Lee Griffiths

07843 119 789 lee@lee-griffiths.net

Available for work in Hertfordshire, Cambridgeshire or surrounding areas. **Willing to relocate**.

http://uk.linkedin.com/in/LeePGriffiths https://bitbucket.org/Poddster http://stackexchange.com/users/23045

Digital version of CV, containing *much* more detail can be found at http://www.lee-griffiths.net/cv/

Overview

Software developer. **7 years' experience** writing WDDM/**DirectX** 2D/3D **graphics drivers** in semiconductor industry for *Imagination Technologies*. Left in Aug 2015 for a career break, now looking to get back into work.

Worked on large-scale GPU projects, involving many HW and SW teams, targeting fixed and expensive silicon fabrication deadlines. Experienced full development life-cycle whilst writing **embedded, low-level, real-time code** for **drivers and firmware**, and **writing tools** for use by developers. Completely comfortable writing **non-driver code**: desktop apps, 3D apps, utility scripts. Targeted various OS (Windows, Linux, custom RTOS) on different form-factors (simulated GPUs, tablets, phones, desktop).

Can design, code, test and document software to a high standard. Take pride in professional responsibility, and the thorough design and testing of software to ensure its correctness. Capable of working on **solo projects**, in teams, or as part of a **300-person effort** with relatively **little supervision**. Capable of independent action to get information and work from other teams without hand-holding. Not daunted to work on projects that many other people rely on. Gained some experience in leading and supervising junior developers in their work and training others.

Key skills: C, Python, graphics drivers, debugging IEEE 754 floating point (C, GPU shader code)

Professional Experience

Imagination Technologies, Kings Langley, Hertfordshire August 2008 - August 2015

Leading Software Design Engineer

Graduate S.D.E. (Aug 2008 – Aug 2010) S.D.E. (Aug 2010 - Jan 2015) Leading S.D.E. (Jan 2015 - Aug 2015)

Imagination's PowerVR GPUs used by leading-brand smart devices (e.g. **all Apple iPads & iPhones**). DirectX driver shipped on Windows-based laptops and tablets, and helped in validation and verification of all shipping GPUs.

- Worked as **a senior member** of the WDDM team (~30 people), writing WDDM/DirectX 9, 10, 11 drivers for Windows XP/Vista/7/8/10. User-land and kernel mode C.
    - Experienced full software development life-cycle. Driver started as **0-file project** and delivered to customers *(Intel, Allwinner*) with **full WHQL pass** on Windows (Vista, 7, 8.1), on Intel x86, x64 and ARM based platforms.
    - **Mentored** new/junior engineers. **Supervised** feature implementation. Helped **train**, educate and present DirectX topics to SW/HW engineers. Advised HW on DirectX spec requirements. **Signed-off** some HW specs.
    - Indirectly **supported customers** via technical answers in tickets (IPGear, TeamTrack). Have been "responsible engineer" for WDDM/DX team when cores are taping-out, or when customer's validation failed on a DX test.
    - Debugged/Reverse-engineered DirectX API output of many 3D apps, games & game engines to fix faulty images.
    - Debugged **IEE 754 floating point** problems in shader code, HW FPU designs, software FPU implementations.
    - Found/debugged **many hardware bugs** in simulated, prototyped- and live- hardware designs, all would have been **very expensive** to fix if not found. Created driver workarounds for "won't fix" HW problems.
    - Worked with various build systems for driver and tools. nmake/msbuild/Visual Studio.
    - Understood the maths involved in creating 3D and 2d graphics and used this to write driver code and 3D apps. Helped educate junior team members in terms of related maths problems.
- Responsible for **design, architecture and implementation** of large sections of the DirectX driver codebase:
    - "Shader stack" (compilation, execution, data/code memory management), Dx11 tessellation, GPU command stream, reading "apphints" from windows registry and .ini files, log file dumps.
    - Had significant input into all other areas of the driver. Worked on every file in the driver codebase.

