CS4803DGC Design and Programming of Game Consoles

Spring 2011

Prof. Hyesoon Kim

Georgia Tech | College of Computing

What Will We Learn?

- ~~Game programming~~
- ~~Game architecture~~
- ~~Graphics programming~~
- Computer architecture
- Computer architecture case studies → Game consoles
- Emerging architecture
- Graphics processor hardware
- CUDA/OpenCL programming
- Embedded processor programming (Nintendo DS programming)

Game Console

- CPU
- GPU

Who Should Take ?

- If you want to develop
  - Xbox 720 Xbox 1440?
  - Playstation 4,5,6 ?
- If you want to program efficiently using those hardware.
  - ARM Processors, Nintendo DS, PSP
- Background
  - CS2200

Course Info

- http://www.cc.gatech.edu/~hyesoon/spr11/index.html

Changes from Last Year's Course

- Nintendo DS programming (ARM) programming

- More material on ARM, less material on computer architecture background

- More Lab oriented.

- Final Project

Why Game Consoles?

Effects of Game Industries

- Leading the industry
- Game processors are used for other applications
  - GPGPU:
  - Medical image processing
  - Scientific applications
- Movie industries

Requirements for GC

- Time constrain
- Lots of Data
- Heavy use of graphics
- Both Integer/floating point operations are important
- Floating point → low precision
- Stream applications
- Embedded systems
- Various I/O devices
- No comparability issues (no reason to support legacy code)
- All the platform is stable:
- Platform optimizations

Lab Classes

- Every Friday

- Lectures on CUDA Programming (first half of the semester)

- Nintendo DS programming (the second half of the semester)

- No TA
  - Help each other!
  - Use newsgroup to post and ask questions
- Newsgroup activity is counted as class participation points.
- **cs4803dgc-2011**@**googlegroups.com**

Presentations

- Form a team (2 members) and choose a target game console/consoles
- Game architectures/programming/available information
- Examples
  - iPad, wii, PSP, iPhone, Android Phones, etc.
  - 3D game future?

Final Projects

- Typically game programming
- Using GPUs
- Nintendo DS Programming
- Using Tegra 2 Platform
  - Potentially provide full Tablet features
  - Dev development kit will be provided
  - No lab assignments

GAME WORKLOADS

Game Workload Charcterizations

- Still graphics is the major performance bottlenecks
- Previous research: emphasis on graphics
  - 2D/3D graphics tests
- Example: 3DMark Vantage
  - CPU and GPU tests
  - visual and game-play effects
  - CPU side: Physics Simulation and AI
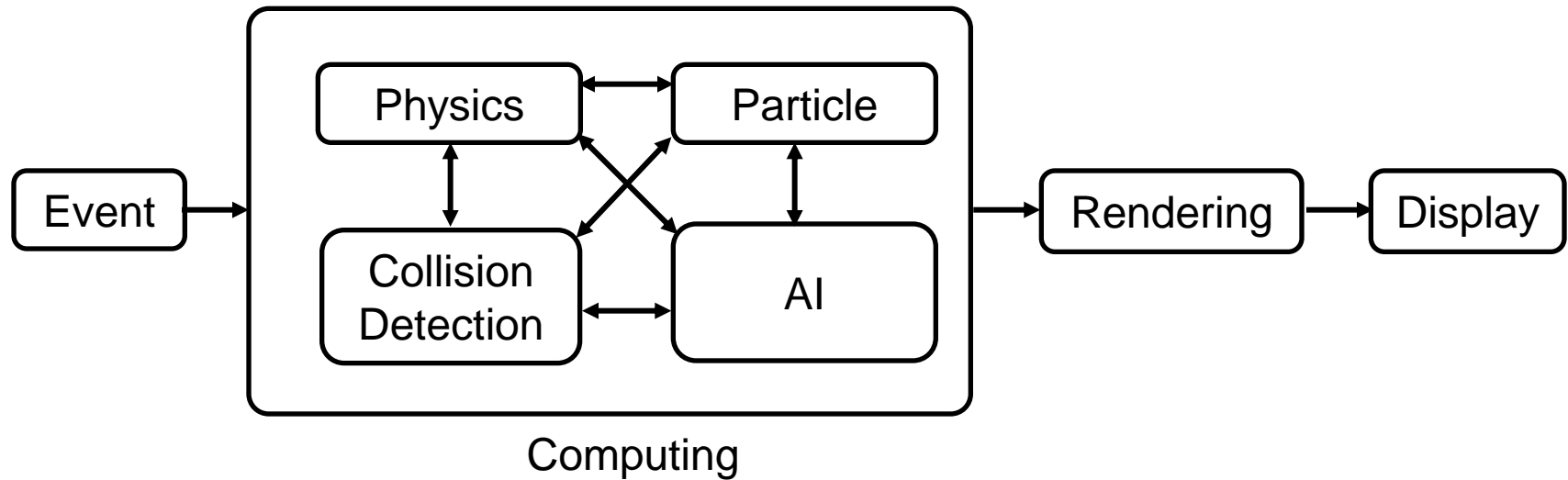 - Rendering tests (both CPU and GPU)

