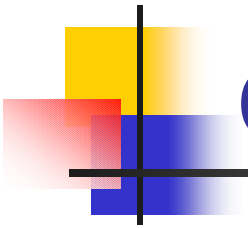
Got MIPS?
The NEED FOR SPEED in On-line Games

Wu-chang Feng

Portland State University

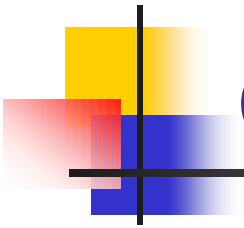
SC|05 MASTERWORKS

Sponsored by: intel IBM



Games

- A big business
  - $25.4 billion market in 2004
  - $54.6 billion market in 2009 (projected)
- Drives advances in computing platforms
  - Intel vs. IBM
    - PC platform vs. console platform
  - This talk
    - What functions do these platforms need to support for future games?



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds

Humans as input devices

- Physical gaming
  - Blurring the real and virtual
    - Physical motion initiating virtual equivalents
    - Prevalent in high-end video arcades in Asia
  - Faster CPUs at clients enabling richer HCI
    - Real-time image and sensor processing
    - Used for traditional video games & augmented reality games

EyeToy

- Entire body as input
  - Real-time image processing
  - Arm, leg, head tracking
  - Embedded in game or driving game actions

Karaoke Revolution

- Voice pitch as input
- Not enough MIPS to detect enunciation
  - The humming cheat
    - BNL's-"One Week" or REM's-"It's the End of the World ..."
    - Simon would not be impressed
      - But humming works in the American Idol game, too!

Human Pacman

- Physical location as input
  - Virtual overlaid on physical via goggles
  - Similar to NFL first-down markers

Future directions

- Higher-resolution input
  - Real-time speech recognition
  - Stereo EyeToy for depth
    - Motion capture akin to current production of sports games
    - Obviate the need for motion-sensor suits?
  - Facilitated by 100-fold increase in processing PS2 to PS3

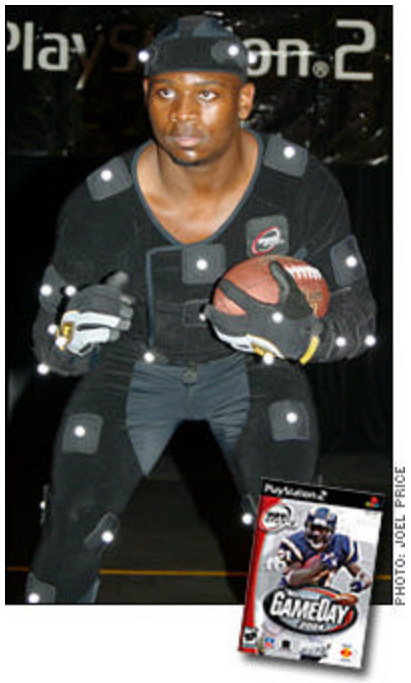


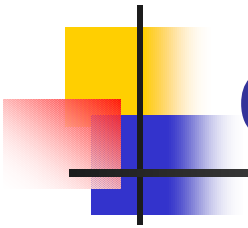
PHOTO: JOEL PRICE

Future directions

- Multi-modal input
  - Karaoke Revolution Party
    - EyeToy
    - DDR pad
    - Microphone

Future directions

- Other input
  - Psychophysiologic sensors
    - Sensing and using emotional state via passive monitoring
  - Gesture-based input
    - Accelerometer tilt sensors
    - Gyroscopic motion sensors (Nintendo Revolution magic wand controller)
    - Not far from a “Minority Report” interface



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds

Procedural content

- Run-time generation of audio and visual effects
- Why?
  - Artists are a huge part of budget
  - Higher resolutions exacerbate problem
    - Increases development time and cost
      - Content generation dominates cost of MMORPG after launch
    - Increases storage and/or bandwidth costs of game
      - Example: Everquest 2 on 10 CDs!
      - Xbox 360 games on 4 dual-layer DVDs?
    - Send new “tree generation algorithm” vs. new trees
      - Procedurally generate all objects, textures, and sound
      - Demo coders can generate a 3D game in 64KB

Generate character aging

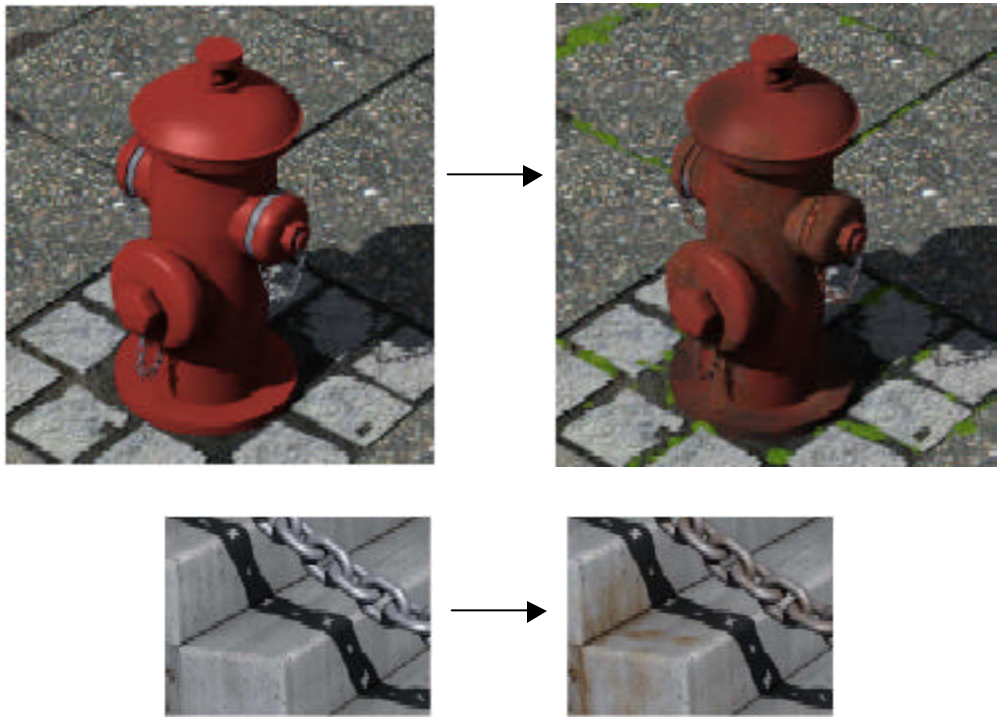
- Versus static pre-rendered models of discrete ages
  - 3 pre-rendered ages of Sims in original (baby, kid, adult)
- Simulate aging and continuously update model
  - Avoid 1000 renderings of same character

