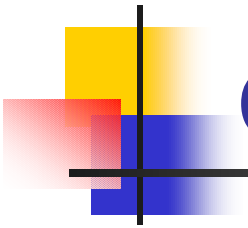
Got MIPS?
The  in On-line Games

Wu-chang Feng

Portland State University

Sponsored by: intel IBM



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users

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
  - 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

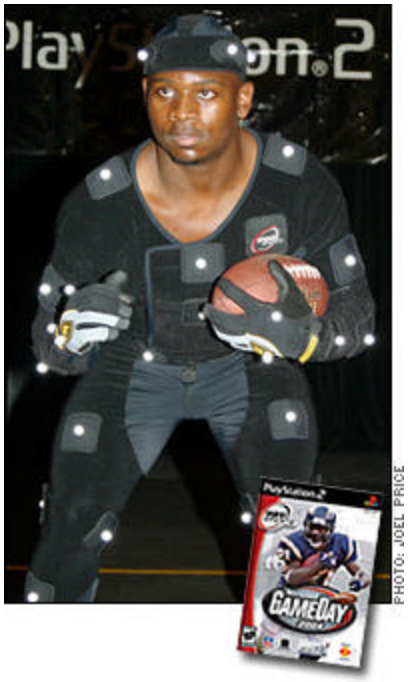
- Physical location as input
  - Virtual overlaid on physical via goggles
  - Picking up real cookies
  - Catching ghosts

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

Future directions

- Higher-resolution input
  - Real-time speech recognition
  - Gesture-based input
    - Accelerometer tilt sensors
    - Gyroscopic motion sensors (Nintendo Revolution magic wand controller)
  - 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, Xbox, Gamecube ~5-10 GFLOPS
    - PS3, Xbox 360 ~1-2 TFLOPS



Future directions

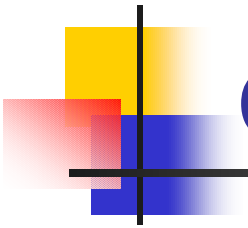
- Multi-modal input
  - Karaoke Revolution Party
    - EyeToy
    - DDR pad
    - Microphone

Future directions

- Multi-modal input
  - Weight sensors for balance
  - Not too far from a “Minority Report” interface..

Future directions

- 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

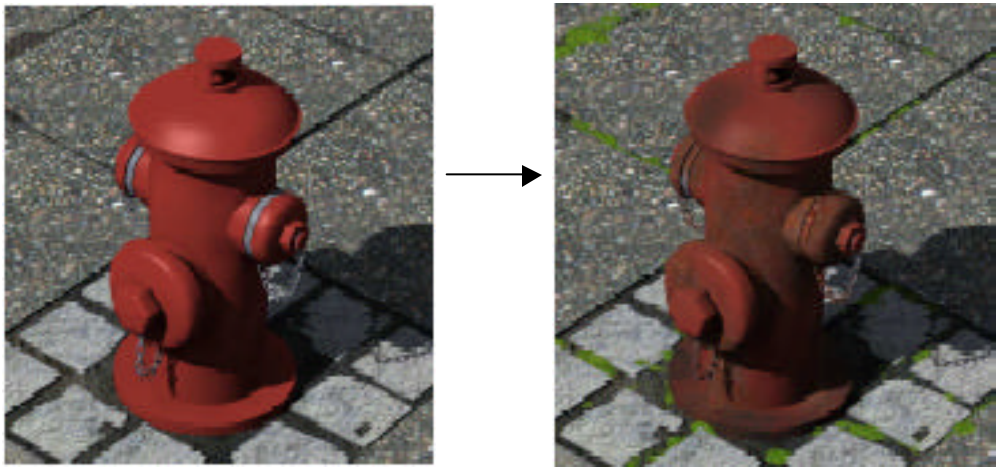
- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users

