Thom Chiovoloni — Rust Expert

chiovolonit@gmail.com • https://shift.click • thomcc
@at_tcsc

Professional Experience

TCDI

Senior Software Engineer *Oct. 2022 – May. 2024*

- Implemented several APIs in PGRX, a Rust framework for implementing PostgreSQL extensions.
- Critical in implementing PL/Rust, a PostgreSQL extension that offers a procedural language handler for the Rust language.
- Maintained a stripped-down fork of the Rust standard library for use inside PL/Rust.
- Designed and implemented a Rust static analysis framework that catches known soundness holes in the Rust compiler (for use inside PL/Rust).
- Implemented cross-compilation support in PGRX, ‘cargo-pgrx’, and PL/Rust (useful for deploying PL/Rust on PostgreSQL clusters with heterogeneous architectures).

Ockam

Lead Rust Engineer *Aug. 2021 – Jul. 2022*

- Leader of the Rust team at Ockam.
- Redesigned many APIs in Ockam’s user-facing Rust libraries.
- Oversaw and participated in bringing the Ockam Rust libraries to general availability.
- Rewrote the ‘ockam’ CLI for improved security, maintainability, and production readiness.

Cloudflare

Systems Engineer *Oct. 2020 – Jan. 2021*

- Worked on the JavaScript runtime for Cloudflare Workers in C++.

Mozilla

Sr. Platform Engineer *Mar. 2016 – Aug. 2020*

- Worked on Firefox Sync for both Desktop and mobile.
- Designed and implemented the “sync telemetry ping” which collects anonymous usage and error data from Sync users who enable Firefox telemetry.
- Did various performance and optimization tasks around the browser during the Firefox Quantum effort.
- Helped lead the effort to consolidate our 3 separate implementations of sync (Desktop, iOS, Android) into a shared set of Rust libraries (the “Rust Components”) that exposes idiomatic JavaScript, Swift, and Kotlin APIs.
- Designed and implemented the “Megazord” system, reducing the binary size overhead of the “Rust Components” by both deduplicating shared code (Rust stdlib) and allowing downstream applications to choose a “megazord” which contains only the components they need.
- Pioneered techniques and wrote a library for safely and easily exposing Rust to Android and iOS. This used Rust’s type-system to minimize unsafe code and keep the core of our components in idiomatic and safe Rust. As part of this, wrote lots of documentation, so that the whole team could write FFI code.
- Created “viaduct”, a HTTP library with pluggable backends, routing requests through “GeckoView” on Android, “Necko” in desktop Firefox, or a pure-Rust HTTP stack when running local unit tests.
- Implemented the “logins” Rust component, providing storage and sync for password managers, used Lockwise and mobile Firefoxes. This was the first Rust Component, and laid groundwork for the others.
- Implemented a number of other smaller pieces of the “Rust Components” including: The Sync 1.5 API client, optimized history search behind Fenix’s “awesome bar”, our logging adapter, the Swift and Kotlin APIs for Bookmarks (and FFI).

Muzzy Lane **Newburyport, MA**
Game Programmer *Feb. 2014 – Feb. 2016*

- Gameplay and engine programmer on multiple successfully shipped games.
- Implemented games and tools using Unity3D, HTML5, or our custom C++ game engine (Locust, which we eventually abandoned as it required a browser plugin).
- Implemented game client and server for educator-focused game authoring tools, using HTML5, WebGL, React, and other web tools.
- Designed and implemented a high-level node-based gameplay scripting system which was used by game designers to author content across several Unity-based games.
- Added a number of optimizations and features to the C++ game engine (text styling, garbage collected scripting, ...).
- Lead engineer on *Practice Spanish: Study Abroad*, a Spanish language learning game written for the McGraw-Hill Practice series, consisting of over twelve 10-20 minute “quests”. Shipped for Web (HTML5), iOS, and Android, using Unity.
- Lead engineer on *Practice Spanish: Mini-Games*, a set of companion games to *Practice Spanish: Study Abroad*. Written using pure HTML5 (no engine), shipped on the Web, iOS, and Android.
- Lead engineer on *Practice Medical Office*, another educational game focused on training employees in a medical office. Shipped for Web (HTML5), iOS, and Android, using Unity.

UConn Office for Sponsored Projects **Storrs, CT**
Web Application Developer *2011 – 2013*

- Responsible for the creation of internal web applications.
- Applications were written mainly in JavaScript, but used ColdFusion and Oracle SQL, or PHP and MySQL for the backend.

UConn School of Engineering (Computer Science Department) **Storrs, CT**
Undergraduate Teaching Assistant *Fall 2011*

- Undergraduate teaching assistant for UConn’s Introductory Computer Science course.
- Taught in Scheme using the textbook Structure and Interpretation of Computer Programs.

Miscellany

Rust:

- Expert of the Rust language with experience dating back to 2016.
- One of the maintainers of the official Rust standard library implementation, on the libs and testing-devex teams.
- Maintainer of many popular Rust crates (many millions of combined downloads), including ‘cc‘, ‘rusqlite’, and several others.

Technical Interests: Optimization, FFI (calling functions in one language from another), Data Compression (both lossy and lossless), Unicode, Databases, Low level programming, Concurrent programming, Threading/Thread-safety, Memory-safety and “unsafe Rust”, and much more.

Languages: Rust, C++, C, JavaScript, Kotlin, Swift, SQL, Scheme, C#, Java, Objective C, quickly picks up new languages.

Platforms: More than passing familiarity with: MacOS, Linux, Windows, iOS, Android, Web (JS/HTML/CSS), Web Assembly, Unity3D, and more.