Hardware Overview

Steve Rabin

Software Development Support Group

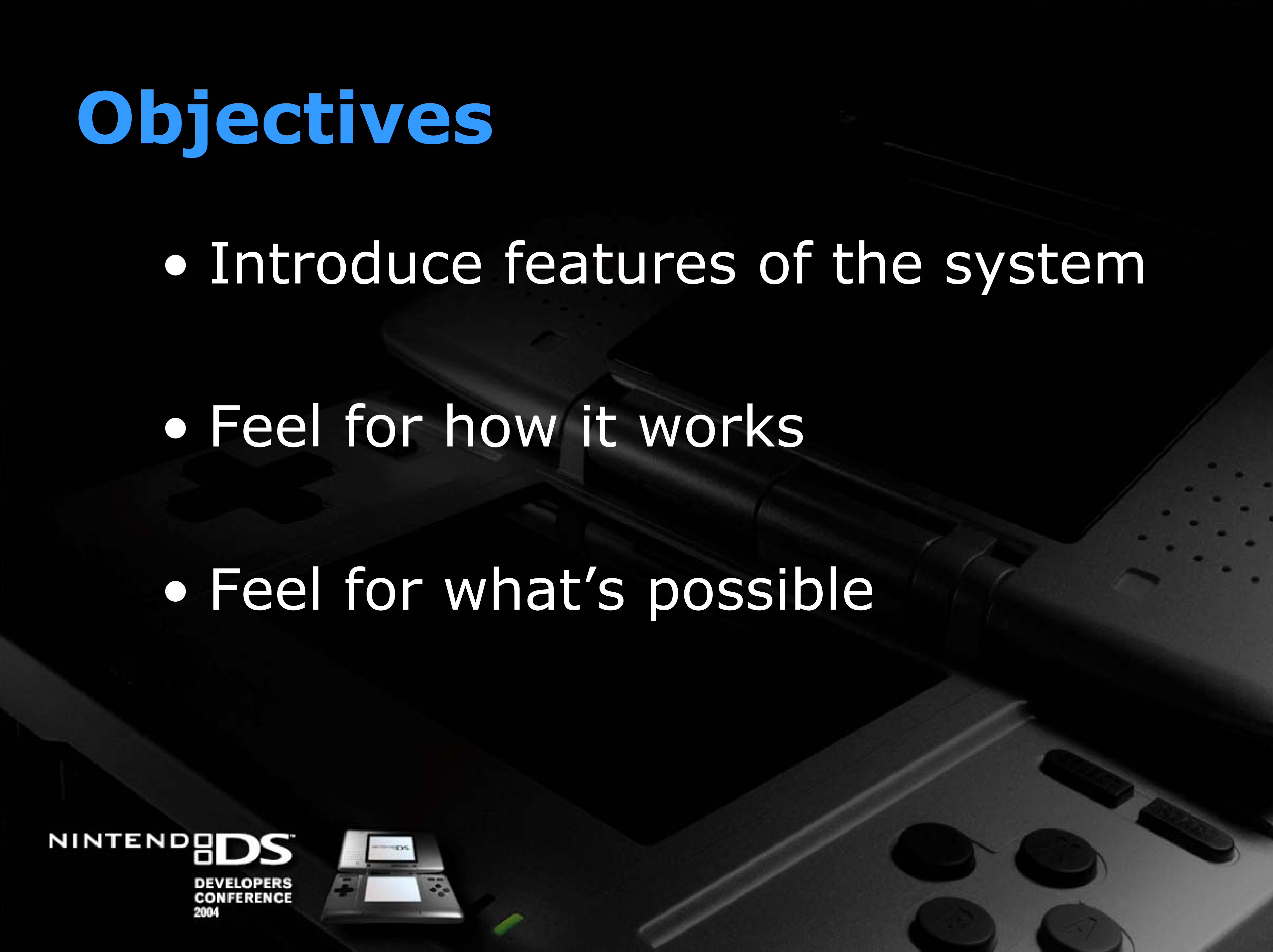
Nintendo of America Inc.

NINTENDO DS

DEVELOPERS CONFERENCE 2004

Objectives

- Introduce features of the system
- Feel for how it works
- Feel for what's possible



Defining Hardware Features

- Dual Backlit LCDs
  - 256x192, 18bit color (256K colors)
- 3D + 2D Graphics
  - 3D comparable to N64
  - 2D better than Game Boy Advance
- Touch Panel
  - Bottom screen, pixel-level resolution
- Built-in Microphone
- Wireless Connectivity (10m range)
  - Local mode up to 16 units (wireless boot possible)
  - IEEE 802.11 Wireless LAN System

More Hardware Features

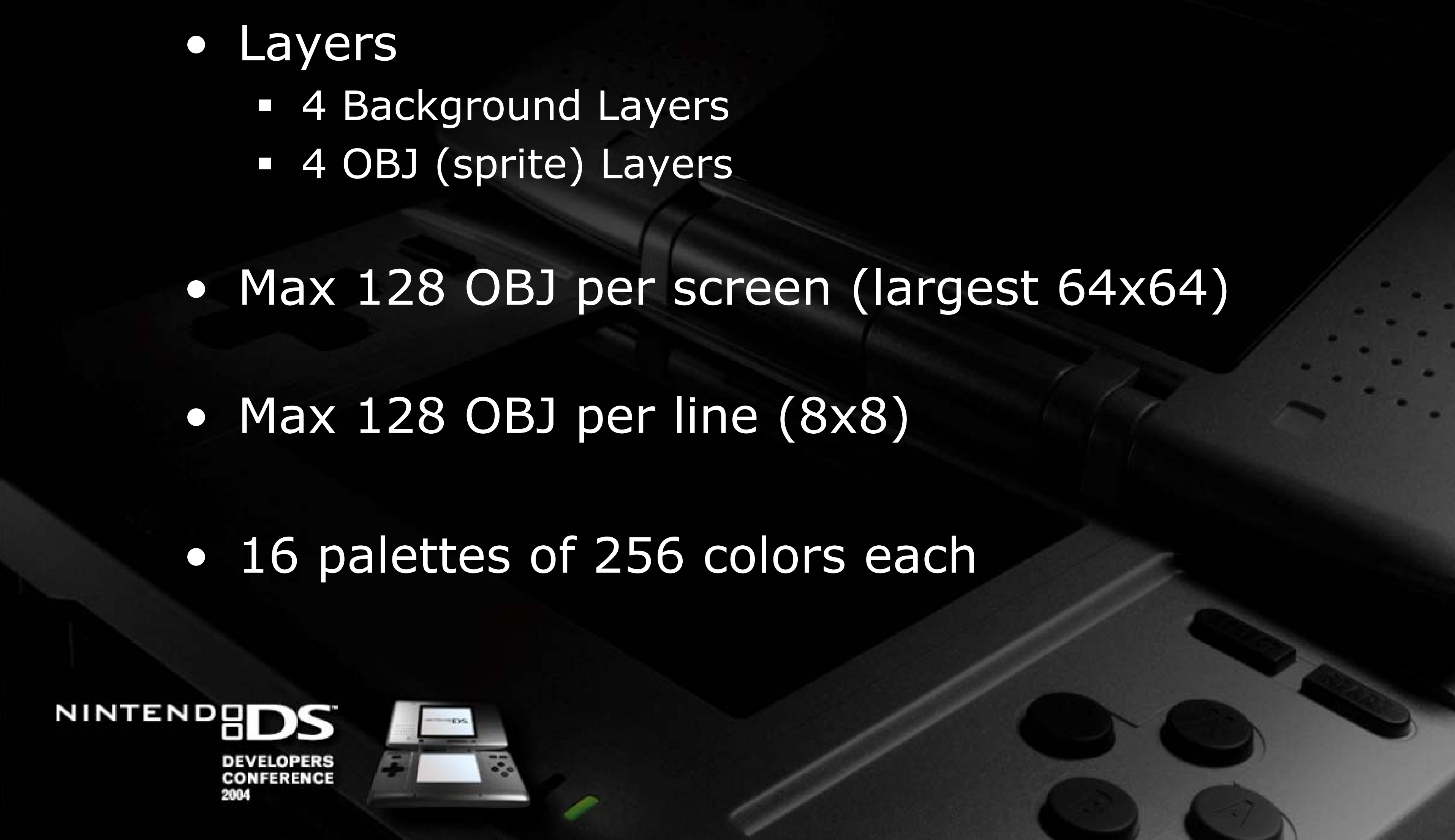
- Backward compatible with AGB games
- Dual ARM CPUs
  - ARM9 (66MHz) main CPU
  - ARM7 (33MHz) sub-CPU for processing:
    - Sound, wireless, touch, mic, buttons, RTC
- 4MB RAM, 656KB VRAM
- Card sizes: 8MB to 128MB (64Mb-1Gb)
  - EEPROM backup: 0.5KB, 8KB (4Kb, 64Kb)
  - Flash backup: 256KB (2Mb)

