

Special Topic Course Proposal

ECE 4xxx: Multicore and GPU Programming for Video Games

Instructors

Aaron Lanterman (ECE), Hsien-Hsin Sean Lee (ECE), David Bader (CoC)

Course Objective

This class provides the multicore and GPGPU programming skills needed to meet timely demands of the multimedia, visualization, and gaming industries. The course also bridges the gap between our current generic computer architecture courses and the video game design courses offered by CoC. The class covers two major aspects. First, we will discuss the state-of-the-art general purpose graphical processing unit architectures (GPGPUs) and multicore architectures from application and hardware design perspectives. Then, the course will consider programming models using examples from the algorithmic needs of modern 3-D games (e.g. geometry processing, shading algorithms, physical modeling, collision detection, and artificial intelligence), as well as techniques for adapting such inexpensive architectures for potential use in scientific applications.

Technical Interest Group

Computer Engineering

Proposed Course Units

2-3-3

Target Audience

Junior or senior undergraduate students.

Target semester

Fall 2007

Prerequisites

- ECE3035: Mechanisms for Computation, or
- CS2210: Computer Organization and Programming or equivalent

Course Materials

Primarily lecture notes and literature. Recommended textbooks:

- GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics by Randima Fernando, Nvidia.
- GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation by Matt Pharr and Randima Fernando, Nvidia.

Assessment

Grades will be based on several programming projects (80%) and one final exam (20%).

Projects

Students will be expected to undertake several projects to gain real programming experience on Cell processors, the Xbox 360, and NVIDIA or ATI graphics cards as part of this course.

Topics

1. Overview of real-time computer graphics techniques (1 week)
2. Overview of multi-core processors (1 week)
3. Graphics architectures (2 weeks)
 - a. Power PC/AltiVec used in Xbox 360 and STI Cell BE
 - b. Emotion Engine in Playstation 2
 - c. STI Cell Broadband Engine and SPE
 - d. Nvidia GPU
 - e. ATI GPU
4. Programming tools and environment (2 weeks)
 - a. Standard API: OpenGL and Direct3D
 - b. XNA for Xbox 360
 - c. Cell programming development SDK
5. Programming models (2 weeks)
 - a. Low-level GPU programming
 - b. High-level GPU programming (e.g. Cg or Brooks)
6. Compute-bound game algorithms (3 weeks)
 - a. Physical modeling
 - b. Collision detection
 - c. Particle effects
 - d. Artificial intelligence
7. Game programming on Xbox 360 and Playstation 3 (2 weeks)
8. Implication of CPU/GPU partitioning (1 week)
9. Future graphics/gaming architectures (1 week)

Other Information

Please visit <http://users.ece.gatech.edu/~lanterma/hpvg.html> for more information.