



1st Playable
PRODUCTIONS

Real-time Graphics: Issues and Trends in Games

Tobi Saulnier
CEO, 1st Playable Productions

October 17, 2005
Computer Graphics, Fall 2005



Games as a Graphics Application

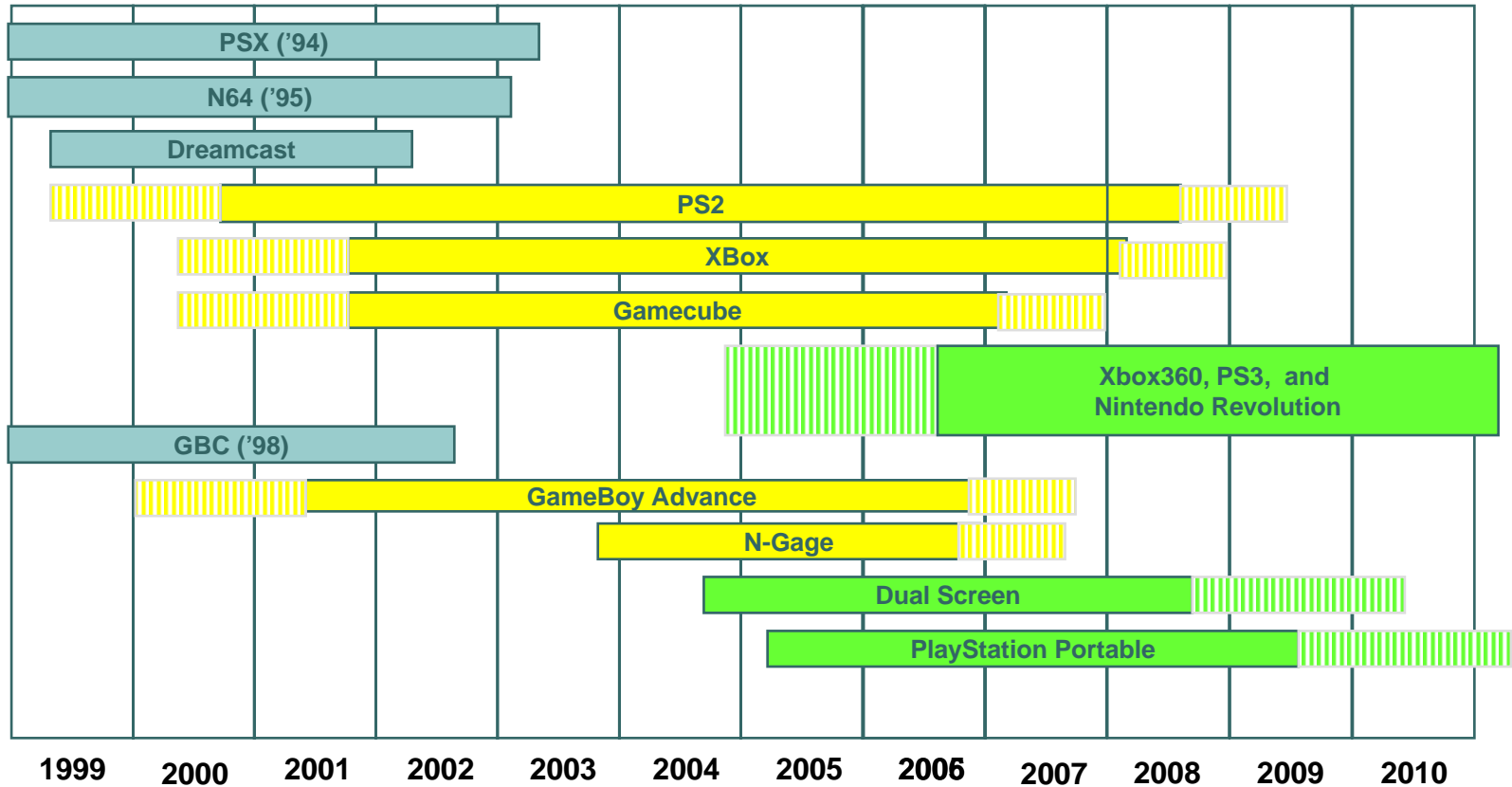
- Real-time graphics
 - Actually, a blend of prerendered and real-time rendered
 - Migration of graphics techniques as technology advances
- Each game platform has a unique architecture, optimization techniques need to be tailored for each
 - Genre enforces priorities for processing
 - Game art and design production creates bottlenecks