More Hardware Features

- Stereo speakers built into unit
  - Stereo headphone jack built-in
  - 16 channels of ADPCM audio
- Language setting
  - English, Japanese, French, German, Italian, Spanish
- Real-time clock and player info
  - Nickname (10 unicode characters)
  - Birthday, user color (16 choices)

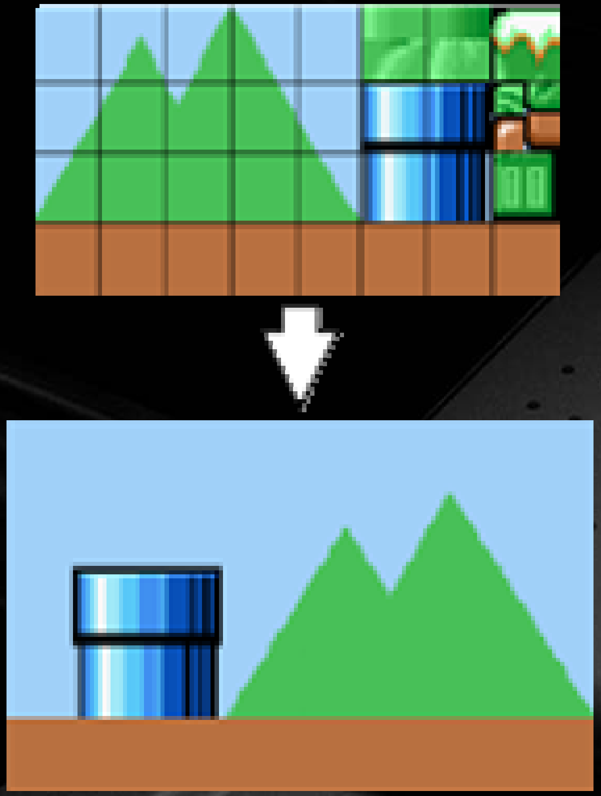
2D Graphics Overview

- Layers
  - 4 Background Layers
  - 4 OBJ (sprite) Layers
- Max 128 OBJ per screen (largest 64x64)
- Max 128 OBJ per line (8x8)
- 16 palettes of 256 colors each



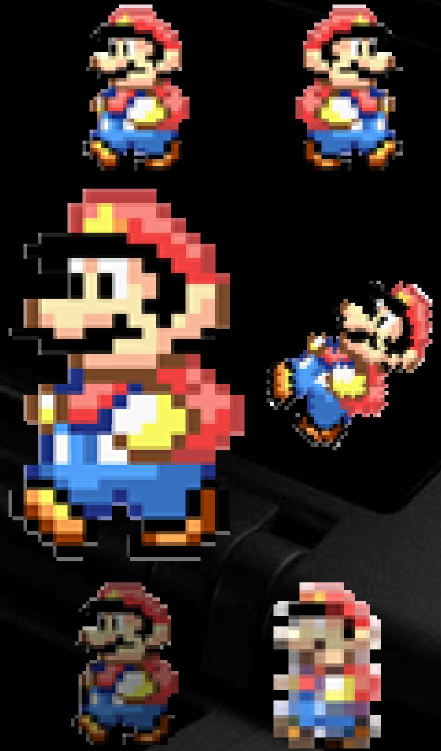
2D Background Features

- Bitmap method
  - Single image
- Character method
  - Tile approach
- Backdrop
  - Single background color



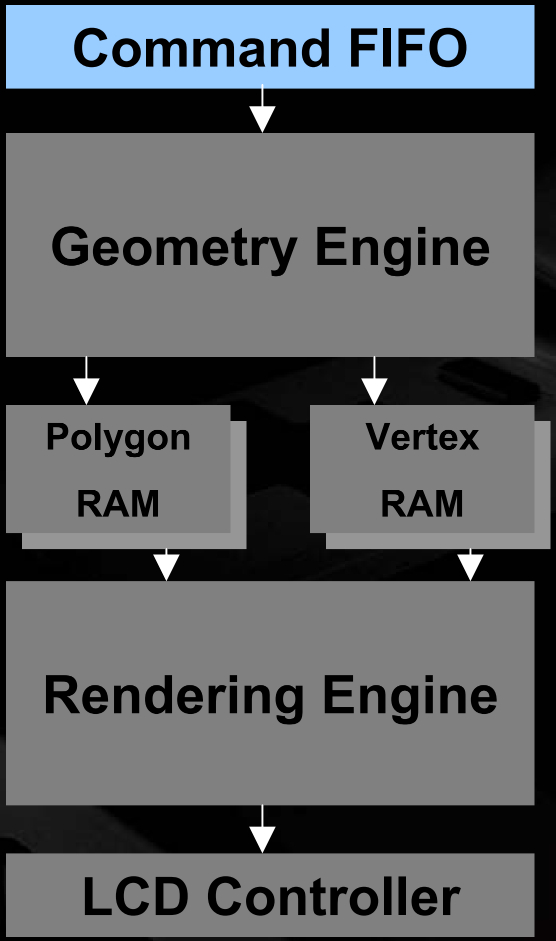
2D OBJ (Sprite) Features

- Two types of OBJ
  - Character and Bitmap
- Features
  - Horizontal / Vertical Offset
  - Horizontal / Vertical Flip
  - Affine Transformations
    - Scaling, Rotating
  - Translucence
  - Mosaic
    - Pixelated abstraction
  - Priority Settings



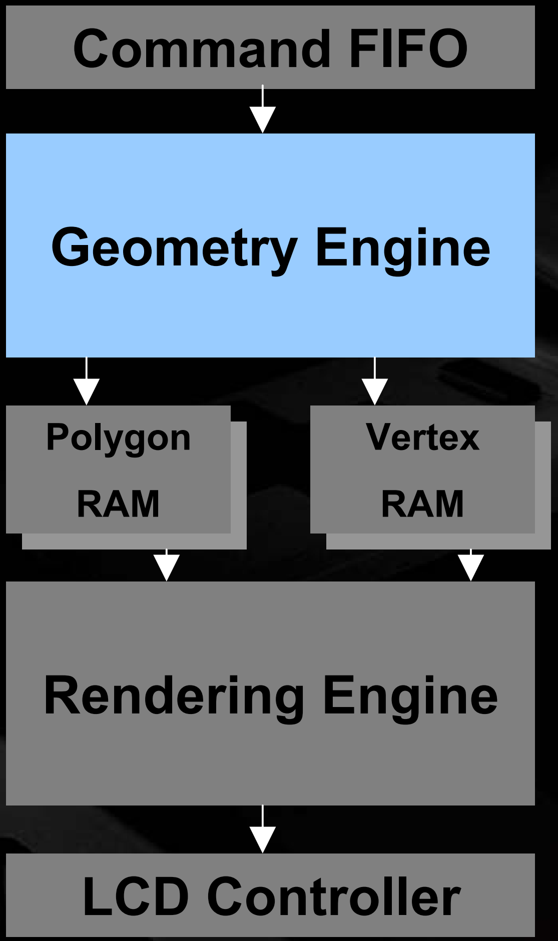
NINTENDO DS
DEVELOPERS
CONFERENCE
2004

3D Engine: Command FIFO

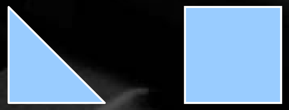


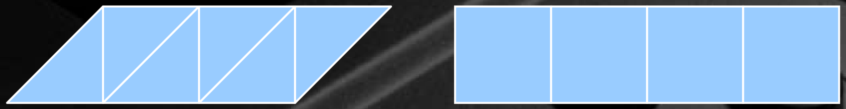
- Direct with function calls
  - Direct writing to GX command registers
  - Costly: load/store overhead
  - Costly: GX idle until next one
- Display lists
  - DMA ~3 cycles for 32b data
  - Commands can be packed
  - During DMA, do other work

3D Engine: Geometry Engine

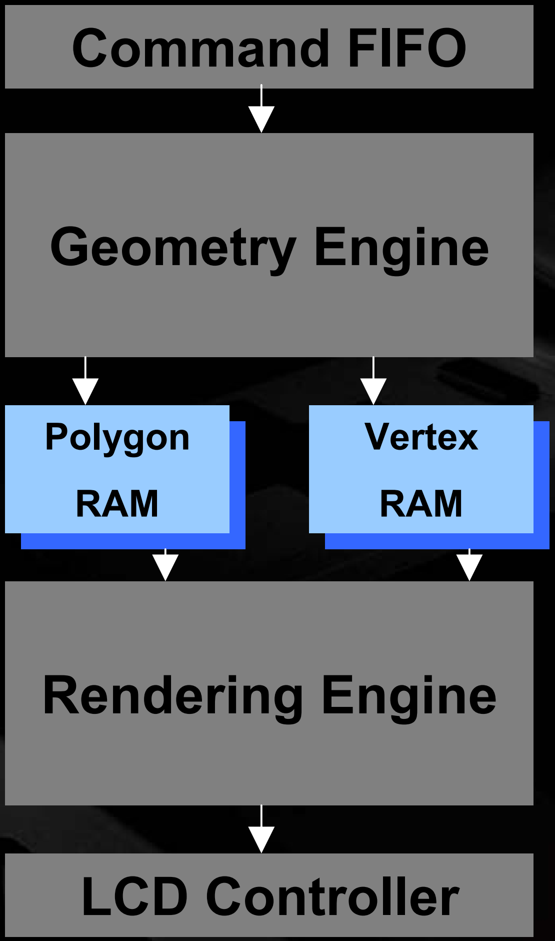


- Coordinate transformation
- 4x4 matrix computation
- Lighting (4 parallel lights)
- Backface culling
- Clipping
- Matrix stack
- Box test, position test
- Primitives
  - Triangle, quads
  - Tri-strips, quad-strips





