The Pedagogy of Low-Latency Distributed Interactive Multimedia Collaboration Systems

Using Video Games to Motivate Computer Science

Dylan McNamee
eXaNetworks, Inc.

“Video Games”

- Are dirty words in academia
- Yet they are
  - Ubiquitous - *really!*
  - A huge industry
  - Worming their way into curricula
- Euphemisms abound
  - *Interactive multimedia*
  - *Low-Latency Distributed Interactive Multimedia Collaboration Systems*

- Using euphemisms to “cloak” video games is missing an opportunity!
- A CS curriculum that includes video games is
  - Relevant
  - Exciting
  - Challenging
- Isn’t that exactly what we need, especially at the intro to a CS program?

A Humble Proposition

- *It could be "a good thing" to introduce a* touchstone *course on video games into a CS curriculum*
- *Touchstone:* A survey course with a theme
- *A good thing:*
  - good for the CS department
    - help increase enrollment and diversity
  - good for the students
    - tie core CS concepts to concrete and exciting applications

“Are You Nuts?”

- Video Games should be taken seriously
  - $17 *billion* / year business
    - This is 2x annual movie theater sales
  - Pervading our culture
  - Economists, social scientists, ... are seriously studying video games

“Are You Nuts?” ...

- Many games are objectionable
  - The controversy is somewhat justified
  - Even these games are fascinating inside
- We are in a *Golden Age* in terms of Computer Science inside video games
  - Ten years ago, it was all assembly code and “hitting the hardware.”

Video Games use an amazing breadth of Computer Science

- Graphics, of course
- Algorithms and complexity
- Data structures
- Languages, compilers and interpreters
- Networking
- Architecture
- Software engineering
- AI, UI, ... (you get the idea)

Complexity and Video Game Performance

- Games strive for 30-60 frames per second
- Most games have many elements that need to be searched/sorted, etc., *each frame*
- Given different algorithm-complexity, how much work can be done per frame?

Complexity
O(n)
O(n^2)

Impact of Faster Hardware

- Students have heard of Moore's law...but probably don't *get it*
- Many people assume algorithmic complexity doesn't matter when machines are ~10x faster every 3-4 years
- This gets their attention!

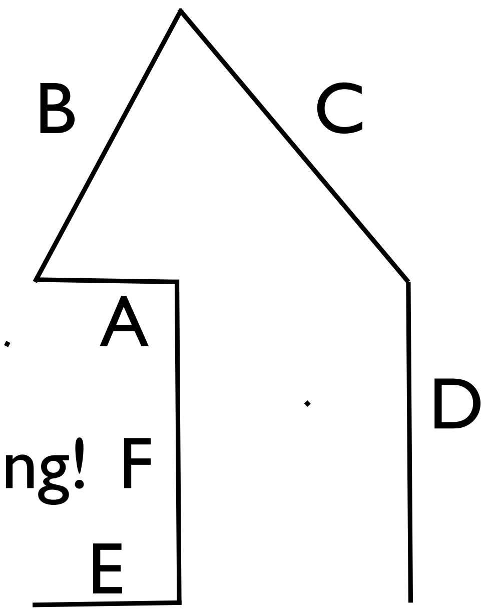
Complexity	Problem size per frame	...with 10x faster HW
O(log n)	~infinite	$\sim\text{infinite}^{10}$
O(n)	2,000,000	20,000,000
O(n log n)	118,650	1,003,201
$O(n^2)$	1,414	4,472
$O(2^n)$	21	24

Data Structures

- You name them, games use them
  - Linked lists
  - Hash tables
  - Trees (very interesting ones!)
  - Graphs
  - Arrays

BSP Trees

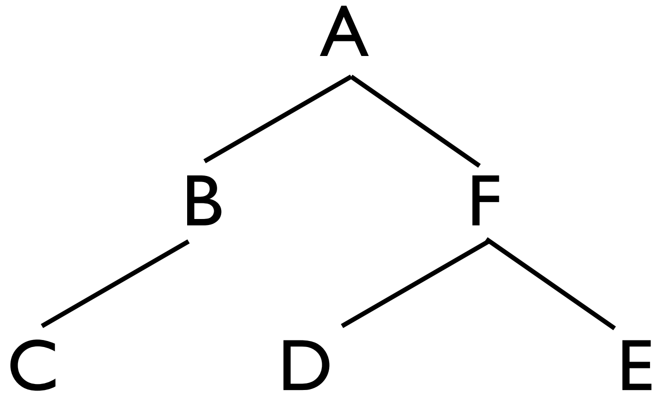
- Rendering 3D scenes using the painter's algorithm requires a depth-sorting from the viewer's POV



Step 1: Place the Camera

Step 2: Traverse the Tree

Step 3: Inorder Traversal = Depth Sorting!



