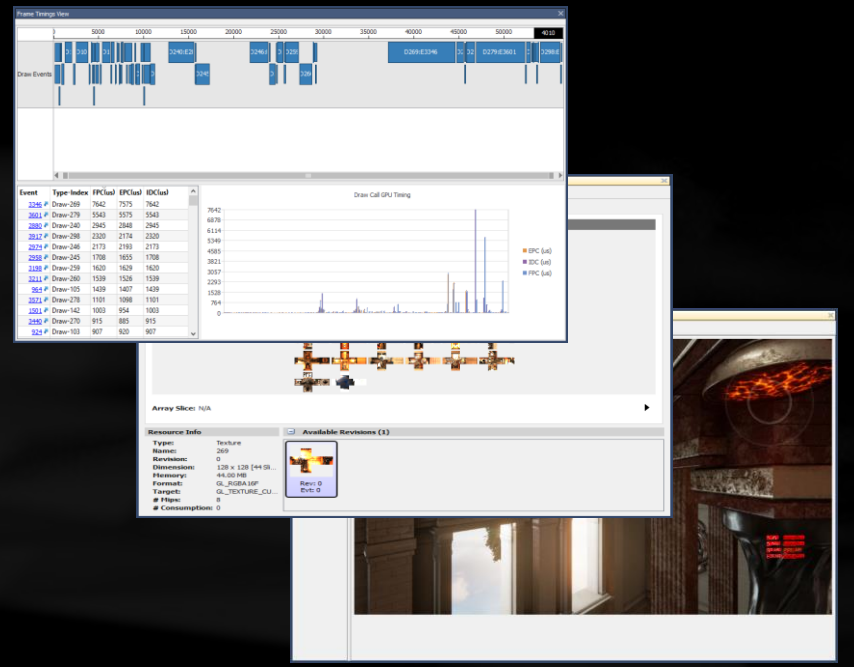


NVIDIA GAMEWORKS™

NVIDIA® TEGRA GRAPHICS DEBUGGER

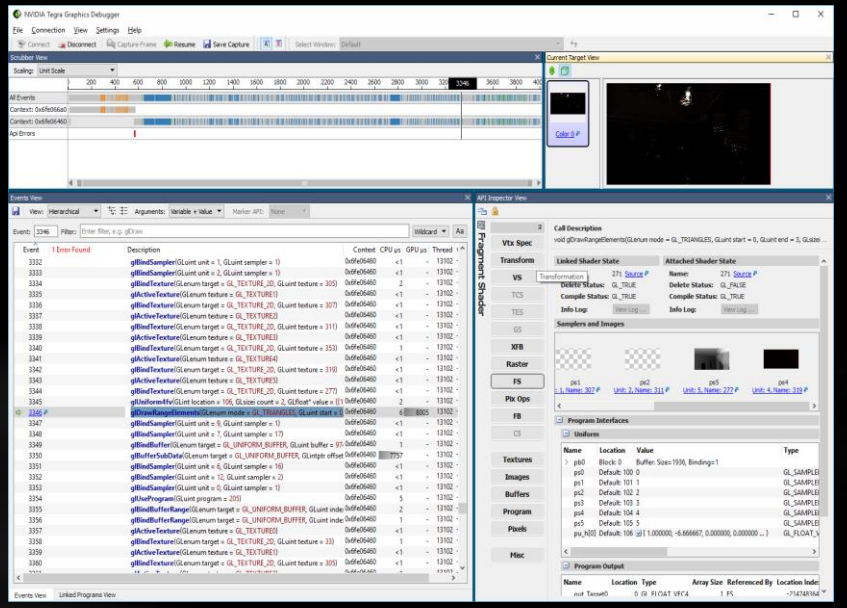
Accelerating Visual Computing Development on *Mobile Platforms*

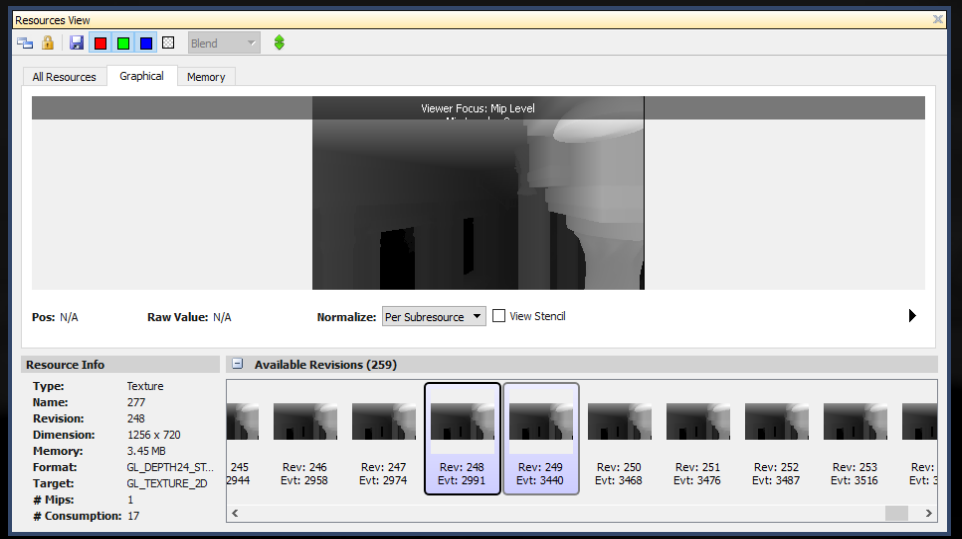
- Supports cutting edge OpenGL
- Debug and profile graphics workloads
- Live capture of a single rendering frame
- Automatic GPU bottleneck analysis
- Advanced timings for draw calls
- Standalone tool on Windows/Mac/Linux



NEW IN VERSION 2.0

- Supports OpenGL ES 3.1 and OpenGL 4.5
- Support Android 5.0 and Android TV
- Support arm 32bit/64bit Android
- Serialize captured frame
- Target HUD





PROFILING UE4 SUNTEMPLE DEMO

Connect to Application

Target: SHIELD Tablet - 1740CA1091000000009060340

Launch APK | Attach

Search for an APK...

Remove Debug

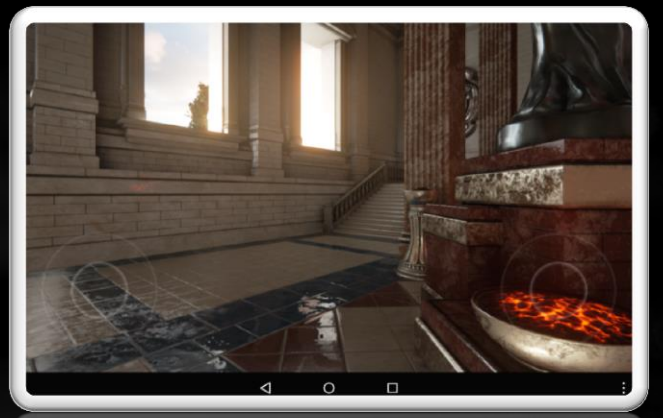
Package Name	Package Location
com.nvidia.ParticleUpsampling	/data/app/com.nvidia.ParticleUpsampling-1/base.apk
com.nvidia.UnityWindmill2ES3001	/data/app/com.nvidia.UnityWindmill2ES3001-1/base.apk
com.nvidia.demo	/data/app/com.nvidia.demo-1/base.apk
com.nvidia.main__2015_07_27__23_26_33	/data/app/com.nvidia.main__2015_07_27__23_26_33-2/base.apk
com.nvidia.main__2015_07_28__15_17_24	/data/app/com.nvidia.main__2015_07_28__15_17_24-1/base.apk
com.nvidia.tegraprofiler.tools	/data/app/com.nvidia.tegraprofiler.tools-1/base.apk
com.ru.golf	/data/app/com.ru.golf-1/base.apk
com.ue4.SunTemple47	/data/app/com.ue4.SunTemple47-1/base.apk
com.vectorunit.purple.googleplay	/data/app/com.vectorunit.purple.googleplay-1/base.apk
com.walnut.striveNvidia	/data/app/com.walnut.striveNvidia-1/base.apk
net.kishonti.compubench.gl.v10500.corporate	/data/app/net.kishonti.compubench.gl.v10500.corporate-1/base.apk

