

Paul Olteanu

☎ 647-449-6597 | ✉ p.a.olteanu@gmail.com | 🌐 github.com/PaulOlteanu

Work Experience

#PAID INC

Toronto, ON

Software Developer

January - April 2018, May - August 2019

- Architected **OTP** application to maximize application stability and to parallelize data processing
- Created a persistent priority queue in **Mnesia**, for use with the **Honeydew** job runner
- Configured **AWS EC2** instances for staging and production use, and set up **Squid** proxy servers for request caching
- Processed **30,000+** users/day, generating insights on audience circles allowing for better campaign targeting
- Led project updating multi-step form for clients to create campaign proposals, increasing amount of data collected from RFPs
- Key Technologies: **Elixir, Ruby on Rails, AWS EC2 & IAM, Postgres, Mongo**

Nymi Inc

Toronto, ON

SDK Developer

September - December 2018

- Developed 3-part async SDK supporting multiple Nymi Enabled Applications and Nymi bands, usable in any language supporting a **C FFI**
- Refactored processing of incoming messages using **Serde**
- Optimized **Phoenix** server communications to minimize latency in **Citrix** environments
- Consumed **Windows APIs** to support applications running locally with the Nymi Runtime on Citrix servers
- Key Technologies: **Rust, Elixir + Phoenix, Windows APIs**

Projects

Personal Website

September 2019

- Project data loaded from .ron files, dynamically creating routes
- CI/CD done with the **Github Actions** beta
- Implemented in **Rust** using **Rocket**, and hosted using **DigitalOcean**

VSCode Syntax Theme

August 2019

- Custom port of the Railscasts syntax theme
- Per-language highlighting separated into multiple files for proper separation of concerns, and combined with a build script
- Published in the VSCode extension marketplace, with **10,000+** downloads

Ray Tracer

December 2018

- Multiple cameras able to be arbitrarily positioned for every scene
- Supports multi sample antialiasing (MSAA), and both Lambertian and specular reflecting surfaces
- Able to capture and export scenes as PNGs
- Implemented in **Rust**

CHIP-8 Emulator

October 2017

- Handles all 35 opcodes + legacy mode for opcodes that have traditionally been misimplemented
- Interprets binary .ROM files (CHIP-8 machine code)
- Implemented in **C** using **SDL2**

Version Control System

January 2017

- Supports commits with messages, reverting, and full logs
- Stores commits as diffs to reduce storage footprint, with complete backups every n commits to increase revert speed
- Implemented in **Python** using **Click**

Proplist

October 2015

- Contributed to library adding binary (string) key support to keyword lists by re-implementing the Keyword API
- **Published** on Hex (Elixir's package manager), with **~1,000 installs**
- Implemented in **Elixir**

Education

University of Waterloo

Expected Completion Apr. 2022

Candidate for B.A.S. in Computer Engineering. Currently 2B