

Technical Skills

Programming Languages:

C/C++, Objective C, Python, C#, Java, ActionScript 3, ARMv7/8 asm (Neon), Pentium asm (SIMD: SSE, MMX), PowerPC and Altivec VMX intrinsics, STL, Boost, HLSL, GLSL, Cg, DirectX shader assembler, PlayStation2 MIPS assembly, VU microcode and MMI, Xbox pixel shader combiners, HTML/CSS, CGI, JavaScript, Unix C-Shell, Lisp (Scheme), BASIC

Tools/Libraries:

DirectX 8-11, OpenGL ES 2/3, Flash 11/Stage3D, Unity 3D, Unreal Engine, libgcm & libspurs (PS3), RenderWare, IrisGL, Glide (3dfx raster library), MS Visual Studio/Visual C++, Xcode, Android Studio, IntelliJ, FlashBuilder, Xamarin, CodeWarrior, Perforce, SVN, Git, Emacs, VTune, PIX (PC, Xbox 360), Instruments, GPAD & SN Tuner (PS3)

Operating Systems:

Windows, Mac OS, DOS, Linux, Unix, iOS, Android

Work Experience

10/14 – present: ATAP Performer, [Google](#) (Advanced Technology and Projects) – Mountain View, CA

Spotlight Stories 3D interactive story telling on Android, iOS and VR

- Novel 360 panoramic video format (CubeRect) - codec and renderer
 - improved quality and reduced bandwidth relative to equirectangular video
- Multi-platform rendering engine improvement and maintenance
 - features, performance, architecture, parallelism
- Android and iOS profiling and optimizations (CPU and GPU)
 - Shader optimizations, ARMv7 and ARMv8 Neon asm micro-optimizations
- Porting 32-bit Android app to 64-bit architecture and iOS
- Streaming video player (iOS, Android and Mac)
 - Supports hardware decoding and OpenGL rendering of videos

R&D:

- Depth buffer compression algorithm (for compositing, relighting, etc.)
 - Variable compression, fast decompression

8/11 – 10/14: Lead Graphics Engineer, [Rumble Entertainment](#) – San Mateo, CA

Kings Road high-end online game for browsers and mobile devices

- 3D rendering pipeline using Flash 11 Stage3D for hardware accelerated 3D graphics in a browser
- Deferred shading for dynamic lighting
- Streaming of large detailed worlds encoded into multiple textures (depth & color)
- Developed depth encoding algorithms, to store world depth with very low memory usage
- Math libraries in ActionScript 3 (vector, matrix, quaternion, dual-quaternion)
- Asynchronous asset loading (worker objects time sliced from main frame time)
- Shader tool for converting HLSL shaders to AGAL shaders for Stage3D, and GLSL for mobile
- Various shaders and permutations for skinned characters, environment, water, world objects, etc.
- Visual effects pipeline, GPU particles, animated sprites, refraction effects, emissive effects
- Post processing filters, including bloom, blur, vignette, color adjustment, S-curve tone adjustment
- Profiling, optimizations, and compatibility on various platforms (Windows, Mac, iOS) and browsers
- Wrote various art pipeline tools (depth encoding, vfx editing, image processing, mesh and animation compression)

4/10 – 8/11: Principal Graphics Engineer, [Turbine \(Warner Bros\)](#) – Boston, MA (working remote from CA)

Lord of the Rings Online and **Dungeons & Dragons Online** Massively Multiplayer Online (MMO) games for the PC

- Multi-platform rendering support, including PC DX9, DX10, DX11, PS3 and Xbox 360
- Optimizations of various components, including skinning, rendering pipeline, core parallelism
- Vertex compression improvements on all platforms
- Created a framework for asynchronous SPURS job management
- Synchronization between PPU (core CPU), RSX (GPU), and 6 SPUs
- SPU skinning and culling with Edge libraries
- Architected new shader system, with more code sharing, fewer permutations, and more features
- Implemented HDR rendering pipeline for all platforms, with gamma correct rendering
- Created a post processing system, to enable us to more easily add new filters
- Investigation of features to improve the graphics and performance of our engine

12/09 – 2/10: Chief Technology Officer, [VisiDeck](#) – Warrington, UK (working remote from CA)

Real-time visualization software and services

- Analyzed various game engines to find best fit for our requirements and needs
- Competitive analysis, to ensure we provide a better product and services
- Responsible for most of our software and tools development and features
- Miscellaneous fire fighting, helping with asset creation, video presentations, etc.

