Computer Graphics in Games: present & Future

http://www.pong-story.com/

MIT EECS 6.837
Frédo Durand
Course evaluation

- HKN underground guide survey
- http://sixweb.mit.edu/
- until 11:59pm Friday, December 10
Final exam

• Wednesday December 17 1:30-4:30
• 4 pages (2 double-sided sheets) of notes
• Everything we saw this semester
  – With emphasis on 2nd part
• Similar to quiz 1
• Will target a 2-hour subject
Games & Graphics
Gears of war 2

- (Unreal engine)
- Smart conservative collision detection
- Blades of grass using vertical textures
- Anisotropic mipmapping not bad
- Physics
- Visibility culling
Tim Sweeney’s slides


- [Image of a person]
  - [Image of a creature]
Unreal Engine

- Mostly for First Person Shooters
- Animation, rendering, sound, etc.
Rendering in Unreal Engine 3

- Multithreading
- Visibility culling
- High Dynamic Range (more than 0-255)
- Pixel shaders: normal map, Phong; anisotropic effects; displacement mapping; etc.
- Platform abstraction layer
- Lighting with spherical harmonics
- Shadow volume, shadow map, PCF
- Volumetric effects (fog)
- Texture streaming
- Reflection with dynamic environment maps
- Image post-processing: bloom, tone mapping, etc.
- Lots of shader support
Umbra’s occlusion culling


A view of a 3d-scene

The same view seen from higher perspective.

The same scene with Umbra culling off the nonvisible objects. And it works real time!
Animation

- skinned mesh rendering (up to 4 bones / vertex)
- Mesh and bone Level of Detail
- Keyframed animation
- Particle system
- Rigid-body physics
- cloth simulation
- Soft-body physics
- Fracture
- Collision detection
See also Valve source engine

• Slides from Jason Mitchell
• http://www.pixelmaven.com/jason/
Battle for Middle Earth

- Shadow map artifacts
- Near plane
- Animated texture coordinates
- Billboard trees?
- Polygonal models
- Path planning
Fight Night round 3

- constrained environment -> can be optimized
- Uncanny valley
- mocap
  - http://www.youtube.com/watch?v=IF3vrR_wrLo
- depth of field
- some 3D scanning
- hdr
- specular highlights on skin
- some physics simulation for impact, ropes, shorts
- normal maps for body
- cloth and skin BRDF
- some view frustum culling bugs in overhead view
Oblivion

- Free world, good loading
- LoD popping
- Plants: billboards
Test Drive

- LoD popping
- Car reflection
- Some aliasing
- Free-world game. Pretty good loading.
Game main components (e.g. FPS)

- UI,
- House keeping, game logic
- Sound
- Animation, AI
- Rendering
- Network, loading, streaming
- Assets
- Authoring and workflow tools
Valve's graphics challenges

Compiled for you by Jason Mitchell

- Dynamic global illumination effects This could include in-between steps such as dynamic ambient occlusion
- Omnidirectional soft shadows with contact hardening
- Surface detailing (solutions include displacement mapping, wrinkle mapping etc)
- Motion Blur
- Procedural modeling to increase productivity (foliage etc)
- Character animation Proper foot placement during walking/turning/running (while still having immediate response from player input) Walking/running animation on uneven surfaces (stairs, hills, etc)
- Volume preserving deformations, particularly during object-object interactions
- Cloth animation (physics is more challenging than rendering)
- Hair rendering
- Fluid rendering
- Local adaptation for tone mapping.
- Large scale dynamically modifiable/destructable environments
• Big issues: Low memory budget, multiprocessing
• Local information, no access to the scene → Recast problems as local
• Deferred rendering/shading
  – multiple passes, decompose the problem according to components, polynomial approximation, each coefficient in a pass and accumulate them
• Parallelism
  – Load balance, teapot in football field (moving forward) PS3 (G80 similar to Cell, extrapolate forward, hypercube bus)
  – Plan to not using HLSL in the future, move to something like Cuda, UPC, Peakstream, accelerator to do the rendering and more (compositing layers), physics
  – Lucas Art is unusual (most companies don't do water/physics on GPU) cloth, water, ductile fracture. in announced star wars, indiana jones, hair cloth, water, FEM for soft body, ductile fracture (a couple ms/frame at 60hz)
  – Question: PPU: window of opportunity has closed, purpose too limited, optional
  – Rigid physics, collision, etc one ms of CPU time
    • Next gen GPU will be so much more powerful, not planning to use PPU
Nick Porcino, LucasArt