Generate character animation

- Versus manually generating static animations
  - Example: The Sims 2 with 22,000 different animations
- Procedural animation based on player's character design
  - Will Wright's Spore
  - GDC 2005 talk

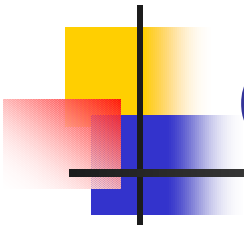
Generate weathering effects

- Versus pre-rendered images of discrete levels of decay
- Simulate rust, stains, and moss growth
  - See Chen et. al. SIGGRAPH 2005
  - MasterWorks talk “Computing Visual Effects is like Compiling Code”



Generate lighting

- Versus fixed levels of lighting in virtual worlds
  - Shadows and lighting pre-rendered in textures/maps apriori and shipped to client
  - Example: Counter-Strike
    - Two pre-rendered versions of a tunnel in cs_militia

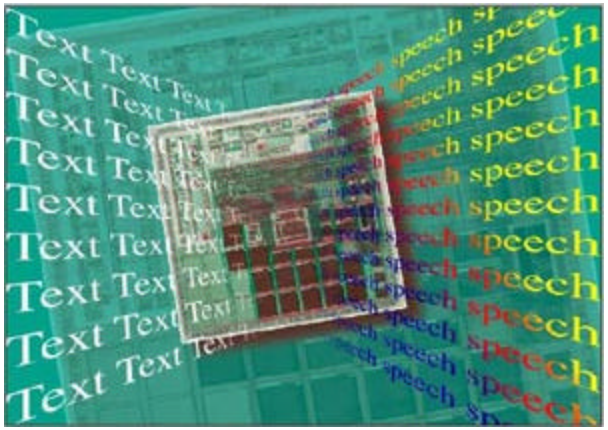


Generate lighting

- Simulate lighting effects dynamically
  - Lighting to reflect time of day and available light sources
  - Global illumination simulation (photon mapping)
    - Photon maps calculated at server based on virtual world
      - Allow server to control time of day on map
      - Allow player “flashlights”, shooting lights out, opening doors, etc.
    - See Henrik Jensen’s animations at http://graphics.ucsd.edu/~henrik
    - More later in talk...

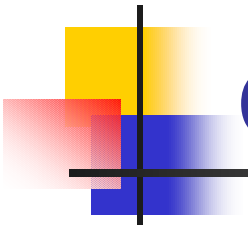
Generate character voices

- Versus static pre-recorded dialogue
  - Example: Call of Duty 2 battle chatter system (10/2005)
    - 20,000 lines with static levels of hoarseness and tones
    - Takes up more space than original CoD!
    - 8% of $14.5 million budget on audio
- Run-time speech synthesis
  - Epson/Fonix 5 language TTS chip (11/2005)
  - http://www.tmaa.com/tts/engine_listing.htm



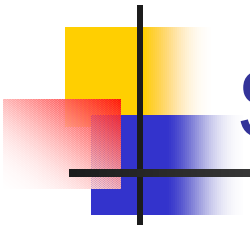
Future directions

- Better algorithms
  - Can we write good artwork generators?
    - Need trees not fractals
      - SpeedTree http://www.speedtree.com/
      - Film CGI tools in games
        - NaturalMotion Ltd.'s endorphin http://www.naturalmotion.com
        - Maxon
        - Lucasfilm
    - Need human not computer voices



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds



Simulation

- Low-level modeling of objects and activity
  - Done at run-time (a form of procedural content)
- CPU advances driving simulations at all levels
  - Physics,
  - People
  - Civilizations

Serious games

- Large-scale simulations for education, training, and forecasting
  - Made possible by recent CPU advances
    - Can finally simulate something useful
    - Pioneered by the military (largest consumer)
  - Economy simulator
    - Japanese Finance Minister game: http://www.mof.go.jp/zaisei/game.html
  - Trauma care center trainer
    - Healthcare games http://gamesforhealth.org/
  - University simulator
    - University management game http://www.virtual-u.org/
  - Iraqi cultural simulator
    - Interaction with allies and local communities (USC/DoD)

Serious games

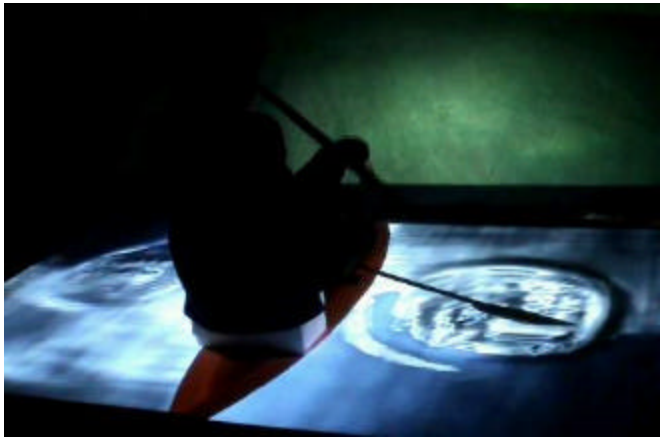
- Non-entertainment games for training and education
  - Military
    - 24 Blue : flight deck operations simulator
    - Public safety personnel trainer
    - Secret service building security analysis
  - Leadership
    - Virtual leader: http://www.simulearn.net/
  - Business
    - Business War Games: http://www.prisim.com/
  - Planning and government
    - Traffic simulators http://www.traffic-simulation.de/
    - Government (Civilization, SimCity, Diplomacy)
  - Biology
    - ImmunoAttack (see next talk)

