



**TAKE  
CONTROL**  
[www.gdconf.com](http://www.gdconf.com)

MARCH 5-9  
**2007**  
SAN FRANCISCO

MOSCONE  
CENTER



CMP



# 3D Tricks: Engineering Innovation on the Nintendo DS

Chuck Homic & Greg Oberg  
Vicarious Visions



# Who are we?

- Vicarious Visions
  - An Activision Studio
  - 10+ years in handheld and console development
  - Handheld experience in Game Boy, Game Boy Color, Game Boy Advance, PSP, **Nintendo DS**
  - Console experience in PlayStation, N64, Dreamcast, PS2, Xbox, Game Cube, PS3, Xbox 360, Wii



# Who are we?

- Greg Oberg
  - Extensive handheld experience
- Chuck Homic
  - PC, console heritage
  - Applying lessons learned to DS



# Who are we?

- We love the DS
  - As capable as 1<sup>st</sup> gen 3D consoles
  - Lightweight transparent OS/SDK that helps you, doesn't get in your way
  - Battery friendly
  - Lots of cool hardware features
  - SDK source code provided by Nintendo!



# Agenda

- Nintendo DS Graphics Hardware
- Some DS Features in brief
- Toon Shading and Edge Marking
- Fog
- Texture Matrix
- Environment Mapping
- Texture Streaming
- 2D Billboards
- Questions



# Nintendo DS Graphics Hardware





# Nintendo DS Graphics Hardware

- Layer based
- Game Boy Advance 2D hardware
  - Sprite layers
  - Tiled or bitmap layers
- Custom 3D graphics system
  - Output is just another layer
- Layers are sorted and blended into final image





# 3D Graphics Limits

- 2048 polygons per frame\*
- 6144 vertices per frame\*
- N polygons per scan line
  - Can draw between 4 and 8 times per pixel depending on # of polygons
  - Exact formula in programming manual
- Up to 512K for textures

\*Can be reduced due to clipping



# Don't Be Limited

- DS is falsely regarded as an underpowered platform
- Developers are used to throwing a lot of polygons at the problem
- There are a lot of other features available
- Use them!



# Some DS Features

- Dynamic lighting
  - Directional lights
  - You can simulate point lights
- Alpha
  - Auto sorting available
- Stencil Shadow
  - 1-bit stencil buffer
  - That's all you need for convex shadow volumes
  - Can create really nice detailed shadow intersections



# Toon Shading and Edge Marking



# Toon Shading / Edge Marking

- Crisper / more visual “pop”
- Can be used to stylize game
- Takes advantage of the strengths of the DS
- It’s free!



