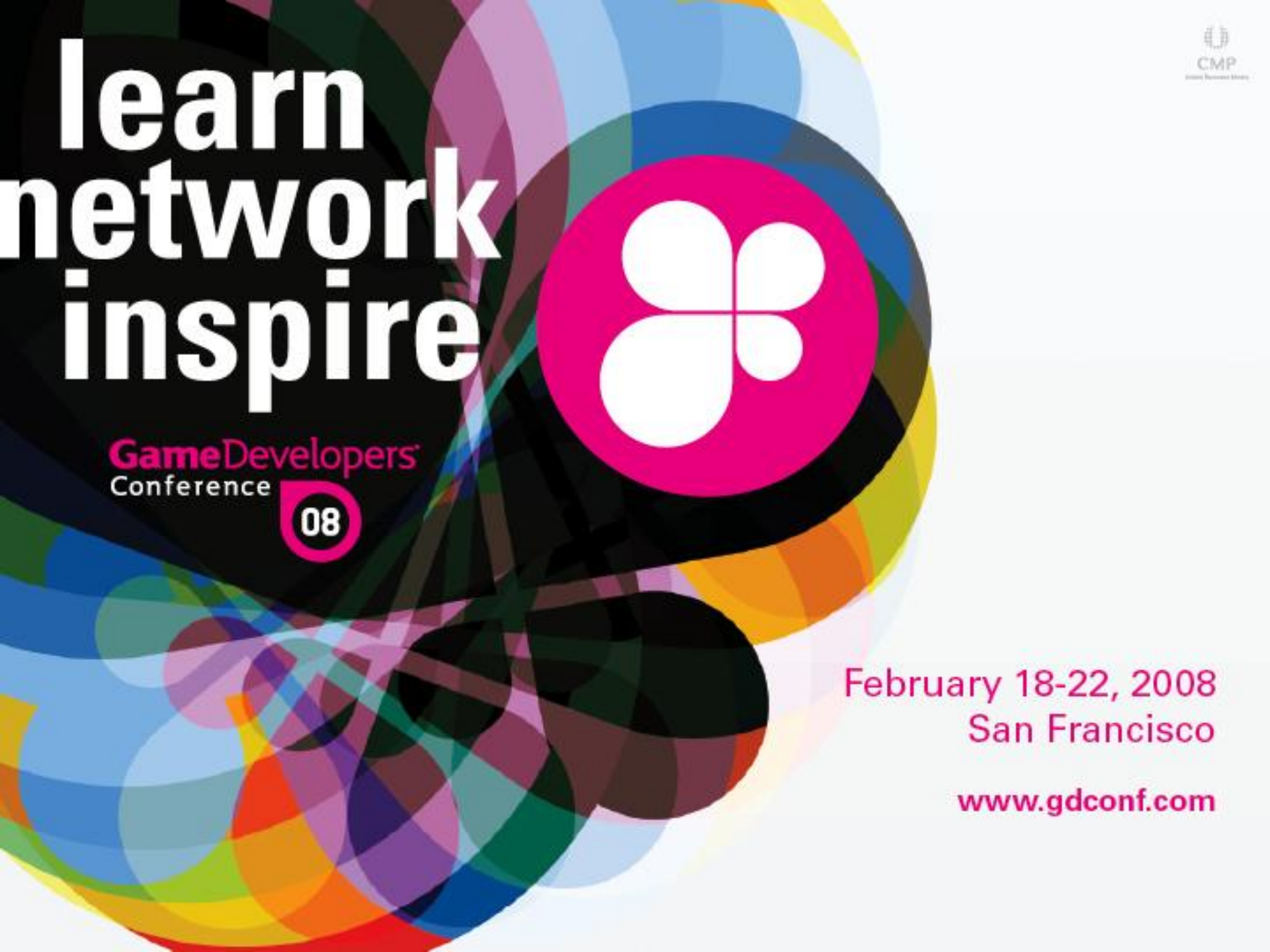

learn
network
inspire
GameDevelopers
Conference
08
CMP
February 18-22, 2008
San Francisco
www.gdconf.com

DirectX 10.1

"DirectX 10 and then some..."

Richard Huddy
Richard.Huddy@AMD.com

With thanks to Chris Oat of AMD who put together most of this material...

Introduction to DirectX 10.1

- Super-set of DirectX 10 functionality
- Adds new features
- Adds new minimum requirements
- Backwards compatible with DX 10
- Ships with Windows Vista SP1

Creating a DirectX 10.1 Device

- **`D3DX10CreateDevice(…)`**
  - First tries to create a 10.1 device
  - Falls back to a 10.0 device
- **`D3DX10GetFeatureLevel1(…)`**
  - Tells you if you created a 10.0 or 10.1 device
- The 10.1 device interface inherits from 10.0
- The Direct3D 10.1 DLLs support both 10.0 and 10.1 hardware
  - Some function calls may fail on 10.0 devices where new functionality is not supported

New Minimum Requirements

Feature	DirectX 10.0	DirectX 10.1
Shader Model	4.0	4.1
MSAA Samples	None	4x
VS Inputs	16	32
Blending	**INT8** **FP16/32**	**INT8/16** **FP16/32**
Filtering	**FP16**	**FP32**

New Features

- Cube map arrays
- Separate blend modes (per-MRT)
- PS can export coverage mask
- 32 shader inputs/outputs
- SSAA by running PS at sample freq.
- New shader instructions
- Standardized MSAA sample patterns
- **`CopyResource`** restrictions lifted
  - Single-sample depth/stencil surfaces
  - Multi-sample color & depth/stencil surfaces

New Possibilities!

