

Making the Tiger Woods Golf Engine Truly Multi-Threaded

Who I Am

- Jose Caban, Software Engineer
 - GeorgiaTech, Fall 2002 – Spring 2007
 - B.S. in Computer Science, B.S. in Mechanical Engineering
 - System, Software Engineering, and Graphics Specializations
 - CoreTech Engineer
 - Systems, Graphics, Online, Rendering, the game even runs
 - Build Engineer
 - In charge of keeping the game stable, submitting builds to go to MS/Sony
 - Work on Memory System, Low-level systems, external dependencies, and others
 - Work under the TD as a Mr. Random Task

Overview of Multithreaded Changes

- Work took place across year
 - Initial development work about 5 weeks
 - Bugs fixed throughout the development cycle
- Multithreaded system implementation handled by Senior Engineer
- MT Loading System by Technical Director

The Tiger 2008 Engine

- Archaic, Polling-Based engine originally designed for the PS2
- Many static members scattered throughout source
- Multiple copies of the same data in memory, unsynchronized
- Unknown systems that no one has ever touched

Changes in Tiger 2009

- Simulation and Rendering split into different threads, pipelined.
- Loading is multi-threaded
 - File ops have always been asynchronous
 - Load control runs on a different thread than Rendering
- Game Objects handle scheduling of function calls on multiple threads
 - Scheduling is a simple per-thread game object FIFO queue
- Most the stuff from the previous slide still there

Discussion of Changes

- Sim/Render Split
 - Notification System required to delay function calls
- Multi-threaded loading
 - TD had to revert changes numerous times after turning on MT loading
 - Significantly slower than Single-Threaded loading in the common case
- Memory System upgrades
 - More than halved time to allocate chunks

(Relatively) Common Issues

- The Bottleneck a.k.a. the GPU
- Memory Corruption
- Race Conditions
 - Streamed movie player race conditions
- Priority Inversion
- Deadlocks
- SpawnThread(); WaitForThreadToFinish();
 - No, seriously.

Benchmarking Results

- Total Processor Utilization
 - $(\%CPU - \%Wasted\ CPU) / Total\ Possible\ CPU$
- Marker Based Profiling
- Function Based Profiling
 - Sampled
 - Frame
- PIX

Questions

- No seriously, that's it, ask anything
- Example Questions:
 - How was crunch time?
 - What did you do to utilize the SPU's in an MT environment?
 - How were the differences between platform Operating Systems handled?
 - How do I apply?
 - What happens when you break the build?