- Worked on many internal **useful tools**. E.g. Python scripts for: C-codegen, syncing different source-control.
- Regular contributor to **vital** debug tool/fast GPU simulator used by all engineers in PowerVR (~300 people)
  - **Performance sensitive** C code (.dll, .so) with various frontends: text (Python) & graphical (C#/DirectX/OpenGL).
  - Helped code review all patches for objanal. Contributed to design and architecture discussions.
- Helped innovate WDDM/DirectX team with **modern and efficient working practices.**
  - Helped introduce *mandatory* peer **code-review** system into the team (via **ReviewBoard**)
    - Acted as 'back stop' to ensure every single commit on ReviewBoard was reviewed, usually by myself.
  - Helped introduce and implement a **continuous-integration regression test environment** for the driver.
    - QA/testing tools: automated, **distributed**, multi-threaded tool for test scheduling/execution; test image comparison and error reporting; email delivery reports, system configuration parsing scripts. All **in Python.**
    - A **dll extension to Python**, using the Python/C API, for DX10 API. A set of Python scripts and 'shell' environment, using comtypes and ctypes libraries, for DX9 and DX11 APIs. Allowed 3D test-apps to be written in Python, rather than in C or C++ using COM/DirectX.
    - Wrote hundreds of DirectX 9, 10, 11 3D test apps. Majority in Python, few in C or C++.
    - Maintained report hosting website. Django (earlier PHP) for dynamic content generation.

CS Department, University of Manchester *Summer Vacation Student* June 2007 – September 2007

Paid to start work early on third year project. Developed materials for new second-year systems-level course.

PEVE group, University of Manchester *Summer Vacation Student* June 2006 – September 2006

Created a new website for PEVE group to match “new style” template on CS department's webpages (cs.man.ac.uk).

Academic Qualifications

2005 – 2008	**University of Manchester, School of Computer Science**	B.Sc. Computer Engineering (Honours), Class 2:1
2002 – 2005	**Cardinal Newman College, Preston**	A-Levels in Computing(A), Maths(B), Environmental Science(B), General Studies (C), Accounting(D)
1997 – 2002	**St. Bede's High School, Lytham**	11 GCSE grade A-C

School of Computer Science, University of Manchester August 2005 - July 2008

Usual CS topics (algorithms & data structures, databases, AI, etc); Micro-controllers, Digital design of hardware (circuits, CPUs and systems on chip); Operating system design; Programming language theory & compilers

Example of Academic Projects

- Simple multi-threaded servers: e.g. telnet, ftp, a "hotel booking" system. (Java)
- Client software to book slots from a server running a booking database. (Java)
- FAT12, process scheduler, virtual memory implementations. (Java)
- MPEG audio encoding software intended for mobile devices. (Java)
- Developed a compiler & interpreter for an artificial, C-like language (lex, yacc, C)
- Developed an I^2C (I2C) peripheral for an AT91 (ARM7TDMI) in Verilog. Synthesized onto Xilinx Virtex FPGA.
- Developed "Stump": An ARM-like 16-bit RISC processor in VHDL. (Only simulated, too big for FPGAs available in CS department at the time).
- Full custom CMOS layout for a ripple carry adder (Cadence suite)

Final Year Project

Hand-picked by University staff to create hardware platform for new 2nd year course. Worked with one other student. Large freedom of design, single requirement was to use two existing boards, one ARM and one PIC based. Asked to design a complete system and sketch feasible applications that could be run on it. Delivered technical report (~20k words) as final year dissertation. Also produced handover report documenting the structure and function of the code/system for the course organisers. Full development life-cycle, taking it from a basic spec to a full working system.

The course apparently used the platform and codebase, with modifications, for a few years with no problems.

Final design: Camera module, rotatable around two axes. Analogue-digital conversion done via VDEC1 co-processor. PIC controlled rotation via stepper and brushed-DC motors. ARM processed images. FPGA on ARM board helped accelerate image operations. All talked via I^{2}C. Working example provided to staff was simple colour space conversion.

