



中国游戏开发者大会

CHINA GAME DEVELOPERS CONFERENCE

2014

议程

- 感谢Croteam (Dean Sekulic, Alen Ladavac, etc.)
- Nvidia Developer Tools 和 Nsight总览
- Nsight 帧调试工具
- GPU 优化工具



编译...调试...优化...

Microsoft
DirectX

OpenGL

nVIDIA
CUDA

OpenGL ES

ANDROID
NDK

nVIDIA
GAMEWORKS™

开发工具

ID集成

Visual Studio

独立工具



硬件支持
CPU与GPU调试/优化



Nsight, Visual Studio 版本 桌面系统上的图形调试、优化与 系统跟踪工具

优势

调试

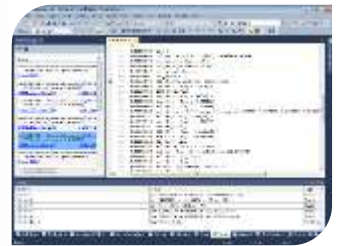
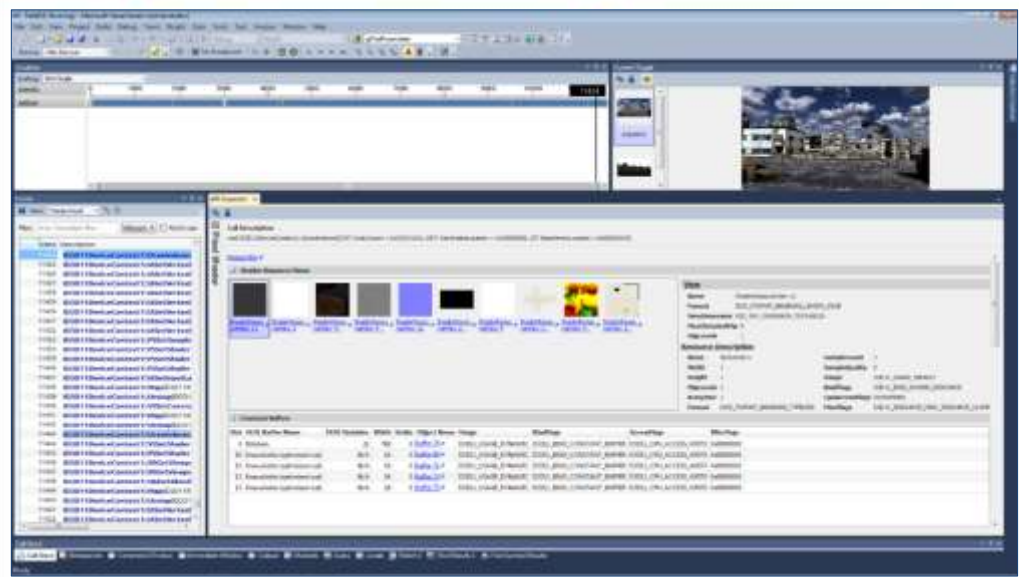
调试draw call及相关的状态和资源
帧捕捉与回放（生成D3D9/11源码）
直接在游戏画面上拖动和侦测draw call
硬件GPU shader调试和GPU内存视图
硬件单GPU shader调试，支持复杂条件断点
像素绘制历史

优化

自动GPU性能瓶颈分析
先进的时间线系统，用于图形和通用计算的时序显示
D3D性能特征和渲染状态组群或通过CPU/GPU对支持OpenGL和
Direct3D的系统跟踪活动进行分类。
相关的进程，调用堆栈，API调用和CPU工作流程
系统跟踪支持并发执行的draw call和追踪内存传输

APIS

Direct3D 11/11.1/9 , OpenGL 4.2+



“ *Once you get hooked on Nsight it's hard to live without it!*
— Dan Ginsburg | President, Upsample Software, LLC

”

实例1：总览



实例2: CPU与GPU分析



练习 2: 总结

- 使用性能分析工具查看GPU/CPU的总体工作流程
- 在总结视图中查看总体的使用率
- 时间线显示
 - 线程使用情况
 - GPU使用率
 - 工作负载如何在处理器间流动
- 通过图表查看API的调用
- 调用堆栈追踪



实例3：我的瓶颈在哪里？



实例 5：总结

- 一旦确认是GPU瓶颈，使用帧分析器来查找瓶颈所在
- 使用帧时间线来进行快速分析
- 运行性能分析器得到更深层次的瓶颈信息
- 对Shader做出修改后，锁定并重新运行
- Shader编辑和A/B切换可以用来检测新的渲染错误和性能提升

let me share
2014