

Maurice Bos

m-ou.se@m-ou.se

Jobs

- | | |
|------------|---|
| 2007–today | <p>BosByte — Owner</p> <p>Since 2007, I'm running a small one-man business. My activities include software development for a wide range of platforms (including web, desktop, embedded systems), but also graphics design, electronics (hardware), robotics, and publishing a children's book.</p> |
| 2014–2015 | <p>Google</p> <p>Software engineer and site reliability engineer at Google Switzerland.</p> |
| 2013–2014 | <p>Aanmelder.nl</p> <p>Part time Python developer.</p> |
| 2011–2013 | <p>DUT Racing Team</p> <p>Software engineer at the Formula Student team of the TU Delft.</p> <p>The DUT Racing Team is a team of about 60 students working in their free time to design, build and test a fully electric formula-style race car in exactly one year. Each summer, we compete with hundreds of teams from other universities all over the world. In this competition, we've taken the first place in 2010, 2011, 2012 and 2013. Also, with the car built in 2012, we got an official Guinness World Record for the fastest accelerating electric car (0–100 km/h in 2.134 seconds).</p> <p>I worked in the team for several years, but was especially active in 2012. In the team of 2012, I have spent months working close to a hundred hours a week on the project. I was responsible for the design and all implementation of the software running on the ARM Cortex M microcontrollers controlling the entire four wheel driven electric 100kW race car.</p> <p>All software was written in C++11.</p> |
| 2010–2013 | <p>Student Assistant</p> <p>Assisting at several programming related lab courses at the Delft University of Technology.</p> <p>This includes designing/writing the assignments, checking handed in assignments, and most importantly, explaining the subject to students individually.</p> |

2012 **Cellolo**
Lead Embedded Software

In 2012, I was responsible for the software in an international project of Cellolo, a company based in Israel developing hardware for spectacular light shows. For several months, we worked in a team of four engineers on an embedded systems project for the light show of a big concert. Unfortunately, even though the prototypes worked perfectly, the project got cancelled due to budgetary problems.

All software was written in C++11.

Projects

2015 **Whiteboard Clock**
A small 3d-printed wirelessly controlled robot that regularly erases and (re)writes the current time on a whiteboard using a whiteboard marker. A video of it went viral in February 2015: Just search for 'whiteboard clock' on the internet.

2013–2014 **C++ preprocessor**
A fully standards-compliant unicode-aware C++11 preprocessor, written from scratch using only C++11 without any non-standard libraries. It tokenizes and preprocesses source code, and can produce detailed error messages.

2013–2014 **Moggle**
Moggle is a C++11 library to make it easy to use OpenGL. It provides simple wrappers for objects like OpenGL buffers and shaders, but also has more advanced features such as the generation of source code for shaders.

In 2013, I started with a D version of Moggle, as this language seems to fit better with the goals of Moggle.

Moggle is open source and hosted on GitHub. The C++ version is currently used in a few small games.

2013 **PICP** — A PIC16F145x based USB PIC programmer.
Software for a PIC16F145x microcontroller to turn it into a USB programmer for PIC microcontrollers.

Both the software for the PIC and the command line tool to use it are open source (GPL3) and hosted on GitHub.

2012 **Delftgit.nl**
A hosting service for Git repositories for students of the TU Delft. (Like a simplified GitHub.) Currently, it has 7 users.

- 2012 **BliepGeit**
 A computer running Linux and software written in C++ to let the members of the DUT Racing Team pay for drinks and food they take from the team refrigerator by scanning the items with a barcode scanner.
 The software is implemented in an overly complicated way, because I used the project to learn more about multithreading and drawing 2D graphics. It used to gain a new feature about every month.
- 2011 **TMT**
 A command line tool to simulate and debug single tape deterministic Turing machines, written in C++03. It is open source and hosted on GitHub.
- 2010 **x86502**
 A 6502 processor emulator, completely written in x86 assembly, which runs without any operating system. (It is bootable.) It comes with a game, written in 6502 assembly, to show off the capabilities of the emulator.
- 2010 **LoopMixer**
 A Java library that allows the user to easily mix audio loops together. It takes care of slowing down and speeding up the samples to match a common rhythm, which can be changed at runtime.
 In 2010 and 2011, a lab course on the TU Delft required the students to use this library for its assignments. LoopMixer is open source and hosted on Google Code.

For more projects, see my GitHub page: <https://github.com/m-ou-se>

Other activities

- 2014 **Linux/UNIX System Programming Training**
 A five day long course on the low-level system interface of Linux and UNIX, by Michael Kerrisk, the author of *The Linux Programming Interface*.
- 2010–2012 **ACM ICPC**
 For several years, I competed in the ACM International Collegiate Programming Contest. In 2010, I won the preliminary contest at the TU Delft. Later, in 2012, I was part of the team that took the first place in the North West European Regional Contest. We got the 36th place in the world finals.
- 2011 **Raytheon BBN Technologies**
 In 2011, I worked with 'Proto', a language for Spatial Computers developed at MIT and maintained at BBN Technologies. I wrote my own implementation of the virtual machine for the language, and visited BBN Technologies in Boston for a month to replace their virtual machine with mine.

Skills

- Since 2008 **C++ (Programming Language)**
Most of the code I wrote is in C++, which is the programming language I know most about.
- Apart from writing all kind of programs in C++, I like writing libraries. Examples include a compile-time unit-checking library and an implementation for a proposal for `std::variant`.
- I spend a lot of my free time on an IRC channel about C++ (Freenode's `##C++`), discussing the language and helping others. Also, I try to contribute to the (future) standardization of C++. I have read most of the ISO standard, discuss proposals for upcoming revisions of the standard on mailing lists, have written proposals, and visited the ISO C++ committee meetings to present and discuss proposals. I always try to use the newest features of the language (C++14, or C++17 where possible), and try to get others to use it as well. I have written a few articles on C++, which are available on <http://blog.m-ou.se/>.
- Since 2015 **Haskell (Programming Language)**
Most of the code I have been writing recently is written in Haskell.
- Since 2013 **D (Programming Language)**
If I had more time, I'd be writing a lot of code in D, too.
- Since 2012 **Other Programming Languages**
On a regular basis, I write small things in Javascript, Python and shell scripting language (Zsh/Bash). For fun, I sometimes write something in assembly (x86, 6502 and ARM). I have experience with many other programming languages which I rarely use, and aren't really noteworthy.
- Since 2011 **Electronics**
I know basic electronics, how to work with tools like logic analyzers and oscilloscopes, I've designed simple printed circuit boards, created small robots, and know how to operate a soldering iron.
- Since 2010 **Linux**
I only use Linux on my desktop, laptops, and do almost everything in the terminal.
- Also, since 2011, I have several Linux servers running for hosting websites, files, e-mail, DNS, and many other services for both others and myself.

Education

- | | |
|-----------|---|
| 2009–2013 | Computer Science and Engineering
Bachelor at the Delft University of Technology.
Graduated with an average grade of 8.14 (on a scale of 10). |
| 2009–2013 | Industrial and Applied Mathematics
Next to my computer science bachelor, I finished all courses of the mathematics bachelor at the TU Delft. |
| 2011–2012 | Honours Programme
A research project worth 15 ECTS next to the regular courses. |