**TAKE CONTROL**  
March 5-9, 2007 in  
San Francisco

# Toon Shading / Edge Marking




# Regular






Game Developers  
Conference 07

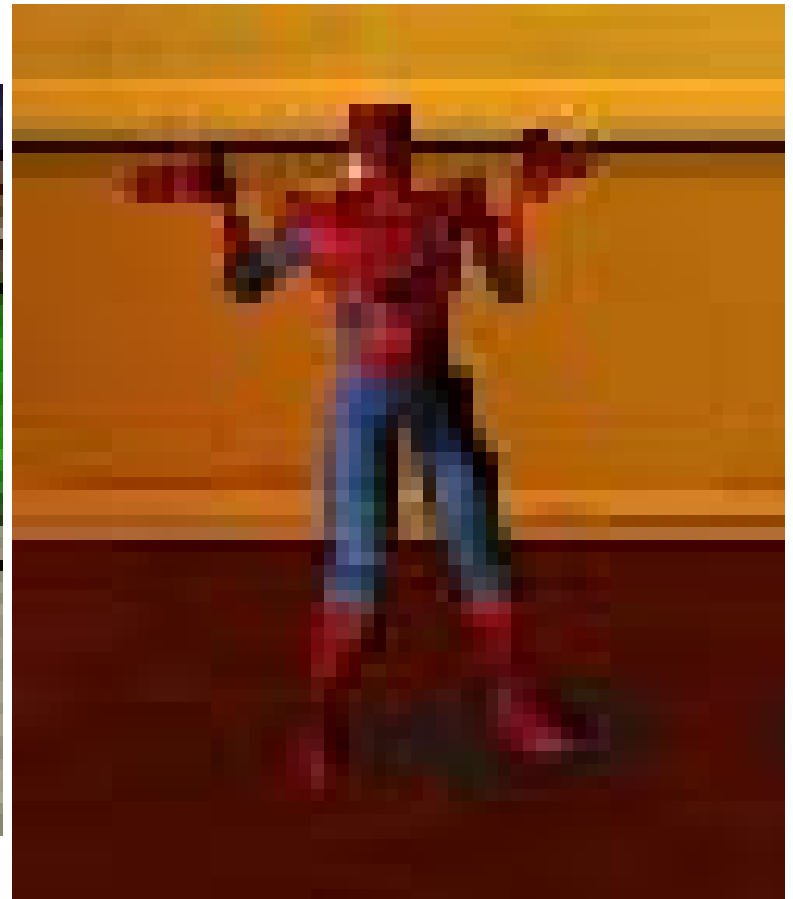


**TAKE CONTROL**  
March 5-9, 2007 in  
San Francisco



CMP  
Computer Music Products

# Toony and Not Toony





# Toon Shading

- Create a toon color table
  - The final vertex color will come from this table
- The hardware will...
  - Calculate initial vertex color value with lights & normals
  - Use the intensity as an index into the toon color table
- Customize toon table values to achieve desired visual effect



# Edge Marking

- Create edge color table
- Assign polygon(s) an ID
- The hardware will...
  - Draw an edge when different polygon IDs are adjacent in screen space
  - Determine the edge color by looking up the PolygonID of the closer polygon in the edge color table



# Edge Marking Techniques

- Create an “ink” effect by making entire edge color table black
- Highlight objects without using any precious polygons
- Change the edge color table every frame



# Movie

- 01\_oth\_edge\_highlight.avi
  - Shows the use of edge marking to highlight interactive objects
  - Interactive objects have a polygon ID which maps it to a particular entry in the edge color table
  - The color value for this entry is changed every frame



**TAKE CONTROL**  
March 5-9, 2007 in  
San Francisco

# Edge Marking Artifact





# Edge Marking Artifact Close-up







# Edge Marking Artifact

- Edge marking artifact at screen edges
- It's a known issue
- Can use DS window feature to hide (almost)
- Hide the top row with sprites or BG

# Edge Marking Artifact Repaired





# Fog



# Movie

- 02\_oth\_day.avi
  - Normal scene from Over The Hedge. No fog, because the game did not use fog.



# Movie

- 03\_oth\_night.avi
  - Hypothetical “night mode” in Over The Hedge. The real game did not have night mode, but this demonstrates that it could be done with absolutely no art or code changes, other than enabling fog.



# Fog

- Use as general purpose color filter
- Can be enabled or disabled on specific objects
- Allows an object to be emphasized



# Fog to Highlight

- *Due to legal reasons, we can't show the original examples*
- *We will demonstrate the concept via the following mock ups*





# No Fog





# Fog to Highlight





# Fog

- Distance based
- Differs from DirectX or OpenGL
- Traditional fog is applied to polygons
- DS fog is rendered after all polygons based on Z depth
- Can interfere with alpha

