VINCENT ESCHE

STAFF SOFTWARE ENGINEER

PORTFOLIO | 2025-05-14

in/vincent.esche
github.com/regexident
vincent.esche@gmail.com

RUST PROJECTS

The following projects represent a curated selection from a broader portfolio:

CARGO-MODULES

RUST

AUTHOR

ACTIVE PROJECT

A command-line tool designed to help Rust developers visualize and analyze the internal structure of their projects. By offering these insights, cargo-modules aids developers in understanding and managing the architecture of their Rust projects.

The tool provides commands to display a crate's module hierarchy as a tree, map internal dependencies as a graph, detect cycles within the crate's dependency graph, and detect unlinked source files within the crate's directory.

I've been actively developing/maintaining `cargo-modules` since 2016 with near-weekly releases. It recently crossed the 1000 Github star mark and has accumulated over 160,000 downloads on crates.io over the years.

code-analysis | graph-analysis | rust | rust-analyzer | visualization

TRYEXPAND

RUST

AUTHOR

ACTIVE PROJECT

A snapshot-testing harness for macro expansion.

Invoke `cargo expand` on a set of test cases, asserting that the macro-generated source code, or any resulting error messages are the ones intended.

cargo-expand | procedural-macro | rust | snapshot-testing | test-harness | unit-testing

ENUMCAPSULATE

RUST

AUTHOR

ACTIVE PROJECT

Safe casting for newtype enums and their variants.

A proc-macro crate that provides all the tools you might need for working with enum state polymorphism, providing derive macros for safe casting between enums and their (type-unique) variants.

encapsulation | enum-polymorphism | procedural-macro | rust

COMMUNITY

MEETUPS

I am an active member of the Rust Berlin Meetup, attending regularly and have also given talks. From 2016 – 2019 I was co-organizer of the renowned CocoaHeads Berlin meetup, focused on Apple's software technologies, with well over 1000 members, for which I organized and moderated monthly meetup events with an average attendance of 70-90 people.

MENTORING

I have taught Ruby at the RailsGirls and RustBridge beginners workshops several times in Frankfurt, as well as Berlin and was a team coach for the RailsGirls Summer of Code in 2016.

SWIFT EVOLUTION

I am co-author of the "Hashable Enhancements" (SE-0206) Swift evolution proposal, which improved Swift's security, performance and extensibility, replacing the existing error-prone legacy Hashing API with a modern and robust Rust-like API for the Swift stdlib. It was accepted and publicly announced by Apple at WWDC 2018.

SIGNALO

RUST

AUTHOR

ACTIVE PROJECT

Signalo provides the basic building-blocks for low-level real-time filtering pipelines, based on zero-cost, zero-allocation abstractions, which can be assembled via composition.

I developed "signalo" as part of my work at Runvi / NWTN Berlin, leading both, the iOS app development, as well as development of the Bluetooth-connected on-device human gait analysis DSP engine, which was based on signalo's DSP primitives. The DSP engine was written by me in pure Rust and integrated into an existing embedded C firmware. It replaced an existing simplistic C implementation that had been plagued with frequent crashes.

algorithms digital-signal-processing embedded real-time rust

QUADRATURE

RUST

AUTHOR

ACTIVE PROJECT

Implementation of a logic-level quadrature decoder, as well as hardware-level encoder.

A quadrature decoder with error detection, smoothing, and full-, half-, and quad-stepping support ensures accurate, noise-resistant motion tracking. It improves reliability in noisy environments and allows tuning resolution and responsiveness to fit specific motor control or rotary input needs, which can be crucial given the often noisy output of cheap rotary decoders.

embedded embedded-rust hardware-driver quadrature rust state-machine

ALLEN-INTERVALS

RUST

AUTHOR

ACTIVE PROJECT

An efficient implementation of Allen's interval relations for Rust's range types.

Allen's interval relations are used in temporal reasoning to describe how time intervals relate to each other, such as whether one interval occurs before, during, or overlaps with another. They are commonly used in artificial intelligence, natural language processing, and scheduling systems to represent and reason about temporal information.

constraint-programming rust temporal-logic temporal-reasoning time-intervals

SOFTWARE SECURITY

I reported a security exploit affecting the Swift Package Manager (SPM) to Apple in 2017, which would allow an attacker to execute arbitrary code during compilation. Apple resolved the issue by hardening the SPM build process via sandboxing, as publicly announced on stage by Apple at WWDC 2018.