Status:

```
Device Product Model: SHIELD Tablet
Making sure adbd has root permissions
Waiting for the device to become available.
Mounting device for read/write.
Detected current device is running Android L
Setting SELinux policy to permissive.
SELinux policy successfully changed to permissive.
Located 32-bit driver libraries in /vendor/lib
Host supports 32-bit graphics debugging
Detected NVIDIA tegra device drivers
Graphics debugger version is correct.
Searching for packages installed on the current device.
Scanning for attachable processes on SHIELD Tablet - 1740CA1091000000009060340...
Scan for devices complete.
Process scan complete.
```

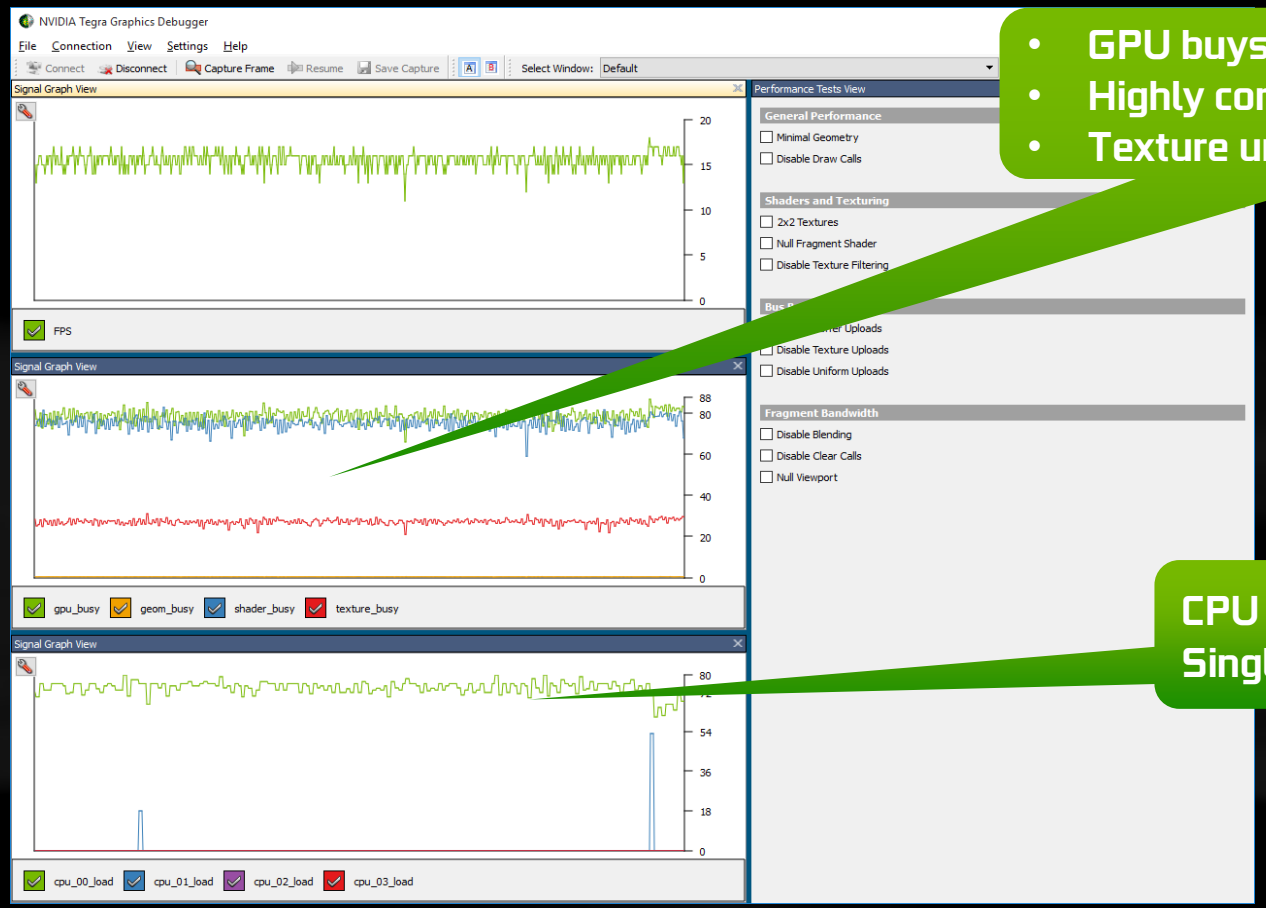
ADB: F:/NVPACK/android-sdk-windows/platform-tools/adb.exe Browse... Cancel **Launch**

APK Name



Launch!