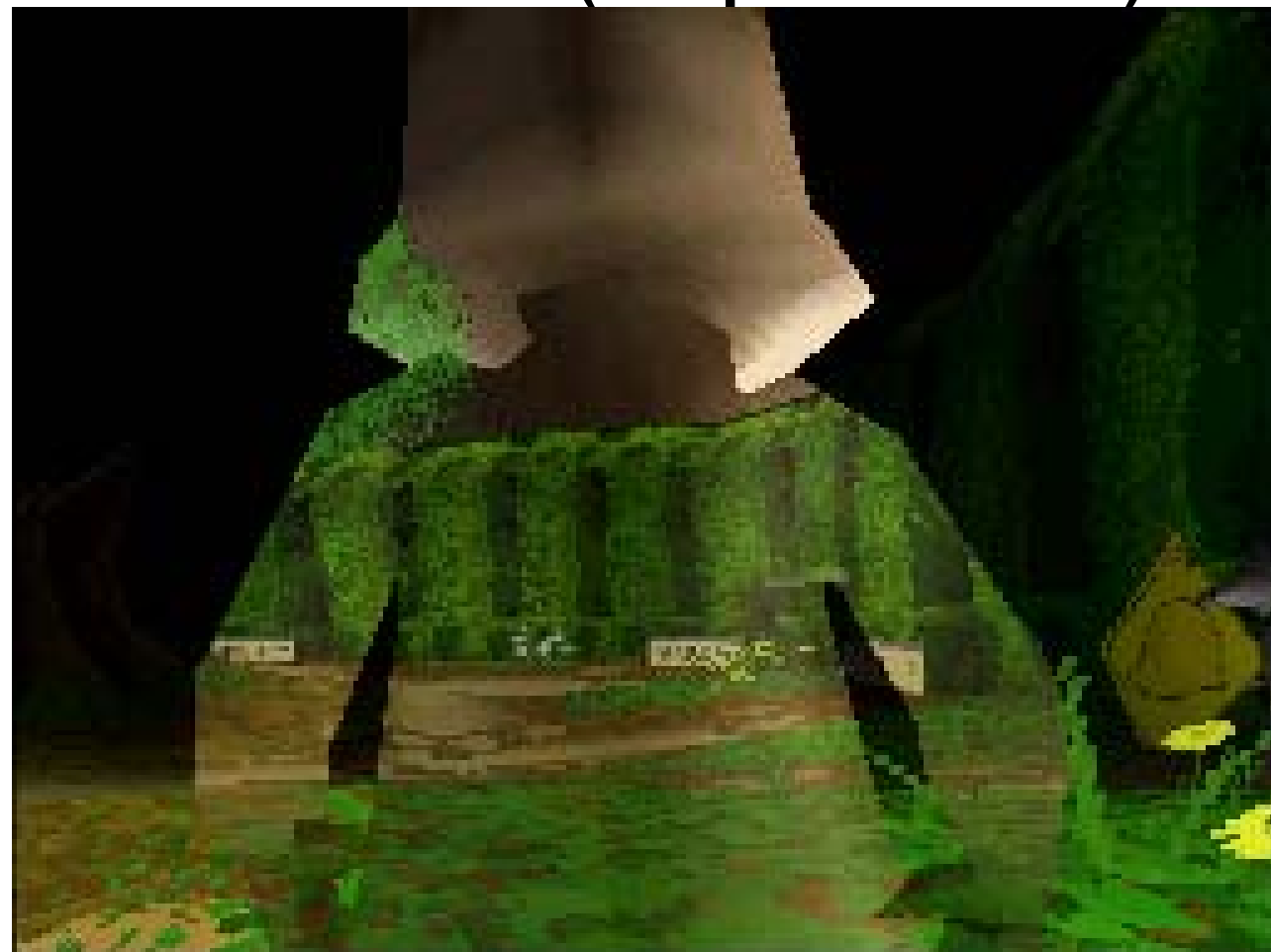
# Fog With Alpha Conflict (no depth write)





**TAKE CONTROL**  
March 5-9, 2007 in  
San Francisco

# Fog With Alpha Conflict (depth write)



# Fog With Alpha Conflict (depth write)





# Textures



# Texture Matrix Effects

- Low cost texture effects
- Scrolling, scaling, rotation





# Movie

- 04\_thdj\_river.avi
  - This is a simple texture scroll effect from Tony Hawk's Downhill Jam.



# Movie

- 05\_oth\_tv.avi
  - This is a more advanced example from Over The Hedge. The animation on the TV is done entirely with texture matrix animations. This dramatically reduces the amount of VRAM that would have been used to make this animated TV screen.



# Environment Mapping

- Vertex normals can be used as texture coordinates
- Produces a metallic reflection
- Not quite necessarily what we want



# Environment Mapping

- Use multi-pass rendering
  - Second pass with DTE bit
- Can be rather expensive use of limited polygons
- No additive blending mode
- Compensate with a clever texture



# Movie

- 06\_envmap.avi
  - This is a prototype of an environment mapping effect using multiple passes on a character from Over The Hedge.
  - The environment map is programmer art.
  - There are some vertex precision errors due to the haste in producing the demo. These errors are unrelated to environment mapping.