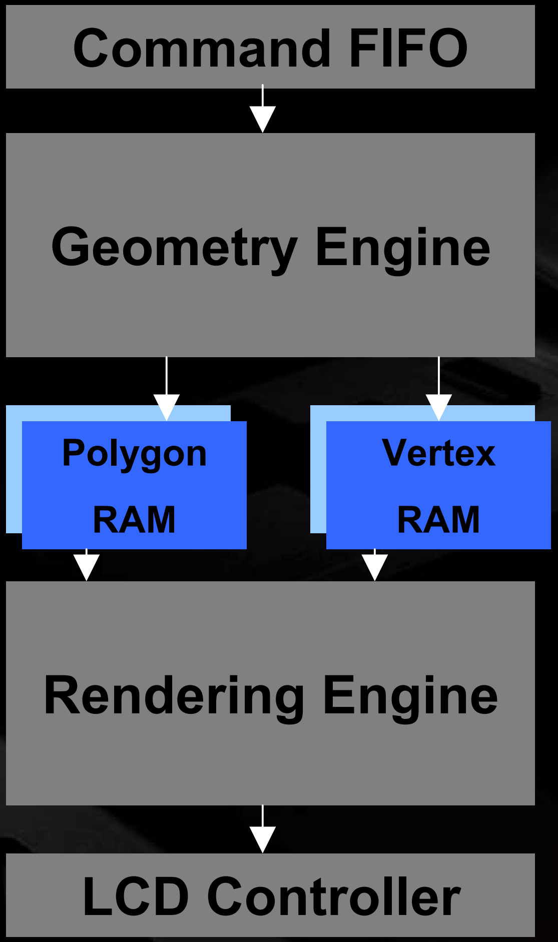
3D Engine: Poly/Vert RAM



- Polygon RAM
  - 2048 tris per 1/60th frame
  - 1706 quads per 1/60th frame
- Vertex RAM
  - 6144 verts per 1/60th frame

- Both double buffered
- Limits don’t include
  - Culled polygons/vertices
- Limits do include
  - Clipped polygons/vertices
  - Can add up to 350 polys if well-stripped

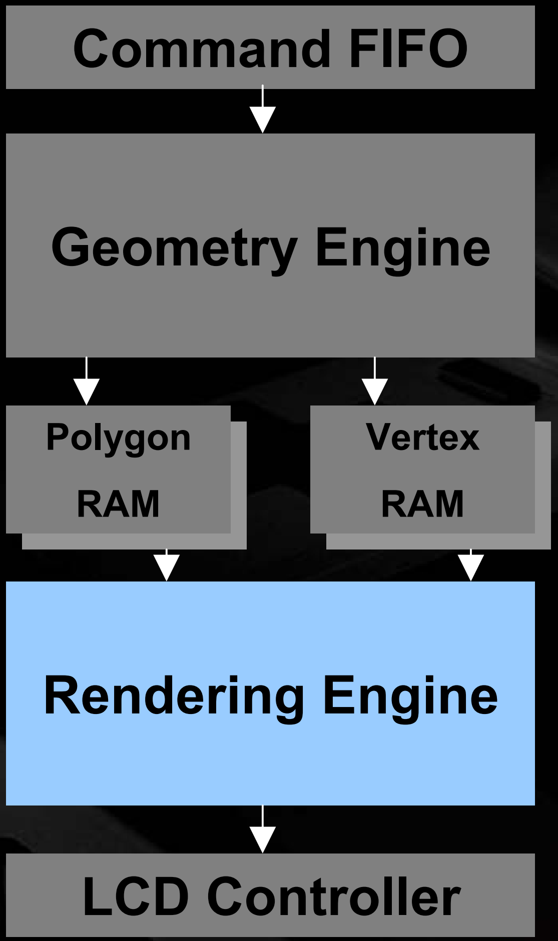
3D Engine: Poly/Vert RAM



- Polygon RAM
  - 2048 tris per 1/60th frame
  - 1706 quads per 1/60th frame
- Vertex RAM
  - 6144 verts per 1/60th frame
- Both double buffered
- Limits don't include
  - Culled polygons/vertices
- Limits do include
  - Clipped polygons/vertices
  - Can add up to 350 polys if well-stripped

NINTENDO DS DEVELOPERS CONFERENCE 2004

3D Engine: Rendering Engine



- Renders tris and quads
- Depth processing (Z/W buffer)
- Texture Mapping
  - Textures (8x8 to 1024x1024)
    - 4, 16, 256 color palette
    - Direct 5-5-5-1 RGBA
    - A5I3, A3I5 alpha textures
    - 4x4 texel compressed
- Alpha-blend, fog, anti-aliasing
- Toon shading, edge marking
- Highlight shading, shadow
- Clear image feature
  - Pre-store color/depth from VRAM

3D Engine: Line Buffer Method

- There is no framebuffer
- 48 lines at a time are rendered

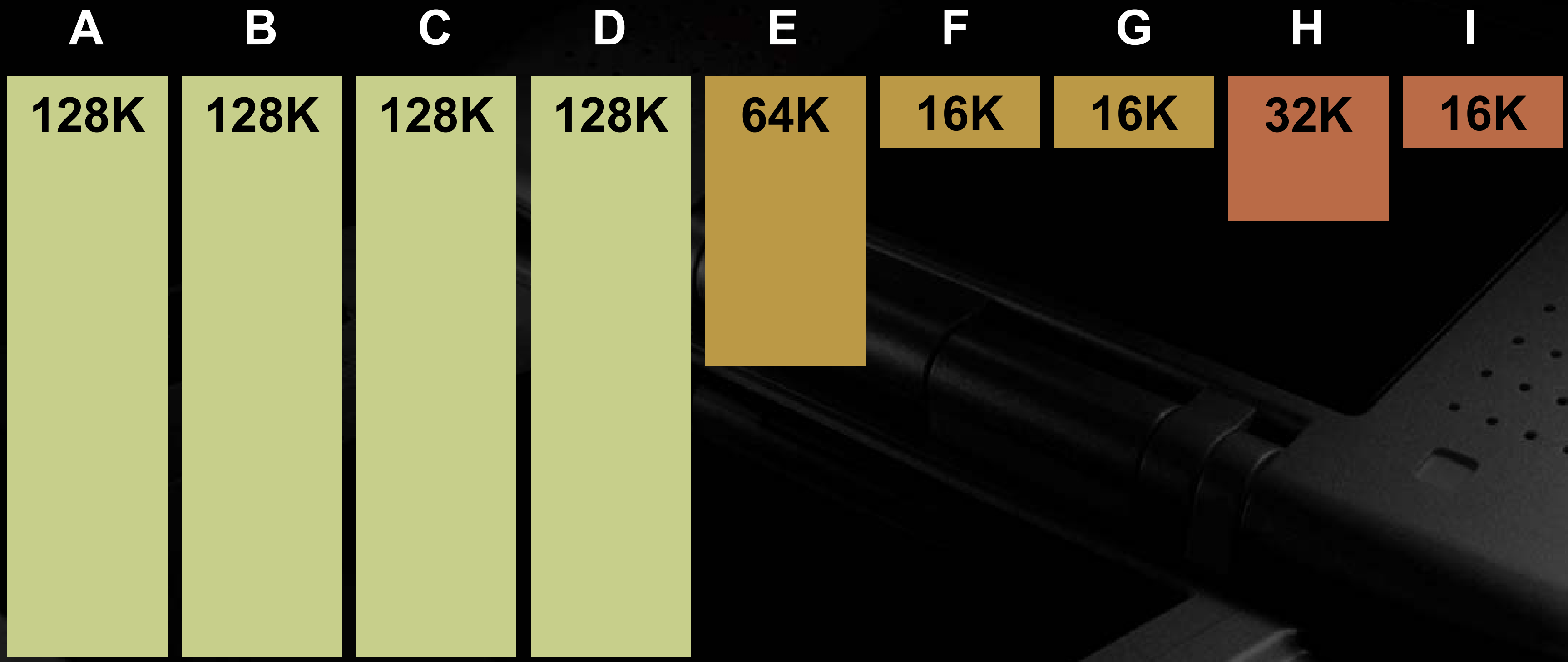
- Eliminates contention over main memory during rendering
- Ramifications
  - Must use capture feature to accumulate over multiple frames
  - Drops line if rendering takes too long
    - Completely dependent on fill rate
    - Can be detected after frame is drawn