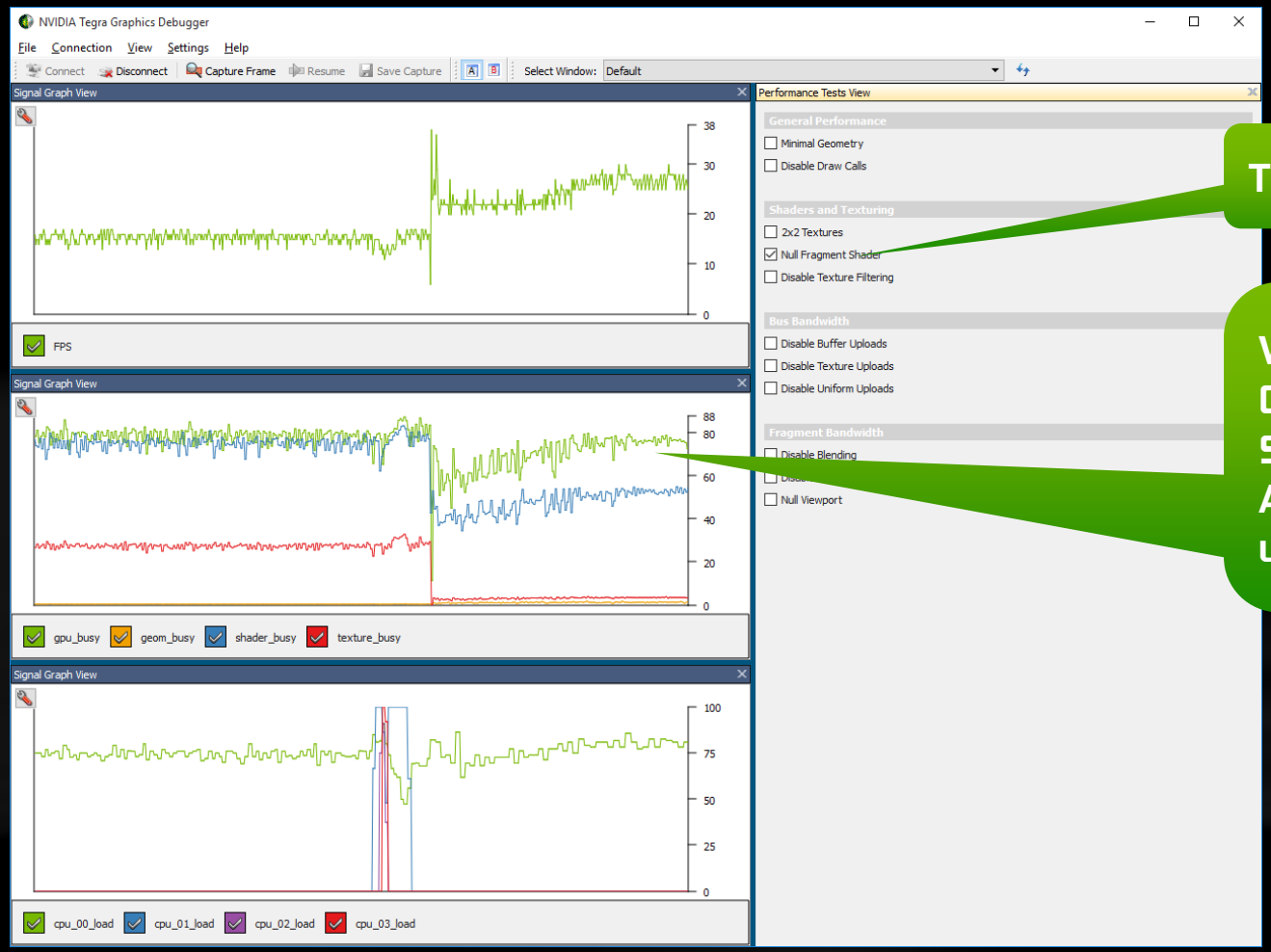
PROFILING UE4 DEMO



- GPU buys is 80%
- Highly correlated to shader unit busy
- Texture unit is not so busy 30%