Serious games

- Non-entertainment games for training and education
  - People
    - PsychSim http://www.isi.edu/isd/carte/proj_psychsim/
  - Bio-terrorism
    - University of Chicago: http://www.uic.edu/sph/cade
  - Science
    - Aerospace simulators
  - Vehicles
    - Cars (GT4, Need for Speed)
    - Airplanes (Flights Simulator)
- More information on serious games
  - http://www.seriousgames.org/
  - http://www.gamasutra.com

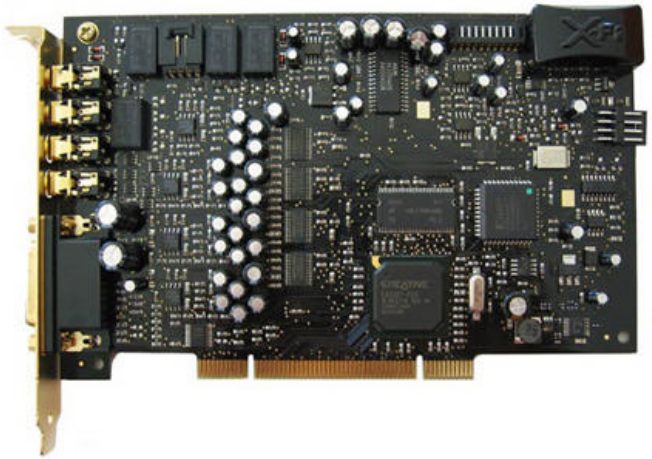
Physics simulation

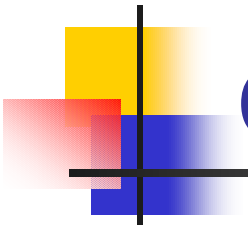
- Scaling collision detection
  - Per-polygon collision detection
    - Polygons increasing rapidly
    - CPU becoming a severe bottleneck
  - Inverse Kinematics (tackling in Madden ’06)
  - Large particle systems (rain, fire, etc.)
- Custom physics
  - Vehicles (cars, planes)
  - Weapons (recoil, ricochets, shrapnel spray)
  - Fluids (water, wind)
  - Activities (parachuting, sailing, snowboarding)
    - SIGGRAPH virtual canoe with algorithm-driven fluid resistance oar



Audio

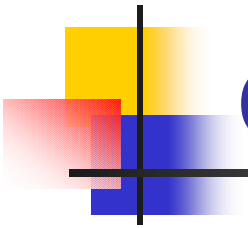
- High-fidelity, 3D audio
  - Environmental effects on sound propagation
  - Doppler effects (bullet localization in Dolby 5.1)
  - Reverberation and echo effects
  - Non-repetitive procedural sounds (footstep sounds of FPS games)
  - Per-player VoIP mixing based on virtual player positions
- Creative X-Fi (Extreme Fidelity)
  - 51 million transistors, 10,000 MIPS





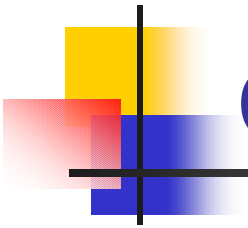
Graphics

- Next-generation graphics
  - Consoles leading the way
  - 3 year window ahead of PC platforms
- Key feature: Highly programmable shaders
  - Dominates current generation of graphics hardware
  - Examples: UE3, Xbox 360, PS3
    - Dependent texture mapping (procedural generation of textures)
      - Realistic fluids, cloth, sweat
      - UE3's water demo
    - Per-pixel processing ("per-frame" Photoshop)
      - Soft shadows
      - Depth of field



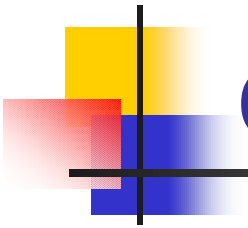
Graphics

- Highly programmable shaders
  - Dependent texture mapping
    - Realistic fur and grass
      - Concentric layers with programmable textures based on motion
    - Alpha-blending (transparency effects)
      - Per-pixel color/texture combination
  - Specific pixel effects “per-frame PhotoShop”
    - Z-buffering (depth of field)
      - Per-pixel blurring based on focus of player
    - Stencil-buffering (soft shadows)
    - Blooming and starring effects
      - Light bending around objects
    - Tone mapping
      - Blinding effects based on eye adjustment time
    - Anti-aliasing
    - Radiosity
      - Simulating reflected light
    - Motion blurring



Graphics

- Highly programmable shaders
  - Requires massive memory bandwidth and close CPU/GPU coupling
    - GPU is main memory controller of Xbox 360
    - Up to 100 instructions per pixel in shader (versus 1)!
    - Accesses across 5-10 textures per pixel (versus 1)
    - Memory bandwidth from CPU to GPU
      - Xbox 360 = 25 GB/s
      - PCI-E = 4.1 GB/s
      - 8x AGP = 2.1 GB/s
    - Internal GPU memory bandwidth
      - PS3 and Xbox 360 = ~256 GB/s
        - Xbox 360 = 10MB of EDRAM for per-pixel processing