Technical skills used in the project

- Made basic **PIC RTOS** for student’s use. **Drivers** for PIC and ARM: **Motor control** (stepper, brushed DC), communication (**I^{2}C**, RS232 and custom protocol), Character LED, basic user-space stack-tracer for debugging for ARM. Custom build + deploy process and FPGA interfacing (Bash scripts, C, ARM assembly).
- Programmed: PIC18, AT91SAM9 (ARM9), video decoder (VDEC1/ADV7183B), flash EEPROM, Xilinx Spartan 3.
- Experience with Microchip's MPLAB software for programming & debugging the PIC.
- The project's software was written in C (PIC RTOS and drivers, Linux tools), PIC assembly, ARM assembly. The image processing/colour-space conversion running on a Xilinx Spartan FPGA was written in **VHDL**.

Personal Projects

Super-Sleuth https://bitbucket.org/Poddster/super-sleuth

Prototype for procedurally-generated detective game featuring the "realistic" simulation of people, their emotions and crimes. (~4k kloc Python). Documentation and design for full game underway.

Programmable Flight Computer for Kerbal Space Program https://bitbucket.org/Poddster/ksp_pfc

Attempt at "full system" computer emulation inside of KSP. CPU emulation done. Circuit simulation in research + design stage. LLVM backend planned. ~12kloc lines of C#, ~4.5kloc of test programs written in ksp_pfc assembly.

More projects available on https://bitbucket.org/Poddster

Technical Skills

Programming languages, technical skills, and technology use

Advanced Use	C (c89, c99), DirectX 9/10/11, HLSL (shaders)
Everyday Use	Python, IEE 754 (C & GPU HW), Drivers, WHQL/WHCK/WLK, concurrent data access
Frequent Use	Assembly (x86, x86-64, ARM, GPU), Direct2D, doxygen, nmake (Microsoft version), msbuild, CMake, MinGW, MSys, SDL (via both PyGame and C API), C#, C++ (C++98, C++03)
Infrequent	Java, c11, OpenGL 4.5, GLSL 4.5, gdb, POSIX/Gnu/Linux utils (grep, sed, awk, etc), Win32 API, COM, XNA 4.0 (+ MonoGame), HTML 5.0, CSS 3, Event Tracing for Windows (ETW), MISRA C
I used this once to do something useful and still vaguely remember how to use it	Android, DirectX12, Vulkan, OpenCL, Pic18 assembly, Perl, MATLAB, MPLab, Apple's Metal API, AMD's Mantle API, SystemC, Ruby, Verilog, VHDL, Modern C++ (C++11 C++14), PHP, Oracle SQL & MySQL, LLVM backend + LLVM language ref, Lex, Yacc, Bash scripts

Tools

Advanced Knowledge	git, Pix For Windows
Configure/use	Mantis bug tracker, ReviewBoard, CVS, SVN, Mercurial (Hg)
Everyday Use	Windows, Linux, Visual Studio, WinDBG (kernel + user-mode), Perforce, MKS, Microsoft Project (Gannt chart), RenderDoc, Serena business mishaps (formally TeamTrack), CA Clarity PPM, Openproject, Bugzilla, Eclipse, Jenkins, Microsoft connect, Microsoft Driver Verifier,
Used a Few Times	GPUView, PVRTune, Intel VTune, ApiTrace, OllyDbg, Cadence Tools (Schematic & CMOS Layout editors. NC-Verilog/NCSim & Verilog XL simulators), Mentor Graphics (Schematic), SoftICE, XPerf, LaTeX

Hardware

Micro-controllers	All PowerVR GPUs from MBX to Rogue (series 3-7), parts of PowerVR Series 8, some PowerVR video & display cores, Intel Poulsbo (GMA 500), Intel Atoms, AT91SAM9621 (ARM9 based), AT91M40800 (ARM7), PIC18LF452, Xilinx Virtex & Spartan FPGAs, Analog Devices ADV7183B (Digilent VDEC1 video decoder chip). Some time with Arduino and Raspberry Pi.
Peripheral Hardware & Protocols	PCI bus, I2C, UART, USART, RS232, SPI. Analogue and digital PIOs. Interrupts, Power Management controllers etc. Motor control, Character LCDs. Watchdog/One-shot timers, 555 timers, etc.