CPU is 80% loaded
Single core is utilized

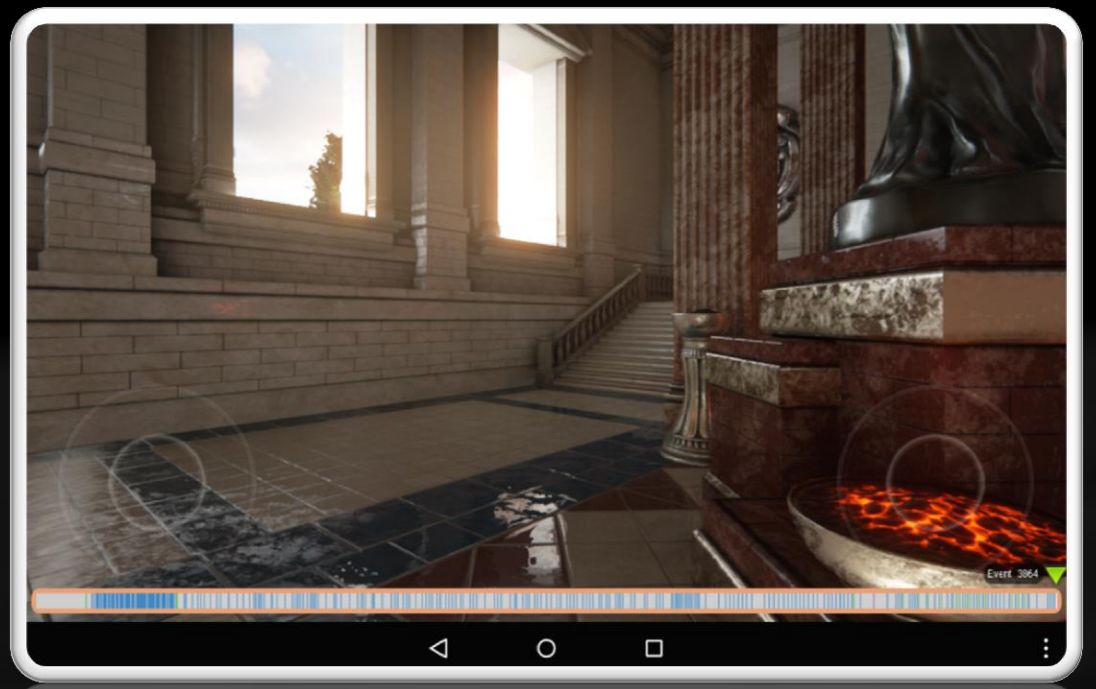
QUICK EXPERIMENTS

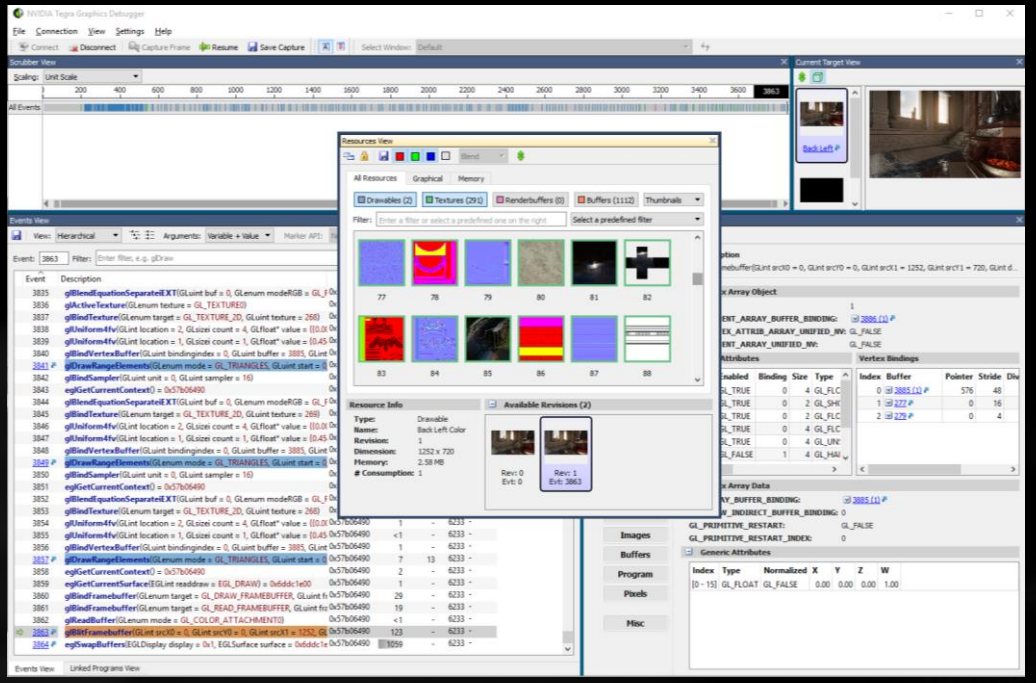


Toggle Null Fragment Shader

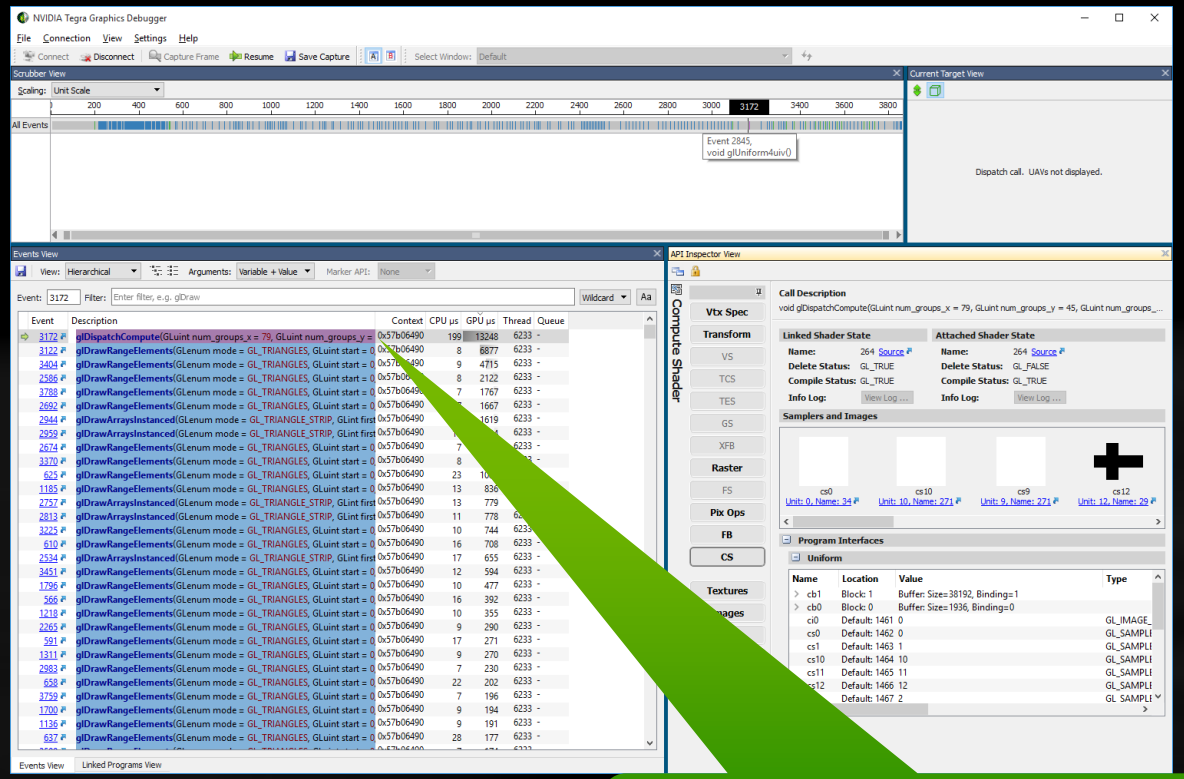
What?
GPU is still busy?
Shader unit is still busy?
Answer: Compute shader is used!!!