- DirectX 10.1 features are important for improving image quality:

- Deferred shading with MSAA
- Alpha test (billboards) with MSAA
- Better reflections/GI approximation

Deferred Shading: Review

- Shading calculations depend on parameters like: normals, positions, and albedo
- Strategy:
  - Store parameters in image space (called G-Buffer)
  - Run complex shaders as post-process
- Hence, the shading is deferred
- The advantage:
  - Decouples shading complexity from geometric complexity
  - No expensive shading computation done on occluded samples

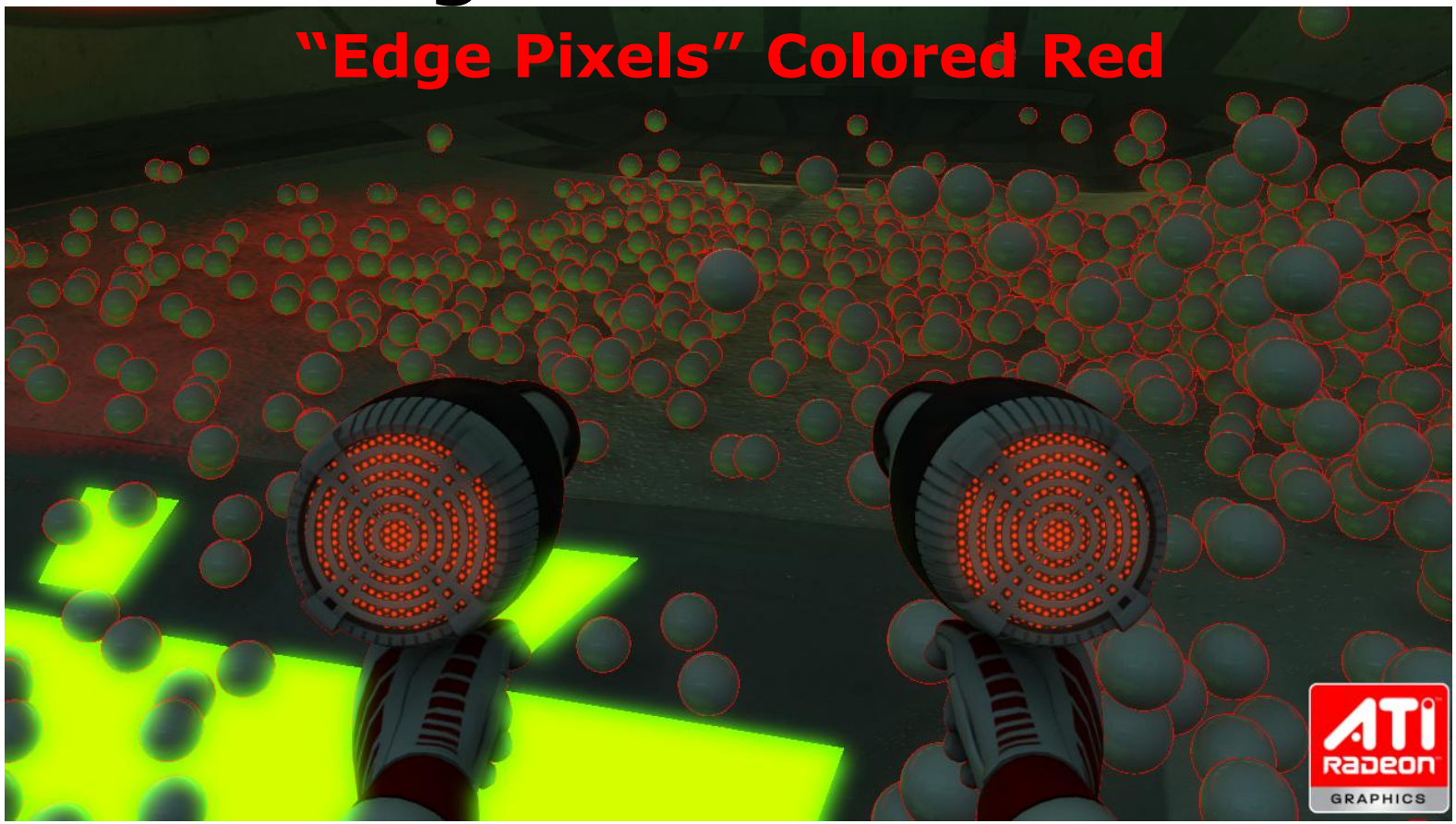
Deferred Shading: Compare

- On DX10.0 gaining access to the AA samples requires considerable extra effort
  - (Write them to a single channel MRT colour buffer as you update the Z buffer, then re-use that color buffer in subsequent passes)
  - [Wasted extra memory this way.]
- DX10.1
  - Write the Z once, no duplication of data, no extra shader variants to copy Z to the color channel

Why Now?