DROPTEST

RUST

AUTHOR

ACTIVE PROJECT

A Rust helper crate for testing drop-semantics.

The library was specifically developed by me as part of my FFI work at Daily and provided the necessary tools for testing and thus ensuring correctness of otherwise very tricky memory semantics (of wrapped C++ WebRTC objects crossing the FFI), especially the mismatch of Rust's move semantics and C++'s move constructors.

memory-management rust unit-testing

GIT-ASSIST

RUST

AUTHOR

ACTIVE PROJECT

Git-Assist automatically skips all internal commits in pull requests, when running `git bisect`, making its use smoother and more efficient on repositories using the "Rebase and Merge" scheme.

With Git-Assist, you can enjoy a simple, linear history while still using git bisect to find bugs quickly. By skipping internal commits in pull requests, it combines the best of "Rebase and Merge" for linear history and "Squash and Merge" for efficient bug isolation.

git-bisect linear-history rebase-and-merge rust

RUST CONTRIBUTIONS

The following projects represent a curated selection from a broader portfolio:

RUST-ANALYZER

RUST

CONTRIBUTOR

ACTIVE PROJECT

The the official, modular compiler frontend for the Rust language.

Over the years I've contributed several (20+) pull requests (mostly to the public API of its high-level intermediate representation, aka the HIR) as part of my work on "cargo-modules" (and more recently a yet-to-be-released successor-project).

code-analysis compiler-frontend language-server-protocol official rust

ASCENT

RUST

CONTRIBUTOR

ACTIVE PROJECT

A logic programming language (similar to Datalog) embedded in Rust via macros that I have made over a dozen contributions to, focussing on ergonomics, testability, etc.

I've recently been using ascent in the context of recursive graph analysis, specifically for things like calculating the dominator-tree, transitive closure, reachability, neighborhood, ego-graph of a graph. I've made over a dozen contributions to the project, focussing on improving user ergonomics, API robustness, and improving overall code quality.

datalog graph-analysis logic-programming rust

WEBRTC-RS

RUST

CONTRIBUTOR

ACTIVE PROJECT

A pure Rust implementation of the WebRTC stack, which I contributed about 25k lines of code to, putting me in the top 3 contributors of the project, focussing mostly on the WebRTC data-channel and media-constraints.

My most significant contribution to the project was complete and spec-compliant implementation of the WebRTC constraints API based on the corresponding Media Capture and Streams W3C spec (25.000+ words), resulting in +9000 near-fully tested lines of code (240+ unit tests): #356.

algorithms audio constraint-solver rust video webrtc

RUSTFMT

RUST

CONTRIBUTOR

ACTIVE PROJECT

The the official, modular compiler frontend for the Rust language.

I made several of contributions to the project, back in 2017, when it was still very actively developed.

My most significant contribution to the project was the initial `Configuration.md` (#1474), that ended up morphing into the official configuration page.

code-formatter official rust

CBINDGEN

RUST

CONTRIBUTOR

ACTIVE PROJECT

CLI tool that generates C/C++11 headers for Rust libraries which expose a public C API.

I've over the years contributed several pull requests to the project, most significantly I've contributed support for `#[repr(transparent)]`: [185] (https://github.com/mozilla/cbindgen/pull/185).

c | code-generation | ffi | rust | tooling

SWIFT PROJECTS

The following projects represent a curated selection from a broader portfolio:

XMLCODING

SWIFT

AUTHOR

INACTIVE PROJECT

A clean and modular implementation of the Codable protocol for XML.

An efficient implementation of a push-parser-based reader, a visitor-based writer, a formatter for pretty-printing, a type-safe DOM document abstraction and last, but not least a flexible pair of encoders/decoders for working with XML.

codable | serialization | swift | xml

CACHE

SWIFT

AUTHOR

INACTIVE PROJECT

Useful caching data structures in Swift.