# Texture Streaming



# Texture Streaming: Why??

- Artists always want more!
- Some VRAM may be used for other purposes
- Make up your own excuse



# Texture Streaming: The DS was made for it

- Snappy ROM performance
- No seek time
- Clever DMA controller can transfer between ROM and VRAM
- No need to waste main RAM





# Texture Streaming: Potential Pitfalls

- Cannot render a texture bank while accessible to DMA
- Content must be designed for streaming
- Don't let the user see textures loading



# Movie

- 07\_oth\_texturestream.avi
  - In this demo, we have allowed the character to run through the walls (in order to break the texture streaming prediction). This allows us to see the texture streaming in action (which has been slowed down to enhance the effect).
  - 07b\_oth\_texturestream\_part2.avi if you want to see more of the same

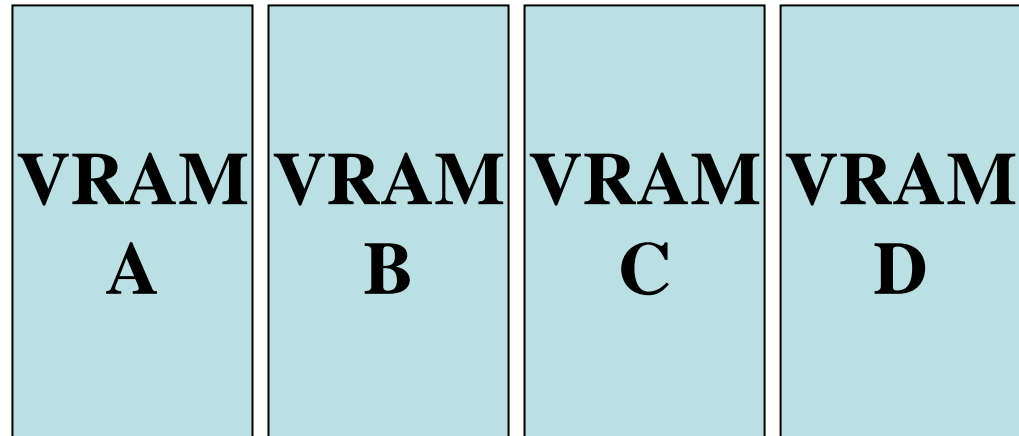


# Texture Streaming: Method 1

- Double buffer VRAM banks
- Render from bank(s) configured for texture use
- Load into bank(s) configured for DMA
- Textures load very fast
- Limits rendered textures
- We didn't actually do it



# Texture Streaming



**Frame Start**

**Frame End**





# Texture Streaming: Method 2

- All VRAM is available for drawing
- Load textures during “render blank”
  - What is render blank?
  - Is that different than vblank?
- It’s safe to access texture memory during render blank
- It takes several frames to load a bank



# Texture Streaming: Method 2 w/ One Time ROM

- What is One Time ROM?
  - Alternative write-once media format
  - Too slow for DMA during render blank
- Must load to RAM buffer during frame
- DMA from RAM to VRAM during render blank as usual

