Brian Richardson
BrianRichardson@knowhere.net

Objective
To become a computer game programmer or obtain a position that involves computer graphics.

Summary of Qualifications
- Professional software developer with nine years of experience shipping software mainly using C++, C# and Delphi.
- Expert on using the Windows platform to develop multi-component applications, and making them all work together reliably.
- Possesses excellent debugging skills. Able to use debugging tools to discover how undocumented processes work.

Professional Experience
Senior Software Engineer
GarageGames, Eugene, OR, June 2007-Current
Project: Juggernaut / Torque 2
- Refactored the lighting code to allow for new lighting systems to be used at any time
- Prepared the rendering engine for a DX10 backend by adding state blocks and shader constant blocks.
- Fixed a lot of rendering and system bugs.

Technical Application Developer
Ensequence Inc., Portland, OR, January 2006-June 2007

Project: MLB.TV Mosaic 2007, Technical Emmy Winner: Outstanding Achievement in Advanced Media Technology for Best Use of "On Demand" (Consumer Scheduled or Programmed) Technology Over Broadband Networks for Active "lean-forward" Viewing (PC)
- As technical lead for this year's version of the mosaic, I was the customer contact point and directed much of the internal development.
- Developed the ability to do multiple regions of pan and scan for the Windows and Mac rendering engines.

Project: MLB.TV Mosaic Mac
- Created a multi-threaded streaming video engine with graphics compositing based on Quicktime and OpenGL that was used by a Java client via a JNI interface.
- Implemented application auto-update mechanism based on Sparkle

Project: MLB.TV Mosaic Windows, Emmy Nominated Application for "Outstanding Achievement for Enhanced or Interactive Programming: New Delivery Platforms"
- Added streaming video capabilities to an existing DirectX based player application. This involved debugging deadlocks introduced by network latencies. Also added the ability to do multiple streams at the same time.
- Created a DirectShow source filter based on the Windows Media Format SDK which performed better than the default WMV filter provided by DirectShow. Also provided other features that were not available such as multiple audio track support.
- Helped develop data gathering module in C#. This required interfacing to a website without a webservice exposed, and reverse engineering it.
- Implemented an application auto-update mechanism based on Microsoft Best Pattern's and Practices

Project: Television Set Top Box Development
- Created a DVR component for use in Interactive Advertisements

Senior Software Developer

Retail Dimensions, Portland, OR, August 2005-December 2005

- Developed plugins for the product Retail Pro using Borland Delphi.
- Developed various programs to scrape web sites, that involved looking at HTTP packets and interfacing with other systems.

Independent Contractor
August 2004-Current

Project: Shelled!
- Added tank functionality
- Enhanced audio code using the OpenAL library
- Modified and implemented GUI for game
- All modifications done to Torque Game Engine (OpenGL & DirectX) in C++

Project: GarageGames Bounties
- fxFoliageReplicator port, using DirectX 9 and HLSL vertex/pixel shaders, I ported a billboarding particle system designed to fill game levels with foliage objects.
- Rendering modifications, using DirectX9 and HLSL vertex/pixel shaders, I made various modifications to the engine to allow easier debugging.

Project: The Nisha Chronicles
- Exported vehicle models for use in Torque Game Engine
- Modified vehicle physics code to support a motorcycle.

Project: Virtual World Prototype
- Added support for P5 Virtual Reality Glove to the Torque Game Engine.
- Added custom skeleton code to link the glove to the player object.
- Created demo level that resembled a virtual mall.
- Created interactive items for the user and scripted it all together. Items were created with Blender.
- Created multiple skins for the two available characters.
- Tweaked joystick support, the user navigated the level using the joystick and glove as the interface.

Senior Software Developer
Action Systems Inc., Silver Spring, MD, April 1999-July 2005

Project: "Restaurant Manager Backoffice", and other supporting programs. Released and supported three revisions of the software.

- Developed and maintained Backoffice setup and reporting systems using Borland Delphi.
- Developed efficient and easily maintainable user interfaces.
- Gathered reporting requirements and implemented them with QuickReports.
- Developed a custom reporting system that allowed the quick generation of new reports. System allowed factoring common tasks such as sorting, filtering, and grouping out of the individual report code.
- Repackaged the reporting system to be used by other applications in our system. Implemented as a DLL.
- Worked with third parties to interface software to three different credit card processing solutions, fingerprint recognition, and debit card readers.
- Interfaced to various pieces of hardware, using serial and parallel port interfaces.
- Used localization components and Windows API to make sure the backoffice would work in multiple languages.
- Refactored a large options dialog that was getting very hard to maintain into a dynamically created tree of options based on an XML definition file. This allowed other programmers in the company to easily add new options and provided a consistent interface for the users.

Project: ASI Handheld POS. This was a PocketPC based restaurant POS system. It used 802.11b to communicate with our standard POS system. Released and supported two revisions of the system.

- Using Borland Delphi, I developed a multithreaded server for the handhelds to communicate with the core software. The server consisted of two components: The first was an ISAPI DLL that was used by IIS to implement the SOAP interface. The second was a COM server that provided the actual functionality and state management. The SOAP interface was described with WSDL to document an interface for potential third party development.
- Server included a process queue based on thread events to provide access to a non-thread safe library. The event based implementation is very CPU efficient and scaled very well.
- Designed the server to have the ability to load balance the handheld clients with other servers if necessary. It is also designed to be reused in the future for other client types.
- Using C++, developed the communication layer for the handheld running Windows PocketPC.
- Handled multithreaded code on the handheld client. This allowed the user to continue taking orders while the handheld was communicating to the server.
- Made sure we made efficient use of the PocketPC's resources. For example, I implemented certain operations using memory mapped files instead of normal file I/O to prevent duplication of data in the PocketPC's memory.
- Implemented load testing to make sure the system would perform well under stress. This test is also used to discover memory leaks and other issues caused by load or concurrent transactions.
- Was able to quickly create an interface for a third party POS system for prototype purposes. I took a large body of 15 year old code that was poorly documented and was able to interface our handheld software to it in three days.