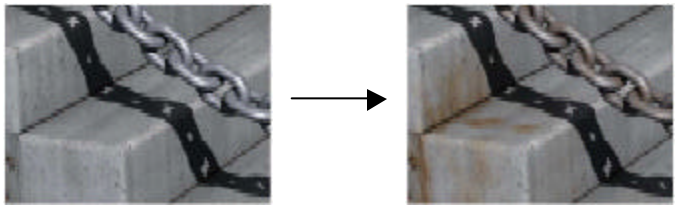
Procedural content

- Run-time generation of content
- 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!
  - Send new “tree generation algorithm” vs. new trees
    - Procedurally generate all objects, textures, and sound
    - Demo coders can generate a 3D game in 64KB

Procedural content

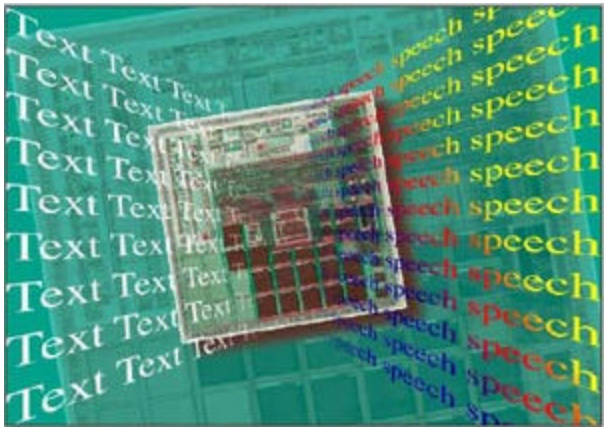
- Examples
  - Generate weathering effects
    - Versus static pre-rendered images of discrete levels of decay
    - Simulate rust, stains, and moss growth
      - See Chen et. al. SIGGRAPH 2005
      - MasterWorks talk “Computing Visual Effects is like Compiling Code”





Procedural content

- Examples
  - Generate character voices
    - Call of Duty 2 (10/2005)
      - Battle chatter system with 20,000 lines of dialogue
      - 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 multilingual TTS chip (11/2005)
        - 5 languages
      - http://www.tmaa.com/tts/engine_listing.htm



Procedural content

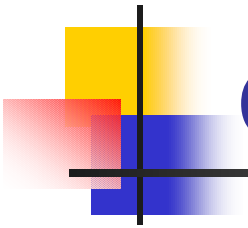
- Examples
  - Generate character aging
    - Static pre-rendered models of discrete ages
      - 3 pre-rendered ages of Sims in original (baby, kid, adult)
      - Black & White fixed age titan models
    - Simulate aging and continuously update model
      - Avoid 1000 renderings of same character

Will Wright's Spore

- Procedural character generation and aging
  - Sims 2: 22,000 animations
  - Spore: dynamically generated animations based on player's character design
    - GDC 2005 talk

Future directions

- Better algorithms
  - Can we write good artwork generators?
    - Need trees not fractals
    - Applying film CGI tools to games
    - Film and game production tools
      - NaturalMotion Ltd.'s endorphin http://www.naturalmotion.com
      - Lucasfilm