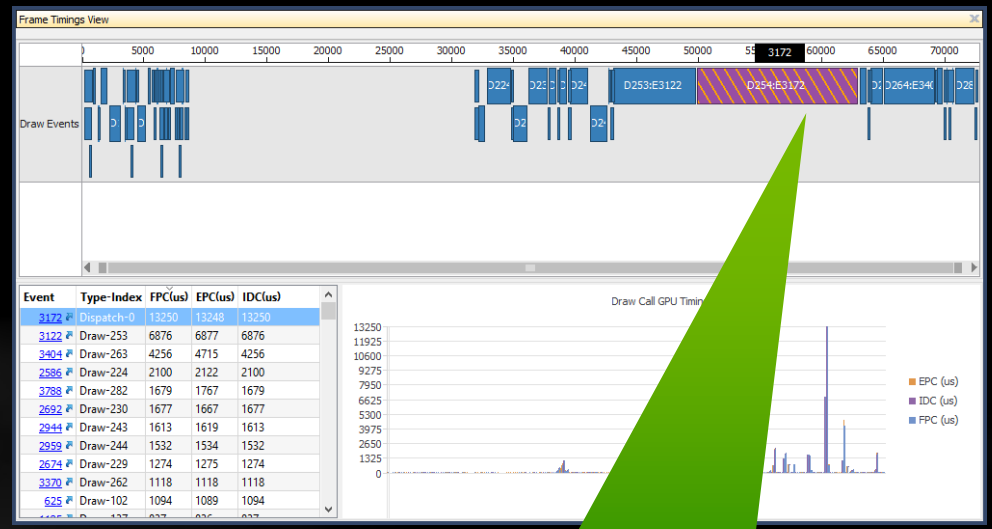
CAPTURE A FRAME





INSPECT EXPENSIVE APIS

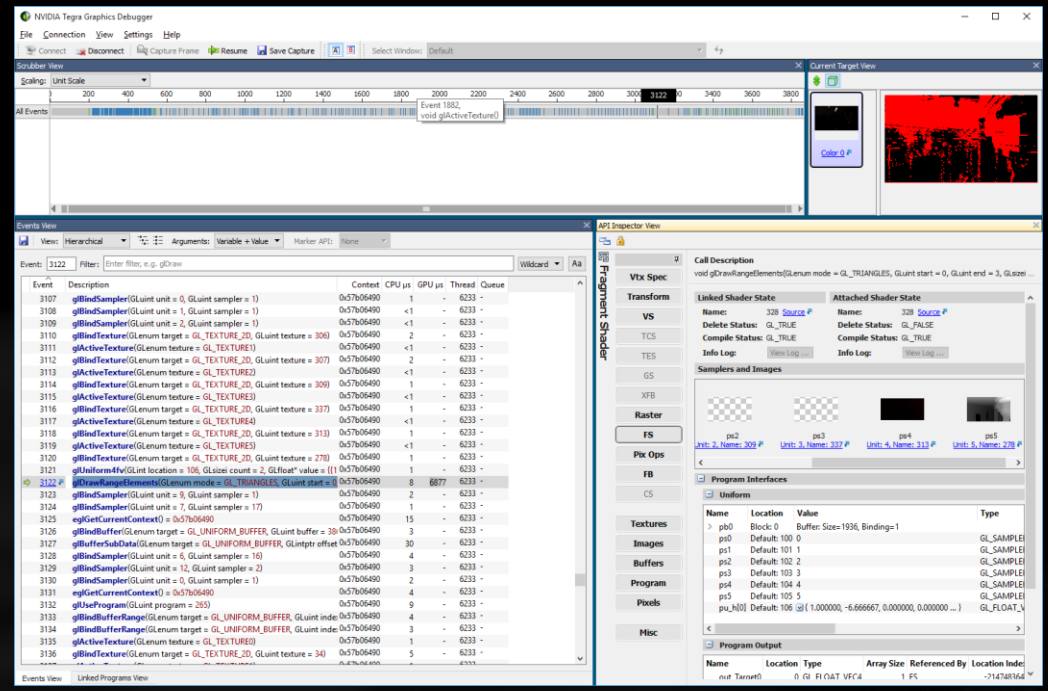




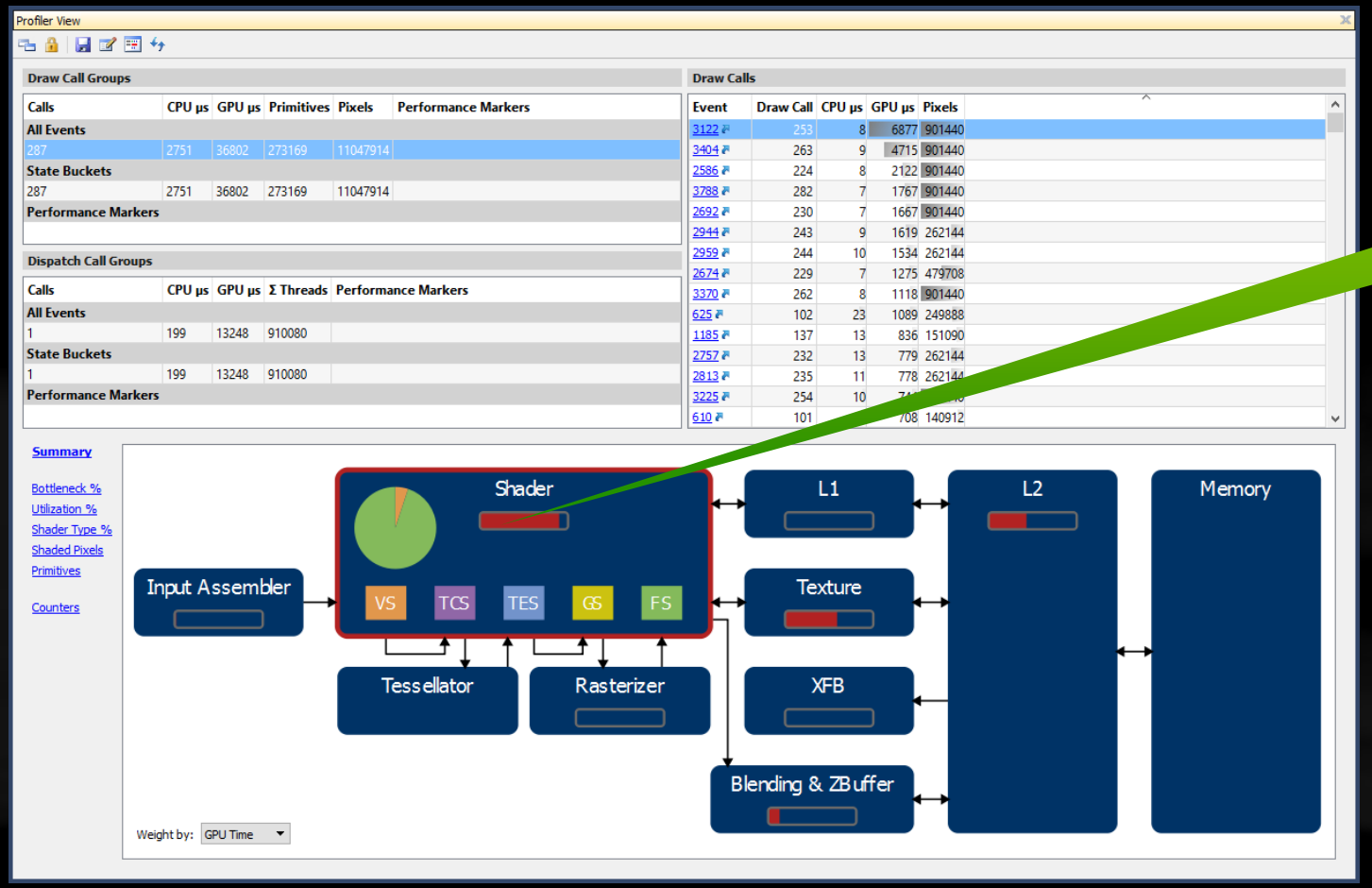
No parallel GPU workload at all!

Most expensive API is compute shading!