Swift implementations of a dictionary-like cache data structure with support for three fundamental caching strategies: FIFO (First-In, First-Out), LRU (Least Recently Used), and RR (Random Replacement), enabling efficient memory management by allowing developers to control cache eviction policies, thereby optimizing performance and resource utilization in applications.

cache-eviction-policy | caching | swift

SYNC

SWIFT

AUTHOR

INACTIVE PROJECT

Useful synchronization primitives in Swift.

Safer and more ergonomic synchronization primitives for Swift by encapsulating shared data directly within thread-safe types. This high-level design helps prevent common concurrency bugs by eliminating manual lock management and making synchronized access more intuitive and less error-prone.

locking mutex swift synchronization thread-safety

EVENTBUS

SWIFT

AUTHOR

ARCHIVED PROJECT

A safe-by-default pure Swift alternative to Apple's `NotificationCenter`.

EventBus is a safe-by-default alternative to Apple's `NotificationCenter`. It provides a type-safe API that can safely be used from multiple threads. It automatically removes subscribers when they are deallocated. It is to one-to-many notifications what a `Delegate` is to one-to-one notifications.

event-bus observer-pattern pub-sub swift

FOREST

SWIFT

AUTHOR

ARCHIVED PROJECT

A collection of persistent immutable trees, in pure Swift.

A collection of persistent, immutable tree data structures implemented in Swift, including binary trees, AVL trees, and Red-Black trees. Designed as functional persistent data structures, they ensure thread-safety and support maintaining historical versions of data.

avl-tree binary-tree persistent-data-structure red-black-tree swift value-types

KALMANFILTER

SWIFT

AUTHOR

ARCHIVED PROJECT

An efficient implementation of a Kalman Filter in Swift, using SIMD.

I became interested in signal filtering and control theory around 2018, specifically Simultaneous Localization and Calibration (SLAC). Recognizing a lack of existing Swift implementations of Kalman Filters I decided to implement my own, based on my other project jounce/surge.

algorithms digital-signal-processing kalman-filter simd swift

PARTICLEFILTER

SWIFT

AUTHOR

ARCHIVED PROJECT

An efficient implementation of a Particle Filter in Swift, using SIMD.

I became interested in signal filtering and control theory around 2018, specifically Simultaneous Localization and Calibration (SLAC). Recognizing a lack of existing Swift implementations of Particle Filters I decided to implement my own, based on my other project jounce/surge.

algorithms digital-signal-processing kalman-filter simd swift

STRATEGIST

SWIFT

AUTHOR

ARCHIVED PROJECT

Algorithms for building strong, immutable AIs for round-based games.

Strategist provides implementations of several well-known AI algorithms for round-based games, such as Minimax and Negamax Tree Search (both with alpha-beta pruning), as well as Monte Carlo Tree Search.

algorithms game-ai simd state-space-search swift

SWIFT CONTRIBUTIONS

The following projects represent a curated selection from a broader portfolio:

SURGE

SWIFT

CO-MAINTAINER

INACTIVE PROJECT

A highly popular (5300+ GitHub stars) Swift library that uses Apple's Accelerate framework to provide high-performance functions for energy-efficient computation on the CPU by leveraging hardware acceleration via SIMD.

I became co-maintainer around 2019 and quickly emerged as the 1 contributor of the project, spearheading efforts to modernize, clean up, and extend the library, as well as significantly improving its test-coverage.

matrix-multiplication parallelism simd swift

SNAPSHOTTESTING

SWIFT

CONTRIBUTOR

INACTIVE PROJECT

A powerful library for snapshot testing of UI components, data structures, and more by comparing current output against stored reference snapshots. It is especially useful in iOS and macOS development to catch unintended changes in views, JSON, or other serializable outputs, ensuring UI consistency and data correctness over time.

I've made several contributions to the project, focussing on adding support for SwiftUI, improving user ergonomics, API robustness, and improving overall code quality.

snapshot-testing swift swiftui

TYPESCRIPT CONTRIBUTIONS

The following projects represent a curated selection from a broader portfolio:

LAYERCHART

TYPESCRIPT

CONTRIBUTOR

ACTIVE PROJECT

Composable Svelte chart components to build a large variety of visualizations.

I have repeatedly contributed to the project, focussing on bug fixes, usability improvements and most significantly a complete refactor of the force-simulation implementation, resulting in significant performance improvements, bug fixes and improved reliability.

d3js force-simulation frontend svelte typescript visualization