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users

AI in breadth

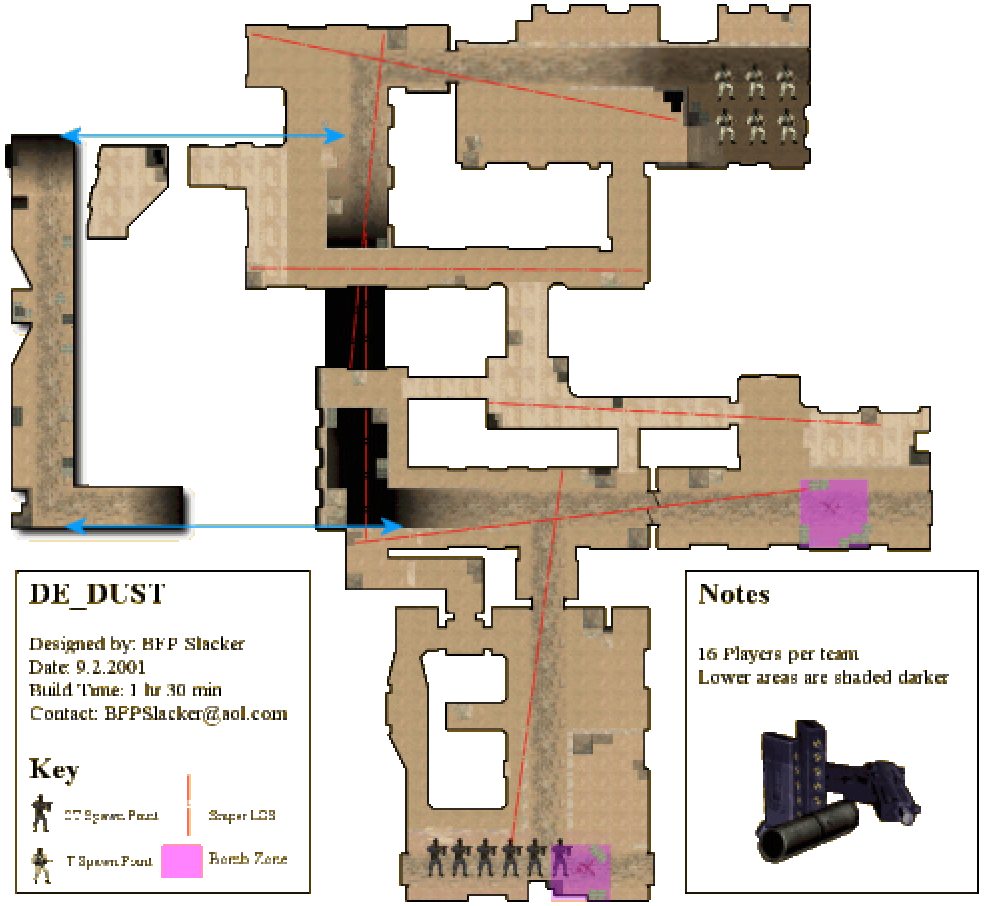
- George Lucas at SIGGRAPH 2005
  - “The next breakthrough in games will be artificial intelligence and voice recognition”
    - “..where you can talk to the game and it will talk back”
    - “..change games from first person shooters narratives to intelligent and challenging first-person shooter type dramas”
- 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 with 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



“Unscripted” AI

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

Role-based intelligence

- Cooperative (allies)
  - Coordinating attacks in Halo
    - Providing cover fire
    - Flanking maneuver
  - Protection in Munch's Oddysee
  - Automating roles in WoW?
    - Fix class imbalance
    - Everyone wants to be the hero
    - No one wants to heal the hero

Role-based intelligence

- Adversarial (opponents)
  - Synthesizing tactical strategy in Killzone
    - Suppression fire to pin down opponent
    - Tactical withdrawal
  - Based on skills, experience, equipment, race, etc.
    - Beyond scripted AI of WoW
    - Richer sets of MMORPG characters
  - Opponent-specific adaptation
    - Play at the level of the person paying you money
    - Play to keep customer happy
      - FreeCell 70% win threshold

Goal-oriented activity

- Enabling free-will button in The Sims
  - Sims taking care of themselves

Personality training and acquisition

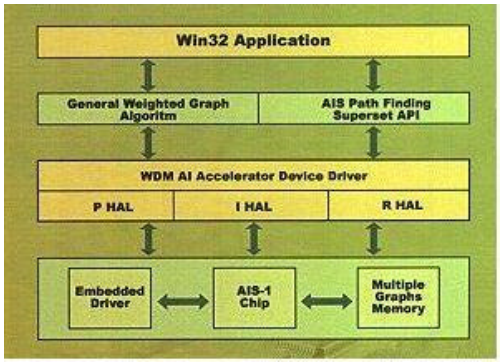
- Pavlov dog training on a grand scale
  - Mimicry and penalty-reward training
  - Nintendogs
  - Black & White Titans (pets)
    - Continuous training to train titan (pet)
      - Glutton, killer, care-taker, athlete, etc.
      - Eat enemy villagers but not your own
      - Heal your own villagers but not the enemy's
    - 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"

