Overview

- ToyShop system overview
- Parallax Occlusion Mapping
- Rain rendering
  - Rendering rain precipitation
  - GPU Water effects
  - Wet reflections via reflection impostors
- Results
- Conclusions
Imitating Reality: Background Research

• We wanted to create a moment in a dark city, downtown, during a rainy night.

• Fortunately, having started on the concept in the middle of October in Boston, we had a lot of opportunities for research.

• Gigabytes of videos of rain, droplets, streams, … and here we are:
Demo: ToyShop
Zen and The Art of Shader Writing

- This demo is a study in combination of complex computations with inexpensive yet effective shaders for a variety of effects
  - Perspective-correct extruded surfaces with SM 3.0 parallax occlusion mapping
  - Water and raindrop simulation effects
  - A variety of post-processing effects for glow, blurry reflections and rain
  - Many custom effects for rendering rain
  - HDR illumination with lightning and shadows
  - Many more…
ToyShop Constraints

• Our goal was to fit this demo into 256MB of video memory

• This was a challenge considering the rich complex environment

• In the end we used 240Mb for the entire demo, including:
  – 54Mb for back / depth buffers and offscreen buffers (many were 101010-2)
  – 156Mb for texture memory
  – 28Mb for object vertex and index buffers
Rich Detailed Worlds Require Complex Shaders

- We created ~500 custom unique shaders for ToyShop
  - >1/2 of those are rain / water / wet surfaces shaders
  - ~1/3 draw depth only or render reflections
  - ~1/6 relate to post-processing effects
  - Particle effect shaders
  - We used ~20 various shared include files (reflection / shadow mapping / HDR / math functions / lighting / lightning / etc)
Data and Rendering Statistics

• 3Dc+ texture compression was crucial
  – We went from **478 MB** texture memory total to **156Mb**

• We used **dec3n** vertex data format to reduce memory
  – Gives 3:1 memory savings
  – For vertex normals / tangents / binormals / other data
  – Available on ATI Radeon line since Radeon 9700

• A lot of work went into optimization of draw calls:
  Averaging 250

• Every frame we render 200K – 500K polygons and 0 – 22K particles
Parallax Occlusion Mapping versus Normal Mapping

Scene rendered with Parallax Occlusion Mapping

Scene rendered with normal mapping
Surface Details in the ToyShop Demo

• Parallax occlusion mapping was used to render extreme high details for various surfaces in the demo
  – Brick buildings
Surface Details in the ToyShop Demo

- Parallax occlusion mapping was used to render extreme high details for various surfaces in the demo
  - Brick buildings
  - Wood-block letters for the toy shop sign
Surface Details in the ToyShop Demo

- Parallax occlusion mapping was used to render extreme high details for various surfaces in the demo
  - Brick buildings
  - Wood-block letters for the toy shop sign
  - Cobblestone sidewalk
- Using multiple lighting models
  - Some just used diffuse lighting
  - Others simulated wet materials
  - Integrated view-dependent reflections
  - Shadow mapping was easily integrated into the materials with parallax occlusion mapped surfaces
- All objects used the level-of-details system
Parallax Occlusion Mapping

- Inverse displacement mapping via ray casting in texture space
  - High precision of height field – ray intersections
- Dynamic real-time lighting of surfaces
  - With soft shadows due to self-occlusion under varying light conditions
- Directable level-of-detail control system
  - Smooth transitions between levels
- Motion parallax simulation with perspective-correct depth
Would Like More Details?