3D Engine: Rendering Lines

- 2126 available cycles to draw a line
- Cost: 1 cycle per pixel drawn
- Cost: 8 cycle overhead per polygon
- Headroom: 4 to 8 times overdraw

Horizontal Pixels	8	16	32	64	128	256
Number of Polys	132	88	53	29	15	8
Fill Rate (Pixels)	1070	1422	1702	1894	2006	2062

VRAM



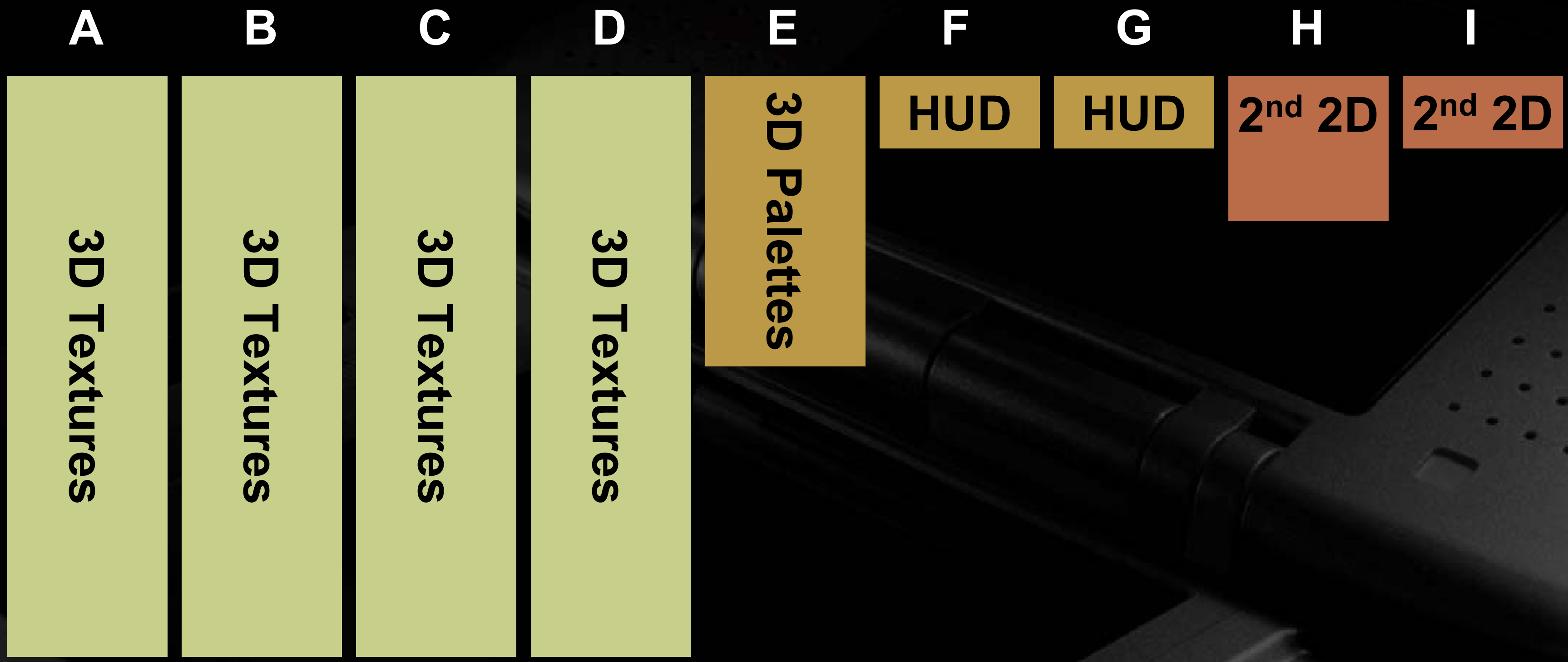
A	B	C	D	E	F	G	H	I
128K	128K	128K	128K	64K	16K	16K	32K	16K

VRAM

		A	B	C	D	E	F	G	H	I
3D Engine	Texture	🟩	🟩	🟩	🟩					
	Palette					🟨	🟨	🟨		
2D Engine A	BG	🟩	🟩	🟩	🟩	🟨	🟨	🟨		
	OBJ	🟩	🟩			🟨	🟨	🟨		
2D Engine B	BG			🟩					🟧	🟧
	OBJ				🟩					🟧

NINTENDO DS™
DEVELOPERS
CONFERENCE
2004

VRAM –3D Game Example

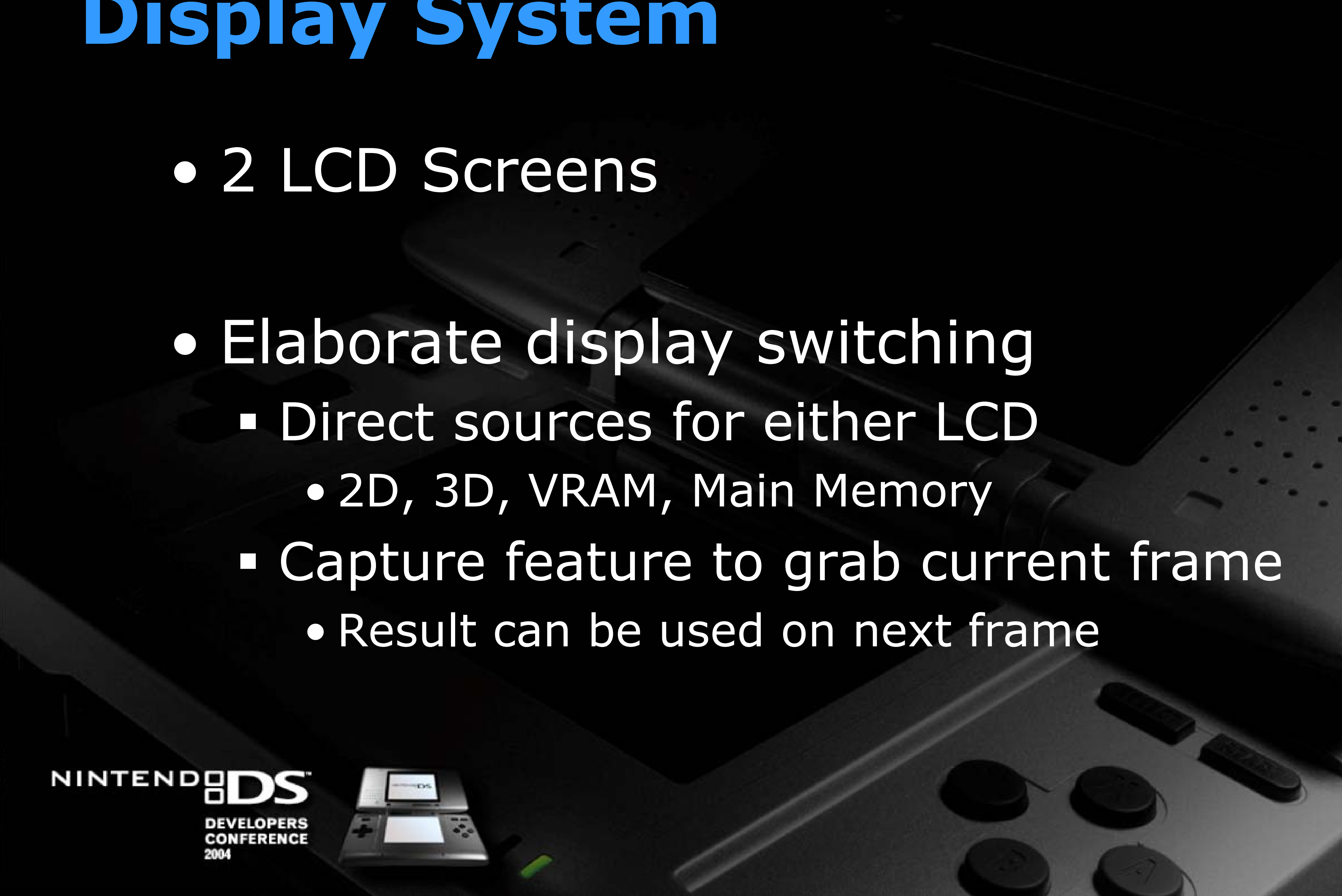


A	B	C	D	E	F	G	H	I
3D Textures	3D Textures	3D Textures	3D Textures	3D Palettes	HUD	HUD	2nd 2D	2nd 2D