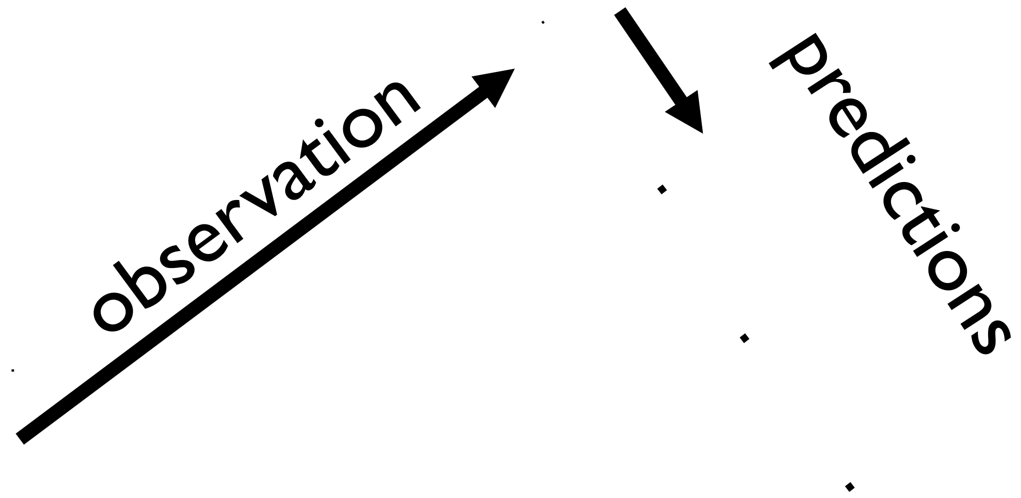
D F E A C B

Lots of offline work turned this into O(N) + O(log N) work!

Networking

- Getting networked games working at all is a feat!
- Clients communicate via server with star topology
- Clients interact with each other using projectiles
- Network latencies can be high and/or variable
- Common technique: client caching

Client-Cached Motion Vectors



Problem: Cache Consistency

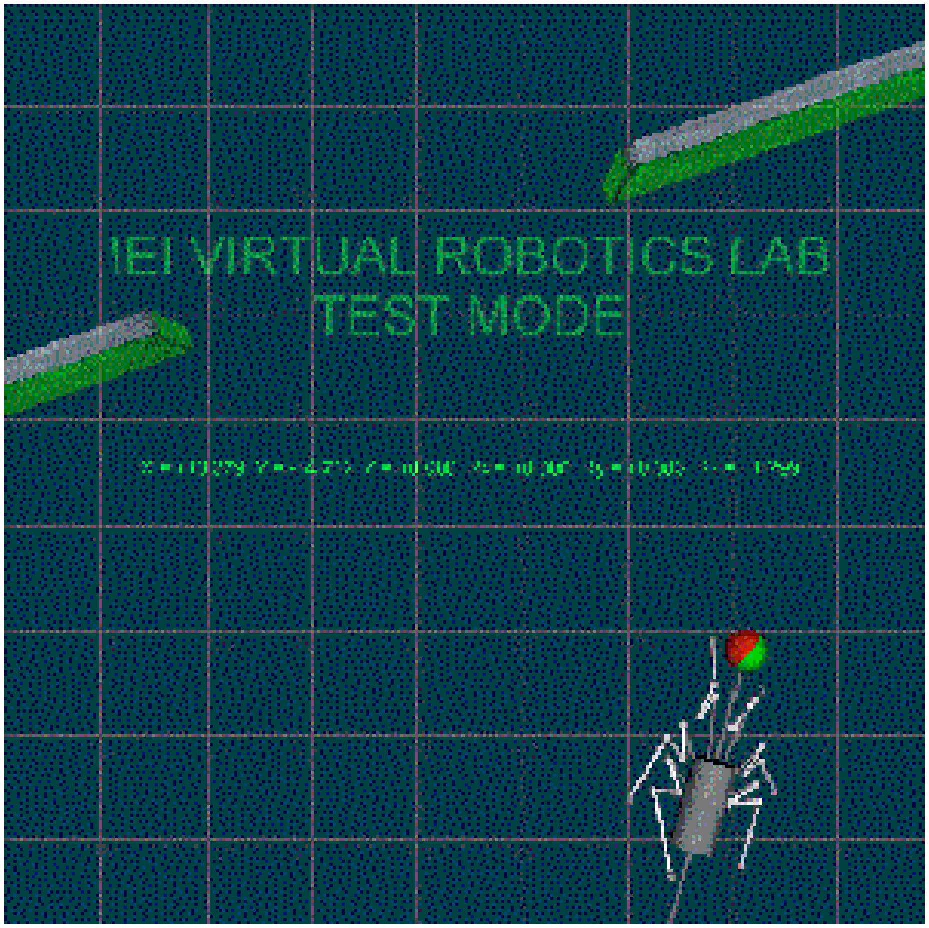
- Sharp turns or changes in velocity invalidate cached state
- Client collision detections are not final
- Server arbitrates disputes adaptively
  - Field-weapons: slow, visible, broad damage
  - Spot weapons: fast, invisible, accurate
- Dynamic, adaptive relaxed consistency policy (!)