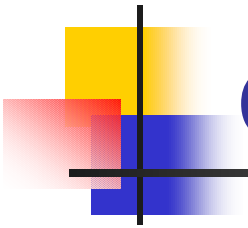
Game Master replacement

- Humans that keep game running at a huge cost to publishers
  - 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 AMX plug-in for Counter-Strike for griefers
    - See it in action at `cs.mshmro.com`

Future directions

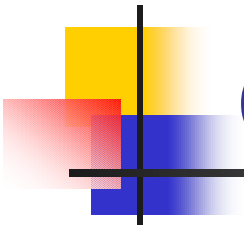
- Hardware acceleration
  - AIseek AIS-1 “AI chip”
    - Path-finding and terrain analysis
    - Squad formation and movement
  - Is there an AI ISA that works across games?
    - Counter-Strike vs. Chess
    - Civilization vs. Nintendogs
- Combining a variety of techniques
  - http://www.cgf-ai.com/links.html





Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users



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

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

Information exposure cheats

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

Information exposure cheats

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

Information exposure cheats

- Counter-measures (MIPS to the rescue!)
  - Remote rendering
    - Currently, games symmetrically designed with both sides running 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

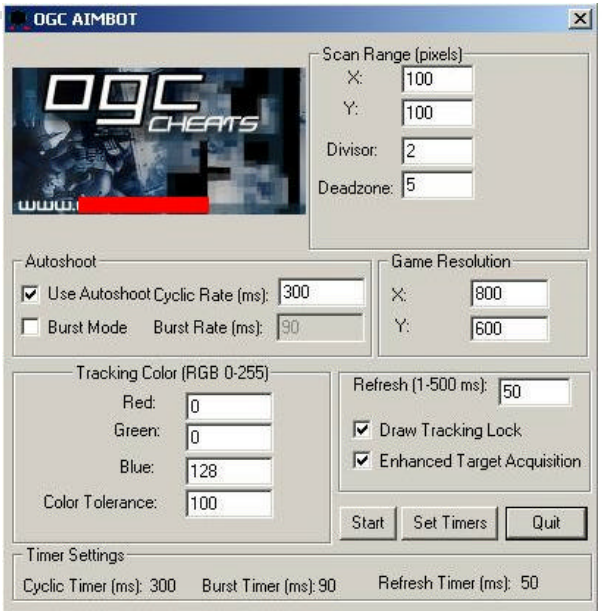
- Counter-measures (MIPS to the rescue!)
  - Data culling
    - Cull data based on player's location and field of view
      - Example: Cheating-Death (client only)
    - Does not work for P2P games
      - No authoritative server (Starcraft)?
      - No trusted third-party (i.e. non-player)

Information exposure cheats

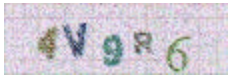
- Counter-measures (MIPS to the rescue!)
  - Bit-commitment
    - Distributed game simulation (can't trust each other)
    - Data culling combined with move and game state signing followed by post-game verification
    - Shameless plug #1: see our work in NOSSDAV 2005! ☺
      - http://www.thefengs.com/wuchang/work/cstrike

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 academic idea: secure mice/keyboards
    - Hardware signing its movement and clicks
      - Solve a hard problem in robotics and image processing to cheat!

Bots and farming cheats

- 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

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)

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

Future directions

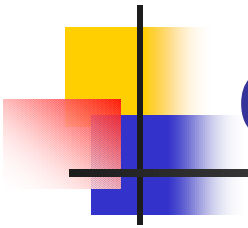
- Generic solutions
  - Use of cryptographic mechanisms
    - Authenticity
    - Attestation
    - Non-repudiation
  - Scanners
    - Continuously scan memory for foreign libraries and cheats
      - Steam and VAC, PunkBuster
      - Heuristics not perfect: Steam and modified OpenGL drivers

Future directions

- Generic solutions
  - Trusted computing
    - Software
      - Ensure no other foreign library is loaded
      - Binary modification detection (Second Life)
    - 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
      - Work currently being sponsored by Intel

Future directions

- Generic solutions
  - Machine learning, clustering, anomaly detection
    - Learn normal behavior, flag abnormal
      - HLGuard for reaction-time (see Bots and Farming cheats)
      - Wealth acquisition for MMORPG (see Bug Exploitation)
    - Security folks: sound familiar?