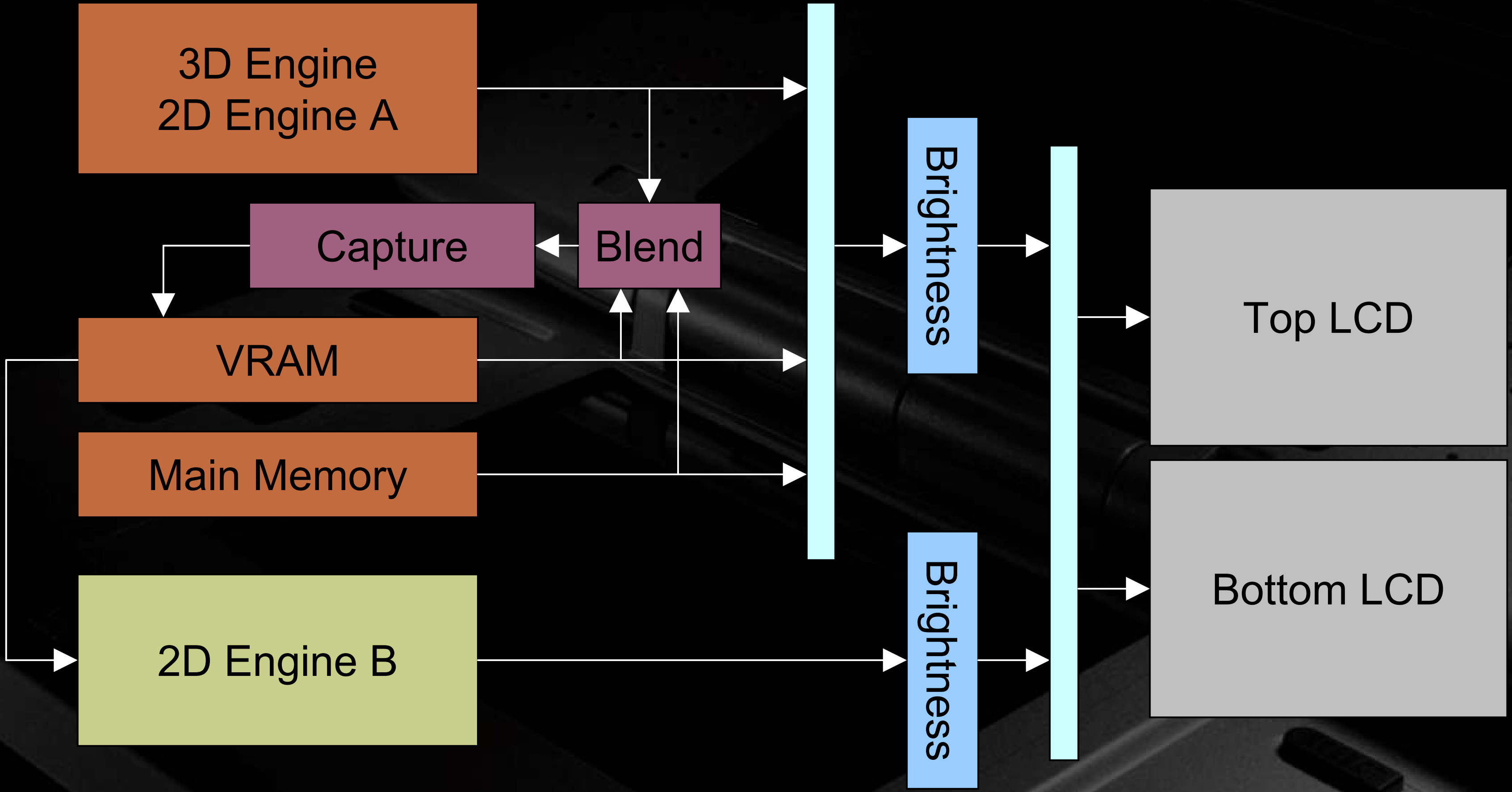
Display System

- 2 LCD Screens

- Elaborate display switching
  - Direct sources for either LCD
    - 2D, 3D, VRAM, Main Memory
  - Capture feature to grab current frame
    - Result can be used on next frame



Display System

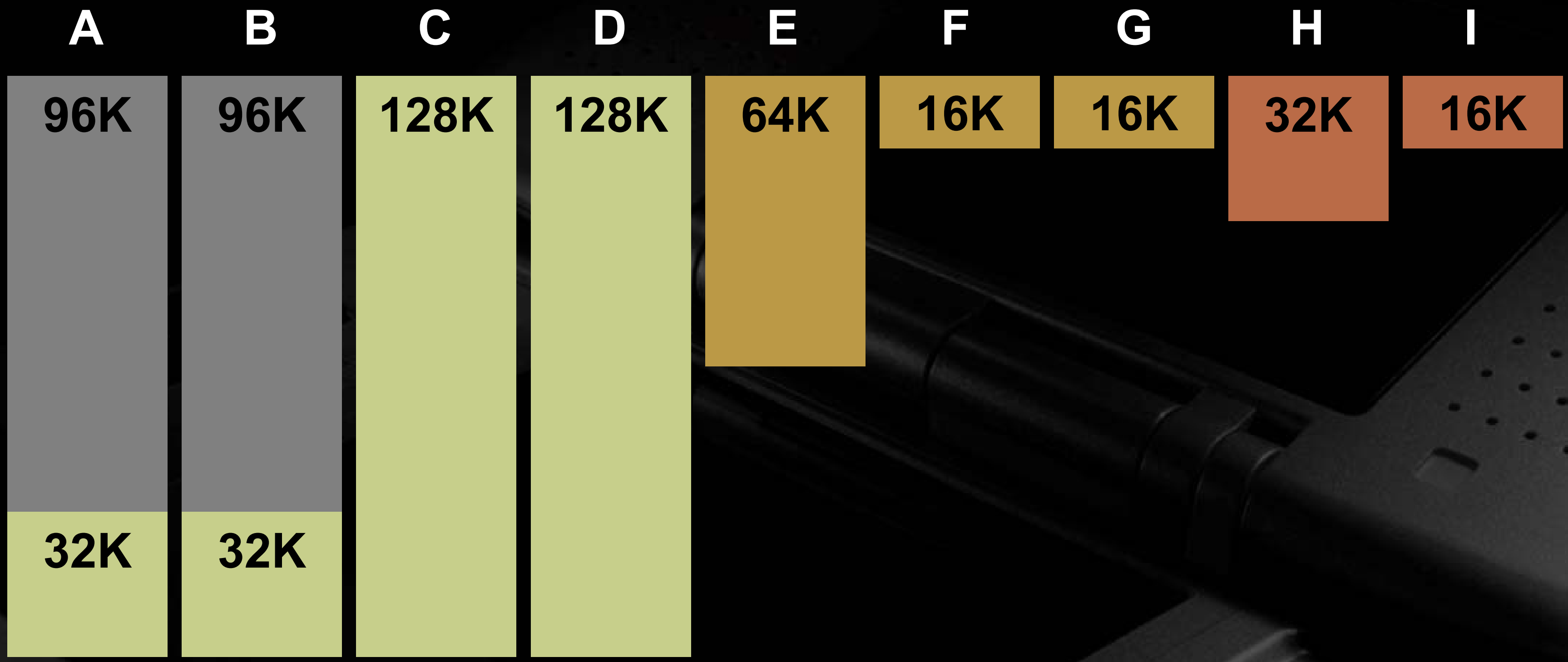


Display System

- Tricks
  - 3D on 2 screens
  - Double Polys/Verts on 1 screen
  - Full Screen Motion Blur on 1 screen

- Requires
  - Max 30Hz frame rate (can’t do 60Hz)
  - 192K of VRAM
    - Leaves 320K for 3D textures (vs 512K normally)

VRAM – Capture Memory



A	B	C	D	E	F	G	H	I
96K	96K	128K	128K	64K	16K	16K	32K	16K
32K	32K							

Graphics Tips

- One bone per vertex
  - Index into pre-built display lists and alter individual matrices
- Two or more bones per vertex
  - Blend on CPU using TCM
- Double poly trick: great for non-occluding background geometry
  - Sports games: Draw field and stadium 1st, draw players 2nd
  - Street fighting: Draw buildings/ground 1st, draw players 2nd
- Use 2D w/scrolling for skybox/skyline in the background

Graphics Tips

- Can’t combine vertex colors and hw lighting
  - Try changing material settings to tint
- Decide up-front whether to use 4x4 texture compression
  - More planning required
  - Fragments VRAM
- 60fps easy to achieve, but 30fps is reasonable
  - 30fps allows double the time for CPU computations

