Sharadh Rajaraman

+44 7512 070176 | r.sharadh@outlook.sg | sharadhr | sharadhrajaraman

Summary

Software engineer working on graphics processor drivers for the Google Pixel smartphones. Previously contributed to high-performance compression and filesystems for genomics and media/entertainment applications.

Highly proficient in modern C++, with a strong foundation in computer graphics, systems programming, compiler design, and low-level programming. Have deep experience in desktop application development for macOS and Windows with Cocoa and Win32 frameworks, as well as embedded systems programming on STM32 microcontrollers.

Supports the tech community through open-source contributions. Has a strong passion for looking under the bonnet of software and hardware, and approaching problems with a scientific mindset. Effective communicator and team player, with a strong interest in mentoring and teaching.

Work Experience

Google

London, United Kingdom

Software Engineer (Pixel Graphics) — *October 2025 – Present*

- Learning about the Pixel GPUs and their supporting software.
- Working with senior engineers on the team to identify bugs and performance issues.
- Developing testing and analysis tools for Pixel GPUs.
- Working with teams across Pixel, Android, and Google to ensure that their software takes advantage of the Pixel GPU's capabilities.
- Collaborating with partners around improvements and new features.

PetaGene Ltd

Cambridge, United Kingdom

Software Engineer (C++, Objective-C, macOS and Windows desktop programming, CMake, Jenkins, C#/Avalonia, Windows driver programming) — *August 2023 – August 2025*

- Spearheaded build tools transition from GNU Autotools and shell script to vcpkg and CMake. Reduced build and CI run times by 10×.
- Introduced optimisations for thumbnails in Windows Explorer and macOS Finder by reverse-engineering system libraries with IDA Pro. Improved responsiveness for media and entertainment customers by reducing network traffic by 90%. Used by a major film studio.
- Implemented local-file caching with macOS Finder extension.
- Applied static analysis and code quality tools like clang-tidy on the codebase, improving code quality, safety and maintainability.
- Represented the company at SuperComputing 2023, the largest HPC conference in the world in Denver, CO. Attracted ~100 leads, and several new customers.

Government Technology Agency, Singapore (GovTech)

Singapore

Embedded Software Engineering Intern (Sensors and IoT Division: C/C++, CMake, STM32) — *May 2022 – August 2022*

- Implemented a C++ wrapper over Linux Serial Peripheral Interface (SPI) syscall interface. Reduced wheel-reinvention, and improved linkage for other projects using C++.
- Reused COVID-19 contact-tracing tokens using STM32 microcontrollers, by rewriting firmware in C++ to emulate a Trusted Platform Module (TPM) over I^2C for Raspberry Pi (rPi). Reduced costs by recycling and reusing existing hardware.

Skills

Languages	C++, Objective-C, C#, Python, Java, LaTeX, GLSL/HLSL, F#, TypeScript, PowerShell
Frameworks & Tools	AWS SDK, Foundation, Win32, OpenGL, DirectX, Vulkan, OpenMP, CUDA, IDA Pro

Projects

Personal projects and contributions

vulkan.cppm (**Merge request**)

C++20 module for Vulkan-Hpp (C++, CMake)

- Adapted a code generator to output a C++20 module interface file for the Vulkan-Hpp wrapper library.
- Improved type safety and performance by exporting C macros and function-like macros as `constexpr` variables and functions.

Coursework

Oat Compiler

Compiler for statically-typed, C-like Oat language with Python-like list comprehension (OCaml, Menhir)

- Front-end outputs a subset of LLVM IR; back-end lowers to a subset of x86_64 assembly.
- Static single assignment (SSA) design to enable optimisations.
- Implements a type system with type inference, expression and statement typing, and type covariance/contravariance.
- Compile-time optimisations e.g. constant and expression folding, dead-code elimination, and register allocation with graph colouring.

Static Program Analyser

Lexer and parser for a C-like toy language (C++17)

- Lexer implemented with `std::regex` state machine; straightforward greedy algorithm.
- Recursive-descent LL parser, with left-recursion elimination and operator precedence parsing.
- Inserts information such as variable declarations, function calls, and control flow into a database about a given program written in the toy language.

cache-sim (**Repository**)

Quad-core cache-coherence simulator (C++20, CMake)

- Implements MESI, MOESI, and Dragon cache-coherence protocols.
- Correctly simulates cache-coherence behaviour of a real quad-core CPU, is configurable (cache size, associativity), and outputs statistics in `.csv` format.

Education

National University of Singapore

Singapore

Bachelor of Computing (Honours) in Computer Science; 2nd Major in Physics *August 2018 – June 2023*

- Computer science: parallel computing; real-time computer graphics; operating systems; compiler design.
- Physics: astrophysics; quantum mechanics; solid-state physics.

Extracurriculars

Indian classical music

Carnatic vocal, violin *September 2003 – Present*

- Completed diploma in Carnatic vocal music in 2012 and violin in 2013
- Performed solo since 2010 in Singapore, India, Australia, and the UK
- Conducted workshops and classes on Carnatic music theory and practice for beginners