Outline

- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users

Large-scale sims

- Scaling users in a FPS
  - FPS games historically monolithic, single-server games
    - Limits maximum player numbers to 32-64 on a server
    - Planetside ~200-300
  - Parallel and clustered FPS server implementations
    - Parallel Quake II (Glenn Deen, OptimalGrid, IBM Research, ICPP 2005)
      - Clustered implementation with 70ms transition between nodes
      - http://www2.dnd.no/icpp2005/keynote_icpp2005.pdf

Large-scale sims

- Scaling users in an MMORPG
  - Single server limited to thousands
    - Socket, thread, memory limitations
    - Must resort to realms, sharding, and instancing
    - WoW ~ 8,000 per continent in a realm
  - Fundamentals of MMORPG server internals
    - Common use of scripting languages
      - Programmability, maintainability, extensibility
    - Thread per-player or per-object (> 10k)
      - Event-driven approaches too difficult to debug and maintain
    - Examples
      - Lua (WoW)
      - UnrealScript (Lineage II, Unreal games, America's Army, Deus Ex)
      - Python (Civilization, Eve On-line, Kaneva engine)

Large-scale sims

- Scaling users in an MMORPG
  - Improving MMORPG server internals
    - Issues with massive threading and scripting
      - Scheduling and synchronization overhead
      - Memory footprint 1MB per pthread if OS threads used
      - Inability to migrate threads
    - Next-generation scripting with Stackless Python
      - http://www.stackless.com/
      - Cooperative user-level multithreading with heap-based stacks and an O(1) RR scheduler
        - Tasklets, microthreads (i.e. user-level threads and co-routines)
        - Minimal need for objects to synchronize
      - Massive thread scaling with slight heap overhead
      - Add “pickling” (think Java serialization) to swap to disk and to migrate to other processors

Large-scale sims

- Scaling users in an MMORPG
  - Database limitations
    - DB performance forces MMORPGs to restrict players in realm/shard
    - Most games use relational DB backends, but don't need flexibility of relational model
    - Application-specific databases (XML and hierarchical DBs)
      - Restrictive queries and data representations but extremely fast
      - Hierarchical DBs no longer taught in database classes, but used prevalently in credit card transaction processing
      - Large MMORPGs need transaction rates akin to credit cards

Large-scale sims

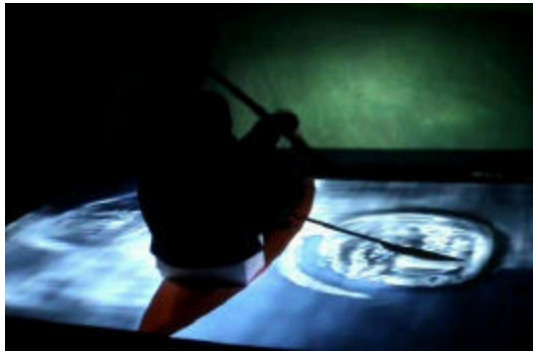
- Hosting infrastructure for large workloads
  - Enabling games on grids
    - IBM on-demand services for games, Butterfly.net
    - Dynamically allocate resources based on predicted load
    - Reduce risk in hardware purchasing at game launch
    - Shameless plug #2: See our game workload studies at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

Large-scale sims

- Supporting “emergence”
  - Process of complex pattern formation from simpler rules
    - Characterized by
      - Unscripted interactions
      - Varying game play under same set of rules
      - Large number of outcomes and end-states
    - Keeps game interesting for players
  - CPU advances enable certain modes of emergence
    - Create larger behaviors by simulating many smaller interactions
    - Occurs at all levels and genres
      - SimCity, Civilization, Black & White
      - Counter-Strike, World of Warcraft, The Sims
      - Spore