- We finally have everything necessary to do this correctly
  - Can output to MRTs (form G-buffer)
  - Can output to high-precision surfaces
  - MSAA problem solved by DirectX 10.1
- DirectX 10.0
  - Allows access to Multi-Sampled Color Buffers
- DirectX 10.1
  - Allows access to Multi-Sampled Depth Buffer
    - Use depth and inverse view-projection matrix to compute per-sample world space position
    - Detect per-sample depth discontinuities
      - Shade at fragment rate when depth samples are coherent
      - Shade at sample rate when depth samples differ

Resolving the G-Buffer

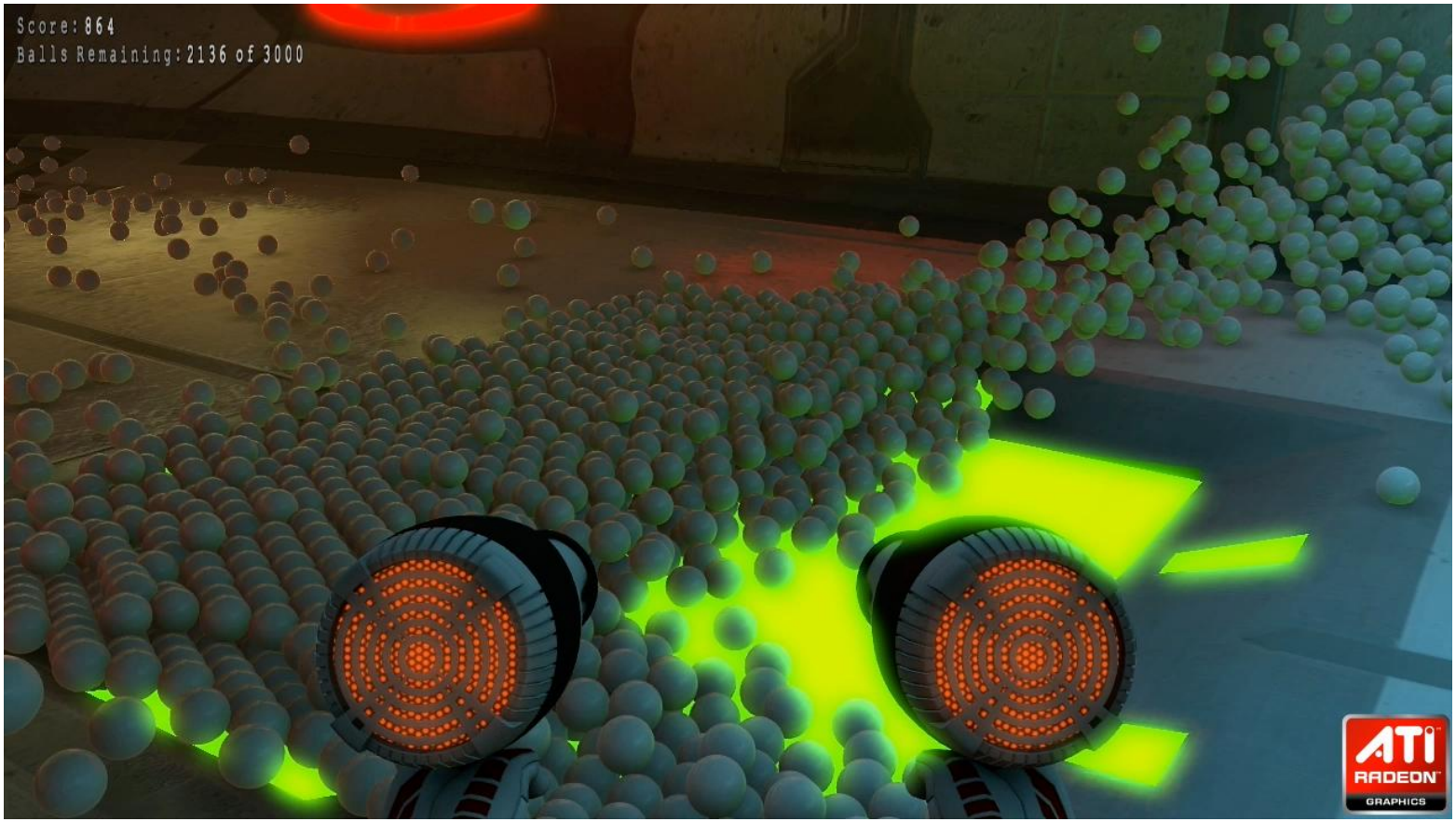


- Compare depth value at pixel's samples
- Find edge: Shade samples then average
- No edge: Shade a single sample