Types of platform issues

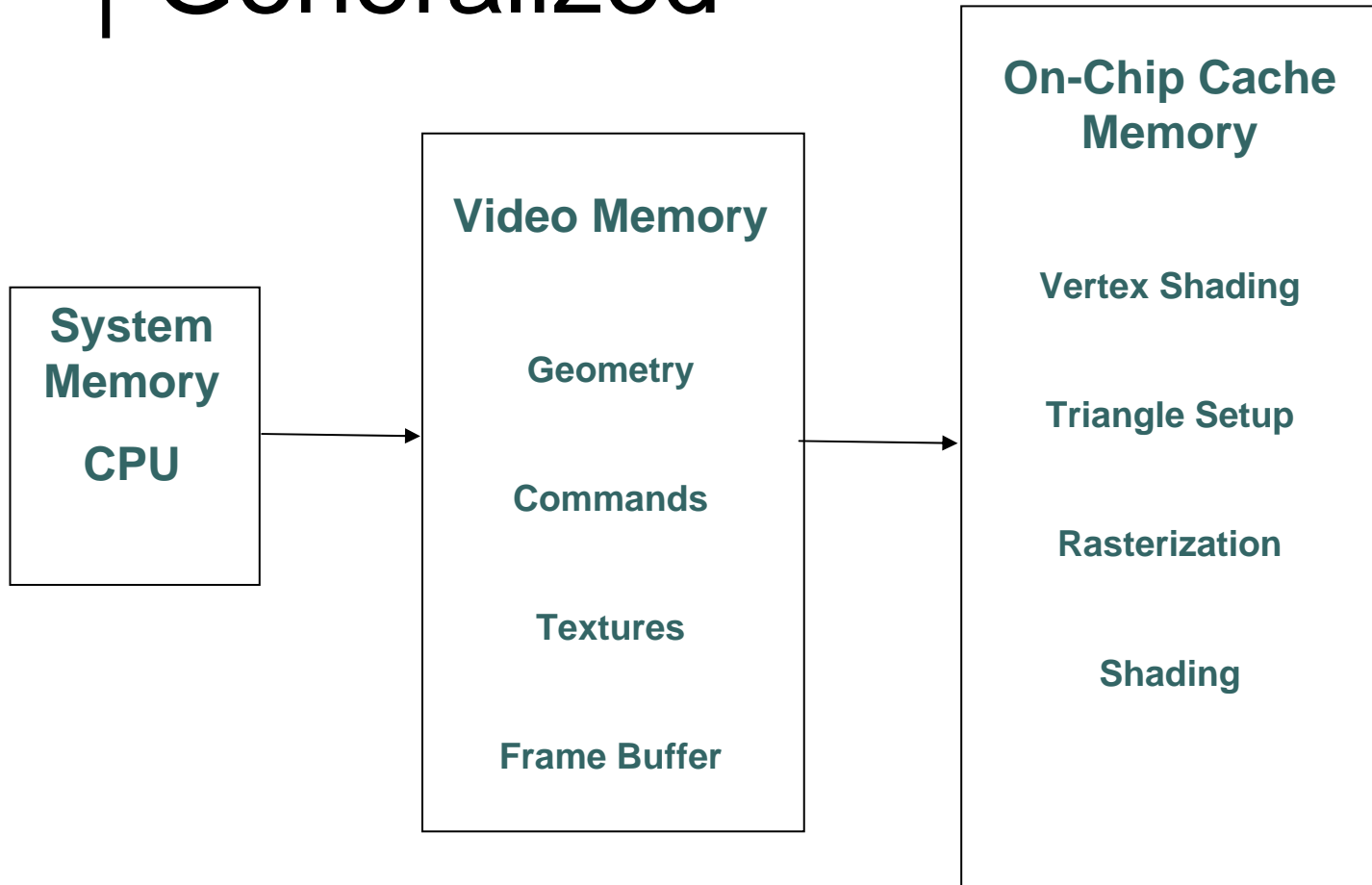
- Embedded platforms – do more with fixed platform
 - Optimize for specialized needs = differentiation
- General purpose platforms
 - Phones, PC's
 - Graphics frameworks to isolate from low-level dependencies

Console Platform Transitions ...



Video Games technology introduction is driven by intense industry competition.

Graphics Pipeline Generalized





Graphical Data Types

- Scene data
- Character data
- Special Effects data
- Lighting
- Weather effects
- Particle systems



Why Graphics Optimization?