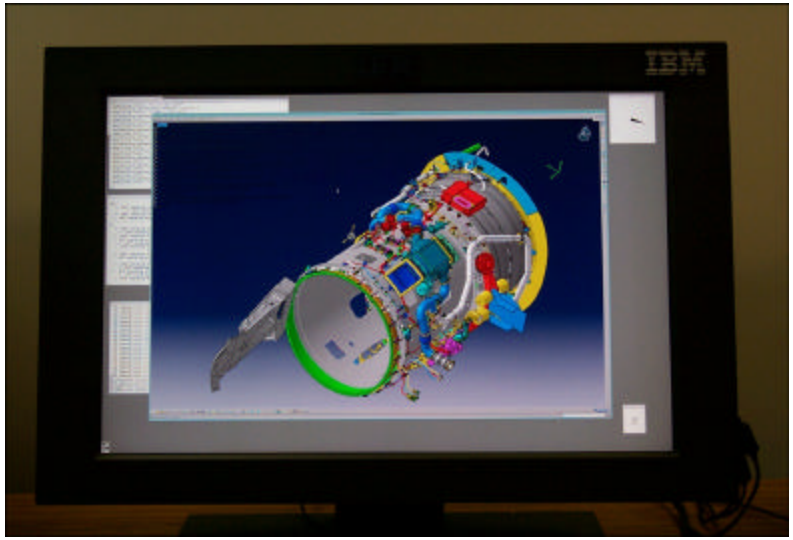


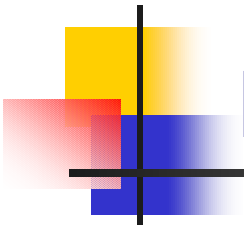
Graphics

- Highly programmable shaders
  - Unified shader architecture of Xbox 360
    - Observation
      - Resource consumption of different stages changes with scene
    - Dynamic ALU allocation between vertex/shader operations
      - Similar to Intel IXP μ-engines
      - 3 banks of 16 shaders
        - Each shader with 4 ALUs
        - 64 hardware threads

Graphics

- Increasing polygon counts
  - UE1 (200 polygons)
  - UE2 (2000 polygons)
  - UE3
    - 4,000,000 polygons (Pre-rendered, cut-scenes)
    - 7000 polygons (real-time)
  - Avenues of improvement
    - Parallel rendering on clusters
      - Chromium (Stanford)
    - Out-of-core rendering
      - Large maps and worlds
      - UE3 streaming from disk



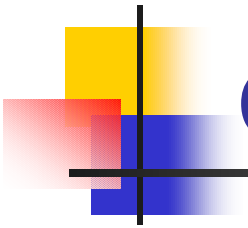


Future directions

- Hardware acceleration
  - Identify common functions, put in specialized hardware
  - Ageia PhysX physics acceleration chip
    - Hardware acceleration for physics and collision detection
      - 120 million transistors
      - Go from 200 simultaneous objects to 32000
    - Supported in UE3
      - Falling rocks demo
    - Rumored hardware support in PCs (Dell?)
      - A threat to Intel?
  - Havok
    - GPU-based physics acceleration

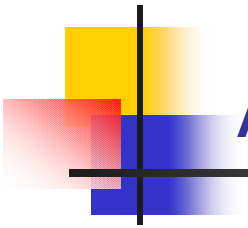
Future directions

- Physiological simulation
  - Face simulation
    - Simulate facial motion based on audio phenomes
    - Examples
      - FaceFX face animation (UE3)
      - Di-o-matic LipSync and Facial Studio (http://www.di-o-matic.com)
  - Muscle simulation
    - Simulate human muscular and nervous system to synthesize animations
    - NaturalMotion Ltd.'s endorphin http://www.naturalmotion.com
- Evolutionary simulation
- Other simulations?



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds



AI in breadth

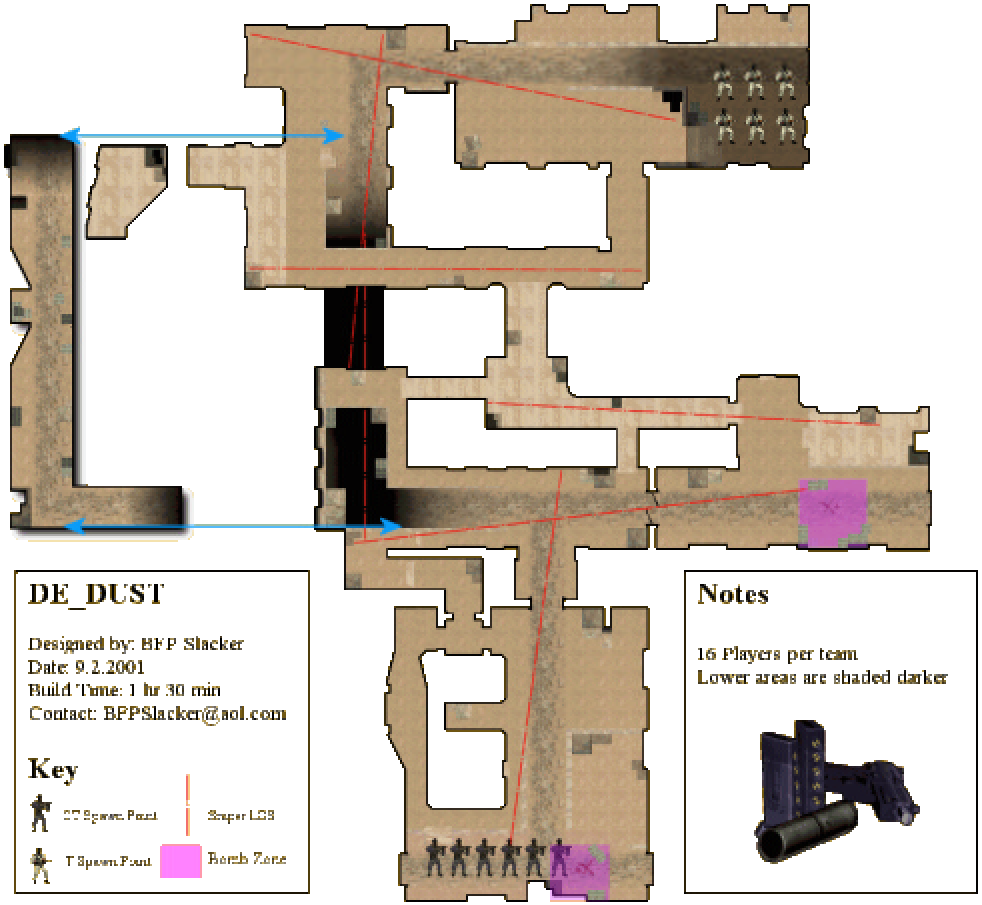
- George Lucas at SIGGRAPH 2005
  - “The next breakthrough in games will be artificial intelligence and voice recognition”
- AI providing more separation in games vs. graphics and art
  - 7-fold increase in CPU time devoted to AI since 1997
  - Killzone (PS2) devotes 12% of CPU to AI
  - NPCs no longer need to “cheat”