INSPECT EXPENSIVE APIS

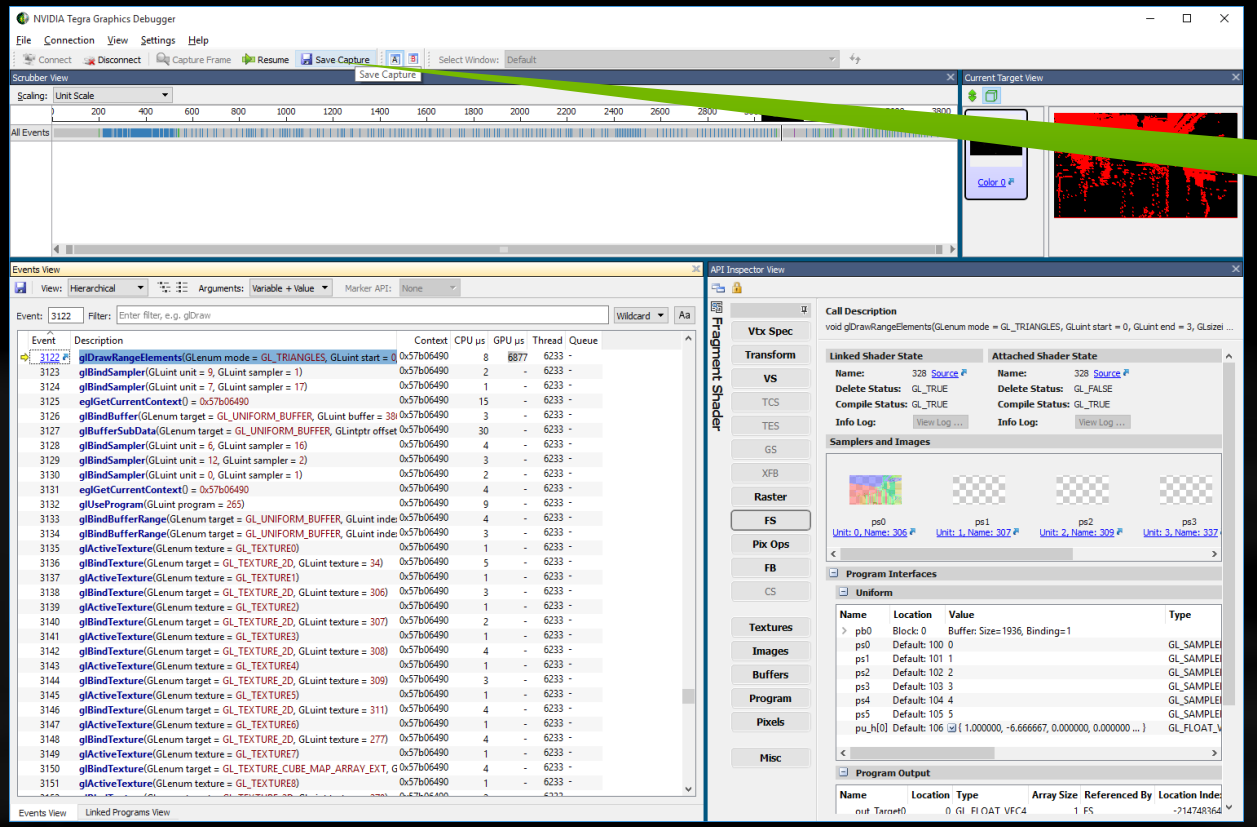


RUN PROFILER



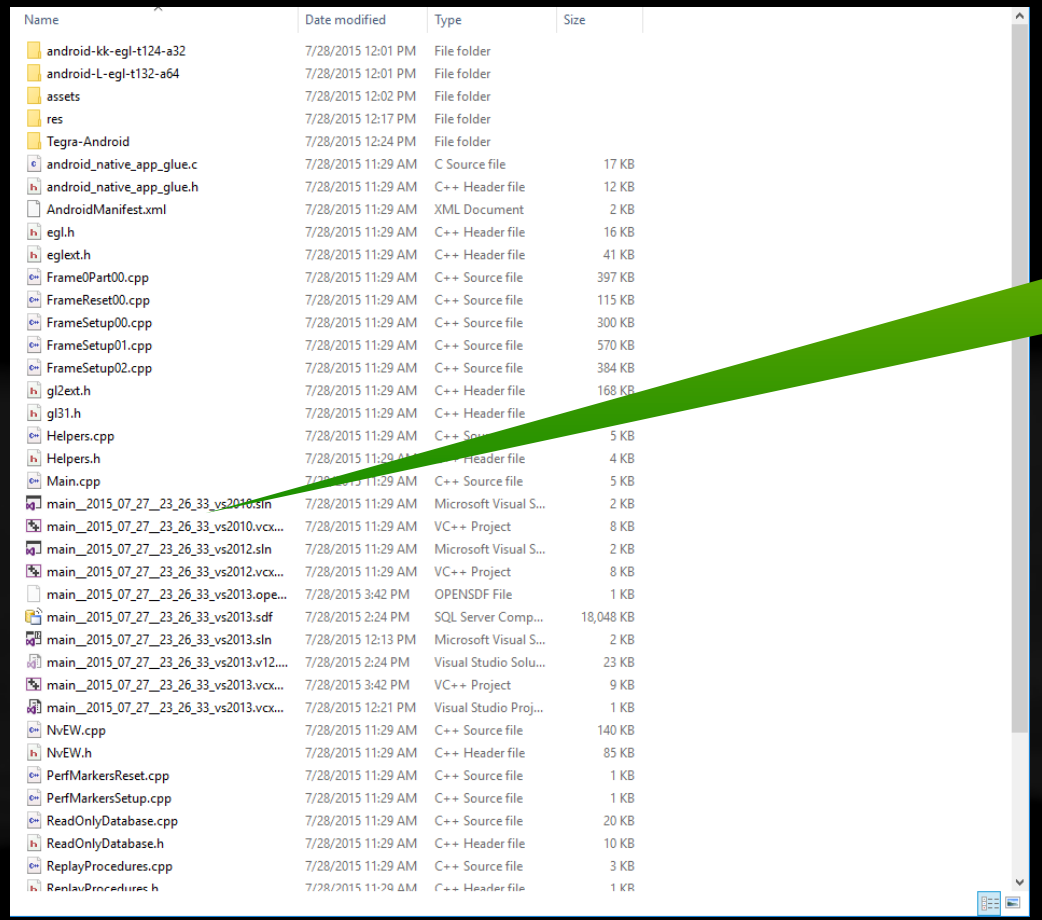
Almost all shader bottleneck

SERIALIZE A FRAME



Almost all shader bottleneck

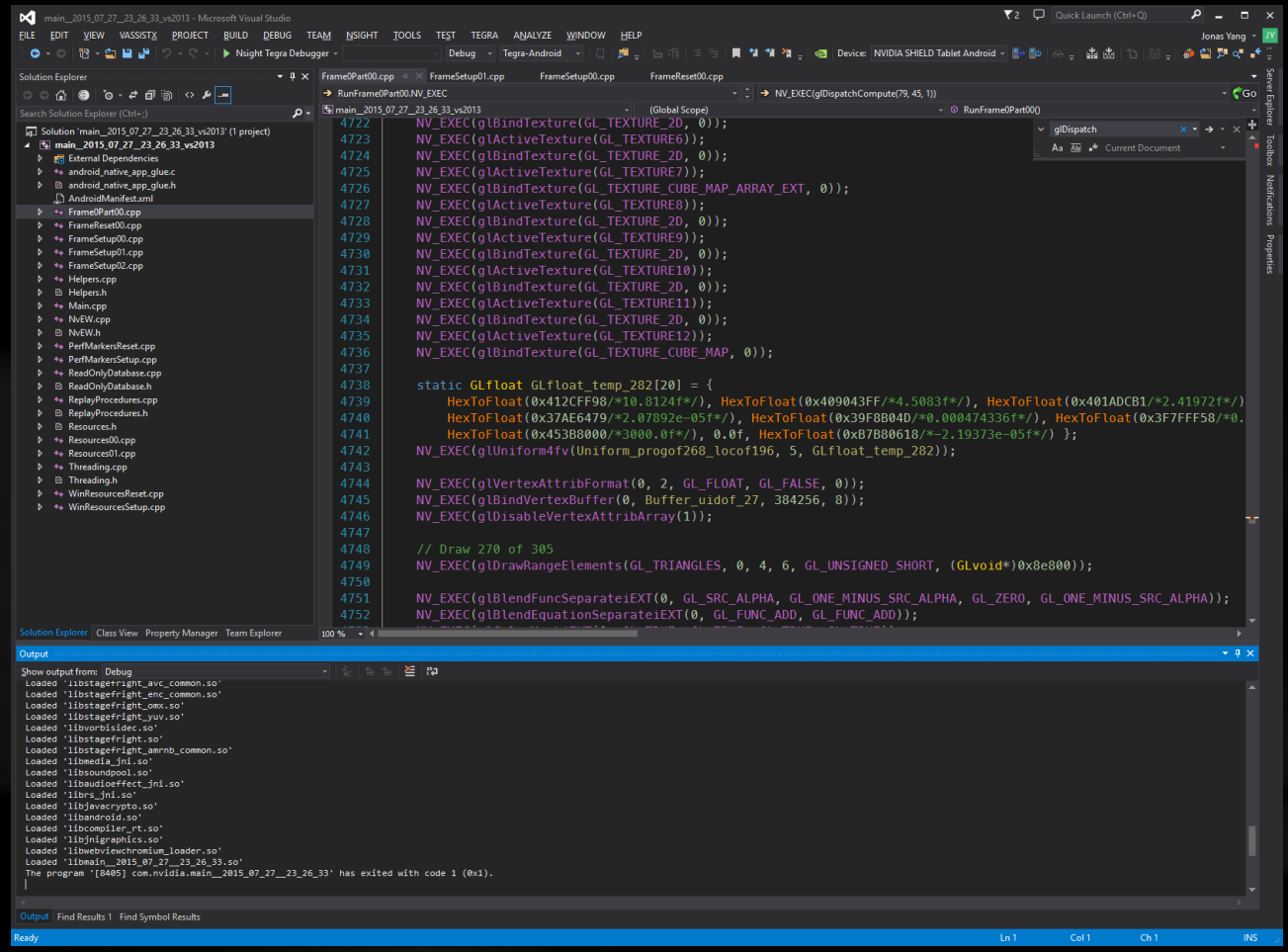
OPEN THE SOLUTION FILE



Name	Date modified	Type	Size
android-kk-egl-t124-a32	7/28/2015 12:01 PM	File folder	
android-L-egl-t132-a64	7/28/2015 12:01 PM	File folder	
assets	7/28/2015 12:02 PM	File folder	
res	7/28/2015 12:17 PM	File folder	
Tegra-Android	7/28/2015 12:24 PM	File folder	
android_native_app_glue.c	7/28/2015 11:29 AM	C Source file	17 KB
android_native_app_glue.h	7/28/2015 11:29 AM	C++ Header file	12 KB
AndroidManifest.xml	7/28/2015 11:29 AM	XML Document	2 KB
egl.h	7/28/2015 11:29 AM	C++ Header file	16 KB
eglext.h	7/28/2015 11:29 AM	C++ Header file	41 KB
Frame0Part00.cpp	7/28/2015 11:29 AM	C++ Source file	397 KB
FrameReset00.cpp	7/28/2015 11:29 AM	C++ Source file	115 KB
FrameSetup00.cpp	7/28/2015 11:29 AM	C++ Source file	300 KB
FrameSetup01.cpp	7/28/2015 11:29 AM	C++ Source file	570 KB
FrameSetup02.cpp	7/28/2015 11:29 AM	C++ Source file	384 KB
gl2ext.h	7/28/2015 11:29 AM	C++ Header file	168 KB
gl31.h	7/28/2015 11:29 AM	C++ Header file	
Helpers.cpp	7/28/2015 11:29 AM	C++ Sou...	5 KB
Helpers.h	7/28/2015 11:29 AM	... Header file	4 KB
Main.cpp	7/28/2015 11:29 AM	C++ Source file	5 KB
main__2015_07_27__23_26_33_vs2010.sln	7/28/2015 11:29 AM	Microsoft Visual S...	2 KB
main__2015_07_27__23_26_33_vs2010.vcx...	7/28/2015 11:29 AM	VC++ Project	8 KB
main__2015_07_27__23_26_33_vs2012.sln	7/28/2015 11:29 AM	Microsoft Visual S...	2 KB
main__2015_07_27__23_26_33_vs2012.vcx...	7/28/2015 11:29 AM	VC++ Project	8 KB
main__2015_07_27__23_26_33_vs2013.ope...	7/28/2015 3:42 PM	OPENSDF File	1 KB
main__2015_07_27__23_26_33_vs2013.sdf	7/28/2015 2:24 PM	SQL Server Comp...	18,048 KB
main__2015_07_27__23_26_33_vs2013.sln	7/28/2015 12:13 PM	Microsoft Visual S...	2 KB
main__2015_07_27__23_26_33_vs2013.v12....	7/28/2015 2:24 PM	Visual Studio Solu...	23 KB
main__2015_07_27__23_26_33_vs2013.vcx...	7/28/2015 3:42 PM	VC++ Project	9 KB
main__2015_07_27__23_26_33_vs2013.vcx...	7/28/2015 12:21 PM	Visual Studio Proj...	1 KB
NvEW.cpp	7/28/2015 11:29 AM	C++ Source file	140 KB
NvEW.h	7/28/2015 11:29 AM	C++ Header file	85 KB
PerfMarkersReset.cpp	7/28/2015 11:29 AM	C++ Source file	1 KB
PerfMarkersSetup.cpp	7/28/2015 11:29 AM	C++ Source file	1 KB
ReadOnlyDatabase.cpp	7/28/2015 11:29 AM	C++ Source file	20 KB
ReadOnlyDatabase.h	7/28/2015 11:29 AM	C++ Header file	10 KB
ReplayProcedures.cpp	7/28/2015 11:29 AM	C++ Source file	3 KB
ReplayProcedures.h	7/28/2015 11:29 AM	C++ Header file	1 KB

