GREGOIRE GEIS

I started programming in middle school, where my skills allowed me to create a website that could help my classmates study for the upcoming exams.
Since then, I have started over 50 programming projects, most of them public, including multiple compilers and assemblers.
I have also contributed to more than a dozen open source projects in several different programming languages.

CONTACT

- hello@gregoirege.is
- gregoirege.is
- github.com/71
- keybase.io/71
- linkedin.com/in/gregoiregeis
- stackoverlow.com/story/gregoiregeis

PROFESSIONAL EXPERIENCE

LRDE *Summer 2018* *C++ 14*

Created a pattern-based LTL formula rewriting engine for Spot. 5,000 lines added.

NOTEWORTHY PROJECTS

lesspass.kt *2018* *Kotlin, Android Studio*

Unofficial LessPass client for Android, using Material Design. Around 1,700 LOC.

Ryder *2017 - 2018* *C#, CIL*

C# library that modifies code generated by the .NET runtime in order to redirect methods freely. Almost 4,000 LOC and **known to be used internally by third-parties**.

Cometary *2017 - 2018* *C#*

C# analyzer that uses the aforementioned Ryder library to hook into the inner workings of the compiler to modify its behavior. Over 20,000 LOC.

ASM. *2018* *Python*

Python program that generates sources for an assembler in several programming languages. Supports ARM, Mips and x86 in C, C++, C#, JavaScript, Nim, Python and Rust. Over 60,000 LOC (most of them machine-generated).

RECENT CONTRIBUTIONS

- CraneStation/cranelift (> 1,000 stars): Added ability to report multiple errors, instead of stopping at the first one. Over 1,000 lines changed.
- fsprojects/FSharp.TypeProviders.SDK (> 180 stars): Added support for the compilation of the `try with finally` expression. 240 lines added.
- TheDan64/inkwell (> 120 stars): Implemented the Kaleidoscope language from the LLVM tutorial as an official example, and added small fixes to the library. Over 1,400 lines added.

PROGRAMMING LANGUAGES

Language	Experience	Description
C#	5 years	Very comfortable, including with the compilers and .NET runtime.
F#	1 year	
JavaScript	6 years	Very comfortable with the language and its ecosystem / standard library. From 100 to 500 hours of work with each language.
Python	4 years	
Nim	3 years	
Rust	2 years	
C, C++	4 years	Comfortable with the language, and most of its standard library.
Nim	3 years	
Kotlin	2 years	
Java	3 years	Comfortable with the language, and some of its standard library.
Go	2 years	
Haskell	1 year	
OCaml	1 year	

EDUCATION

EPITA *2017 - 2022 (Estimated)*

Pursuing a computer engineering degree in the international section.

Lycée George de la Tour *2014 - 2017*

Baccalauréat S with *Bien* honors, and a perfect score in mathematics.

OTHER SKILLS

- Android, Linux (Arch, NixOS), Windows
- Git, Vim, VS Code, Android Studio
- Office, Google Suite, Slack
- Markdown, LaTeX