AI in breadth

- State of AI in games
  - Dominant at static, turn-based games with strict rules of movement
    - Example: Chess and Deep Blue
  - AI developers have difficulty with
    - Games with heavy independent thought and action
    - Dynamic, open-ended games with emergent behavior (Counter-Strike)
    - Games that require diverse virtual characters, allies, and opponents
    - Need better breadth!

Path-finding

- Age-old AI problem still consuming most of CPU
  - Find shortest, safest, most tactically advantageous path
  - Consumes 40% of CPU for bots on FPS games
    - Map complexity
    - Updates every 50ms
  - Path-finding in WoW
    - Too difficult
    - Too many creatures
    - Mobs go directly to points in world
  - SDKs for pathfinding
    - PathEngine



Minimally scripted AI

- Riots in State of Emergency
- Soldiers in Metal Gear Solid
- Police tactics in GTA3
- Platoons in Far Cry

Minimally scripted AI

- The Sims
  - Free-will button allows characters to take care of themselves indefinitely based on goals

Role-specific intelligence

- Better NPC allies
  - Automating WoW?
    - Fix class imbalance
    - Everyone wants to play the hero
    - No one wants to heal the hero
  - Coordination and protection in Halo
- Better NPC enemies
  - Provide diverse opponents
    - Munch's Oddysee: Monster-specific intelligence
    - AI based on skills, experience, equipment, race, etc.
  - Understanding and adapting to players
    - Play at the level of the person paying you money
    - Play to keep customer happy

Personality training and acquisition

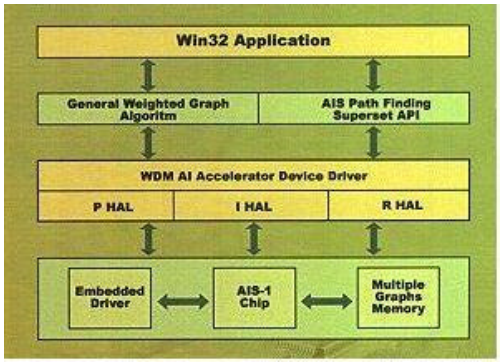
- Pavlov dog training on a grand scale
  - Psychological simulation
    - Mimicry and penalty-reward training
  - Nintendogs
  - Black & White Titans (pets)
    - Continuous training to train titan (pet)
      - Glutton, killer, care-taker, athlete, etc.
    - Non-programmed behavior
      - Eating its own arm when starving
      - “My ape couldn’t find someone to heal. He got pretty upset. So he threw a guy against a mountain. Then he healed him”

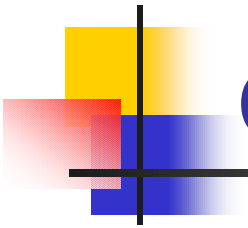
Game Master replacement

- Humans that keep game running at a huge cost
  - Detect and ban cheaters
  - Observe and ban griefers
  - Free players who are stuck
  - Provide technical support
- Game Master automation
  - Currently primitive
  - HLGuard for cheaters
  - Swear filter for griefers
    - The Sims On-line
    - AMX plug-in (In action at cs.mshmro.com)

Future directions

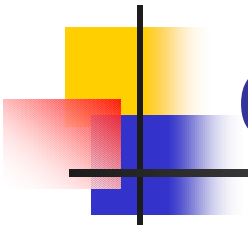
- Hardware acceleration
  - AIseek AIS-1 “AI chip”
    - Path-finding and terrain analysis
    - Squad formation and movement
  - Is there an AI ISA that works across games?
    - Counter-Strike vs. Chess
    - Civilization vs. Nintendogs
- Combining a variety of techniques
  - http://www.cgf-ai.com/links.html





Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds



Cheat detection and prevention

- Cheating impacts bottom-line of any game
  - Wrecks virtual economies
  - Causes legitimate, paying players to quit
  - Creates bad word-of-mouth to discourage new players
- The Achilles heel of the PC gaming platform
  - Must be fixed to compete with consoles

Information exposure cheats

- Server or peer sends complete information to other client
  - Cheat reveals information that should be hidden
- Wallhack
  - Quake 4 – released 10/18/2005
  - Call of Duty 2 – released 10/25/2005 (Server boycott due to cheats)

Information exposure cheats

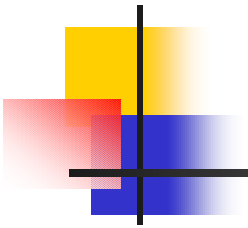
- Maphack (reveal map and enemy units)
  - Warcraft3 without Maphack