10/07 – 12/09: Principal Graphics Engineer, [Shaba Games \(Activision\)](#) – San Francisco, CA

7/09: [DJ Hero](#) for Xbox 360 and PlayStation3

- various rendering optimizations
- eliminated CPU and GPU stalls on the Xbox 360 for improved parallelism
- implemented and improved some post processing filter effects for Xbox 360 and PS3

10/08: [Spiderman: Web of Shadows](#) for Xbox 360 and PlayStation3

- improved and maintained rendering pipeline and shader architecture
- optimized graphics pipeline and shaders
- implemented HDR rendering, with high quality bloom and tone mapping
- implemented new post processing system, with numerous filters, including depth-of-field and radial blur
- implemented image-based lighting for more realistic lighting
- responsible for the majority of the PlayStation3 graphics, nearly full parity with Xbox 360 graphics
- SPU code improvement and increased utilization to relieve the PPU

Graphics R&D:

- efficient deferred lighting, supporting 100s of light sources and still maintaining framerate
- experimented with tiled lighting and pass per light with stenciling
- real-time computation of line light sources, can be later extended to area light sources
- multiple methods of doing real-time ambient occlusion in screen space
- vector and matrix library fully optimized using Altivec VMX intrinsics

8/05 – 9/07: Senior Graphics Engineer, [Perpetual Entertainment](#) – San Francisco, CA

[Gods & Heroes – Rome Rising](#): Massively Multiplayer Online (MMO) game for the PC

- mainly responsible for the rendering pipeline and shader architecture
- improved shadow rendering algorithm by increasing effective resolution of the shadow map
- post-process filtering architecture for easily adding post-render full screen effects (e.g. bloom, radial blur)
- extensive profiling and optimizations of various systems (shaders, rendering pipeline, vector, matrix and math libraries, etc.)
- geometry batching for reduced D3D draw calls, which leads to higher performance
- geometry compression for saving memory
- re-designed the shader architecture and replaced RenderWare's D3D rendering structures
- memory usage optimizations, memory managing, memory tracking
- numerous profiling and visual debugging tools to help programmers and artists improve the game

- multi-threaded asynchronous asset loading / streaming, highly optimized with minimal thread contention
- fully memory mapped file system, with and without packed files, with optional compression

7/00 – 8/05: Senior Graphics Technology Engineer, [Rainbow Studios \(THQ\)](#) - Phoenix, AZ (remote office in Emeryville, CA)

Shipped games:

- **3/05:** [MX vs. ATV Unleashed](#) for PlayStation2 and for Xbox
- **2/04:** [MX Unleashed](#) for PlayStation2 and for Xbox
- **8/03:** [Splashdown: Rides Gone Wild](#) for PlayStation2
- **11/02:** [ATV Offroad Fury 2](#) for PlayStation2
- **8/02:** [Mat Hoffman's Pro BMX 2](#) for PlayStation2 and for Xbox
- **6/02:** [Splashdown](#) for Xbox
- **2/02:** [Star Wars - Racer Revenge](#) for PlayStation2
- **11/01:** [Splashdown](#) for PlayStation2

multi-platform rendering engine (PS2, Xbox, PC):

