Martin Larsson

Stockholm, Sweden ▪ me@erikmartinlarsson.dev ▪ +46 70 551 52 29 ▪ LinkedIn ▪ GitHub

---

Summary

Software engineer with 7+ years of professional experience building games and systems across multiple platforms. Comfortable across the stack, from low-level systems to gameplay and DevOps pipelines. Quick to pick up new tech and focused on delivering practical solutions.

Open to short- and long-term consulting work across web, games, and systems. Comfortable working independently or within a team.

Core Strengths

- Shipped multiple games on multiple platforms and worked on a live service reaching millions of players.
- Contributing across all levels: systems, game engines, end-to-end integrations, and pre-production.
- Learns fast. Gets productive in new stacks within weeks. Good track record of self-teaching and taking ownership.
- Comfortable navigating unfamiliar codebases, often tracing behavior across files when docs are lacking.
- Focuses on simplicity and maintainability. Chooses the right tool for the job.
- Systems-level experience in multithreading, replication, data-oriented design, and embedded firmware.
- Solid Linux knowledge, using reproducible, declarative setups with Nix.
- Terminal-based workflow optimized for speed and control (Neovim, Tmux, task runners, DAPs, automation).
- I like making the computer do the work, not the other way around, and with as few keystrokes as possible.

Technology Experience

Experienced with a wide range of tools, languages, and platforms, including:

Languages: C++, Rust, Zig, C, C#, Go, Lua, Python, Bash, Nix

Engines/Tech: Unreal Engine 5, Unity, Bevy, PlayFab, Vivox, Havok, GTest, FlatBuffers

Platforms: Linux (NixOS/Arch/Fedora/Ubuntu), macOS, Windows, Meta Quest 1-3, PS5/PSVR2

Tooling: Jenkins, TeamCity, UGS, GitHub Actions, LLDB, Taskflow, Perforce

Not limited to the above. Quick to learn new stacks and environments as needed.

Professional Experience

Software Engineer — *Monetisation Area, Farm Heroes Saga at King* — Live service reaching millions of players

Stockholm, Sweden — *August 2024 – September 2025*

- Built internal Bash developer tools to automate development and cleanup.
- Contributed and owned successfully launched game features.
- Led pre-production efforts: risk analysis, task breakdowns, time estimations, and coordinating timelines.
- Participated in "Documentation Heroes", improving clarity and developer onboarding.
- Worked in a data-driven workflow: AB-tested features, tracked impact, and iterated based on performance.
- Development focused on over-the-air delivery and backwards compatibility.

Intermediate Programmer — *Fast Travel Games* — Shipped on Meta Quest and SteamVR

Stockholm, Sweden — *April 2023 – July 2024*

Project: Mannequin (UE5, Multiplayer - Networked replication)

- Implemented multiplayer sessions via PlayFab: server browsing, custom games, and persistent replicated settings.
- Developed X-Ray detection systems used for visibility through walls.

- Built editor tools to help designers place and replicate groups of objects: doors, elevators, and buttons.
- Built internal CLI tooling in Rust to automate local multiplayer testing and simulate PlayFab sessions.
- Set up Vivox with multiple channels (team, free-for-all, proximity) while supporting team swaps and late joiners.

Junior Gameplay Programmer — *Fast Travel Games* — Shipped on Meta Quest, Pico, and PSVR2 as a Launch Title

Stockholm, Sweden — *May 2019 – April 2023*

Project: Cities: VR (Unity, data-oriented, simulation-heavy code)

- Prototyped and finalized VR controls used to interact with core gameplay systems.
- Extended and maintained a complex legacy codebase from Colossal Order and Tantalus.
- Developed PSVR2 features including trophy support and advanced controller haptics.
- Made UX and gameplay decisions in the absence of dedicated design support.
- Maintained TeamCity-based build pipelines.
- Onboarded new developers and contractors into the project.

Other Projects:

- **Rapid Prototyping Team:** Explored and iterated on experimental game concepts using Unreal and Meta Horizon Worlds.
- **The Curious Tale of the Stolen Pets:** Built level scripting logic, an achievement system, and core UI, along with a save/load system (Unity + FlowCanvas).
- **Apex Construct:** Contributed to live patches and bug fixes during internship phase.

Projects & Technical Highlights

Atlas Game Engine (WIP)

- Modular C++ game engine designed for performance, clarity, and cross-platform development, inspired by Unix principles.
- Built a high-performance ECS module with chunked, cache-friendly memory layout (SoA), multithreaded system scheduling, and zero-cost compile-time type resolution.
- Achieves ~70K avgFPS (excluding pre-ticking entity creation) simulating 20M entities with simple but real-world system logic that stresses and invalidates cache.
  - Performance remains stable up to memory limits (~120M entities) with no architectural bottlenecks with non cache invalidating work.
  - M1 MacBook Pro and M3 MacBook Pro MAX used for testing these numbers
- Able to compete against performances of industry ECS implementations like Unity DOTS, EnTT, and Flecs with many improvements still to come.
- Designed for developer ergonomics with clean APIs, thread-safe batching, and predictable system behavior.

Mechanical Keyboard Build

- Hand-wired custom mechanical keyboard with optimized layout for programming.
- Wrote custom firmware in C using QMK with features like home row mods, mod tap, and leader key shortcuts.

Personal Linux Stack

- Daily driving NixOS (desktop) and Asahi NixOS (M1 MacBook).
- Declarative system and home management using flakes.
- Comfortable in minimal, keyboard-driven environments. Daily use of Sway, Vimium, Zsh, and other shell tools.

Education

Vocational College Degree in Game Development
PlaygroundSquad, Falun, Sweden — 2019