VS2010
VS2012
VS2013
Make sure you installed Nsight Tegra!

COMPILE AND RUN



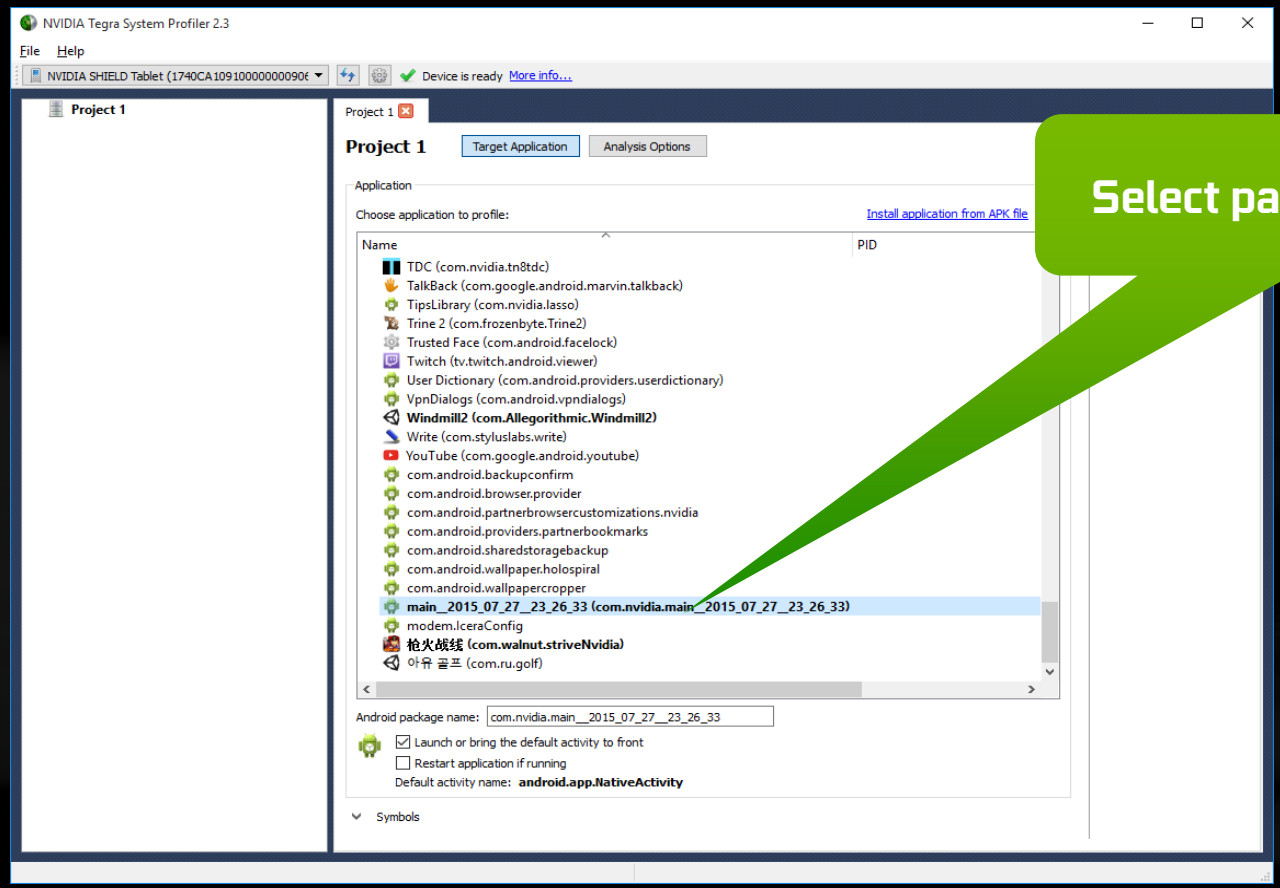
BENEFITS WITH GENERATED SOURCE

- You are denominating everything from bottom to top
  - Modify source and re-compile.
  - All CPU overhead is removed.
  - Add/remove/modify draw calls.
  - Change rendering state on the air.

