Lecture 22: Xbox 360 Debugging & Monitoring

Prof. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Assembly Information

Setting a breakpoint

Or use System.Diagnostics.Debugger.Break();
Snapshot after each “continue”

Remote Performance Monitor

- Basic garbage collector information
- Can tell if you have a GC problem
- Not usually enough to diagnose the cause
- Shows the number of system calls
- Not much help with computational bottlenecks

Launching from the Performance Monitor

Peek at variables on the Xbox 360
“Lies, damn lies, and statistics”

• See Shawn Hargreaves, “Understanding XNA Framework Performance” PowerPoint presentation for tips on interpreting numbers
  – Garbage collection issues
  – Kernel/User mode switches

Publish to PerfMon
Finding PerfMon

Adding a counter

PerfMoning Spacewar
Profiling on Windows

- Inference to the rescue!
  - The XNA Framework is similar on both platforms
  - Measure your game on Windows
  - Results usually also apply to Xbox 360

- There are many great Windows perf tools
  - The CLR Profiler for garbage collection issues
  - Sampling profilers: Visual Studio Team System, ANTS, NProf, OptimizeIt, VTune
  - PIX

Slide from Shawn Hargreaves, "Understanding XNA Framework Performance"

Re-entering the Framework

- GraphicsDevice is somewhat thread-safe
  - Cannot render from more than one thread at a time
  - Can create resources and SetData while another thread renders

- ContentManager is not thread-safe
  - Ok to have multiple instances, but only one per thread

- Input is not threadable
  - Windows games must read input on the main game thread

- Audio and networking are thread-safe

Slide from Shawn Hargreaves, "Understanding XNA Framework Performance"