Information exposure cheats

- Maphack (reveal map and enemy units)
  - Warcraft3 with Maphack

Information exposure cheats

- Counter-measures (MIPS to the rescue!)
  - Remote rendering
    - Games symmetrically designed currently
      - Both sides run exact simulation
      - Client gets all data about world
      - Bad for preventing cheating
        - Sending the deck in on-line poker
    - Server rendering
      - Fixes information exposure problem
      - Can aid mobile game playing as well
        - Client no longer has to be powerful enough to run full engine
        - Example: PS3 to PSP
      - Problems
        - Not scalable, only simple games
        - Latency issues



Information exposure cheats

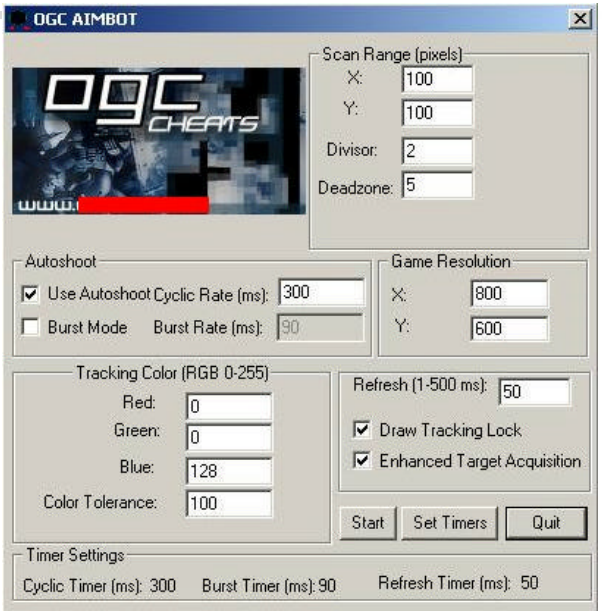
- Counter-measures (MIPS to the rescue!)
  - Data culling
    - Cull data based on player's location and field of view
      - Example: Cheating-Death for Half-Life (client only)
    - Does not work for P2P games
      - No authoritative server to perform culling (Starcraft)?
      - No trusted third-party (i.e. non-player)

Information exposure cheats

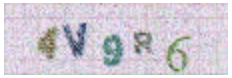
- Counter-measures (MIPS to the rescue!)
  - Data culling with bit-commitment
    - Distributed game simulation (can't trust each other)
    - Data culling combined with cryptographic move and game state signing followed by post-game verification
    - Shameless plug #1: see our work in NOSSDAV 2005! ☺
      - http://www.thefengs.com/wuchang/work/cstrike

Automation cheats

- Automate game activities via Bots
- Aimbots
  - Automate aiming in FPS
- Macros and game bot farming
  - Automate wealth acquisition via programs

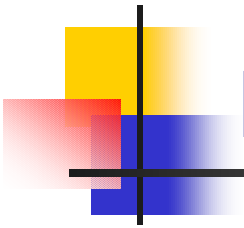


Bots and farming cheats

- Counter-measures (MIPS to the rescue!)
  - Continuous player performance monitoring
    - Example: HLGuard
      - Machine learning of reasonable human reaction time
      - Ban those who react too fast
      - Prone to false positives
        - Cal-I (Cyberathlete league) players
  - Reverse turing tests
    - Captchas 
      - Solve a hard problem in pattern recognition to cheat
  - Personal favorite: secure mice/keyboards
    - Hardware signing its movement and clicks
      - Solve a hard problem in robotics and image processing to cheat!

Bug exploitation

- Exploit inconsistencies and errors in game code
- Item duping
  - Disconnect while dropping item
  - Ambiguity in whether event happened globally
- Other game glitches and errors
  - Magic “pizza” machine in The Sims On-line
  - Vending machine and pawn shop hack in Lucasfilm’s Habitat
- Counter-measures (MIPS to the rescue!)
  - Formal verification
    - NetGames 2004
  - Wealth heuristics
    - Analyze “Gross Game Product” continuously
    - Check per-player anomalies (i.e. use credit card fraud detection algorithms)
    - Personal favorite
      - Monitor currency devaluation on Ebay ☺ (Eve On-line, EverQuest)

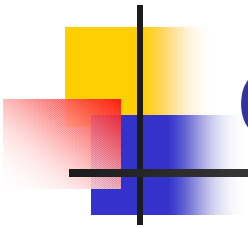


Future directions

- Generic solutions
  - Use of cryptographic mechanisms
    - Authenticity, attestation, non-repudiation
  - Machine learning, clustering, anomaly detection
    - Security folks: sound familiar?
    - Learn normal behavior, flag abnormal
      - HLGuard for reaction-time (see Bots and Farming cheats)
      - Wealth acquisition for MMORPG (see Bug Exploitation)
  - Scanners
    - Continuously scan memory for foreign libraries and cheats
      - Steam and VAC, PunkBuster
      - Heuristics not perfect: Steam and modified OpenGL drivers

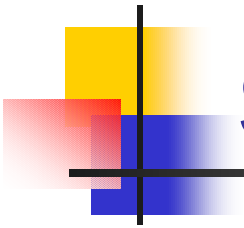
Future directions

- Generic solutions
  - Trusted computing (LaGrande, TPM)
    - Software integrity
      - Ensure no other foreign library is loaded
    - Curtained memory
    - Peripherals
      - Keyboard, mice
      - Secure remote screenshots
        - PunkBuster
        - Sign geometry info or raster output
      - Trusted network output
        - Cryptographic timestamping/ordering
        - Prevent look-ahead cheats
    - Issues
      - Customization vs. Trusted Computing
        - Mods and macros are successful parts of games
        - Counter-Strike, Neverwinter Nights, and Second Life