File System

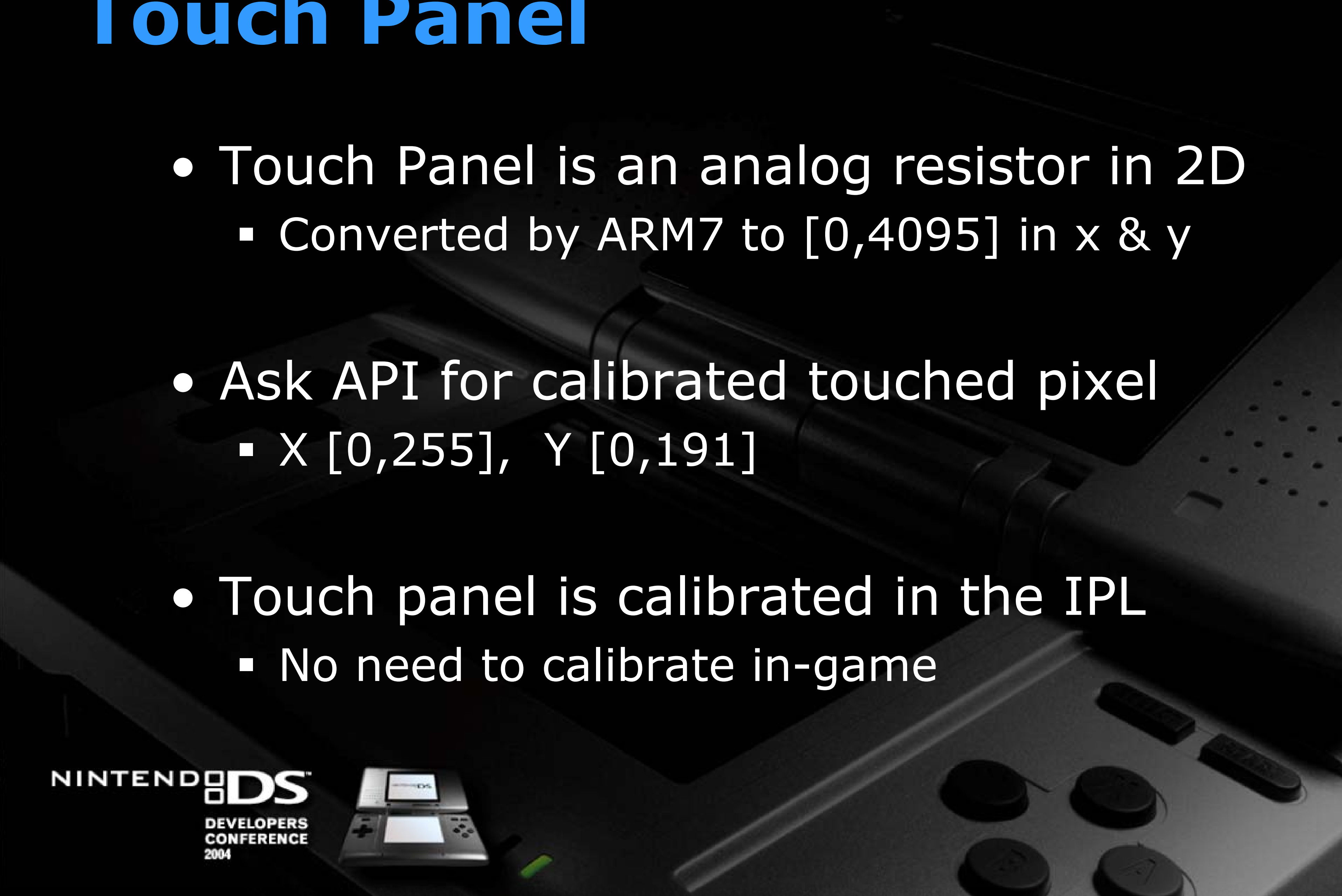
- Card memory can’t be directly accessed
- Must use File System API
  - Treats card as a standard file system
  - Open/Read/Close commands
  - No seek time – concept doesn’t apply
- Theoretical transfer speed 5.96MB/s
  - About 100K per frame at 60fps
  - Compare to 2-3MB/s for Nintendo GameCube

File System Restrictions

- Max number of directories
  - 4,096
- Max number of files
  - 61,440
- File/dir names 127 characters or less
- File/dir names have no upper/lowercase distinction (for speed reasons)
- File/dir names can’t contain:
  - ¥ / : ; * ? “ < > |

Touch Panel

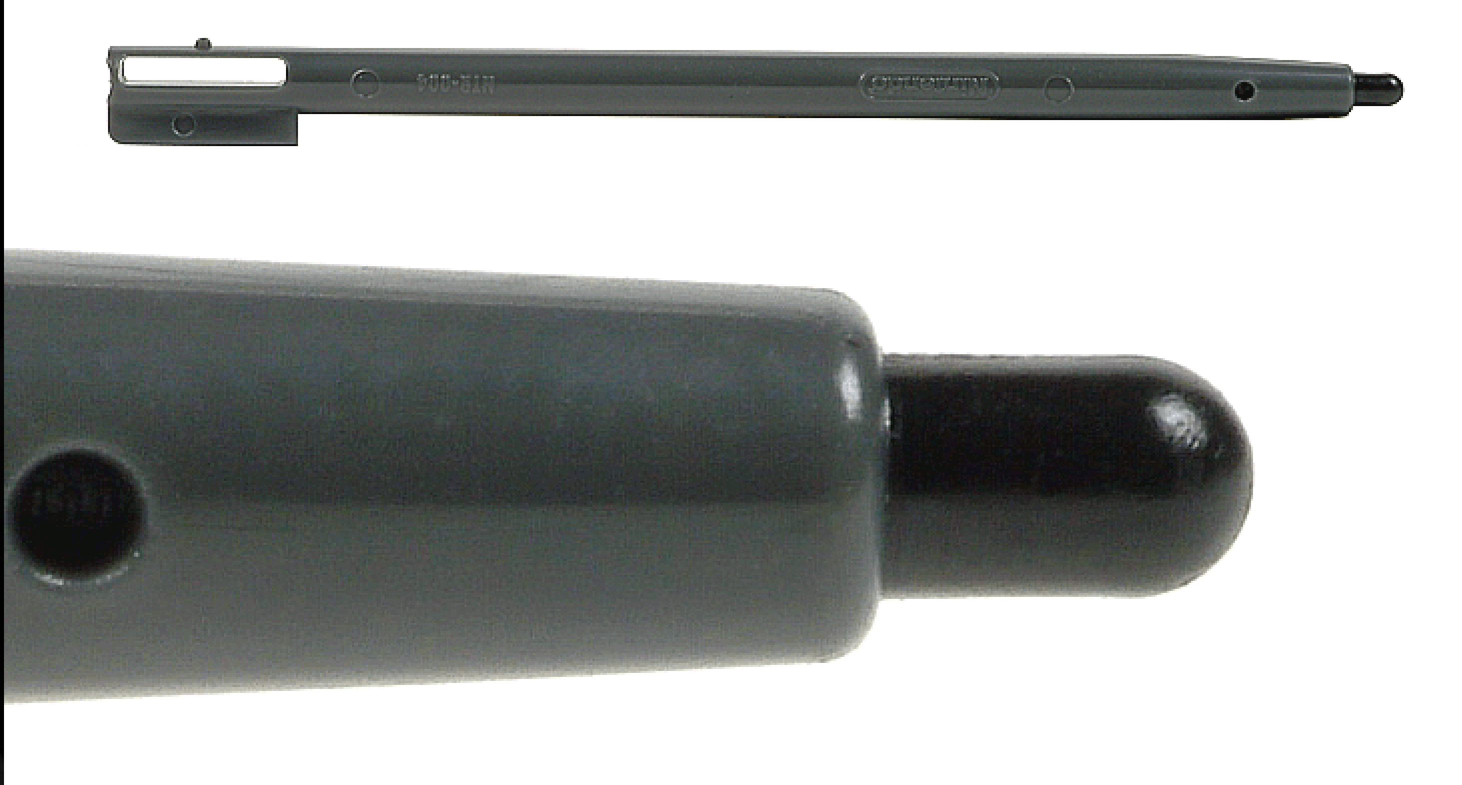
- Touch Panel is an analog resistor in 2D
  - Converted by ARM7 to [0,4095] in x & y
- Ask API for calibrated touched pixel
  - X [0,255], Y [0,191]
- Touch panel is calibrated in the IPL
  - No need to calibrate in-game



