

Samuel Gélineau

✉ gelisam@gmail.com

☎ +1 514-509-5173

📍 Montreal, Canada

👤 github.com/gelisam

📺 [haskellcat.com](https://www.youtube.com/channel/UC8v31111111111111111111)

✍ gelisam.com

🐦 [@haskell_cat](https://twitter.com/haskell_cat)

Senior Software Engineer, Architect, and Evangelist

WHO AM I?

I am a software engineer who specializes in the Functional Programming paradigm and more specifically in Haskell, with which I have 16 years of experience by now. The most well-known packages I maintain are `hint` and `recursion-schemes`. I have contributed to `agda`, `criterion`, `ffmpeg-light`, `fltkhs`, `ghc`, `gloss`, `hackett`, `policeman`, `polysemy`, `servant`, `tasty`, and many more.

I have a broad background, covering academic research, distributed systems, embedded systems, entrepreneurship, graphics programming, mobile development, and web development. Throughout those adventures I have been paid to write Bash, C, C++, C#, Haskell, Java, JavaScript, Python, Ruby, Scala, and TypeScript. I also have experience with Agda, Elm, Racket, and Rust.

I enjoy spreading the gospel by giving presentations and tutoring sessions on the advantages and disadvantages of various FP idioms, for example combinator libraries, difference lists, effect systems, lenses, and `MonadBasedControl` vs `UnliftIO`.

I strive to be helpful to others, by writing researched, detailed answers with plenty of code examples. I like to codify existing practices, to remove ambiguity and to give us the opportunity to self-reflect and improve. I reach out to others, building bridges, collaborating and resolving misunderstandings.

WORK EXPERIENCE

Senior Software Engineer, Architect, Lead Architect, 2017 – 2022

SimSpace

Helped the backend codebase grow to half a million lines of Haskell code. Main author of several components including an error-reporting library, an incremental-update library, a progress notification system, a step-function library, and a yaml templating language. Wrote architectural documents guiding the development of several other components, including a latency-reducing agent, a network-creation wizard, a resource-division system, and a template-sharing system.

Led weekly architectural discussions, made sure everybody's opinion was heard. Introduced several collaboration-based policies including a process for choosing between competing technologies, a process for crowd-sourcing code improvements, a process for finding a shared vision for the future of our codebase, and a process for spreading technical knowledge. Created new roles to solidify our codebase's performance, security, and testability. Democratized our tech debt prioritization process while making sure that important improvements were not neglected.

Course Author, 2016 – 2017

Packt

Recorded an online course about advanced Haskell topics including `unsafePerformIO`, lazy IO, streaming, free monads, Functional Reactive Programming, parallel programming, concurrent programming, and distributed programming.

Programmer, 2015 – 2016

Keatext

Worked on the Scala backend (AWS micro-services) and the React frontend of a Natural Language Processing web application.

Consultant, Tech Lead, 2011 – 2015

Spiria

Worked for various clients including Autodesk, Citi Bike, Shaw, and the NRCC, controlled real-time hardware systems such as haptic arms and frame grabbers, refactored codebases with millions of lines of code, benchmarked desktop and embedded software, ported software to Linux, and implemented Android applications.

Technical Co-Founder, 2011 – 2013, part time

Aero Web

Managed remote employees, connected heterogeneous technologies. Despite a small profit, we closed for lack of passion.

Programmer, 2006 – 2010, part time

Environment Canada

Migrated large, legacy C systems from HP-UX to Linux. Maintained and supported high availability systems used 24/7 by airport forecasters across the country.

Course Instructor, 2007 – 2008, part time

McGill School of Computer Science

Taught students how to code more quickly without sacrificing correctness.

Research Assistant, 2005 and 2010, part time

McGill Software Engineering Lab

Researched transactions, aspect-oriented programming, and multitouch widgets. Published at the AOSD conference.

Programmer, 2002 – 2003, part time

Art Team Images

Wrote online content editors and cross-browser JavaScript widgets.

SELECTED PUBLICATIONS

Möbius: Metaprogramming using Contextual Types, by Jang, Gélneau, Monnier, and Pientka

POPL2022

A programming language guaranteeing that generated code is well-typed.

Klister: Predictable Macros for Hindley-Milner, by Barrett, Christiansen, and Gélneau

TyDe2020

A programming language with type-driven macros.

PRAISE

Edward Kmett, famous Haskell guru

"Hire this guy."

Alex Speaks, Staff System Engineer

SimSpace

"Sam is not only a next-level developer but he is also an incredibly kind and compassionate human. I would work with him again in a second. If you are looking for a skilled functional-programming specialist Sam is the best candidate you will find."

Matthew Russell, Deputy Director of Engineering

SimSpace

"Samuel Gélneau is amazing! One of the best engineers I've ever worked with and an incredible hire for any lucky team."

Michael Beynon, Director of Engineering

SimSpace

"I cannot agree more with the positive comments about Sam. He is a standout engineer, super passionate, loves to teach others, and would fit in on any team."

Employees, who were hired after I left

Keatext

Said they use the code I wrote as a reference for what good FP code looks like. They also complimented the quality of my documentation.

Michel Larivière, my boss

Environment Canada

Admitted that because I seemed over-confident, he gave me their hardest project and expected me to get stuck for the whole summer. He was then pleasantly surprised to see me succeed and then tackle their second-hardest project as well.

EDUCATION

M.Sc. in Computer Science, 2010

McGill University

B.Sc., major in Computer Science, 2006

McGill University

DEC in Natural Sciences, 2003

Collège Édouard-Montpetit