Architecture

- Game hardware is a great demonstration platform for computer architecture:
  - Classic ALU, Cache, Registers, Busses, ...
  - Parallel processing
    - SIMD in graphics cards
    - MIMD on motherboards
- Finding and fixing performance bottlenecks is a true *systems architecture* challenge

Artificial Intelligence

- Video games are an ideal testbed for AI
    - Complex virtual environments are a fantastic space for planning
    - “Virtual robotics”
- Similarly, games’ flora and fauna can be made up of agents with emergent behavior



And so on...

- O O Programming
  - Simula was invented for simulation
- Software patterns
  - Model, view, controller
  - Mediators, observers, ... most apply!
- Operating systems
- Real time systems
- Databases

Course Suggestions

- Don’t cover all of this - cherry pick your favorites
- Project platforms
  - Java and Smalltalk both would work fine
  - StarLOGO is an interesting alternative
- Avoid proprietary high-level toolkits

A Taxonomy of Video Games

- The Classic [Crawford, 1980]:
  - "genre"
    - maze games
    - shoot-em-ups
    - puzzles
    - simulations
- Worked for a while, but is now broken
  - All 3-D games are simulations, most are puzzles, many involve shooting in mazes

An Implementation-based Taxonomy

- Measure attributes of "model" and "view" orthogonally, fit games to matrix:

View

	Text	2D	2.5D	3D
1D	Books	Cartoons, Movies		Plays, "linear games"
2D	Adventure, "goto" books	PacMan, SimCity		Myst, Riven, Mazewar
2.5D			SimCity2K, Zaxxon	Doom
3D				Quake Flight Sims

FAQ’s

- Isn’t this a bit “vocational”?
  - I’ve been assured by members of the industry that all they want is smart, well-rounded students
- Isn’t this a fad?
  - Maybe, but this fad is about 30 years old
- What about capstone courses?
  - They’re a great complement - a touchstone course should draw students in earlier

More FAQ’s

- Won’t this only attract boys?
  - It’s true that games have a gender problem
  - My hope is that drawing more women into CS & games will break that cycle, and result in better games!

References, Resources

- Books
  - *Game Coding Complete,* Mike McShaffry
  - *Rules of Play: Game Design Fundamentals,* Salen & Zimmerman
  - *Physics for Game Developers,* David Bourg
  - *Developing Games in Java,* David Brackeen
- Web sites
  - www.gamasutra.com
  - ludology.org
  - www.mecodegoodsomeday.com/games

Languages, Compilers and Interpreters

- Game applications are written *in* a language
  - Language choice should depend on libraries and abstractions, not absolute performance
- Many games use embedded languages for high-level scripting / level design