Game workloads

- Several genres of video games
  - First Person Shooter
    - Fast-paced, graphically enhanced
    - Focus of this presentation
  - Role-Playing Games
    - Lower graphics and slower play
  - Board Games
    - Just plain boring

Overview of Game Engine

Event → Computing (Physics, Particle, Collision Detection, AI) → Rendering → Display
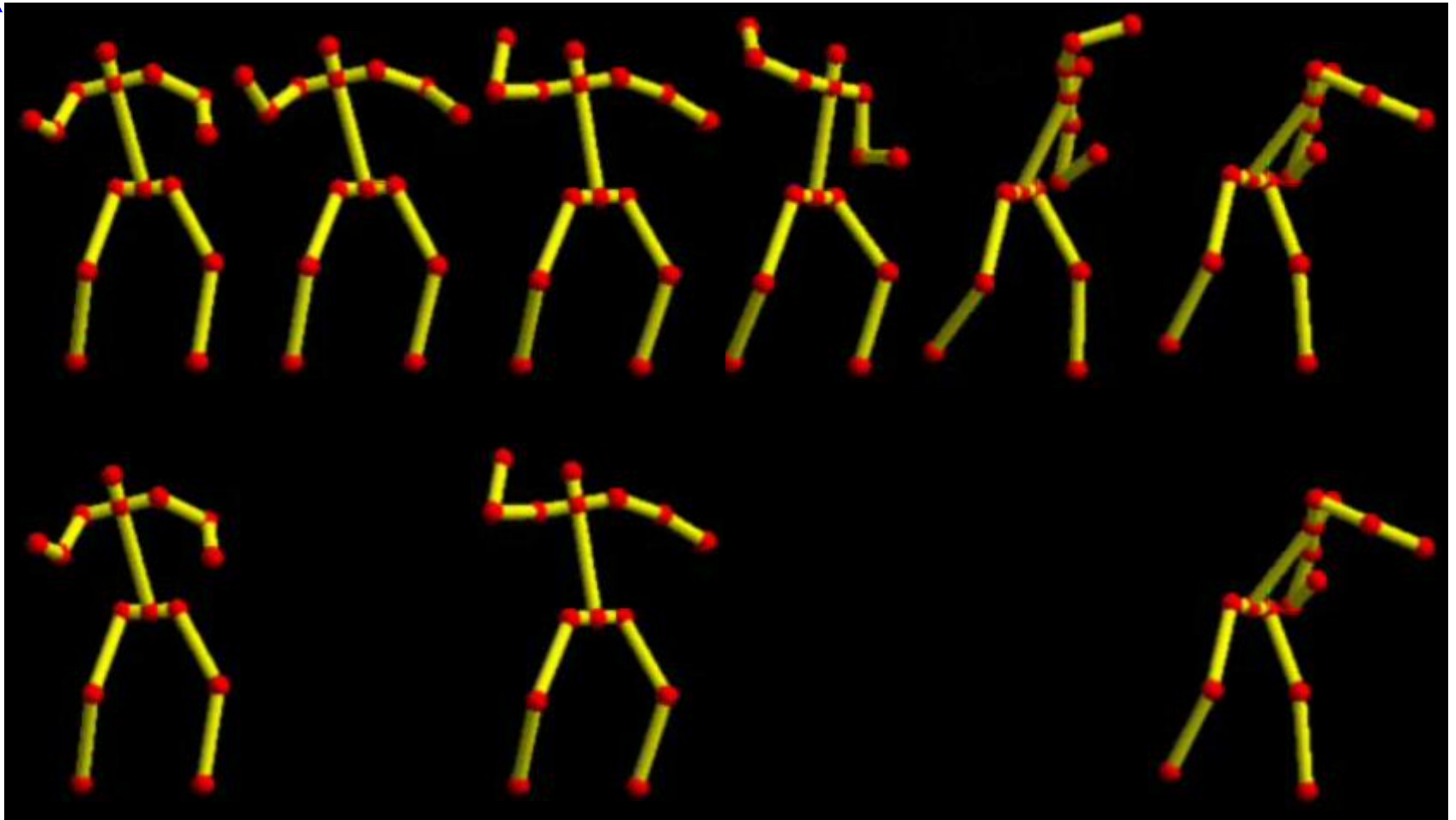
Frame Rates

- Current game design principles:
  - higher frame rates imply the better game quality

- Recent study on frame rates [Claypool et al. MMCN 2006]
  - very high frame rates are not necessary, very low frame rates impact the game quality severely

A First Cut: Reduce Frame Rates

Snapshots of animation [Davis et al. Eurographics 2003]

time

Georgia Tech | College of Computing

Game workloads

Computational workload → **Game workload**

Rendering workload → **Game workload**

Other workload → **Rendering workload**

Rasterization workload → **Rendering workload**

Georgia Tech | College of Computing