- See the paper

- Check out DirectX 9.0c SDK sample (April 2006 and later)
  - Parallax Occlusion Mapping
  - [www.microsoft.com/directx](http://www.microsoft.com/directx)
The Challenge of Rendering Rain

- Rain is *difficult*!
  - Creating convincing impression of rain in rich natural environments is a difficult task

- It is a complex phenomenon
  - Vast diversity of rain components
  - Huge amount of small details

- But - rain rendering greatly enhances outdoor scenes!
  - Many applications in games and motion pictures
  - Challenging to film and even more so to render at interactive rates
Components

• Modeling and rendering rain precipitation
  – Rainfall, raindrops and rain splashes

• Water simulation for puddle rendering
  – Fluid dynamics
  – Approximation effects

• Atmospheric effects
  – Lightning illumination
  – Fog and multiple scattering effects

• Rain droplets simulation and rendering

• Reflections rendering
Rendering Complex Environment: Without Rain
Rendering Complex Environment: With Rain
Novel Algorithms for Rendering Individual Rain Effects

Post-Processing Rainfall Algorithm

Raindrop splashes

Simulation and rendering of water droplet movement on glass surfaces on the GPU

Simulation and rendering of dripping raindrops

Eurographics Animation Festival – Best Of. Vienna, Austria, 2006
Novel Algorithms for Rendering Individual Rain Effects

Misty objects in the rain (Halos around light sources and objects due to light scattering)

Streaming water

GPU-Based Water Surface Simulation for Puddle Ripples due to Splashes

Atmospheric Light Attenuation

View-Dependent Warped Reflections

Eurographics Animation Festival – Best Of. Vienna, Austria, 2006
Creating Rainfall with Multiple Layers of Rain

• Novel image-based algorithm to render composite rainfall
  – The artists specify the rain direction and speed in world-space to simulate different rainfall strength
• Model multiple layers of rain in a single pass with a single texture fetch from the rainfall position placement texture
  – Allows simulation of raindrops falling with different speed at different layers
• The layer of rain is shaded by using a normal map of varied individual raindrop shapes using scene lights
GPU-Based Water Simulation

- Dynamic realistic wave motion of interacting ripples over the water surface
- Water ripples are generated as a result of rain drops falling onto the geometry in the scene
  - Stochastically seeded into the simulation texture as droplet mass
- Entire simulation runs on the GPU using SM 2.0 shaders
  - Solve for water heights using explicit Euler integration
Water Droplet Animation and Rendering on Glass Surfaces in Real-Time on the GPU

• Adopted an offline raindrop simulation system from [Kaneda99] to the GPU
  – Discretize glass surface to simulate on a grid
  – Dynamically animate and render a large number of water droplets on glass surfaces on the GPU
  – The simulation is modified to use a gather pass in the pixel shader, rather than original scatter-based particle system implementation

• The droplet shape and motion is influenced by the forces of gravity and the interfacial tension forces, as well as air resistance
View-Dependent Streaking Reflections

- Realistic streaky reflections increase the feel of rain on wet street and surfaces
  - Very prominent in any rainy scene
  - Appear to elongate toward the viewer
  - Much more saturated for brighter sources
Want More Details?

• See the paper:
  – SIGGRAPH 2006 Course 26 “Advanced Real-Time Rendering”
    • Course notes, Chapter 3
Performance

• Measured on 1GB Dual 3.2 GHz Pentium 4 PC with ATI Radeon X1900 XT with 512MB of video memory

• Achieve frame rates of 26-69 fps
  – Rendering time for the raindrop particles and splashes was limited by the CPU
ToyShop Revealed: Summary

- We presented you some of the technology that was developed for the ToyShop demos
- Rich, complex environments demand convincing details
- Lots and lots of custom shaders allowed us to create a thorough illusion of a dark, rainy night in the city
- We hope to see even better environments in next gen games and interactive applications
The ToyShop Team

Lead Artist
Dan Roeger

Lead Programmer
Natalya Tatarchuk
David Gosselin

Artists
Daniel Szecket, Eli Turner, and Abe Wiley

Engine / Shader Programming
John Isidoro, Dan Ginsburg, Thorsten Scheuermann and Chris Oat

Producer
Lisa Close

Manager
Callan McInally

ATI
Questions?

natasha@ati.com