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds

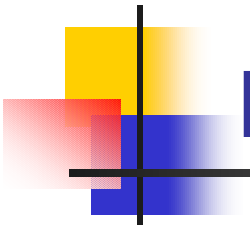


Scaling users

- Goal
  - A single virtual world with everyone in it
- Current games
  - Entire game application replicated into separate instances
    - Socket, thread, memory limitations
  - FPS
    - Single server with 32-64 players
    - Run 20,000 – 50,000 independent servers to support large numbers of users
  - MMORPG
    - Single server and DB with 5,000-10,000 players
    - Run hundreds of independent instances to support large numbers of users

Fixing FPS

- Parallel and clustered FPS server implementations
  - Parallel Quake II (Glenn Deen, OptimalGrid, IBM Research)
  - Clustered implementation with 70ms transition between nodes
  - ICPP keynote http://www2.dnd.no/icpp2005/keynote_icpp2005.pdf

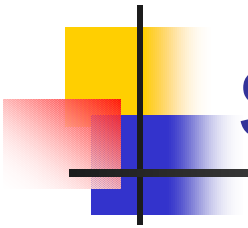


Fixing MMORPG

- Next generation game engines and scripting environments
  - Massive per-entity multithreading (> 20,000)
    - Event-driven programming too difficult
    - Efficient threading, scheduling, synchronization
  - Transparent thread migration between processors
    - Serialization and migration of entity objects
  - Flexible scripting languages
    - Interpreted languages for rapid prototyping and debugging
      - Lua (WoW)
      - UnrealScript (Lineage II, Unreal games, America's Army, Deus Ex)
      - Python (Civilization, Eve On-line, Kaneva engine, BigWorld)
      - Torque (Torque game engine)

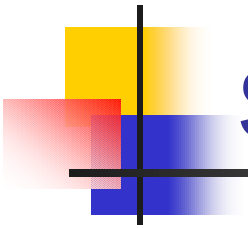
Fixing MMORPG

- Solutions
  - Unreal Engine 3 with UnrealScript
  - Next-generation scripting with Stackless Python http://www.stackless.com/
    - Cooperative user-level multithreading (minimize synchronization)
      - “Tasklets” and “microthreads” (think user-level threads and co-routines)
    - Heap-based stacks (vs. 1MB per pthread for OS threads)
      - Massive threads with slight heap overhead
    - O(1) RR scheduler (minimize scheduling)
    - “Pickling” (think Java serialization) to swap to disk and to migrate to other processors
    - Examples:
      - Eve On-line http://www.eve-online.com/
      - BigWorld game engine http://www.bigworldtech.com/
      - Butterfly.net



Scaling users

- Databases for MMORPG games
  - DB performance a limiting factor
  - Most use relational DB backends
    - May not need flexibility of relational model
  - Application-specific or hierarchical databases
    - Restrict queries and data representations in exchange for speed
    - Hierarchical DBs
      - No longer taught in database classes
      - Used in credit card transaction processing (IMS)
      - Large MMORPGs need transaction rates akin to credit cards

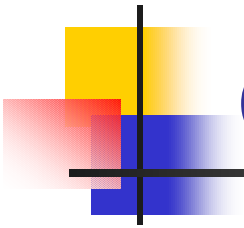


Scaling worlds

- Large, deformable, persistent worlds
  - Currently, server and client share identical maps, textures, & models
    - Small and static
  - Want worlds that are large (do not have to fit entirely in memory)
    - Streaming large maps on-demand (UE3)
    - Does one need to deliver worlds on CD or download them entirely beforehand?
  - Want deformable worlds with persistent and shared effects
    - Persistent "worlds" not just "characters“
    - Terrain that reflects results of the day’s battles
      - Digging a hole and covering it up with grass
  - Enabled via remote rendering, out-of-core rendering?
    - Second Life and a new PSU course
      - www.secondlife.com
      - PSU CS 199: Introduction to Video Game Development (Winter 2007)

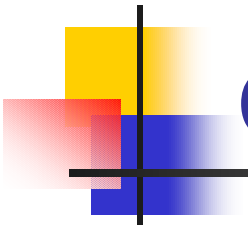
Scaling users and worlds

- Hosting infrastructure for large workloads
  - Enabling games on grids
    - IBM on-demand services for games, Butterfly.net
    - Dynamically allocate resources based on predicted load
    - Reduce risk in hardware purchasing at game launch
    - Shameless plug #2: See our game workload studies at IMC 2005
      - http://www.thefengs.com/wuchang/work/cstrike



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds
- Wrapping up
  - Other areas
  - The future

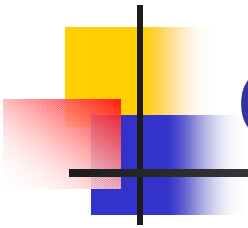


Other areas

- Development engines CPU hungry
  - Engines merging with most games 3D (FPS, RPG, RTS)
    - Unreal Engine 3 (Deus Ex, Lineage II, Splinter Cell, Unreal II, AA, etc.)
    - Modeling, rendering, animation on large clusters
- User content
  - Remote execution of code written by players in VMs (Second Life)
  - Ensuring safety and preventing cheating
    - Virtual machines (Vanderpool?)
    - Language-level mechanisms
    - Code inspection, analysis, and verification
    - Execution monitoring
- OS and networking issues
  - Large number of sockets and threads
  - Large bursts of small packets
  - Efficient handling of one-to-many broadcasts (NOSSDAV 2005)

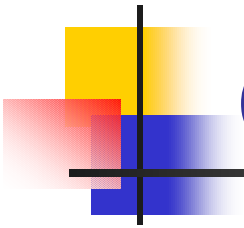
Other areas

- Game services
  - Geographic and multi-player server selection
  - Reputation and ranking systems
  - Game broadcasts and game replay sharing
- Multi-modal output
  - Force-feedback control
    - Electric shock? (Manipulate balance through ear)
  - Sensory surround experience
    - Philips amBX system http://ambx.com
    - Control ambient light, sound, heat, and airflow during gameplay