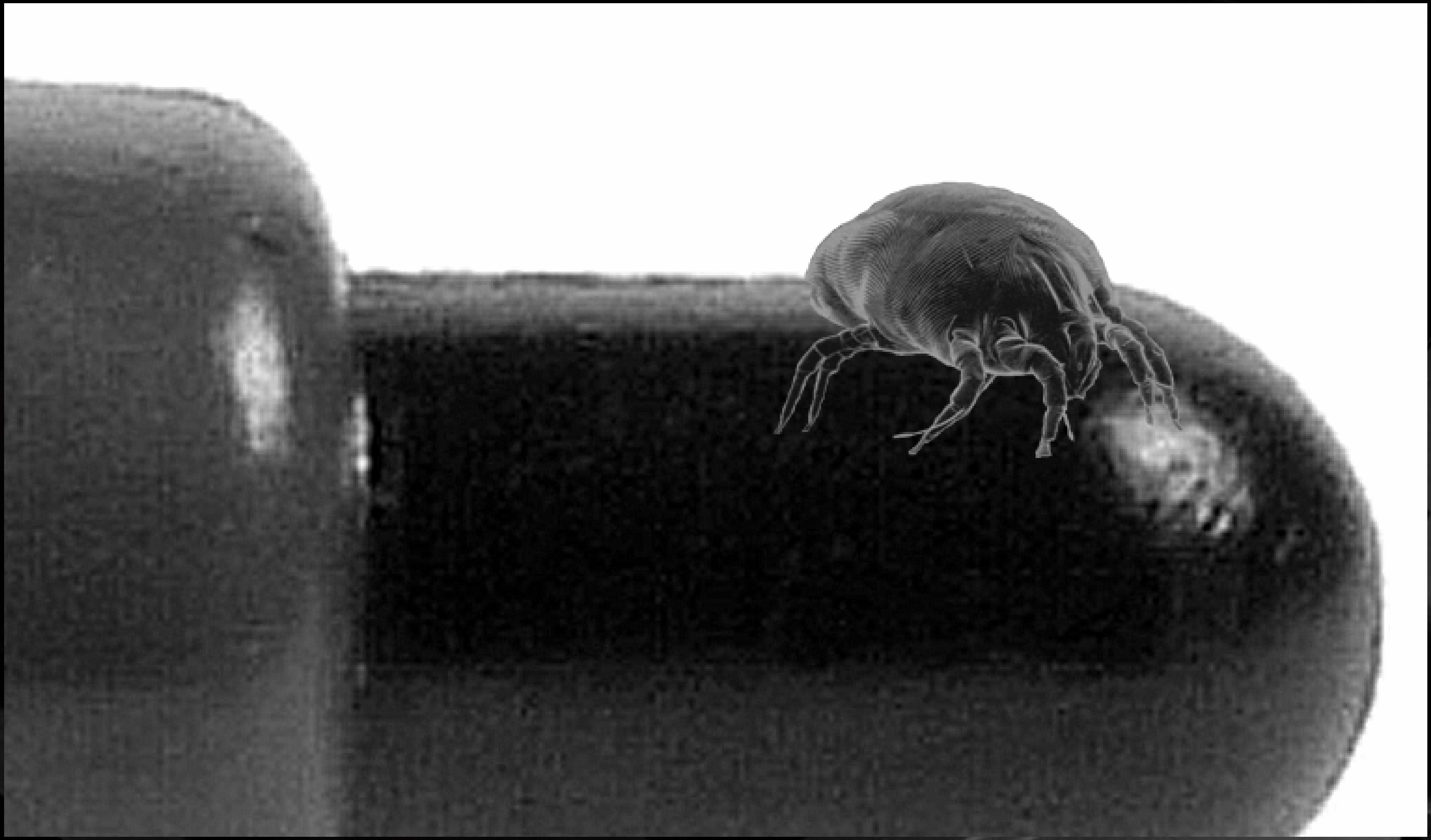
Touch Panel

- Can only detect one location at a time
  - Multiple locations results in average
- Can ask for
  - Instantaneous coordinate
  - Average of 4 samples during last frame
- Can use stylus or finger on touch panel
  - Player should calibrate for their preferred method for precise control