Alpha test with MSAA

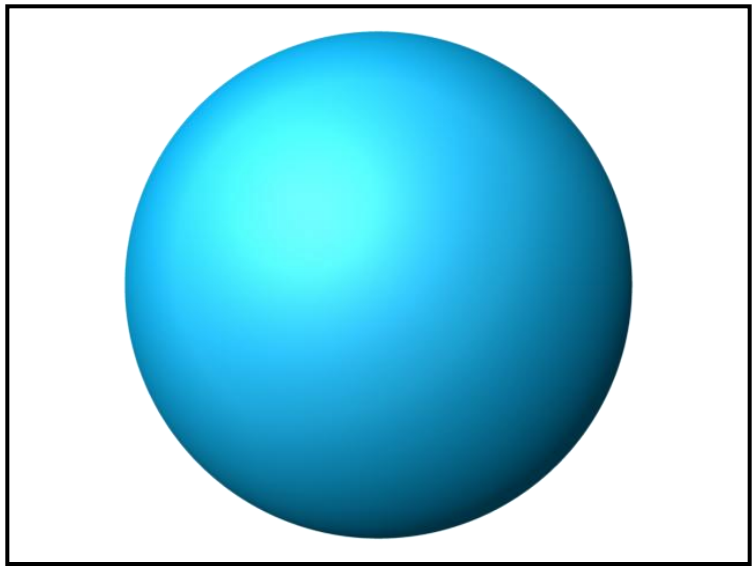
- Sample your alpha mask once per-sample
- Mask off samples that fail the test
  - DirectX 10.1 uses a standardized sampling pattern so you know where the samples are
- You effectively get the alpha test running at the sample rate even though your shader runs at fragment rate
  - Efficient anti-aliased alpha-tested edges

Anti-Aliased Billboards



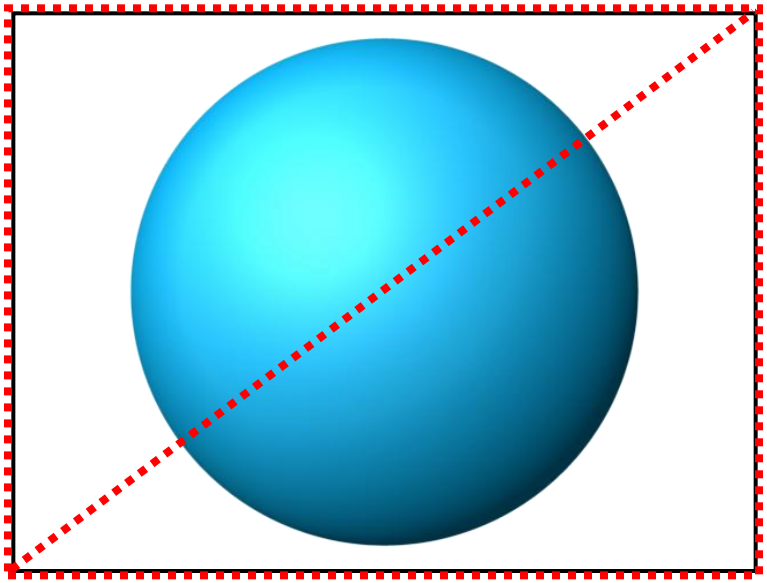
- Thousands of ping pong balls drawn as imposters

Anti-Aliased Billboards



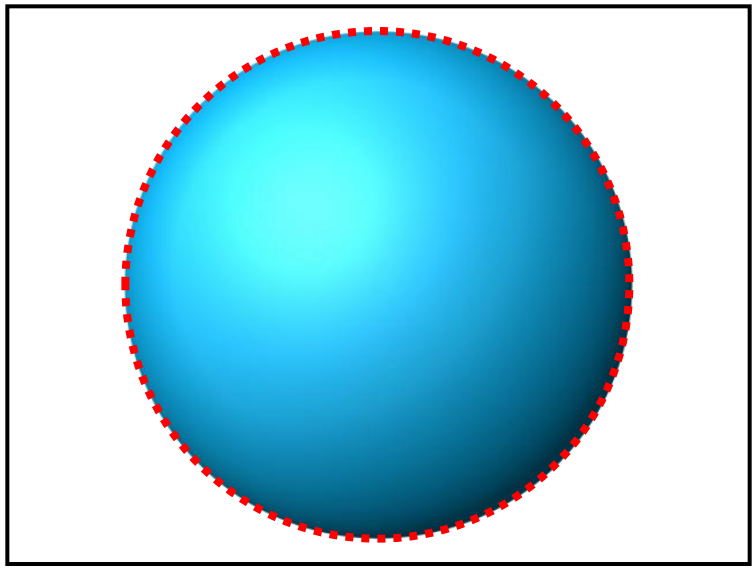
- We want to draw a sphere as an alpha-tested imposter with MSAA