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - Simulation
  - AI in breadth
  - Cheat detection and prevention
  - Scaling users and worlds
- Wrapping up
  - Other areas
  - The future



Crystal ball

- The killer application for multi-core CPUs
  - Both at the client and server
    - All three consoles multi-core
    - PC platform becoming multi-core
  - Most functions in talk are independent
    - Rapidly growing CPU budget allows for interesting new combinations
  - But which is the most appropriate “core” for future games?
    - x86
    - Cell
    - GPU shaders
    - PhysX ALUs
  - Roadblocks
    - Most game engines are single-threaded
    - Most developers are not trained to write parallel code

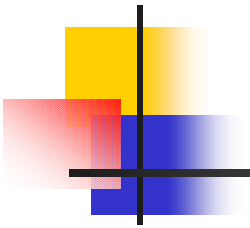
Acknowledgements

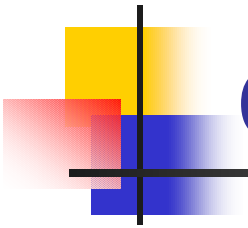
- Sponsors
  - Intel Research Council award
  - IBM Faculty Partnership award
- The Game Group at Portland State University
  - http://games.cs.pdx.edu/
  - Faculty
    - Nirupama Bulusu, Bryant York, Wu-chi Feng, Melanie Mitchell
  - Students
    - Chris Chambers
    - Jim Snow, Francis Chang, Ed Kaiser
- ACM NetGames and `mshmro.com` communities

Final note

- You can upgrade the CPU in the computer, but not the one in the person

- Questions?



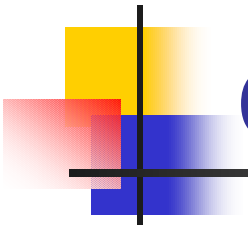


Caveats about talk

- If you’re looking for deep CS research, move along
- Breadth not depth
  - Gaming landscape is massive
  - Talk is not

Doom (1993)

Doom 3 (2004)

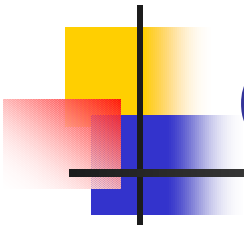


Caveats about talk

- MIPS numbers hard to come by
  - Not really publicized by makers
  - Talk is qualitative not quantitative

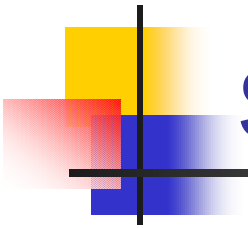
Real Tournament

- Physical location and direction as input
  - Virtual game world displayed on attached iPaq
  - Remote simulation
    - Position and direction of gun determine where shot goes



Current consoles

- PS2, Xbox, Gamecube
  - ~5-10 GFLOPS
- PS3, Xbox 360
  - ~1-2 TFLOPS



Supporting “Emergence”

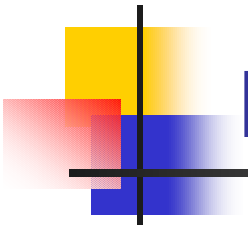
- Process of complex pattern formation from simpler rules
- Create larger behaviors by simulating many smaller interactions
  - Unscripted interactions
  - Varying game play under same set of rules
  - Large number of outcomes and end-states
- Keeps game interesting and new for players
  - Examples: SimCity, Civilization, Black & White, Spore

Bots and farming attacks

- Automating game activities
  - Real-world farming
    - Use workers in third-world countries to generate wealth
    - Counter-measures
      - Machine learning of language and behavior
      - Lawyers
      - http://www.gameguidesonline.com/guides/articles/ggoarticleoctober05_01.asp

Other cheats

- Collusion
  - On-line poker, bridge
  - StarCraft ladders
- Look-ahead (timing) cheat
  - Strategy games
- Speed-hack
  - Half-Life
- Disconnect cheat
  - Madden on-line
- Denial-of-service
  - Time-sensitive P2P games with remote score tracker
  - Go, Chess
- Performance enhancing drugs



Detailed sims

- Teams
  - Sports management
  - Tournament Dreams
    http://www.400softwarestudios.com/tdcb

Generate lighting

- Simulate lighting effects dynamically
  - Lighting to reflect time of day and available light sources
  - Global illumination simulation (photon mapping)
    - Photon maps calculated at server based on virtual world
      - Allow server to control time of day on map
      - Allow player “flashlights”, shooting lights out, opening doors, etc.
    - See Henrik Jensen’s animations at http://graphics.ucsd.edu/~henrik
    - More later in talk...

A word about games in curricula

- Now is the right time to teach Math and CS through games
- Why?
  - Plummeting enrollments in Computer Science
    - Many students turned off by the way CS is being taught
    - Games keep students engaged!
  - Advanced tools allow CS to be taught “top-down”
    - Currently, CS taught “bottom-up”
    - Game engines allow non-CS majors to build a game easily
      - Build a modern game first, then teach underpinnings

A word about games in curricula

- Examples at Portland State University
  - Algorithms, data structures and math via games
    - PSU CS 442: Combinatorial Games
  - Artificial intelligence via games
    - PSU CS 410: Interactive Games and Cognition
  - Systems programming via games
    - PSU CS 200: Computer Systems Programming
  - The “top down” approach
    - PSU CS 199: Introduction to Video Game Development (Winter 2007)
    - Freshman non-major course
      - Second Life www.secondlife.com
        - Leverage built-in physics, graphics, scripting engines
      - Build on top of engine
        - Art, audio, video tools for content generation
        - Introduction to programming