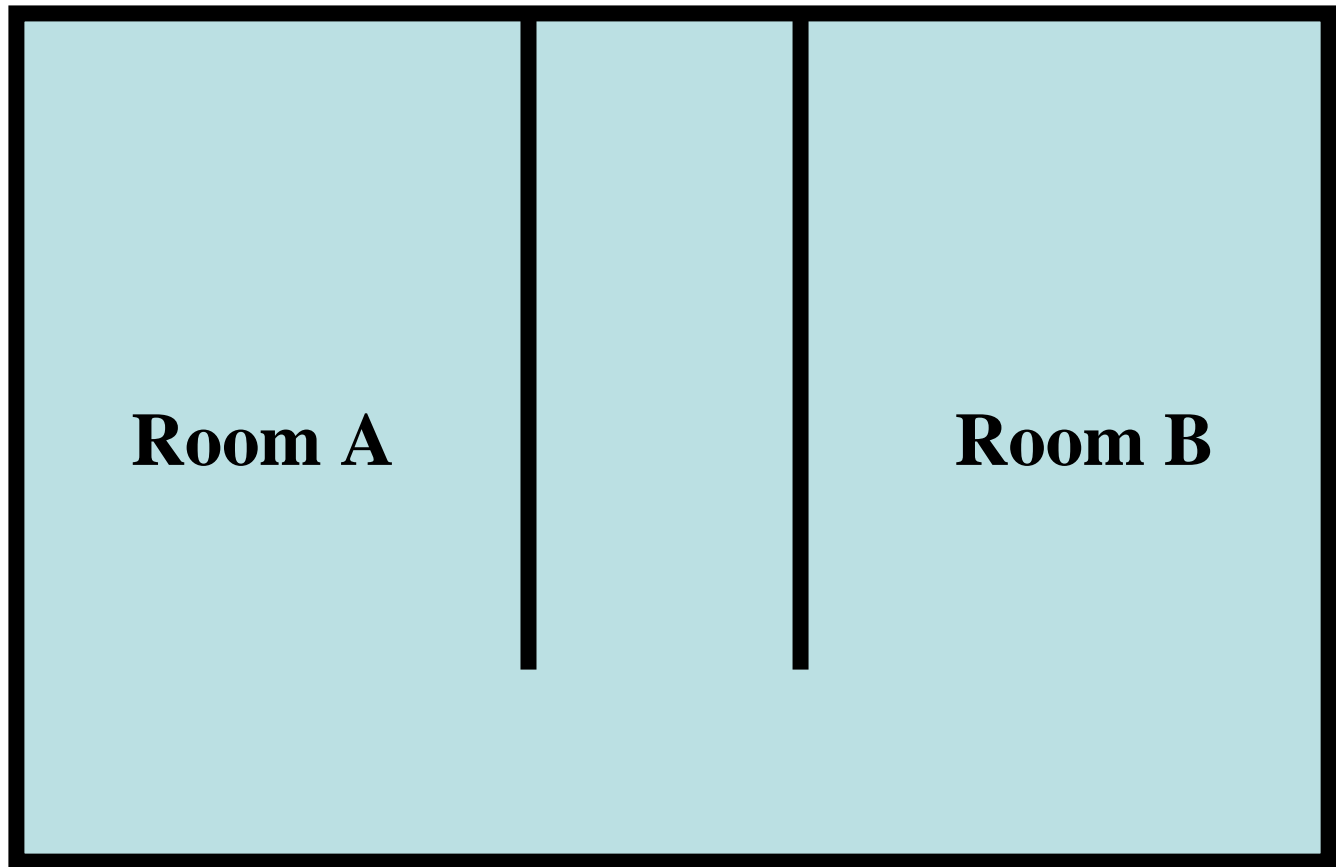


# Texture Streaming: Level Layout

- Your artists and designers need to know the limits
- Limit traversal
  - Don't let the user move to a new area faster than textures can load (half of a hallway)
- Limit line of sight
  - Don't show more textures than fit in VRAM