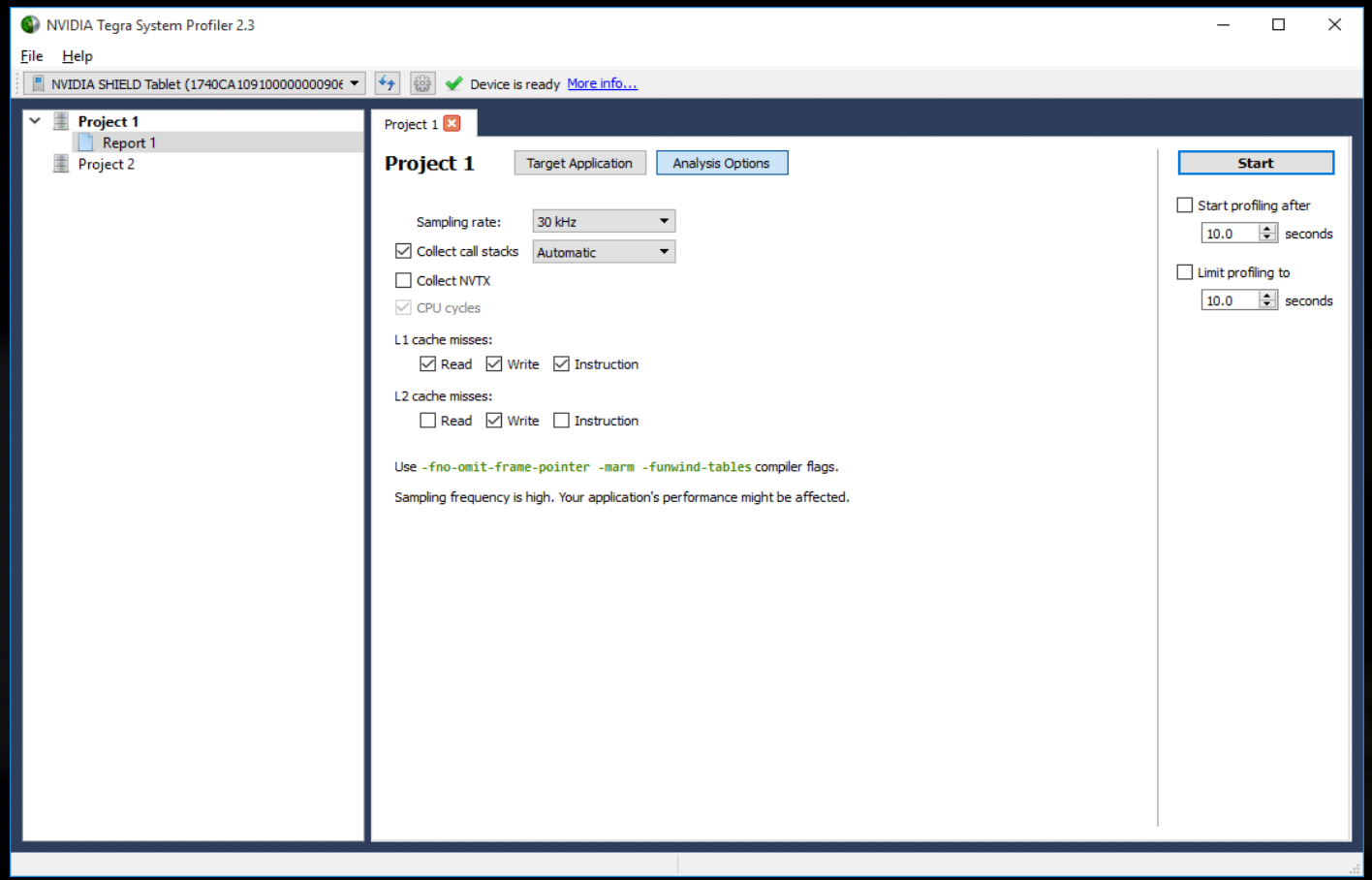
TEGRA SYSTEM PROFILER

- Identify critical functions by sample count
- Call stack analysis
- Standalone tools on Windows/Linux/Mac.
- Support arm 32bit/64bit android

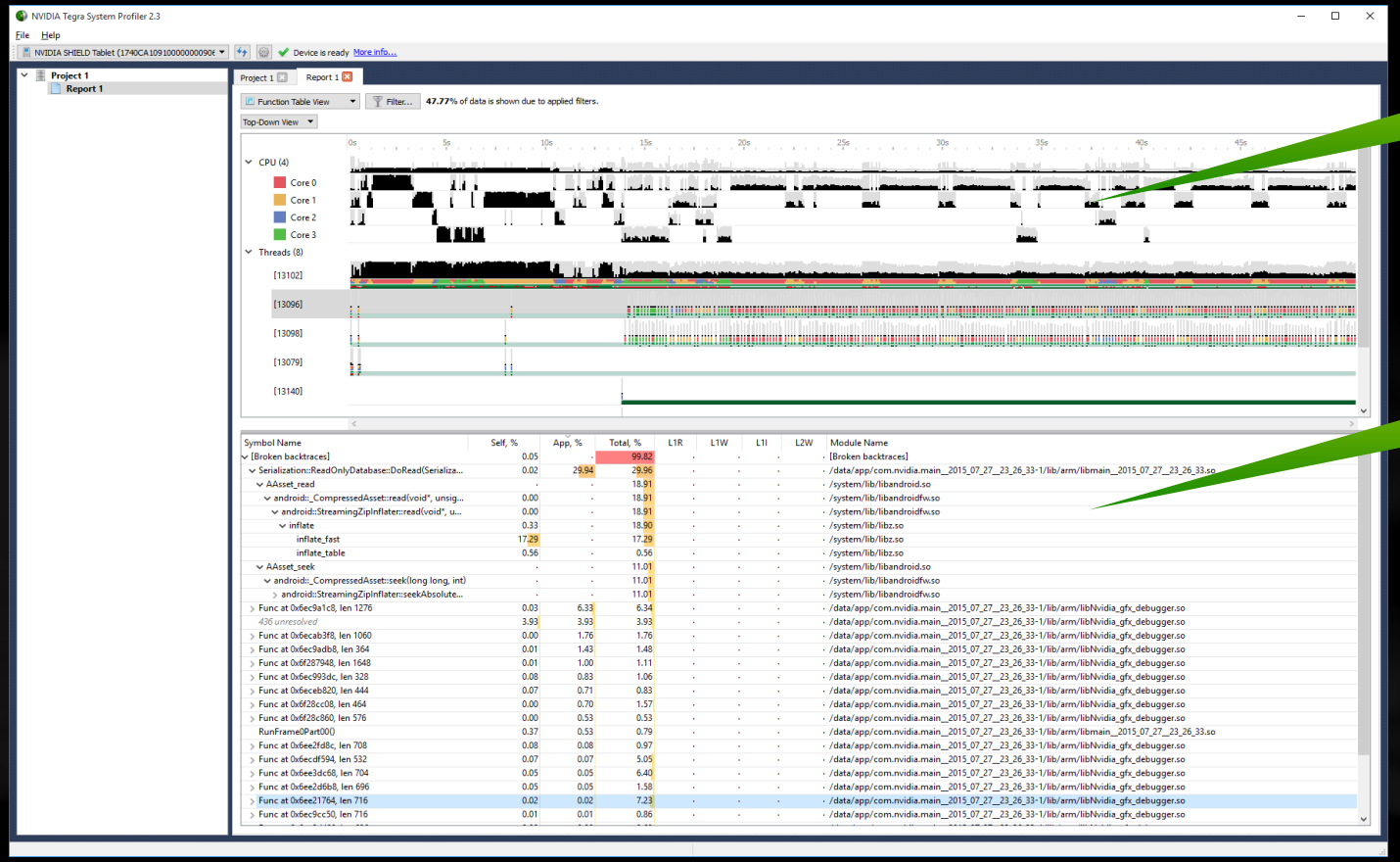
TEGRA SYSTEM PROFILER



TEGRA SYSTEM PROFILER



TEGRA SYSTEM PROFILER



CPU/Threads activity view

Function level profiling info

NVIDIA GAMEWORKS™

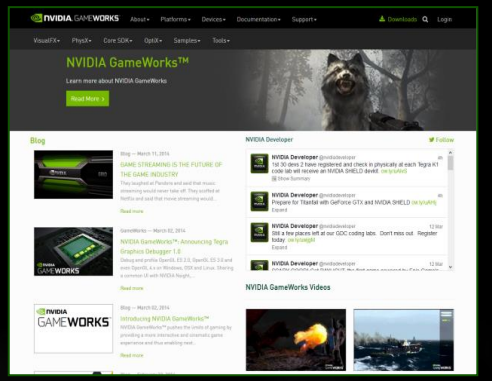
Links…

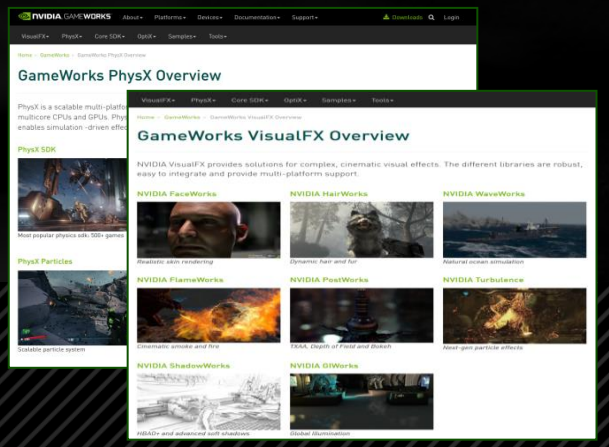
Web: http://gameworks.nvidia.com
Latest info and download options

YouTube: https://www.youtube.com/user/nvidiaGameWorks
Tools, effects, and game integration videos

Twitter: https://twitter.com/nvidiadeveloper
Catch up on up to the minute happenings

Survey: https://developer.nvidia.com/developer-tools-survey-201503





THANK YOU