- More graphics in the game ...
- More AI
- More realism
- More multiplayer
- More characters

Nintendo DS

- Arm 7 and Arm 9 processors
- Tony Hawk DS versus Spiderman DS





Trend in pipelines

- Fixed function pipelines
 - Typical of GBA, DS, PSP, PS2
- Programmable pipelines
 - Pixel shaders
 - Vertex shaders
 - Shader languages
 - Major transition for Xbox 360, PS3

Unreal III Engine (PC)





Platform Architectures

Every hardware platform has quirks and shortcomings, is only partially thought through, and decided well before most developers see the specs

- **Xbox:** 733 MHz, Pentium 3, 64 MB RAM, DVD, GForce3 GPU NV20 chipset, 8GB harddrive, 640x480
- **PS2:** EE 300 MHz MIPS CPU, Vector Unit custom, 32 MB RAM, 2 MB VRAM, Graphics Synthesizer, but no shaders, DVD but 2x slower drive speed than XBox
- **GBA:** ARM7, 16 MHz, 8 MB cartridge, 32K on-chip, 256K off-chip RAM
- **Phones:** the kitchen sink
- **PC:** whatever you put in your PC 😊



Xbox 360 vs PS3

Product Name

Sony PlayStation 3

Microsoft Xbox 360

Release Date

2006

Q4 2005

Controller

Bluetooth Wireless

2.4 GHz Wireless

Graphics Core Clock Speed

550 MHz

500MHz

Graphics Processor

RSX "Reality Synthesizer"

Custom ATI Processor

System Memory

256 MB XDR

512 MB UMA (Shared with GPU)

Video Memory

256 MB

512 MB UMA (Shared with CPU)

Embedded Video Memory

n/a

10MB eDRAM

Resolution

480i, 480p, 720p, 1080i, 1080p

480p, 480i, 720p, 1080i

Memory Bandwidth

25.6 GB/s main memory bandwidth, 22.4 GB/s video memory bandwidth

22.4 GB/s main memory bandwidth, 256 GB/s to eDRAM, 21.6 GB/s FSB



Lighting

- Shadows
- Lighting
- Scene lighting
- Character lighting
 - Resident evil – lighting tied into facial animation
 - Doom – global illumination, shadow volumes

Doom Xbox





Rendering

- Tris, polys
- Multipass, single pass techniques
- Prerendering versus real-time rendering choices
- Architectural impact – GPU, memory, bus





Procedural graphics

- Problem = generation of enough content to utilize hardware capabilities
 - Manual generation is harder
- Terrain generation
- Scene generation
- Spore – real-time generation
 - Trees
 - Returning to the same scene – pseudo-random sequences
- Game challenges – design mapping

Procedural Trees





Other issues related to real-time graphics

- Multi-player - Networking
 - What data to send
 - Compression
 - Multiple players and scenes to manage
- Unpredictable data flow
 - Memory resident scene
 - Loading, prediction



Other topics

- Depth of field
- Field of view
- LODs
- Other data pipeline
 - Animation, physics, collision – visible and non-visible data
 - State data
 - Fluid, cloth, simulation
- Simulation versus emulation
 - Realistic simulation often doesn't achieve the effect needed
- Psychology



Trends

- Fixed function to programmable
- Deferred rendering
- Progression of 3D onto handheld
- Multiprocessing/multipurpose
- Networked play
- Procedural texturing, geometry
- Lighting / cinematics



Takeaways

- Now is an exciting time in graphics for videogames
- No, not all the problems have been solved
- The problems being faced require a new level of engineering discipline, a transfer of knowledge from non-realtime to realtime
- You need to bridge hardware and software to be competitive
- Not just games – Impact of innovation in other important areas



It's not just entertainment

- Training
- Simulations
- Education
- Satellite architectures – launch and use for 20 years
- Medical systems – real-time surgery modalities
- Battlefield systems

- Share similar technical issues!



What else to learn

- Control systems – realtime graphics require conditional processing
- Memory and data flow management
- Software engineering, esp working with existing code
- Profiling/optimization/benchmarking – specs versus reality
- Art
- Psychology of vision/perception



Conclusions

- Master's degree
 - Games as an application has a ton of unsolved engineering and CS problems, with direct relevance to many other industries
 - Rate of innovation, complexity of solutions constantly creates more
- PhD topics?
 - Yes, but harder to say who is funding
 - University funding – at very early stages – technology is often not seen as main reason for success of games, although it is often a reason for failure. An enabler.
- Jobs in games will increase for those with grad degrees, as the complexity of the systems and software continues to grow