

Daniel Prilik

Software Developer

CONTACT

✉ daniel@prilik.com
🌐 prilik.com
🔗 daniel5151

SKILLS

DEVELOPMENT

Rust
C++
C
HTML5 + CSS
JavaScript (ES6)
Typescript
React & React Native

EXPERIENCED WITH

Linux Kernel
KVM
Git
CMake
Travis & Appveyor CI
Visual Studio
Perforce
SDL2
Unreal Engine 4
Wwise

EDUCATION

University of Waterloo
BSE Software Engineering
2015 - 2020

INTERESTS

Exploring Music
Lots of Modern Rock, Psych Rock, and [Future] Funk at the moment

Mechanical Keyboards
Massdrop ALT 65% with MX Blues. Contributor to QMK, the open-source keyboard firmware

Retro Emulation
I keep a close eye on /r/emulation, and enjoy writing my own!

PC Gaming & VR
PUBG, Rocket League, and Fallout are fantastic. I also own a Oculus Rift!

Memes ͡(˘͜ʔ)͡

EMPLOYMENT

Google

Software Engineer - ChromeOS - Rust, C

Mountain View, CA
Winter 2019

- Contributed to *crosvm*, a open-source KVM-based hypervisor written entirely in Rust
- Implemented *virtio-fs*, a paravirtualized shared filesystem device that leverages FUSE and shared-memory to improve filesystem performance by 2x-8x over virtio-9p
- Worked with low-level Linux kernel APIs, including KVM, shared-memory mmaps, and file-descriptors over Unix Sockets
- Gained experience backporting experimental kernel patches, and debugging kernel errors

Apple Inc.

Platform Architecture Engineer - C, C++, Objective-C, Python

Cupertino, CA
Summer 2018

- Replaced hundreds of large, autogenerated modules with a single runtime-configurable module
- Deleted >100,000 LOC from project, reducing compile-time by 30%, and improving maintainability
- Coordinated across teams to develop a long-term support plan for the feature

Microsoft - The Coalition

Audio Engineer on *Gears 5 - C++, Unreal Engine 4, Wwise*

Vancouver, BC
Winter 2017

- Orchestrated a large audio data migration, greatly improving project maintainability
- Identified and intelligently grouped similar assets, reducing asset duplication by a factor of 11x
- Exposed advanced Wwise functionality in the UE4 editor by creating custom UI components, substantially improving audio artist workflow

LCBO | next

Full Stack Web Developer - React, React Native, node.js, Mapbox, Sass

Kitchener, ON
Spring 2017

- Spearheaded the design and development of a complex React SPA
- Architected several RESTful APIs, and assisted in the development of the node.js backend

Polar.me

Front End Web Developer - HTML5, Sass, JS, node.js

Toronto, ON
Summer 2016

- Designed and implemented Native Ads for clients such as Engadget, Wired, and AOL
- Engineered and deployed a build system to streamline and automate Ad development

PROJECTS

ANESE + wideNES - prilik.com/ANESE

- Created a cross-platform Nintendo Entertainment System (NES) emulator in C++11 with SDL2
- Compiles on Windows, macOS, and Linux, with nightly builds available from Travis CI and Appveyor
- Designed *wideNES*, a novel technique to automatically map-out worlds and levels in NES games
- Compatible with many popular NES titles, such as *Super Mario Bros.*, *Legend of Zelda*, *Metroid*, and more

mips241 - prilik.com/mips241

- Implemented an emulator for the MIPS system used in CS 241 at Waterloo
- Features a interactive debugger with step-by-step execution, breakpoints, and instruction disassembly
- Explored compiling C++ to Javascript using *emscripten* to create a Web interface for the emulator

nfinite.space

- Creating a cloud file-storage service to store files without requiring large data centers
- Implementing a protocol to leverage space from connected users to store parts of other users' files
- Developed a proof-of-concept at Hack the North 2016 (winning the AWS Sponsor Prize)
- Actively improving and expanding the project into a Fourth Year Design Project

AWARDS AND RECOGNITION

wideNES Press Coverage

2018

Authored a [blog post](#) about the inner-workings of wideNES, which was featured on *Ars Technica*, *Hackaday*, *Hacker News*, *Kotaku*, and several others sites

Winner of the Amazon Web Services (AWS) Sponsor Prize

2016

Hack the North 2016, QHacks 2017

For work on *nfinite.space* and *fastify* [github.com/Melinysh/fastify] respectively