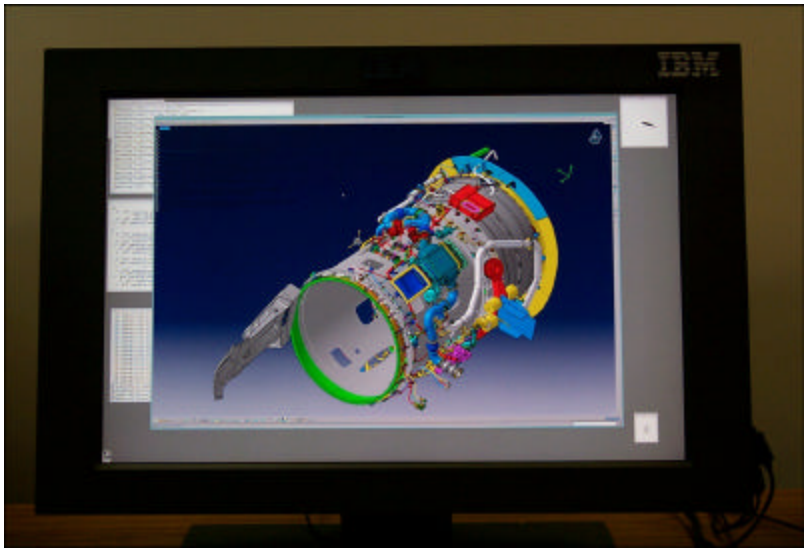
Detailed sims

- Physics
  - Modern game engines do per-polygon collision detection
    - CPU becoming a severe bottleneck
  - Advanced physics for…
    - Vehicles (cars, planes)
    - Weapons (gun recoil, bullet ricochets, shrapnel sprays based on environment)
    - Activities (parachuting, sailing, snowboarding, canoeing)
      - SIGGRAPH virtual canoe with algorithm-driven fluid resistance oar
    - Environmental simulation



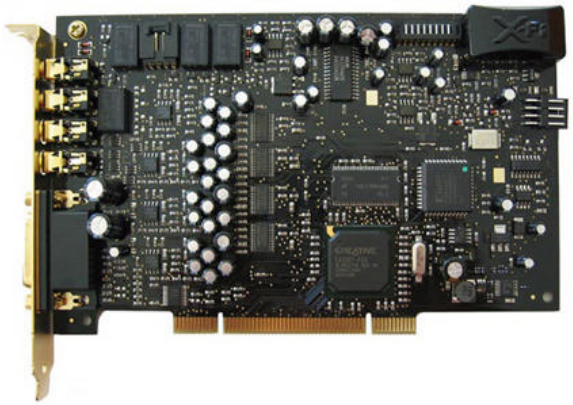
Detailed sims

- Graphics
  - 3D, HD rendering
    - Increased polygon counts
    - Parallel rendering on clusters
      - Chromium (Stanford)
    - Out-of-core rendering
      - Streaming large maps on-demand?
      - Does one need to deliver worlds on CD or download them entirely beforehand?



Detailed sims

- Audio
  - 3D and HD audio
    - Mixing
      - Environmental effects (wind, gunfire, etc.) on sound propagation
      - Doppler effects (bullet localization)
      - Per-player VoIP mixing based on virtual player positions
    - Creative X-Fi (Extreme Fidelity)
      - 51 million transistors, 10,000 MIPS
  - Procedural generation of speech vs pre-recorded scripts
    - See previous slides



Detailed sims

- Serious games (non-entertainment training games)
  - http://www.seriousgames.org/
  - http://www.gamasutra.com
  - Large-scale simulations for education, training, and forecasting
    - Made possible by recent CPU advances
    - Can finally simulate something useful
  - Economic
    - Japanese Finance Minister game: http://www.mof.go.jp/zaisei/game.html
  - Education
    - University management game http://www.virtual-u.org/
  - Health
    - Trauma care trainer
    - Healthcare games http://gamesforhealth.org/
  - Biology
    - ImmunoAttack (see next talk)

Detailed sims

- Serious games (non-entertainment training games)
  - 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)
  - Cultural
    - Voice interaction with allies and (see USC/DoD)

