OpenCL BOF
Siggraph 2012
Neil Trevett, NVIDIA
OpenCL Chair
Welcome!

- AMD - Mike Houston
  - OpenCL 1.2 Intro
  - OpenCL and AMD

- Intel - Adam Lake
  - CLU – OpenCL Utility Library
  - Intel and OpenCL

- Adobe - Eric Berdahl, Dave McGavran and Sarah Kong
  - Using OpenCL

- Q&A
OpenCL Roadmap

**OpenCL-HLM** (High Level Model)
Exploring high-level programming model, unifying host and device execution environments through language syntax for increased usability and broader optimization opportunities

**OpenCL-SPIR** (Standard Parallel Intermediate Representation)
Exploring LLVM-based, low-level Intermediate Representation for code obfuscation/security and to provide target back-end for alternative high-level languages

**Long-term Core Roadmap**
Exploring enhanced memory and execution model flexibility to catalyze and expose emerging hardware capabilities

OpenCL-SPIR Provisional Specification to be released soon for feedback from LLVM community
Expanding Platform Reach for Graphics and Computation

Desktop
- OpenGL
- Interop
- OpenCL
- Full Profile

Mobile
- OpenGL ES
- Interop
- OpenCL
- Full Profile and Embedded Profile

Web
- WebGL
- Interop
- Typed Arrays?

WebGL on majority of production desktops now. WebGL pervasively available on mobile in next 18 months.

WebCL will start deploying in next 18 months.

OpenCL pervasively available on mobile in next 18 months.
OpenCL and OpenGL Compute Shaders

- OpenGL compute shaders and OpenCL support distinctly different use cases
  - OpenCL provides a significantly more powerful and complete compute solution

Enhanced 3D Graphics apps “Shaders++”
1. Fine grain compute operations inside OpenGL
2. GLSL Shading Language
3. Execute on single GPU only

Imaging Video Physics AI
Pure compute apps touching no pixels
1. Full ANSI C programming of heterogeneous CPUs and GPUs
2. Utilize multiple processors
3. Coarse grain, buffer-level interop with OpenGL
Over to Mike..