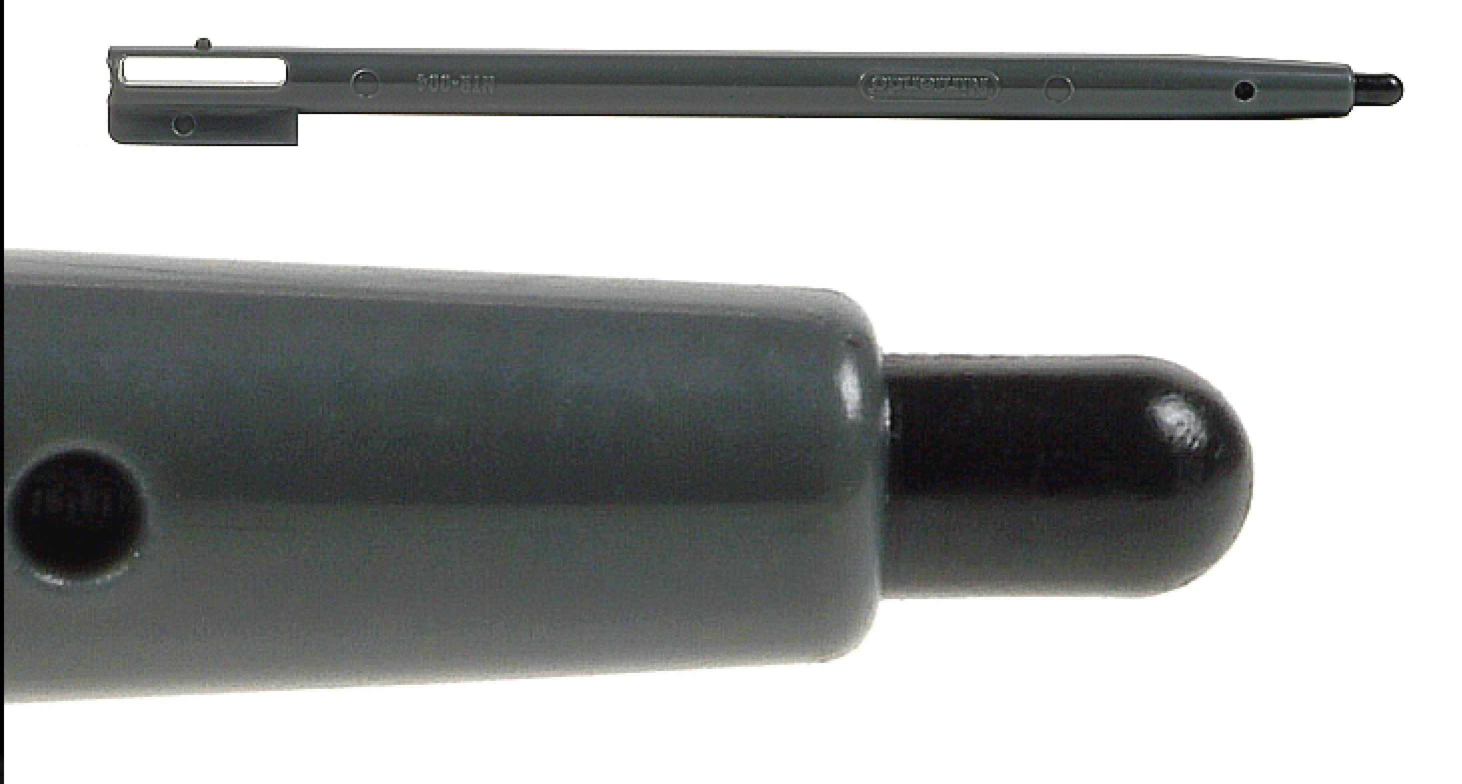
Touch Panel: Stylus



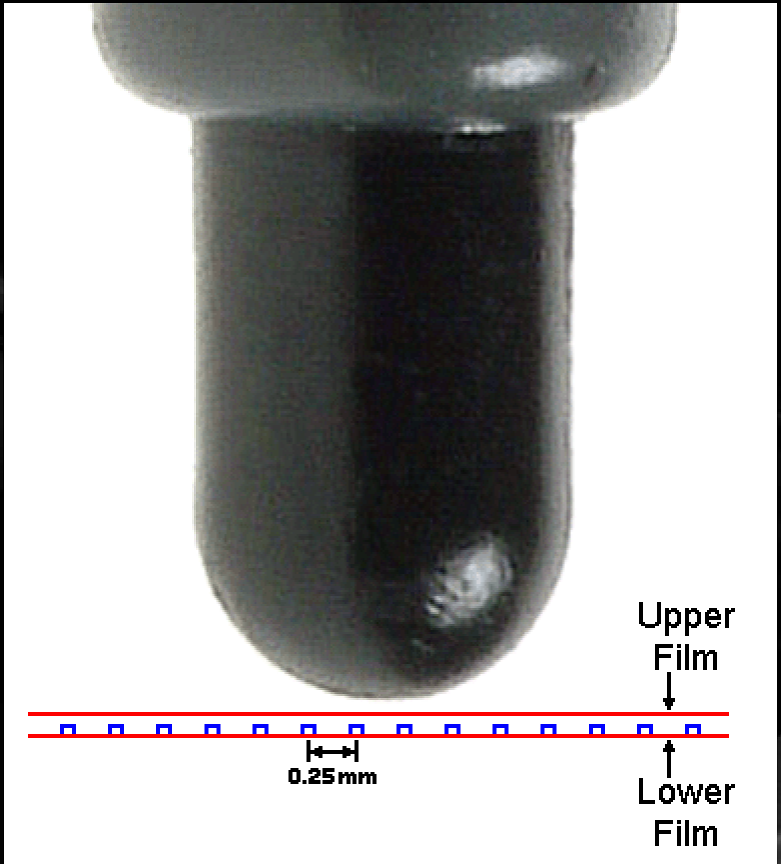
Touch Panel: Stylus



Touch Panel: Stylus

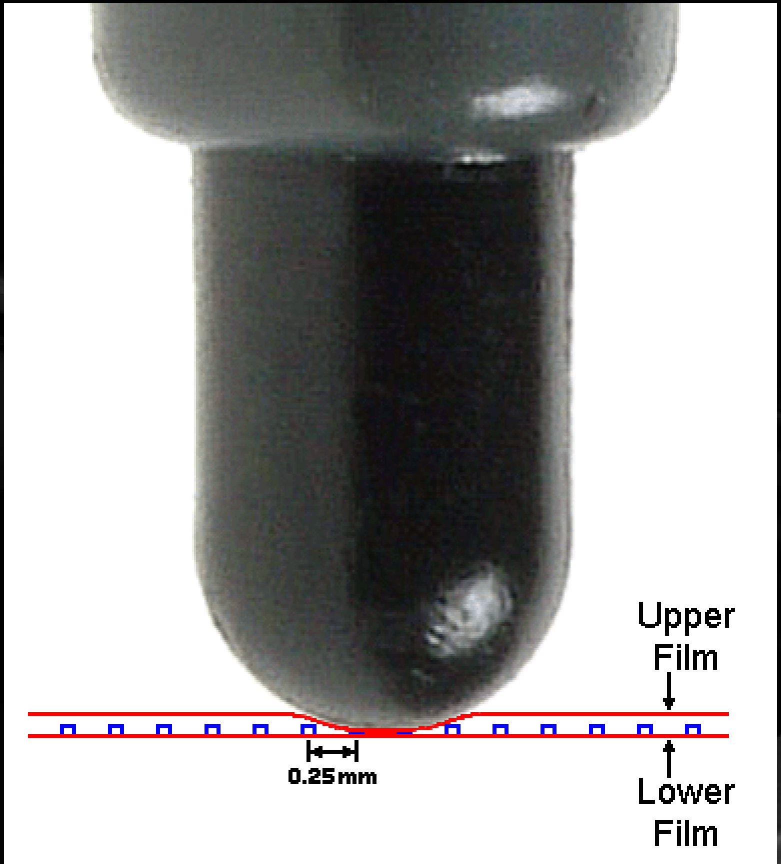


Stylus on Touch Panel

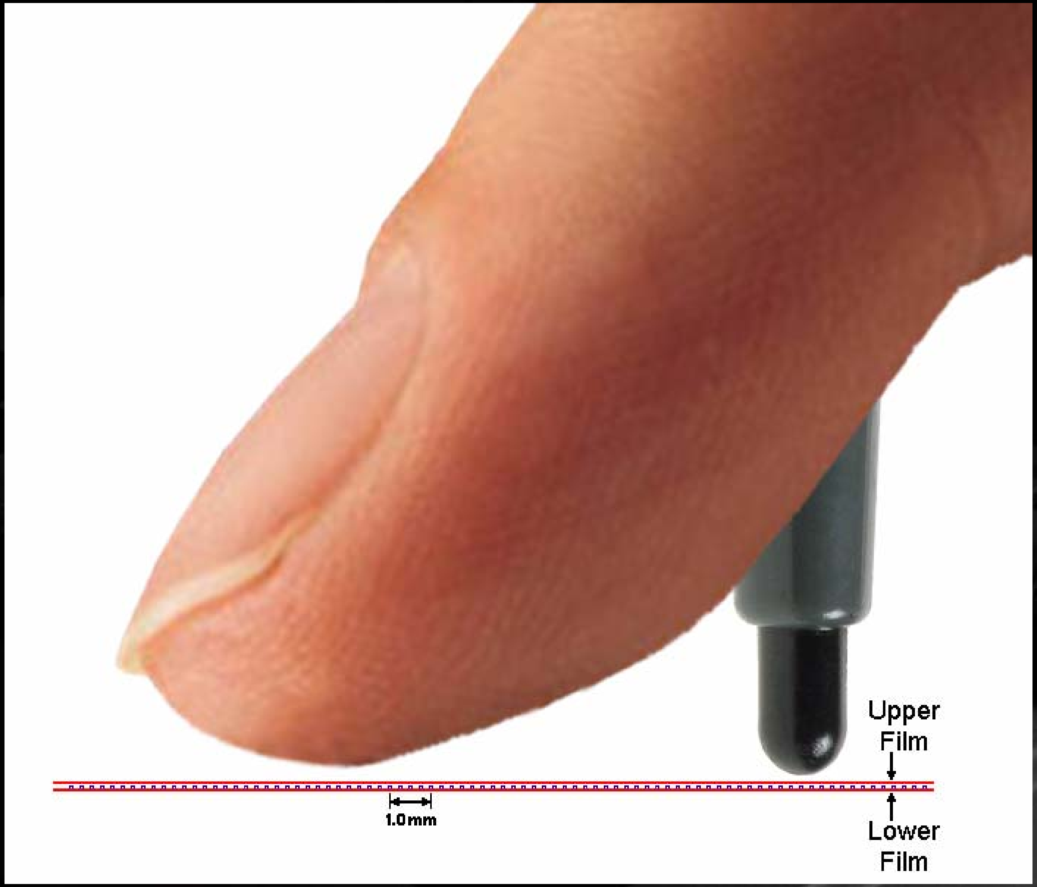


NINTENDO DS DEVELOPERS CONFERENCE 2004

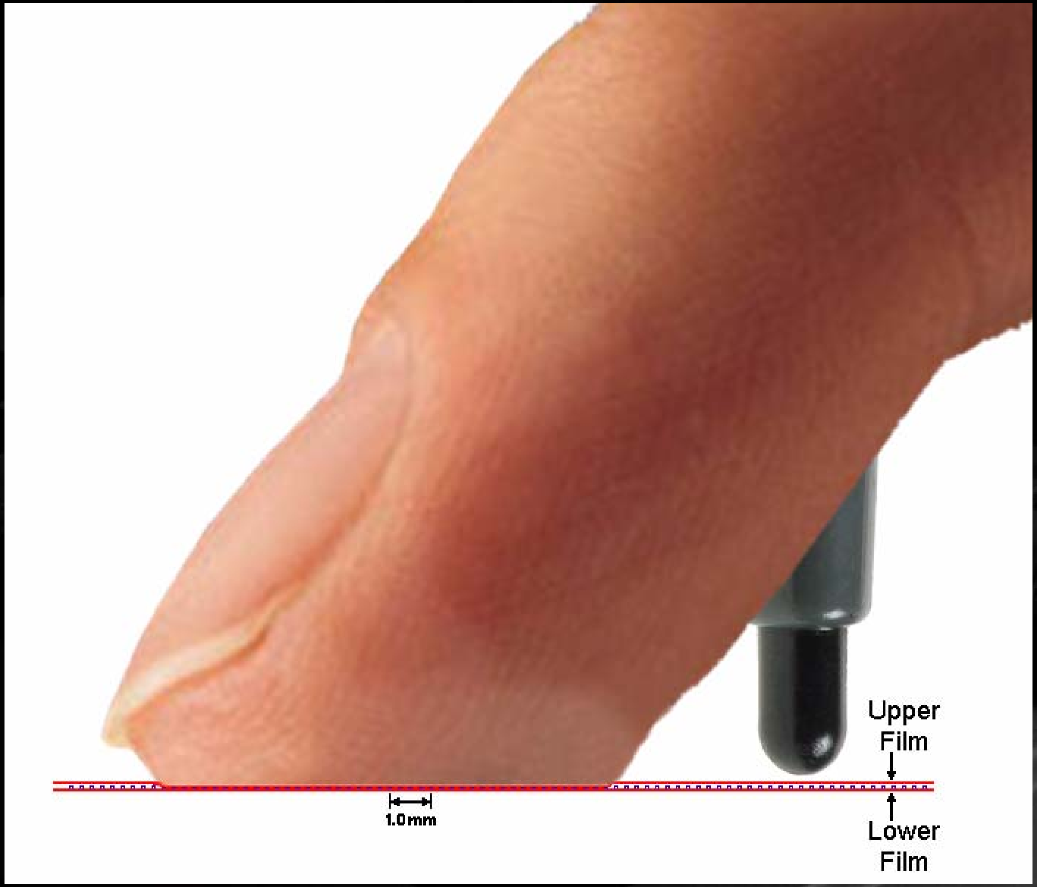
Stylus on Touch Panel



Finger on Touch Panel



Finger on Touch Panel



Conclusion

- Discussed
  - Hardware features
  - 2D, 3D, display system, VRAM
  - File system
  - Touch panel

- Understand the capabilities
- Design toward strengths