

Jason Sun's Resume

jason@sunapi386.ca sunapi386.ca linkedin.com/in/sunapi386 +1 (408) 599-0428 27/07/2018

Profile

- Very driven and motivated with many hobbies (visit my blogs at *blog.sunapi386.ca*). Especially interested computer science, which was discovered after trying two different majors: physics, and business.
- Over twelve years of experience using unix: *Ubuntu, Arch Linux, OS X*, and utilities like *nmap*.
- Experienced working in startup environments, at Velocity Residence and Velocity Garage. Enjoys technology conventions; ones I had attended were: *DEFCON (#22)*, *2600 Hope Number 9*, PennApps 2013, MHacks 2014, HackMIT 2014, and HackZurich 2014.

Education

- *University of Waterloo* Bachelor of Computer Science, 2015.
- Computer Science Exchange Student at *EPFL (Swiss Federal Institute of Technology at Lausanne)*.

Work Experience

- Autonomous Systems Integration Engineer *AutoX Technologies Inc.* in August 2017 – present San Jose, California

Full-stack autonomous system integration, working on both low-speed delivery vehicle and high-speed vehicles. Experience with using SLAM, LOAM, lidars, cameras, radars, ultrasonics. Designed and built vehicle-related product features such as, but not limited to: vehicle remote control (networked video streaming and throttle/steering control), hybrid human-and-AI decision making for solving difficult driving scenarios, vehicle to human interaction methods (currently under US patent application process), design and implement vehicle-to-cloud APIs (LTE networked), embedded systems with touchscreen for human-vehicle interactions. Non-vehicle related projects include automating the build-deploy-release process using dockerized containers, vehicle simulation tests, ROS like message passing system with Protobuf, and leading team efforts to build an API backend and iOS app for a location-based delivery service (LBS). I usually am taking on the general architect role, leading team efforts to design and integrate tech stacks for product feature development, using C++, Python, Javascript.

- Software Engineer *Apple Inc.* in November 2015 – January 2017 (15 months) Cupertino, California

Improved existing product reliability tools, conceived and built full-stack website (Rails API & Ember.js & nginx & MySQL) for managing iOS devices. Created client-sided task runners for delegating test-work, where work jobs were queued from the website. Also worked on internal iOS and macOS apps (Swift & Objective-C).

- Undergrad Research Assistant at University of Waterloo in May 2015 – September 2015 (5 months) Waterloo, Ontario

Worked with my AI professor and a 6 person team to develop a data tool for analysis of chat logs, and built a search engine prototype (using TF-IDF in Apache Lucene). Prototyped a search engine for customer support chat dialogues with Apache Lucene using TF-IDF indexing for feature recognition and searching.

- Software Engineer Intern *Shutterfly Inc.* in July 2014 – August 2014 (2 months) Redwood City, California

Design and implemented a distributed REST API service, in Java/Scala based upon the Apache Cassandra database. The API was designed by myself, with guidance from full-time employee members of the web infrastructure platform team, to be used by other Shutterfly services. Create additional network load tests for our distributed image hosting service.

- Software Developer at *Encircle Inc.* in May 2014 – June 2014 (2 months) Kitchener, Ontario

Worked on the website (CoffeeScript & Python Tornado server) and Android app, at a 3 engineer startup in the *Velocity Garage*, which is a University of Waterloo startup incubator.

- Software Tools Developer Intern at RIM (BlackBerry) in September 2013 – December 2013 (4 months) Ottawa, Ontario

Built internally used features to GitLab (Ruby on Rails) and helped with database migration from Github to Gitlab. Developed a testing framework for integration and regression testing website user interfaces using the Selenium Webdriver.

- Physics Teaching Assistant at *Wilfrid Laurier University* in September 2011 – April 2012 (8 months) Waterloo, Ontario

Developed a spectrometer reading program in Python, using the pySerial library, to automate the reading of lab samples, and generate a spreadsheet file. Supports multiple spectrometer readings in parallel.

Related Classes

- Real-time Operating Systems (CS 452): One project for the entire semester, which is to build a real-time microkernel and write user programs to control model trains on a track. The course is famously called the Trains course.
- Artificial Intelligence (CS 486): Worked on projects involving machine learning, learning probabilistic models, Bayesian networks, search and constraint satisfaction problems.
- Advanced Algorithms (EPFL CS 450): A graduate course in algorithms, learned theoretical techniques and their applications to solve problems. Interesting techniques such as network flow, randomization, dynamic programming.
- Computer Graphics (EPFL CS 440): Half the course was about graphics rendering techniques, and the second half about animation techniques. Implemented rendering methods into a ray-tracer (called *Nori*), and explored animation modeling and some fluid mechanics.
- AI: Intelligent Agent class (EPFL CS 430): Developed intelligent agents to pickup and deliver parcels in a simulated environment, with intelligent behaviours: reactive, deliberative, centralized, decentralized, and auctioning.
- Concurrency & Parallel Programming (CS 343): Multithreaded quicksort, implementation of well known concurrency control mechanisms, such as monitors. Language is in *uC++*, a concurrent dialect of C++, developed at University of Waterloo.
- Distributed Systems (CS 454): Built a remote procedure call library in C++ on top of TCP, both client and server side. Implemented the go-back-N reliable transmission protocol, over UDP using Java.
- Computer Security (CS 458): Created exploits in C using techniques such as buffer overflow, and format strings. Implemented an intrusion detection program which parses output from tcpdump to detect spoofed packets, malicious hosts, and worms.
- Compilers (CS 251): Built a MIPS compiler in C++, parsing a subset of C keywords and generating assembly (MIPS) code.

- Computer Architecture (CS 450): Designed a pipelined CPU in Verilog, supporting 8 instructions for computer architecture class. This is sufficient to run machine code produced by the MIPS compiler, from the CS 251 compilers class.

Awards and Achievements

- *HackZurich 2014* at ETH Zürich, received Tamedia Digital Award (all-inclusive team trip to visit startups in Berlin, Germany) with *.GIFMeIt*: An iOS app that lets a user easily capture and share GIF images.
- *PennApps 2013* at University of Pennsylvania, received Twilio's Communication Award (\$500) with *Marmoset*: A chatbot to respond to your chosen Facebook friends without them knowing.