Detailed sims

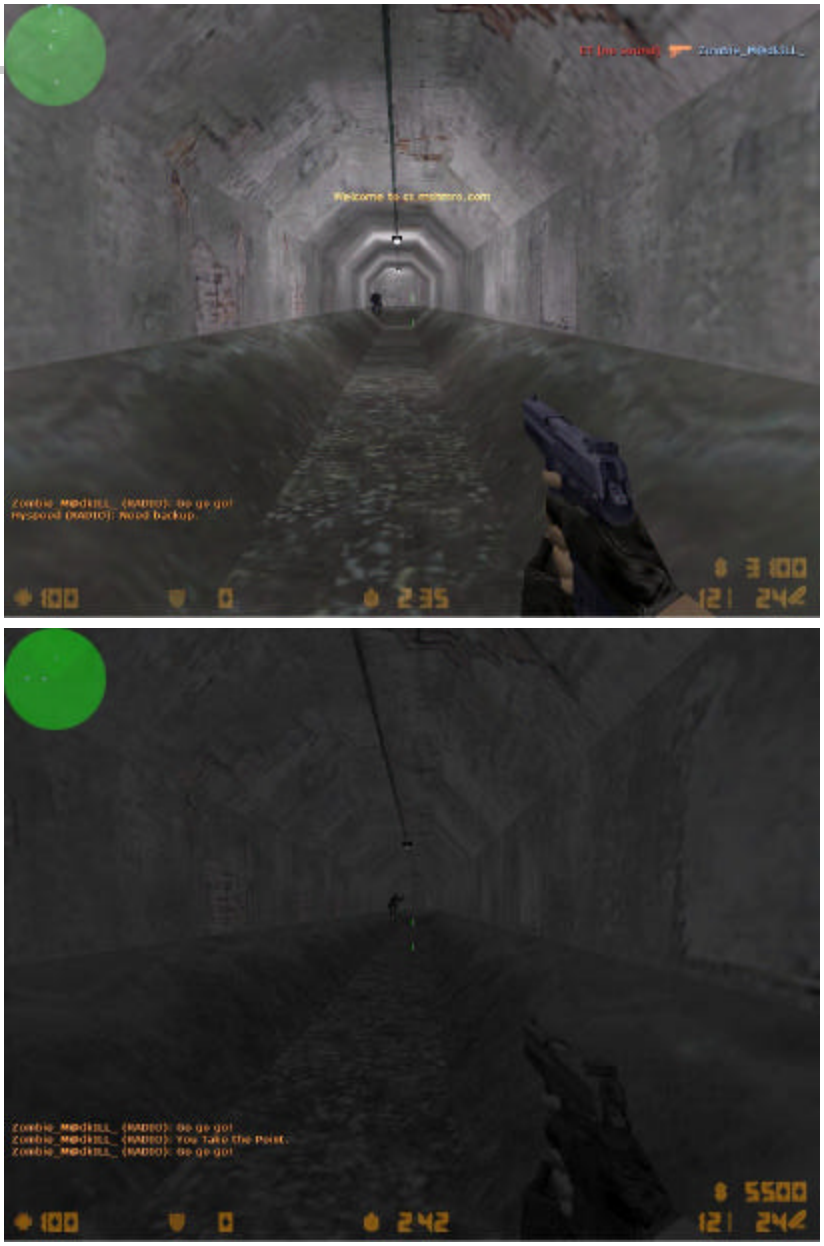
- Serious games (non-entertainment training games)
  - People
    - PsychSim http://www.isi.edu/isd/carte/proj_psychsim/
  - Bio-terrorism
    - University of Chicago: http://www.uic.edu/sph/cade
  - Science
    - Aerospace simulators
  - Vehicles
    - Cars (GT4, Need for Speed)
    - Airplanes (Flights Simulator)
  - Teams
    - Sports management
    - Tournament Dreams http://www.400softwarestudios.com/tdcb

Persistent sims

- Dynamic persistent worlds
  - Deformable maps and worlds
    - Currently, server and client have identical maps, textures, & models
    - Want 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
        - Multiple players interacting with map itself
      - Enabled via remote rendering, out-of-core rendering?

