Developing and Porting PC Level Graphics Games for Android TV

Rev Lebaredian, NVIDIA Senior Director of Engineer
This Talk

- What is necessary to port a AAA PC/Console title to SHIELD?
  - Functionality
  - Performance

- We want SHIELD/Android to be your easiest porting platform!
Borderlands 2 and Pre-Sequel

Images from borderlands2.com and borderlandsthegame.com
Borderlands 2 and Pre-Sequel

- Developed by Gearbox, 2K Games, 2K Australia
- Borderlands 2 (BL2), released September 2012
- Borderlands: The Pre-Sequel! (TPS), released October 2014
- Based on modified Unreal Engine 3 (June 2011 build)
- Originally released on Windows, X360, PS3, Linux, Mac
- PC renderer based on D3D9
  - OpenGL support added later for Linux port (TPS had it at launch)
NVIDIA SHIELD
Unreal Engine 3 and Android

- Existing UE3 support for Android...
  - Base engine, kind of
  - Libraries
  - ES2 mobile renderer
  - Some sample apps

- What we can re-use...

- What we want to add/replace...
Libraries

- The newer your engine build, the better

- For us (June 2011)
  - Google protobuf
  - AkAudio/Wwise
  - PhysX
  - Bink
  - FaceFX
  - Scaleform!!!

- Integration varies in difficulty
Our building blocks
- D3D9 Renderer
- OpenGL 3.2 PC Renderer
- OpenGL ES2 Renderer

Addition: OpenGL ES3.1 backend
PC OpenGL Functionality

- Already have OpenGL support? Good!

- Otherwise...
OpenGL Correctness

- Inversions! Artifacts!
- Or worst of all...nothing...

- Stock UE3 effects
- Custom effects

- Common fixes
  - Texture space inversions \((t = 1.0 - t)\)
    - Static versus dynamically generated versus render buffers
  - Clip space Z difference \((z = 2z - 1.0)\)
  - NaN/inf behavior (abs/epsilon)

- NSight OpenGL will help you out here
More Bug Examples
Android OpenGL Functionality

- UE3’s existing Android support is limited
  - Special case ES2 mobile renderer

- Our solution: convert everything over to desktop GL codepath
  - New RHI GL backend using EGL that supports both GL 3.2 and ES 3.1
Debugging Android

- Does PC D3D work?
  - YES
  - Does PC GL match?
    - YES
    - Does Android GL match?
      - YES
      - Does Android ES match?
        - YES
        - Success!
      - NO
      - Android problem
    - NO
    - GL render problem
  - NO
  - Bug in PC version
Debug View (D3D)
Debug View (PC GL)
Debug View (SHIELD)
Checkpoinnt

- At this point, we have a (mostly) functionally complete, playable game
- Framerate may vary
- Next step is to boost performance to make it more enjoyable
Initial Performance Notes

- A mix of CPU and GPU bound cases in the render thread
- More hitching on Android versus PC

Example scene: main menu
- 15-30FPS, GPU bound on TX1 1080p

Example scene: first level
- 26-28FPS, CPU/GPU bound on TX1 810p
- Null GPU runs at 40FPS

Game thread 10-12ms on TX1 during play
State of the UE3 GL RHI

- OpenGL 3.2, mostly core
- Pros
  - Feature parity with D3D9
  - Ease of use
  - API idiosyncrasies
- Cons
  - Performance and efficiency

- There are many known ways to increase GL API efficiency
  - See “Approaching Zero Driver Overhead” from GDC 2014
OpenGL API Optimizations

- First level test scene
  - 10K+ events reported by debugger
  - 1636 draw calls
- MultiDrawIndirect
  - 1494 draw calls (-8% from start)
- Ongoing work...
- One large UBO indexed by vertex
  - 809 draw calls (-50%)
  - Fewer uniform-related binds
- Indexed bindless textures
  - 502 draw calls (-69%)
GPU Performance

- Ongoing work on GPU bottlenecks

- Possible techniques
  - Z-prepass
  - Post-process quality
  - Anti-aliasing options
  - Resolutions
  - Shader optimization

- Depends on specific needs
Miscellaneous Notes and Pitfalls

- Debug context
- BGRA support on OpenGL ES
- PF_V8U8 bump maps
- ARM NEON support
- Array cookie size
- Memory alignment
Online Functionality

- Adjusted for Android
- Multiplayer
- Cloud saves, achievements
- DLC/add-on content
- Storefront integration
Borderlands: TPS Footage
End