

Mathieu Soysal

Rust Software Developer

Paris, France · +33 6 52 60 01 10

contact@mathieusoysal.dev

github.com/MathieuSoysal · mathieusoysal.dev

linkedin.com/in/mathieusoysal

2,300+

downloads of my Rust CLI on
crates.io

~45k

lines of Rust across 10+
projects

Top 1%

of 320,000 on CodinGame, in
Rust

600+

repositories use my Javadoc
Publisher GitHub Action

PROFILE

Software developer specialized in Rust, with a test-first, tooling-heavy engineering culture: benchmarks in CI, mutation testing, fuzzing, security audits. Author of CG Bundler, a Rust CLI published on crates.io, and of GitHub Actions adopted by third-party projects. Systems-programming foundation from 42 (C, Unix processes and signals, x86-64 assembly) on top of a computer-science degree — at home in performance-critical code that must stay correct and fast under a hard time budget.

SELECTED RUST PROJECTS

CG Bundler — CLI published on crates.io

2025 – present · ~14k LOC

- AST-level bundler that flattens a multi-file Rust crate into one optimized source file for competitive programming — built on **syn / quote / prettyplease** and **cargo_metadata**, with a file-watch mode (notify) and a clap CLI.
- Shipped as a real product: v1.1.18, 2,300+ downloads, integration-test suite, CI with automated releases, **cargo-deny** security audits and a SonarCloud quality gate.

Taskmaster — Unix process supervisor (42 project)

~5k LOC

- supervisor-style job-control daemon: process lifecycle modeled as a state machine over **fork/exec/wait** and **POSIX signals** (libc), declarative YAML configuration, interactive control shell, logging subsystem, optional ratatui TUI.
- Robustness-focused test suite (config, FSM, logger, failure injection); coverage tracked with Codecov in CI.

Mars Lander — genetic algorithm, Rust → WebAssembly

- Trajectory optimizer for CodinGame's Mars Lander: **rayon**-parallel physics simulation, criterion benchmarks, compiled to **WASM (wasm-bindgen)** behind an interactive browser visualizer deployed on Cloudflare Pages.

Ultimate Tic-Tac-Toe engine — cargo workspace

~6k LOC

- Bitboard game engine with precomputed perfect-play tables; separate crates for self-play training and an MCTS visualizer; criterion benchmarks.

CodinGame contest bots — performance-critical Rust

2024 – 2026 · ~15k LOC

- Contest AIs engineered like production code: **continuous benchmarking in CI (Bencher)**, mutation testing (cargo-mutants), fuzz tests, Codecov, security audits — all under the contests' hard per-turn time limits.

OPEN SOURCE & OTHER ENGINEERING

- **Maintainer of reusable GitHub Actions since 2021**, used by repositories beyond my own: Javadoc Publisher (used by 600+ repositories; 57★, 19 forks, external contributors), Jib Container Publish, Replace String In File.
- **Production web engineering on Cloudflare Workers/Pages**: prerendered static sites with an edge API on KV storage, strict CSP, and 20+ automated build-time quality gates (JavaScript); Rust build tooling on a second product (Markdown SSG with pulldown-cmark, SVG rendering with resvg, WASM simulation engine).
- **Java ecosystem**: Maven/Gradle builds, Jib container images, SonarCloud quality gates.

EDUCATION

42 Paris — Common Core

2023 – present

- Peer-reviewed systems programming in C: POSIX shell (minishell), threads & synchronization (philosophers), x86-64 assembly (libasm), raw-socket ICMP (ft_ping), Docker infrastructure (Inception).

IUT — **Université de Montpellier** — DUT Informatique

2018 – 2022

SKILLS

Rust: editions 2021/2024, cargo workspaces, AST tooling (syn/quote), wasm-bindgen, rayon, serde, clap, thiserror/anyhow, criterion, cargo-deny, cargo-mutants, fuzzing
Systems: C (POSIX), processes/signals/pipes, sockets, x86-64 assembly basics, Docker, Linux

CI/CD & quality: GitHub Actions (author), SonarCloud, Codecov, Bencher, automated releases, TDD

Also: Java (Maven/Gradle, Jib), JavaScript/TypeScript, Cloudflare Workers, Git

Languages: French (native), English (B1)