Newton Ni

248–686–8887 nwtnni@gmail.com nwtnni.me

Education

Cornell University - College of Engineering May 2019

- B.S. in Computer Science — 3.93 GPA
- Selected Coursework — Compilers, Programming Languages, Functional Programming, Formal Verification, Algorithms, Computer Systems, Operating Systems, