

Justin McGirr

Summary

I am a passionate software developer with a wide breadth of experience and an even wider breadth of interest. In addition, I have in-depth knowledge of embedded systems, web technology, and software engineering principles.

I am a demonstrated avid self-learner, and would love to learn about your domain, especially if it is one which I am currently unfamiliar with.

Education

2011-2016 **Bachelor of Computer Science, Honours Computer Science / Digital Hardware Option**, *University of Waterloo*, Waterloo, Ontario.

Technologies

Linux	8 Years	<i>LinkedIn, Pebble, Kik, Fitbit, Sweet Tooth, wfd, hobbyist use since 2009</i>
JavaScript, HTML, CSS	8 Years	<i>Kik, Pebble, Fitbit, buildblast, scrawl, Sweet Tooth, freelancing, personal websites, high school's website, and more</i>
C	7 Years	<i>Fitbit & Pebble Firmware, RTOS, video decoder, freelancing, LiteStep Portable</i>
Python	6 Years	<i>Pebble, LinkedIn, and Fitbit build system & internal tools, freelancing, jupyter notebooks, wfd, high school's website, opentower</i>
C++	5 Years	<i>Fitbit & Pebble unit tests, LinkedIn, highrisedev & opentower</i>
Google Go	3 Years	<i>buildblast server, wfd, JSC bindings, small libraries</i>
SQL	3 Years	<i>Pebble, Sweet Tooth, Dexterity Consulting, highrisedev</i>
	1 < n < 3 Years	<i>L^AT_EX, PHP, (TI-)BASIC, Java, ruby, mongodb, C#, VHDL, Verilog</i>
	Curious	<i>Rust, Haskell, Clojure, Erlang/OTP</i>

Employment

January 2020 - **Software Development Consultant, Freelance.**

June 2020 Spent a week in Calgary helping a smart team with that had limited CS knowledge systematically improve the reliability of their React SPA.

Spent 3 months in Waterloo creating factory test fixtures for Monogram Creative Console - physical buttons, sliders, and dials for automating repetitive computer tasks. Created an end-to-end solution to validate the i2c and electrical connections between modules. Wrote about 90% of the firmware, including some driver code, and simple Python to run on an accompanying computer for coordination and firmware updates.

November 2017 - **Firmware Engineer**, *Fitbit*, Waterloo, ON.

February 2019 Worked on the Fitbit Developer Platform, allowing developers to create JavaScript applications to run on their Fitbit.

Read, debugged, and occasionally wrote C firmware code to enable several features, taking the lead on designing and developing our scientific APIs (similar to numpy) and fleshing out the approach for multi-view support.

Was a regular contributor to discussions about our new developer APIs across firmware and mobile, with my contributions being valued enough to make me a gating reviewer, along with the tech leads.

Played a significant role in the successful creation and maintenance of our team's integration tests. Added unit tests for the integration tests, used across Fitbit.

Reduced average JavaScript execution time by 3x, and JSON.stringify from $O(n^2)$ to $O(n)$.

Laid off when the Waterloo office closed, and went off to explore the world, hiking some 900km across Spain, and busing between European cities. Also volunteered locally for a non-profit bike repair shop, and canvassed for the local Green Party!

Fall 2015 **Software Engineering Intern**, *LinkedIn*, Mountain View, CA.

Worked on LinkedIn's next generation distributed graph database, a ground-up rewrite in C++ from the original Java, with a transition from the original proprietary and restrictive graph query language to a more general datalog variant. Aided in designing shared-memory structures for atomicity across processes and consistency against process death.

Took the initiative in improving tools and processes, reducing test run time from several minutes to mere seconds with concurrent execution.

Made incremental builds fast and simple, requiring only a single command, 7 characters in length. Based on a back-of-the-napkin calculation, these changes improved team productivity by at least 5%.

Spring 2015 **Software Developer and Analyst Intern**, *Microsoft Research spinoff - Intentional Software*, Seattle, WA.

Worked under Dr. Charles Simonyi, original architect of Microsoft Word and Excel, on a experimental meta-programming platform.

Added text rendering features in one of the most complicated parts of the codebase, which was not understood and was widely feared by the full-time employees. While adding features, I significantly reduced complexity, removing more lines than I added.

Took the initiative to substantially improve the UI for accepting changed integration tests results. Enabled the use of keyboard shortcuts and for multiple tests to be accepted at once.

Reincarnated the domain tree editor, forming the basis for tools that will allow developers to edit the domain tree while the application is running, and see the changes dynamically reprojected.