- mocap: happy with it. Black sheep of animation. "devil's rotoscope" because data is dirty, they spend more time cleaning.
  - Same technique for games, but need to reuse
  - Lots of interesting problem: smoothing, interpolation, matching

- Q: combine mocap & physics;
  - they do it, difficult. hierarchical rigs. He's disappointed by state of the art, kinematic skeleton only, needs balance skeleton, vertex cluster. Kinematic drives balance skeleton. Create a direction controller, drives kinematic, center of gravity, constrained cloth, dense hierarchy. Only able to afford kinematic skeleton 12 bones, have to map on deformation skeleton: 90 bones. Plausible mapping? Synthesize data for other 90 bones. e.g. spine is easy (from 2 to 5 bones). Shoulder joint can have 9 bones but only arm +scapula in kinematic skeleton. interpolate??? Only dealt with partially. Forced them to standardize.
  - Use natural motion technoloy + Havock.
  - Challenge: getting up. Easy to collapse, but unpredictable. Then how do you stand up? Try to get standard collapsed poses.
Nick Porcino, LucasArt

- Q: procedural modeling & amplification.
  - Mixed feeling. Keen on the technique. Has had to back off and not use. Xbox 360 can do it. Problem: structure of current pipeline, cannot be blind, tesselator becomes the bottleneck. Need access to 1-ring
  - Particle systems, trees, L system. Suffer greatly from lack of global information. L systems look too procedural. Real tree needs knowledge of accessibility to light, etc.
  - Pascal Mueller, city engine project. Can be enforced locally.

- Material appearance
  - 3 components: mechanical application of algorithms: BRDF shading. Capture of data, reduction of model to be tractable. Interesting: polynomial texture maps. surface transfer, did attempt, but not applicable (e.g. PTM too big data).
  - Some techniques: enough memory but data is too scattered. (bad cache performance)

- More and more global illumination? Yes, it is required. ambient occlusion is the current solution. Does not hold under deformation. Search for techniques that encode 1\textsuperscript{st} and second bounce into maps.
  - Geometric algebra. hemicube, attempt to solve initial transfer on hemicube, then remap. Geomerics. Alternative to linear interpolation & slerps.
  - want it, find more and more sophisticated ways to approximate it.
  - Bake more and more static stuff, hope to get more dynamic.
  - Photorealism? Bemused by Photorealism. On the one hand, drives mathematics and advances, but on the other hand, he does not believe it's a useful goal. Starts with bad assumption (name is an oxymoron) realism is what you perceive, photo is mechanical/chemical, it ignores our perception. Realism is about what our perception does. Computer graphics should go there. TV & movies work because they are not photorealistic, they have an interpretive mechanism (director, director of photo). Rules such as contuity editing are designed to skew perception, so that you understand. Example, camera pans over group of people. Position is adjusted depending on camera viewpoint.
  - Glad graphics is spending so much time because they provide toolbox
Frédo's complaints

- Aliasing
- Motion blur, depth of field
- Shadows
- LoD popping (but much better now)
- Character Animation
  - Foot contact
  - Stiffness, dynamics
- Physics simulation
  - Water, cloth, rigid body, deformable
  - coupling with character
- Shader level of detail
- Multithreading
- Collision detection
- Material appearance
- Hair
- Open worlds
- Open scripts
- Detail amplification
- Humans

Wanna be a game programmer?

- Take 6.034, 6.170 (=>6.005), 6.837
- Practice C++
- Make a game to show off
  - or mod a game
  - Microsoft’s Xna is a good place to start
- Learn multithreading & parallel programming
Resources

- http://www.gamasutra.com/
- http://www.pixelmaven.com/jason/