Anti-Aliased Billboards



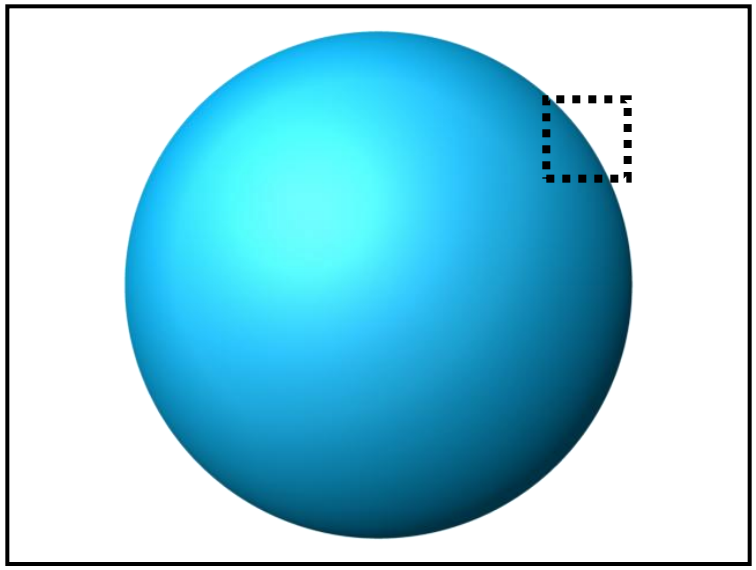
- But MSAA only works on triangle edges

Anti-Aliased Billboards



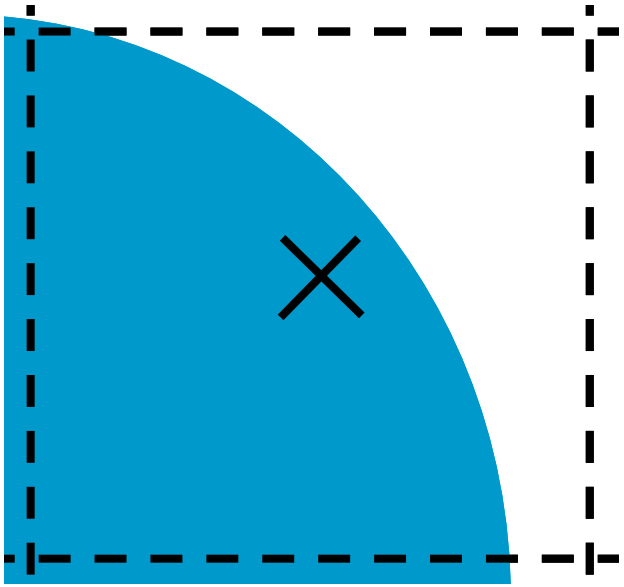
- We want MSAA to happen on the alpha-tested edges of the imposter

Anti-Aliased Billboards



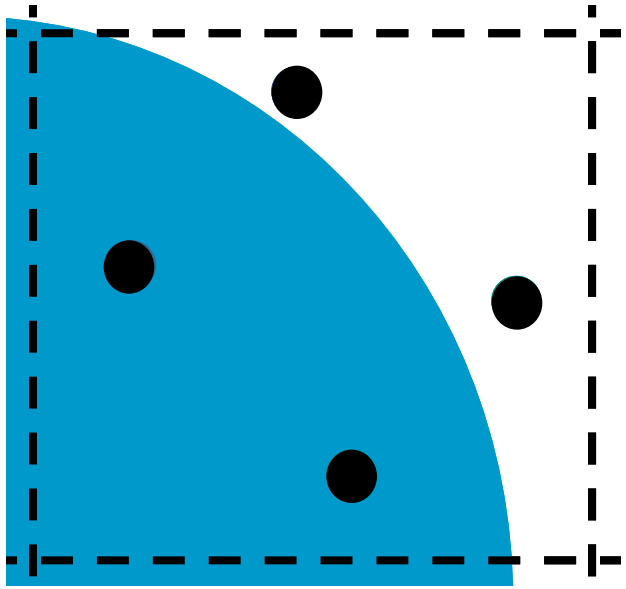
- The hardware considers a single pixel inside the imposter to be fully covered
- Shader is executed at the pixel's center

Anti-Aliased Billboards



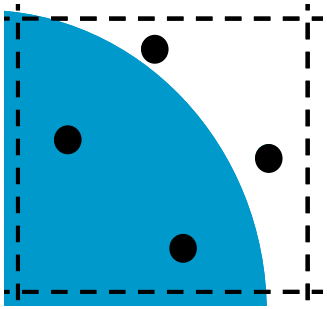
- If the pixel center is “inside” the alpha-tested sphere, then the entire pixel is drawn