Persistent sims

- Dynamic persistent worlds
  - Dynamic lighting
    - Fixed levels of lighting in virtual worlds (FPS and MMORPG)
      - Shadows and lighting pre-rendered in textures/maps apriori and shipped to client
    - Want procedural generation of lighting effects

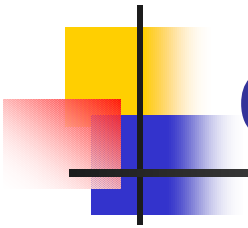


Persistent sims

- Dynamic persistent worlds
  - Dynamic lighting
    - Terrain that reflects time of day and available light sources
      - Dynamically simulated lighting
    - 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
        - http://graphics.ucsd.edu/~henrik
      - Roughly approximate?
        - Pre-generated global illumination simulation compiled into levels

Future directions

- Hardware acceleration
  - Identify common functions, put in hardware
  - Ageia PhysX physics acceleration chip
    - 120 million transistors
    - Go from 200 objects to 32000 (physics and collision detection)
  - Havok
    - GPU-based physics acceleration
- Other lower-level simulations
  - Physiological simulation?
  - Evolutionary simulation?



Outline

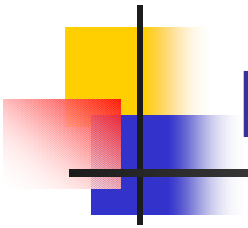
- From client to server
  - Humans as input devices
  - Procedural content
  - AI in breadth
  - Cheat detection and prevention
  - Large, detailed, persistent simulations
  - Content generation via tools and users

Content generation via tools

- Development engines CPU-hungry
  - Most games 3D
    - 3D FPS popularized by Doom (1993)
    - 3D MMORPG popularized by Everquest (1999)
    - 3D RTS popularized by Warcraft III (2002)
  - Applying film CGI tools for games
    - Modeling, rendering, animation on large clusters
    - See previous slides
  - Consolidation of engines for FPS, RPG, RTS (genres blurring)
    - Unreal Engine 3 (Epic)
      - Deus Ex, Lineage II, Splinter Cell, Unreal II, AA, etc.
    - Kaneva, Emergent

Content-generation via users

- Content generation is the largest cost after launch
  - Many successful “user-driven” games
    - Counter-Strike
    - Neverwinter Nights
  - Second Life
    - First MMORPG with user content explicitly designed for
    - Remote execution of code written by players in VMs



Future directions

- Ensuring safety
  - Virtual machines
    - Vanderpool?
  - Language-level mechanisms
  - Code inspection, analysis, and verification
  - Execution monitoring

Crystal ball

- Games have only just begun ...
  - Networks and graphics were key in last 10 years
  - Supercomputing and AI will be key in next 10 years
    - Research from AI and Supercomputing has not been applied to games
- The killer application for multi-core CPUs
  - Both at the client and server
  - Most functions in talk are independent
    - Rapidly growing CPU budget allows for interesting new combinations
  - New system requirement?
    - # of cores along with GHz, RAM, disk space

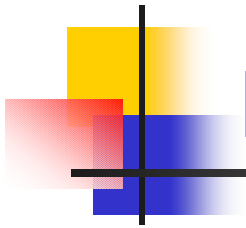
Acknowledgements

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

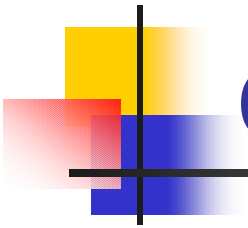
Final note

- You can upgrade the CPU in the computer, but not the one in the person
  - http://www.ebaumsworld.com/asianarcade.html

- Questions?



Extra slides



Caveats about talk

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

Other CPU draws

- OS/Networking issues
  - Lots of tiny packets and low bandwidth connections
  - Unicast work in NOSSDAV 2005
  - Overhead miniscule really
  - Not much here for us networking folks, move along
- Services
  - Geographic server selection
    - http://www.thefengs.com/wuchang/work/cstrike
  - Sharing game replays
  - Spectator networks

Doom (1993)

Doom 3 (2004)