- shader architecture enabling hundreds of combinations of shaders
 - VU1 microcode for PS2, DirectX shaders for Xbox and PC
 - shader generators used to maximize code sharing, and minimize rewrites
 - one can easily and quickly create a new shader for a new effect
 - various lighting techniques (diffuse, specular, iridescent, etc.)
 - various texture mappings (environment mapping, projected texturing, etc.)
- terrain renderer microprograms
- vector and matrix library (highly optimized for Pentium3/Xbox using SSE and for the PS2 using VU and MMI)
- quaternion library
- math library
- code optimization (algorithmic optimizations and instructional optimizations using Pentium and MIPS assembly, SSE, VU, MMI)
- portals for visibility culling
- hierarchy of matrices and bounding boxes for models
- wake water physics simulation optimization in SSE
- VU0 micromode vector and matrix library (runs in parallel with CPU)
- octree spatial subdivision for fast scene search and traversal
- efficient triangle stripping and vertex caching algorithm
- shader convertor that takes PS2 multi-pass blended shaders into single pass Xbox pixel shader combiners (highly optimized)

3D graphics research and development: [\[demos\]](#)

- **Procedurally generated and animated clouds and smoke volumes**
 - Dynamically renders and animates layers of clouds using dynamically generated noise
 - Smoke volume represented as slices of a procedurally generated noise volume texture
- **Post-processing filter architecture**
 - Highly optimal and extensible system for rendering multiple filter passes
 - Includes filters such as bloom, Gaussian blur, radial blur, spherize, and noise
- **Multiple detail layered terrain rendering**
 - Spline for low frequency hill details
 - Displacement map for next level of details
 - Bumpmap for finer details
 - Procedural noise textures for minute details, with numerous levels of detail
- **Water simulation and rendering on the GPU**
 - Computes motion of water with forces applied to it as a spring based force simulation
 - Renders water as a dynamically displaced height field
- ***Sort independent alpha blending***
 - Dissecting the alpha blend formula and using higher precision render targets (16-bits per channel) we can achieve a close approximation to the final color of multiple passes of alpha blending without having to do any sorting
- **Bicubic spline surface rendering**
 - renders bicubic Catmull-Rom spline surface

- could render different uniform splines just by changing basis function matrix
- dynamic lod tesselation
- each surface patch has 4 edge lods and a separate center lod, to avoid cracks between adjacent patches with different lods
- **Deferred lighting**
 - uses floating-point render targets and cube texture maps, in addition to multiple render targets
 - per-pixel lighting and shadowing for multiple point light sources done as a fullscreen post process
 - depth buffer shadow mapping supports self shadowing, and works in a single pass for 4 lights, computed at the same time as the lighting
- **Procedural texture mapping**
 - procedurally generated noise texture maps
 - completely generated in a pixel shader
 - pixel shader generates random colors and computes noise function
 - virtually unlimited texture resolution at essentially zero memory cost
- **Procedural volumetric lightmap with bumpmapping and offset displacement mapping**
 - per-pixel lighting for 8 point lights
 - volumetric lightmap generated procedurally in pixel shader
 - texture coordinates are moved away from view direction to give a visual effect of per-pixel depth using a displacement map (matching the normal map used for the bumpmap lighting)
- **Shadow mapping**
 - depth buffer shadow mapping with self shadowing
 - improved perceptual resolution with a trapezoidal projection warping
 - multiple levels of percentage closest filtering (PCF) for smoother shadow edges
- **Displaced subdivision surface**
 - subdivision surface with displacement mapping
 - triangles are treated as barycentric Bézier patches
- **Slick surface**
 - surface defined by a displaced cylindrical shape following a cubic Cardinal spline as the center axis
 - each section of the base spline is tessellated by a section of a displaced, scaled and twisted cylinder
- **Displacement mapping**
 - mapping a displacement to an arbitrary triangle
 - normals are quickly and efficiently interpolated across the triangle, using spherical linear interpolation (quaternions)
 - interpolated normals don't require re-normalization
 - almost all work is done on the GPU, including transformation to tangent space, displacement, and surface normal computation from the displaced surface
 - CPU is only used for tessellation (dynamic), displacement map lookup, and normal interpolation
 - next generation shader model 3.0 will allow texture lookup in vertex shaders, which will further speed up performance
- **DX9 shader effects**
 - uses HLSL shader language
 - per-pixel effects, including bumpmapped lighting, bumpy environment mapping, parallax displacement mapping