- Fall 2013 & Summer 2014 **Web / Firmware Co-op**, *Pebble Technology Inc.*, Palo Alto, CA.
- Mentored a new hire in the rewrite of our filesystem, and designed & wrote many core storage layers that are now extensively used, practically eliminating storage-related bugs, and eliminating storage-caused UI pauses, which could previously be up to several seconds.
- Completely rewrote the on-device analytics subsystem - which told us what features users were using - to be more robust, maintainable, reliable, and actually work properly.
- Worked on the Pebble OS syscall layer & sandbox, improving the stability and security of the Pebble operating system.
- Wrote ruby scripts to extract and normalize sales tax data from 4 different databases with quickly-written schemas that were all broken.
- Made an assortment of internal web tools to help in battery lifetime analysis, reducing the number of users getting less than a day of battery life by 10x, integrating analytics data from my on-device work to pinpoint features which caused unexpected battery drain in the field.
- Worked on the initial version of a fixed-point 3D graphics library, PebbleGL.
- Spring 2013 **Web Developer Co-op**, *Kik Interactive*, Waterloo, ON.
- Built a fully GPU-accelerated photo viewer, complete with pinch to zoom, panning, and swiping between photos - in HTML. Successfully passed QA, after working around numerous bugs in old (and new!) Android WebKit implementations.
- Created a new theme for app.js (<http://github.com/kikinteractive/app>), which served as the base theme for apps which have been used by tens of millions.
- Summer 2012 **Software Engineering Legend (Co-op)**, *Sweet Tooth Rewards Inc*, Waterloo, ON.
- Created an analytics system that delivered valuable insights into current and potential future revenue, allowing Sweet Tooth Rewards to optimize their pricing model.
- Built "releases pieces" - a deployment tool allowing precise and simple deployment & versioning of related projects together.
- August 2011 **Technology Consultant**, *Dexterity Consulting*, Calgary, AB.
- Designed, tested, and deployed a data mining application, using C# and SQL.

Projects

Projects are also on my website at jmcgirr.com, with pictures, live demos, and source code!

2014 **Creator and Lead Programmer**, *BuildBlast*.

While working at Kik, and continuing through my time at Pebble, I found a renewed fascination with graphics cards, hardware acceleration, fueled by the ability to finally access all this through the web browser. I stumbled upon THREE.js, took some demo code, and started hacking. Today, the code implements a fully multiplayer 3d voxel-based editable world, where you can interact not only with the world, but also other players. It has pvp attacking, with full server verification and lag compensation. You can try it at <https://play.buildblast.com/>

- 2019 **Creator**, *Scrawl (Online Pictionary)*.
Wanted to play Pictionary with some friends online, and grew frustrated with the existing options. Hacked together the initial version in a few hours, and continued to develop it here and there over the years. Both frontend and backend in hackathon style ES6 JavaScript. Explicitly used no libraries on the frontend as a personal challenge. <http://scrawl.jmcgirr.com>
- 2015 **Co-developer**, *Trains (CS452/Realtime Project Course)*.
Given nothing but datasheets, hardware to test on, and an intentionally buggy implementation of `printf()`, my partner and I implemented and thoroughly unit tested a complete microkernel in C and ARM assembly.

Modified our kernel slightly, setup QEMU, and created a virtual train track program in Python, allowing us to develop, test, and perfect changes much faster than other groups.

Built real-time userspace software on our microkernel to control the model train track, running multiple trains at once (*usually* without crashing them into one another, despite somewhat unreliable sensors and switches).

Built a computer vision pipeline that tracked the train's positions, leading us to discover that the acceleration curves given to us by the professor were completely wrong, a revelation which yielded only chuckles from him!
- 2016 **Co-developer**, *Hardware Video Decoder (ECE423 Project Course)*.
Took a basic mjpeg video decoder written in C, ported it to the Nios II softcore (CPU written in VHDL for an FPGA) architecture, and aggressively optimized the code. Created custom instructions in VHDL of our own design for DCT and Huffman decoding. Improved speed from 3fps to 30fps, limited by the speed of the vendor-provided VGA interface. Our implementation was the fastest the professor had ever seen, exceeding his own implementation!
- 2013 **Co-creator**, *Gitdefence*.
After receiving a job offer for Kik, I decided to brush up on my JavaScript skills by making a tower defense game which explored the potential of networks in spreading information and traits (distributed networking model inspired by git, hence the name). You can try it at <http://jmcgirr.com/gitdefence/master>.
- 2011 **Lead Programmer and Designer**, *WFDR Web Framework*.
Wrote the WFDR web framework to solve problems encountered while writing my high school's website, with a team of less-than-perfect volunteer developers. Fault-tolerant and language-agnostic! Github: <http://github.com/crazy2be/wfdr>
- 2010 **Web Developer**, *Highrise Developer*.
Wrote some C++ code, getting the project to compile on Ubuntu and setting up coding standards. After that, I taught myself php, MySQL, and JavaScript in order to write the project's website. <http://highrisedev.sourceforge.net/>
- 2010 **Developer**, *OpenTower*.
Wrote the majority of the code for OpenTower, learning C++, SFML, Object-oriented design principles, and graphical game programming along the way. <https://sourceforge.net/projects/opentower/>
- 2009 **Developer**, *LiteStep Portable*.
Wrote LiteStep portable largely independently, learning C, svn, HTML, JScript, VBScript, and WIN32API along the way. <https://code.google.com/archive/p/litestepportable/>