Anti-Aliased Billboards

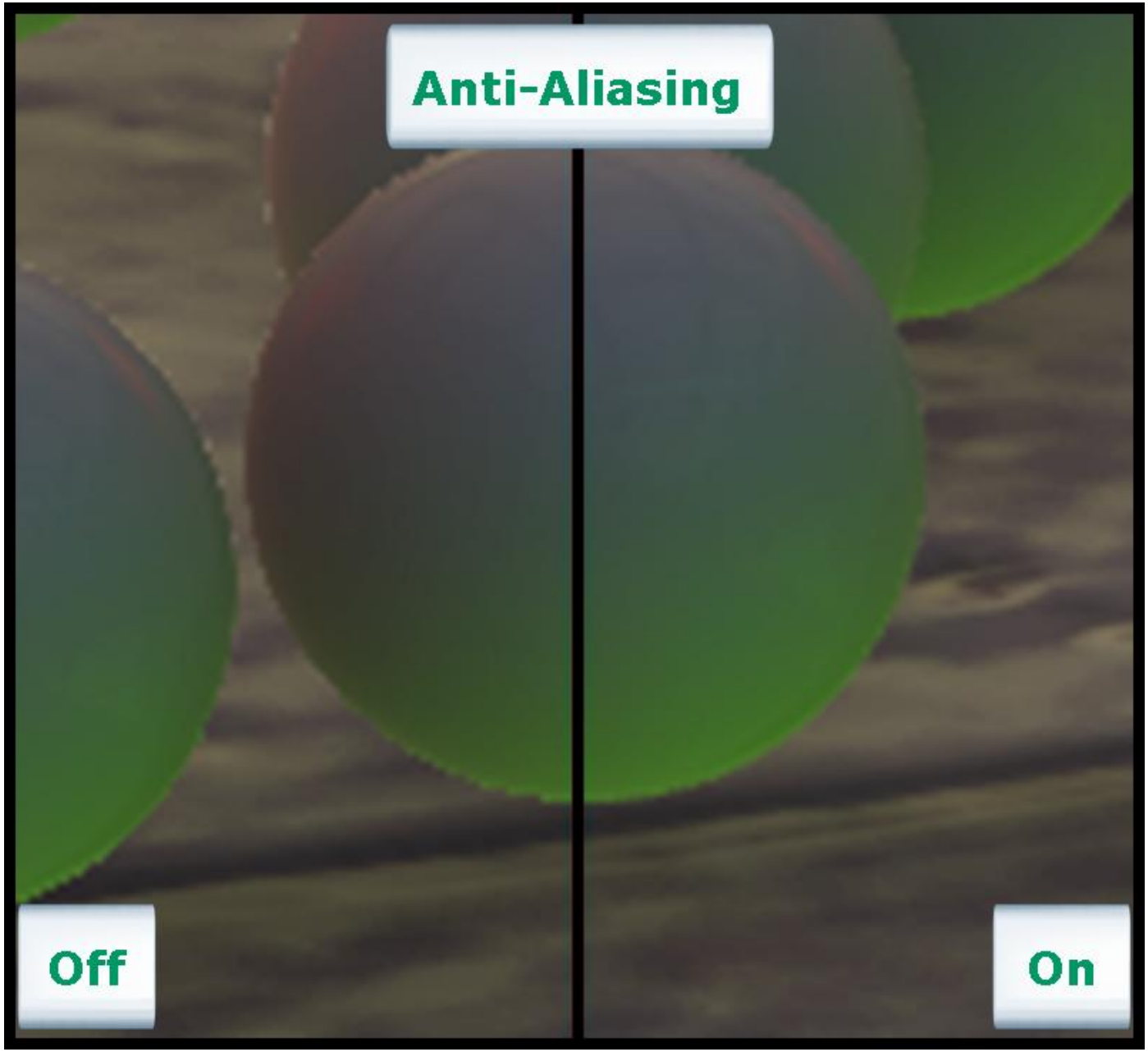


- We would like to alpha test at sample points
- But we do not want to shade the entire billboard at sample frequency

Anti-Aliased Billboards



- Derivative instructions used to find UV coordinates at each sample
  - Sample locations are standardized in DirectX 10.1!
- Sample texture once for each MSAA sample
- Set the sample mask for each sample that passes the alpha-test


