# Deferred Rendering

Erik Gunnarsson & Magnus Olausson

# Overview

- Forward vs Deferred Rendering
- Deferred Shading
  - Pros
  - Cons
- Deferred Lighting
- Real world applications

# Forward rendering

- Traditional method
- Single pass
  - For each object
    - Find all lights affecting object
    - Render all lighting and material in a single shader
  - Shader for each material vs. light setup combination
  - Wasted shader cycles
    - Invisible surfaces / overdraw
    - Triangles outside light influence

**Most of the text in this slide is extracted from a presentation by GUERRILLA GAMES from DEVELOP CONFERENCE, JULY ’07, BRIGHTON**

# Forward rendering (cont.)

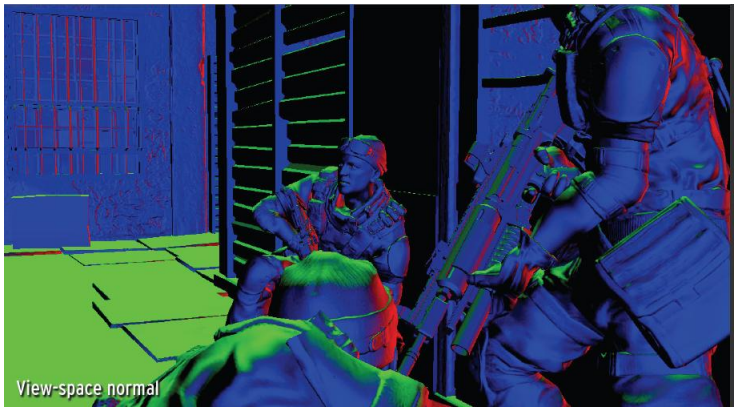
- Solution to material/light combination issue
- Multi-pass
  - For each light
    - For each object
      - Add lighting from single light to frame buffer
  - Shader for each material and light type
  - Wasted shader cycles
    - Invisible surfaces / overdraw
    - Triangles outside light influence
    - Lots of repeated work
      - Full vertex shaders, texture filtering

**Most of the text in this slide is extracted from a presentation by GUERRILLA GAMES from DEVELOP CONFERENCE, JULY ’07, BRIGHTON**

Images in this slide is extracted from a presentation by GUERRILLA GAMES from DEVELOP CONFERENCE, JULY '07, BRIGHTON. http://www.guerrilla-games.com/publications/dr_kz2_rsx_dev07.pdf

# Deferred Rendering

- 1. For each object
  - Render surface properties into the G-Buffer



**Images in this slide is extracted from a presentation by GUERRILLA GAMES from DEVELOP CONFERENCE, JULY ’07, BRIGHTON**

# Cont.

- 2. For each light and lit pixel
  - Use G-Buffer to compute lighting
  - Add result to frame buffer
- 3. Render Transparent Stuff (using forward rendering)

# Deferred Rendering Pro

- Complexity
- Shades only visible pixels
- Few shaders
- Post-processing stuff ready
- Lots and lots of Lights!

# Deferred Rendering Con

- Lots of memory
- Bandwidth!
- Transparency
  - G-buffers store one value per pixel
- Antialiasing
  - MSAA

# Deferred Rendering

- Not always a win
- Type of scene (general usage)
- If you have many dynamic lights!
  - Night scenes
- Not so much with many directional lights/outdoor scenes

# Solutions

- Antialiasing
  - Edge detection
  - MSAA
- Compression
  - G-buffers
    - Extract information
    - Different size on MRT (when possible)
- Which leads us to deferred lighting

# Deferred Lighting

- **Light Pre-Pass**
- normal vector **n** and specular spread factor *m* into a buffer. (depth as well)
- Render “light shapes”, evaluating diffuse and specular shading equations and writing the results into separate specular and diffuse accumulation buffers.
- Render opaque scene geometry a second time, reading the diffuse and specular accumulation buffers from textures, modulating them with the diffuse and specular colors of the surface
- Render any semitransparent geometry

Info from http://www.realtimerendering.com/blog/deferred-lighting-approaches/

# What does it mean?

- Less strain on bandwidth with smaller buffers
- Can manage without MRT

- But, need to render scene twice
- Need two buffers for accumulation pass (one for diff and one for spec) – can use combined buffer with cost in correctness

# Misc

- Both shading and lighting possible on 360 and PS3
- Tiled-deferred: keep track in which tile is which light. Then you render with the GPU all lights per tile. That solves substantial bandwidth and ROP problems

# Which?

- Deferred lighting or deferred shading?
  - Multiplatform?
  - Xbox360 or PS3?

# Deferred lighting

- Uncharted
- Resistance 2
- Crysis 2

# Deferred shading

- Killzone 2
- Starcraft 2

# Deferred lighting or shading

- Battlefield 3 (tiled)
- InFamous
- Little Big Planet
- GTA IV
- Halo Reach
- Dead Space

# Video

- Killzone 2
- Cryengine 3
- Unreal 3

- Also, interesting read with comments:
  - http://gameangst.com/?p=141

# Tasks

- Why do you want Deferred Shading?
- What are the pros/cons?
- Describe the algorithm
- What is the difference between deferred shading and deferred lighting