

Tim C. Schröder

Email: tim@blitzcode.net | Skype: blitzcode | [LinkedIn Profile](#)
Please see www.blitzcode.net/contact.shtml for most recent contact details.

Experienced Software Engineer and Computer Graphics Specialist

Diverse background ranging from 3D graphics for games to physically based rendering. Solving challenging parallel programming problems, large datasets, realtime constraints, researching and implementing computer graphics algorithms on everything from a game console to a shared memory supercomputer. Writing maintainable and efficient code for shipping software products.

Skills

Programming Languages

- Strong C/C++ (Many different platforms / environments, STL, Projects with >1M LoC, 10+ Years)
- Working knowledge of C#, Managed C++, Lua and other scripting languages
- Various GPU Languages
- HTML / XML

Windows

- A decade of Windows programming practice (deep knowledge of tools and platform)
- Win32/64 Expert (Processes & Threads, Memory Management, DLLs & Binary Compatible Interfaces, Networking, DirectX, User Interfaces, GDI, Common Controls, etc.)
- .NET Framework: Windows Forms, Managed DirectX3D
- MFC experience

Linux / UNIX

- Development experience on Linux and others like Darwin / BSD, SGI IRIX / Altix etc.
- Working with standard tools, Makefiles, Vim power user, Highly networked environments (ssh, X11, scp, rsync, etc.), pthreads and other POSIX APIs, Cygwin, ...
- GCC and GNU Binutils, ELF file format, Build optimizations for CellBE SPUs
- Debugging of complex multithreaded applications with TotalView Debugger, gdb and its various front ends, also familiar with SGI ProDev WorkShop Debugger (cvd)
- Intel VTune Performance Analyzer for Linux

Macintosh

- OS X power user with development experience, mostly from a UNIX perspective using multi-platform tools and APIs, Apple Shark profiler, etc.

Graphics Programming

- Everything from realtime 3D for games to physically based rendering, techniques such as Hardware Shader Programming, Path Tracing, (Quasi) Monte Carlo Methods, Photon Mapping, Surface Parameterization, Spherical Harmonics, Progressive Meshes, Shadow Mapping, Spectral Rendering, Tone Mapping, Environment Map Lighting, ...
- 10+ years of experience with DirectX3D and OpenGL
- Graphics experience on multi CPU / GPU systems, Game Consoles, CellBE and GPGPU platforms
- Microsoft PIX and other DirectX SDK tools, FX Composer
- Vertex and Pixel Shaders (Assembler, Cg, CgFX, GLSL, HLSL and FX Files)
- mental ray HW Shaders, Phenomena and .mi Scene Description Language

Software

- Visual Studio power user
- Basic knowledge of DCC tools (Maya, 3ds Max and Softimage)
- PLAYSTATION 3 SDK & GameOS tools (including PA Suite, ProDG debugger), IBM and Sony Linux CellBE SDKs, Classic Xbox SDK, Basic knowledge of the 360 SDK
- VMWare, Doxygen, VTune, Quantify, SVN / CVS, SourceSafe, FogBUGZ, Bugzilla, Jira, Visual Assist, IncrediBuild, Pulse, ...

Work Experience

NaturalMotion (January 2010 – Present, Senior Runtime Engineer)

- Engineer on the PC and console (PS3 / 360) runtimes for morpheme / euphoria game animation middleware (optimization, refactoring, development of new animation nodes, ...)
- Direct3D based renderer (shadows, outdoor lighting) for demonstration application

mental images (May 2004 – December 2009, Senior Graphics Software Engineer)

- Autodesk ProMaterial shader support for iray renderer
- Optimization work on mental ray's BSP2 renderer (memory, intersection)
- Early development of a relighting tool, including algorithm / data structure design, implementation of the GUI and file formats (project put on hold for business reasons)
- Worked with a large range of CellBE platforms: IBM QS20/22 Blades, Sony BCU-100 and PLAYSTATION 3 (GameOS & Linux), implemented advanced concepts such as cooperative multithreading for SPEs, a pipeline for loading and executing MetaSL SPE shader code, a C++ template library for DMA transfers, trained and supported team members in all things CellBE
- Researched raytracing algorithms for CellBE processor using simulator / IBM Blade, successful prototype led to contract for a CellBE port of mental ray
- Developed and documented strategy to port raytracing algorithm from proprietary chip to NVIDIA CUDA platform, including implementation of a CPU based prototype
- Maintained and enhanced mental ray's OpenGL / Cg rendering backend while doing a Direct3D / HLSL / FX File port of it
- Developed scalable rendering software on multi-GPU machines from SGI and PANTA Systems
- Worked on next-generation rendering platform for RealityServer 2.0: GLSL / Cg, Tile Rendering, OpenGL, Texturing, DDS Support, Shadowmaps, Anti-Aliasing, Cubemaps, ...
- Assisted in MetaSL / mental mill development (Cg backend, Direct3D rendering, early evaluation of an VM platform, builds for external employees, CellBE port)
- Early design and implementation of 2D annotation renderer for RealityServer 2.0
- Supported hardware shader writers for 3ds Max and SolidWorks
- Created testsuite to ensure quality of mental ray hardware rendering
- Developed in a UNIX-centric environment using Linux / IRIX / Cygwin

Freelancer (March 2004 - May 2004)

- Implemented Photon Mapping for walkthrough applications as C++ library for KPB
- Created a lightmap compiler and viewer using C++ and OpenGL for S&P Software

Crytek (March 2001 - August 2003, Software Engineer)

- Developer on tools, game and engine code for Far Cry / CryENGINE
- Worked on WYSIWYG game world editor (MFC application, procedural tex. generation & object placement, heightfield editing & lighting, game engine integration, ...)
- Did basic rendering R&D and a proof of concept port of CryENGINE on the Xbox
- Wrote game logic code in C++ / Lua (had to work in multiplayer scenarios)
- Implemented Lua script code debugger integrated in game & editor (including GUI)
- Programmed lightmap system that combines classic lightmaps with bumpmapping

References for these companies are available on request.

Work Examples

To see various examples of my previous work, please visit the projects page on my website (<http://www.blitzcode.net/projects.shtml>).