Anti-Aliasing
Off
On

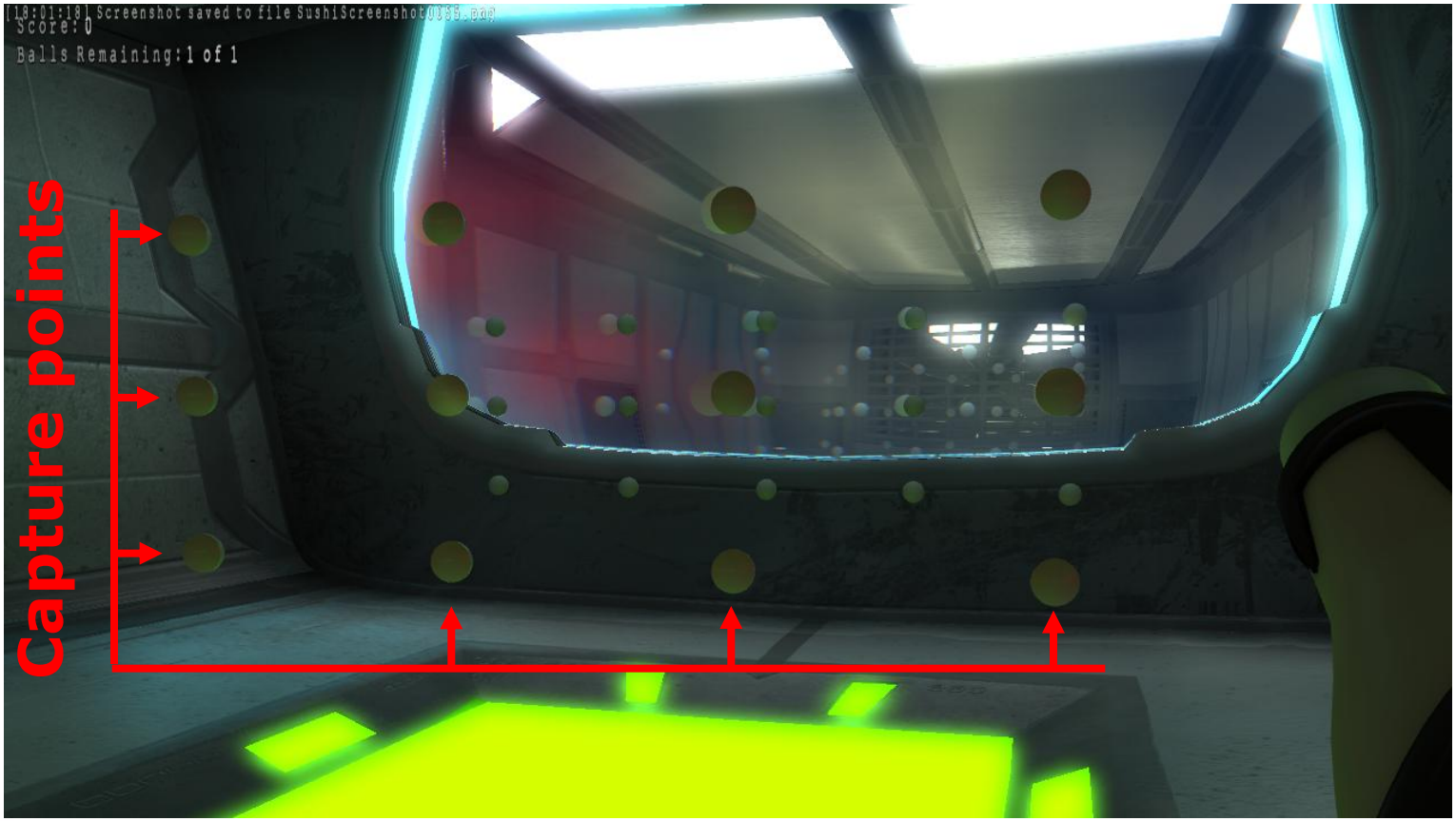
Accurate Reflections

- Cube maps capture environment at a single point
- Used to approximate reflections at many points
- Reflections become less accurate as you move away from capture point
- Changing between cube maps in DirectX 10.0 requires a state change and increases draw call count

Cube Map Array

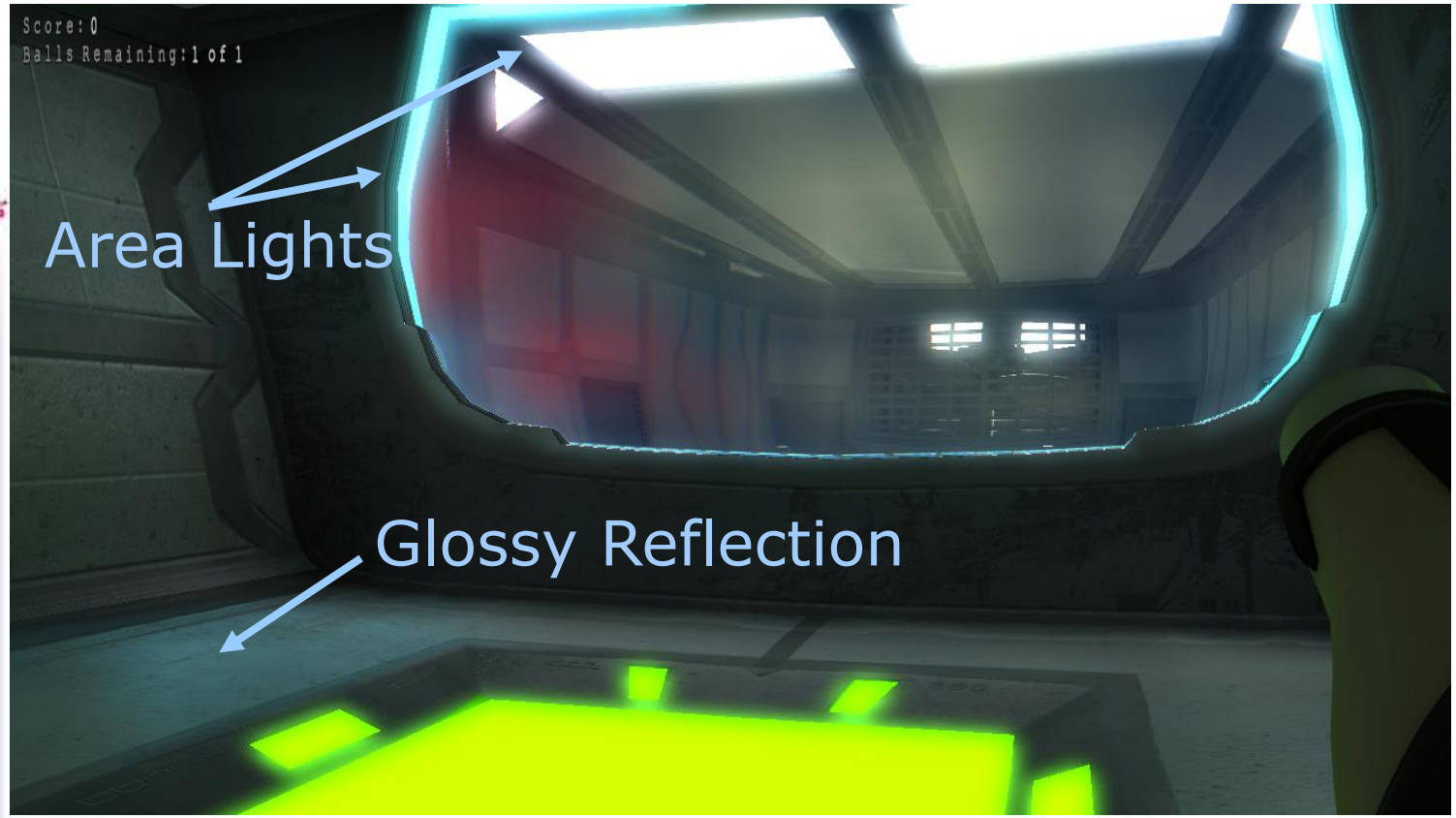
- Store many capture points
  - Shiny objects can pick closest cube map
  - Or blend between the closest *n* cube maps
- Pixel shader has access to every capture point in a scene!
  - No state changes!
- Can be used to approximate Global Illumination