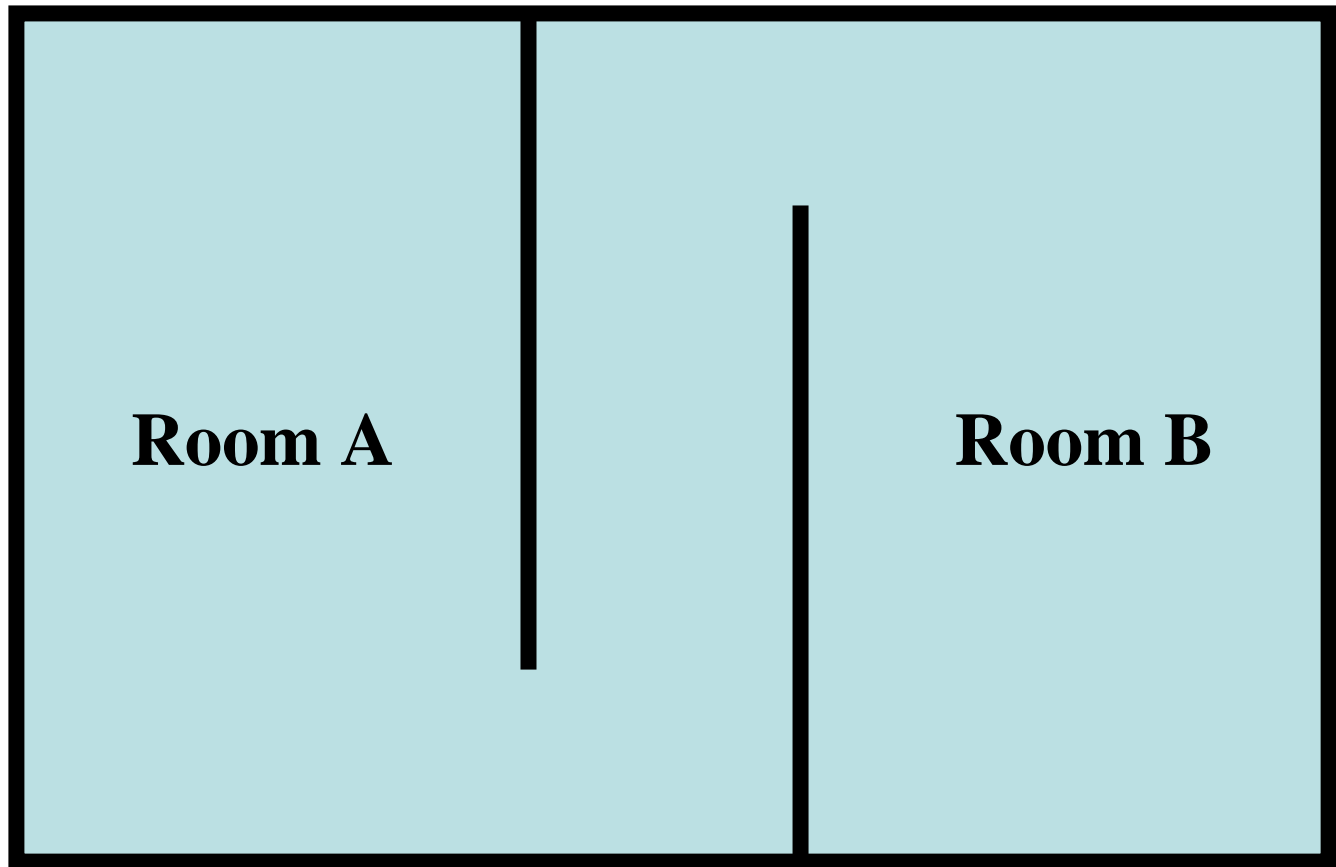
# Bad Level Layout







# Better Level Layout





# 2D Billboards



# 2D Billboards

- What is a sprite?
  - Hardware 2D image
  - Supports scaling, rotation
  - Rendered on a 2D layer
- We like it for light flares



# Movie

- 08\_spidey2\_lights.avi
  - Example of additive light flares in Spider-Man 2.



# 2D Billboards

- Advantages
  - Doesn't use polygons
  - Sprites can be truly additive
- Disadvantage?
  - Because it's a layer, it can't sort into 3D scene
  - We make it an advantage



# Movie

- 09\_spidey2\_lights\_slowmo.avi
  - Slow motion segment of light flare movie, so you can see that the flare sorts on top of all geometry.



# 2D Billboards

- Use PositionTest to find camera relative position
- Scale to screen coordinates
- Details are in NITRO Developer Brief #19



# Summary

- Polygons are not everything
- Take advantage of what the hardware has to offer
- Maximize the visual impact of every DS title





# Questions?

Email:

Chuck Homic [chuck@vvisions.com](mailto:chuck@vvisions.com)

Greg Oberg [greg@vvisions.com](mailto:greg@vvisions.com)

Discuss:

<nntp://news.sdsg.nintendo.com>

[nitro.graphics3d](http://nitro.graphics3d)