Advanced OpenCL Debugging and Profiling

Avi Shapira
Graphic Remedy

Benedict R. Gaster
AMD
Open Physics - OpenCL

- Soft body dynamics
  - AMD developing DirectCompute/OpenCL Cloth acceleration

- Fluid simulation
  - Does not currently exist in Bullet
  - AMD developing OpenCL/DirectCompute SPH implementation

- DMM Finite Element method
  - AMD/Pixelux demonstrated Bullet integration, next step OpenCL

- Rigid body dynamics
  - Erwin Coumans and Sony team developing accelerated versions
Fluids and particle systems basics

- Simply, highly parallel, thus map well to the GPU
- Particles store position, mass, velocity, age, density, etc
- Particles are moved by time stepping:
  - Euler or Leapfrog integration \( \frac{dv_i}{dt} = a_i \)

- Acceleration \( a_i \) has contributions from gravity, pressure gradient, and viscosity
SPH in action
SPH – high level development breakdown

- Design algorithm and structure of implementation
- First pass implementation
- Debug for correctness
  - SPH – printf and gDEBugger (using GL rendering)
  - DX rendering AMD’s GPU Perf Studio
- Debug for performance
  - AMD performance counters and gDEBugger
SPH – Algorithm

• **Build spatial grid on particles**
  - Allow fast neighbor finding

• **For each particle**
  - Find neighbors

• **For each particle**
  - Compute density and pressure

• **For each particle**
  - Compute acceleration

• **For each particle**
  - Intergrate

On the GPU each particle is worked on in parallel

Code Breakdown

• 14 global Buffers
• 9 Kernels
• 17 functions (i.e. not entry points)
• ~1,500 lines of OpenCL C
Debugging and Profiling OpenCL

- Debugging and profiling parallel computing applications are both hard and time consuming tasks
- Companies find it extremely hard to deliver robust and bug-free parallel computing applications
- It is almost impossible to optimize a parallel computing application to fully utilize the system’s resources
• OpenGL, OpenGL ES and OpenCL Debugger, Profiler and Memory Analyzer

• Exposes the internal system information needed to find bugs, optimize graphic and parallel computing performance and memory usage
Acknowledgements and contact info

- DX rendering
  - Saif Ali and Justin Hensley
- SPH Simulation
  - Alan Heirich

www.amd.com
benedict.gaster@amd.com

www.gremedy.com
avi@gremedy.com