2/98 - 7/00: Software Engineer, [3dfx Interactive Inc.](#) - San Jose, CA

3D graphics research and development: [\[demos\]](#)

- **Donut demo:** multi-texturing techniques - environment mapping, bump mapping, specular lighting
- **Surf Demo:** rendering spline surfaces (B-Spline and Bézier) with dynamic level of detail patch tessellation
- **Scene Renderer:** fast high polygon scene rendering (highly optimized for PentiumIII KNI SIMD)
- **Displacement mapping:** mapping a height field to an arbitrary quad (using quaternions to interpolate the normals)
- **Terrain Engine:** rendering a uniform height field terrain
- **Water Simulation:** simulates the motion of water using a highly tessellated grid
- **Skinned Animation:** vertex animation using hierarchical bones and weighted vertices for organic animation effects
- **Lightmapping:** lightmaps generated from multiple dynamic colored light sources
- **Shadow Volume:** real-time pixel accurate shadow casting algorithm

Game porting/helping:

- **Unreal:** helped make the game run properly on Voodoo Rush

- *Half-Life*: ported the MiniGL version of the game to Glide3; also added detail texturing

Driver work:

- *3dfx Splash Screen*: added clipping, anti-aliasing, lighting, and environment mapping; minor optimizations
- *MiniGL driver*: unifying the MiniGL for all MiniGL based games; helping with AMD's 3DNow optimizations

1/96 - 1/98: Software Engineer, Cyclone Studios ([3DO](#)) - San Mateo, CA

Uprising (action/strategy PC game, released 10/97):

- *The Level Editor*: 3D pipeline used to render and modify a terrain database
- *The Game*: Special effects, special weapons, rasterizers, hardware support

Third Domain (3D terrain strategy game - cancelled):

- 3D terrain engine; optimizations

2/95 - 12/95: Software Engineer, [Integrated Manufacturing Laboratory](#) - UC Berkeley, CA

Milling Machine Software Utilities:

- Milling machine graphical simulator and path generator
- Tool picking algorithm for the milling machine

Research/Publications

3/07: [Sort-Independent Alpha Blending](#)

GDC 2007 – using modern graphics hardware, we can render alpha blended objects without needing to sort

3/00: Advanced Lighting Effects - Overbright halos (high dynamic range lighting, aka HDR)

Total Immersion 2000 - using numerous texture passes to create overbright halo effects (simulate Gaussian blur)

3/00: Dynamic Lightmaps

Total Immersion 2000 - multiple colored lightmaps for real-time applications

3/98: B-Spline Surfaces

Total Immersion 1998 - fast rendering of spline surfaces, using variable patch subdivision (adjacent patch stitching)

4/97: [Interactive Generation of Scherk-Collins Sculptures](#)

Proceedings of the 1997 Symposium on Interactive 3D Graphics (SI3D'97)

8/95 - 8/97: [Professor Carlo H. Séquin](#) - UC Berkeley, CA

Complex single-sided [Scherk Surface](#) generator/renderer written on IrisGL and OpenGL

Education

8/91 - 12/95: [University of California at Berkeley](#)

B.S. in Electrical Engineering and Computer Science

1997 – present: [SIGGRAPH](#) Conferences (attended most years)

1996 – present: [Game Developer Conferences](#) (attended most years)

Interests

Video games, Computer Graphics, Programming, Code Optimization, Sports Cars, User Interface, Electronics, Astronomy

Organizations

[Eta Kappa Nu \(H.K.N.\)](#) - Electrical Engineering and Computer Science Honor Society

[Computer Science Undergraduate Association \(C.S.U.A.\)](#)

[U.C. Berkeley Karate Club](#) (nidan - second degree black belt)

Demos and references are available upon request.