Project: ASI Online Web Ordering Prototype

- Built initial prototype in C#.
- Used the server created for the ASI Handheld to interface with the rest of the POS system.

Project: Central Office Manager: This product allowed our users to poll each store and retrieve information for reporting purposes. It also allowed the central office to define menus, employees, and other setup data with central control.

- Used Microsoft DCOM to create a distributed solution for multi-store chain accounts. Supported internet and dialup locations.
- The object oriented design made it possible to later update the software to support SOAP as a transport with minimal code changes. This reduced configuration issues when using the software over the internet.

Project: QuickStaffer: This product allowed users to automate employee scheduling. It allowed one to define staff levels, employee availability, and the scheduling algorithm would generate a schedule.

- Developed as a three tiered application.
- The database layer was replaceable. It could use our default DBF structure or store data in a SQL database.
- The business layer contained no user interface and could be reused for other purposes.
- The user interface layer contained UI to show the user in real-time an employee's availability, the desired staffing level, and the current staffing level. This allowed one to tweak auto-generated schedules, or manually schedule employees with ease.
- Created notification application to notify employees via email when their schedule changed.

Project: RMBrowser: This application reused our report code to allow clients to access their reports via a web browser.

- Refactored the reporting code to work in a web environment.

- Initially developed a custom viewer application similar to a PDF file to display our reports. Later this was switched to generate HTML, CSV, or PDF files. Allowing a user to access their reports without any client side software installation.

Other tasks:
- Developed a application to export data from ACT! to our SQL server.
- Implemented a daily build system.
- Implemented an automated installation build. This eliminated errors introduced by manually performing the steps needed to produce the CD we shipped to clients.
- Started "wiki" for our office to document development processes and other office procedures.

Customer Interaction:
- Taught report training classes given to our resellers.
- Provided tech support to resellers for escalated issues.
- Participated in reseller community forums available on our web site.

Contractor
Maxim Group, Baltimore, MD, November 98-April 99

- Created automated test scripts using Computer Associates QARun tool to test Oracle Financial applications for Baltimore Gas and Electric.

Web Developer
Stock Annapolis, College Park, MD, January 98 - November 98

- Created a website to sell stock photography.
- Developed using Perl and run with MySQL and Linux

Windows Programmer / Intern
Action Systems Inc., Silver Spring, MD, Nov 96-Aug 98
- Initial work of porting DOS based Backoffice and Inventory Clipper applications to Windows using Borland Delphi.

Shareware Author
Super Dooper Swimmer's Logbook
Whiteford, MD
July 1996-1997

- Created a DOS based application to track swimmer's times, weight training schedules, and diet.
- Used Turbo Pascal 7.0 and the TurboVision application framework.
- Sold copies on the United States Swimming Association BBS, and at local swim meets.

Skills

Languages:
- Expert with C, C++, C#, Borland Delphi, Macromedia Flash
- Intermediate with Objective-C, Java, Perl, ASP, PHP, Visual Basic, Clipper / dBase
- Knowledge of Ruby, Intel x86 Assembler, Emacs Lisp, Basic

Databases:
- dBase/Clipper, SQL Server, ADO, ODBC, BDE, MySQL

Technologies:
- Communication/Interop: SOAP, WSDL, DCOM, COM, Berkeley Sockets, XML, DOM
- Application Frameworks: Windows API, MFC, VCL, Motif, GTK, Carbon, Cocoa

- Graphic Frameworks: GDI (Screen, Printer), OpenGL, DirectX (Direct3D, DirectShow)
- Streaming Media Frameworks: Windows Media Framework, QuickTime
- Web: ISAPI, CGI
- Internet Protocols: HTTP, SMTP, POP3, IRC, MMS

Development Environments:

- Delphi 1.0-6.0, Visual Studio 6.0 - .NET 2003, Embedded Visual C++, vi with makefiles, Xcode, Eclipse, ANT, Emacs

Software Products:

- Build/Deployment: InstallShield, Finalbuilder, Nullsoft Scriptable Install System
- Testing / Debugging: VMWare, QARun, Entrek Toolbox (profiling tool for Windows CE), Hex Workshop, Ethereal, Fiddler, TCPTrace, PIX for Windows, NVPerfHUD
- Source Control: Starteam, CVS, Subversion, SourceSafe
- Web servers: IIS, Apache
- Game Engines: Torque Game Engine, Torque Shader Engine, Torque Game Builder

Operating Systems:

- Windows 3.1 – 2003, Windows CE.NET 4.x, PocketPC, Linux (Redhat, Mandrake, Gentoo), FreeBSD, OpenBSD, DOS 3.2-6.0, Mac OSX 10.4

Education

GameInstitute.com, Dec 2004 – Feb 2005, Game Mathematics
This is an online course that reviews vectors, matrices, quaternions, and other mathematics used in video games.

University of Maryland, College Park 1996-1997
Completed 32 credits.