

Ryan Conway

SF Bay Area • ryanmackenzieconway@gmail.com
<http://rmconway.com> • <https://www.linkedin.com/in/ryanmconway> • <https://github.com/PumpMagic>

WORK EXPERIENCE

Stack Labs – *Platform Software Engineer* – Cupertino, CA February 2017 – present

- Developing a spatial intelligence machine learning pipeline built in Java on top of Apache Flink, RabbitMQ, and Google Container Engine.
- Developing a RESTful web service written in Python and deployed to Google App Engine.

Knurld – *Software Engineer 8* – Redwood City, CA June 2016 – August 2016

- Developed the OpenAPI specification for a biometric authentication service.
- Designed and developed the iOS SDK for a web API in Swift.
- Knurld closed its doors in August of 2016.

Arrayent – *Senior Software Engineer* – Redwood City, CA June 2012 – June 2016

- Maintained an IoT load balancer, authentication service, and web API built in Java on top of Spring, Camel, and Tomcat and deployed over AWS.
- Migrated ~300K lines of code comprising a web backend from Java 6 to Java 8.
- Wrote a high-level AWS EC2 Spot Instance manager in Python to cut costs of server load tests.
- Instrumented backend services with health monitoring and metric reporting to simplify operations.
- Mentored software engineering interns.
- Designed and developed an operating system and networking stack abstraction layer in C to decrease the cost of supporting new embedded platforms.
- Refactored an application-transport layer protocol implementation to improve code clarity and modularity.
- Automated customer support and QA tasks by developing Python and Bash scripts.

SELECTED PROJECTS

Ostrich – *personal project* March 2016 – February 2017

- A Nintendo Game Boy CPU and audio unit emulator, with macOS UI, built in Swift. Open-sourced on Github.
- Media player interface written with AppKit enables playback and visualization of Game Boy music.
 - Fully implements Game Boy pulse audio channels and most opcodes of the CPU.
 - Leverages Swift's support for protocols and generic programming to enable quick opcode development.

Twitch Plays Paper Mario – *personal project* June 2015 – Oct 2015

A virtual game controller and chat client that enable the collaborative play of single-player video games online. Built in Rust, open-sourced on Github.

- Controller UI supports intuitive commands like “left” and “b” but also advanced command chaining and durations for nearly frame-perfect analog inputs, such as “25% left 50ms b up+z a 75% right (2s) b”.
- Automatic game state backups and moderation system mitigated trolling.
- UI generation tool enabled rapid support of new controller types.
- Attracted over 3,500 views, 400 followers and a tight-knit community on twitch.tv.

Autonomous Bullfighting Robot – *UCSC mechatronics course, 3-person team project* Winter 2011

An autonomous ping pong ball-shooting robot that competed against others in a public tournament. Built with a Motorola HC12, an array of sensor circuits, CAD software, and a laser cutter.

SKILLS AND KNOWLEDGE

Languages: *primary:* Java, C, Swift; *experience with:* Python, Rust

Tech: SOA / microservices, REST, TDD, Flink, Google Cloud Platform, gitflow

EDUCATION

University of California, Santa Cruz – BS Computer Engineering, minor EE, June 2012. Tau Beta Pi.