Cube Map Array



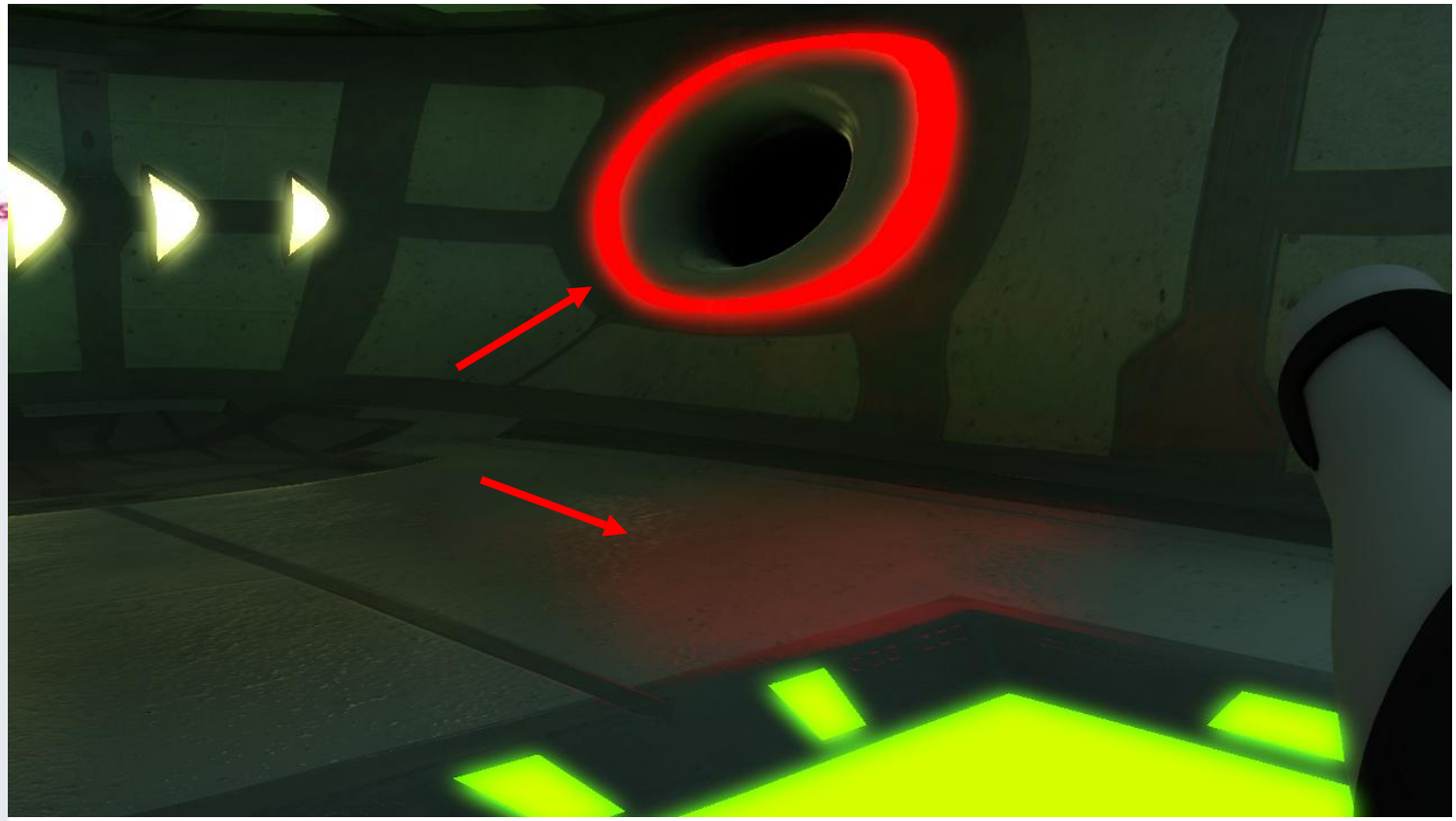
- Environment captured from many points
- Uniform volume of light probes (radiance cache)

Cube Map Array



- Gives spatially correct glossy reflections
- Works with area lights (anything you can render into the light probes)

Cube Map Array



- Gives spatially correct glossy reflections
- Works with area lights (anything you can render into the light probes)

Cube Map Array

- Convert subset of light probes to Spherical Harmonics
- Use for diffuse reflections

Cube Map Array

- Convert subset of light probes to Spherical Harmonics
- Use for diffuse reflections

Access to the AA Z buffer

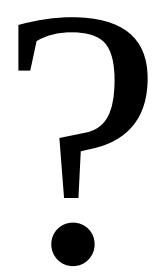
- Application now has access to the AA Z buffer too…
  - For high quality shadows at minimal extra cost
  - Can reduce shader complexity and eliminates extra passes

Conclusion

- DirectX 10.1 adds valuable, simplifying functionality
- Improves image quality
- Improves efficiency
- Insanely easy to add support to your application if you already support 10.0

Thank you!

- Questions?



- Richard Huddy
- Richard.Huddy@amd.com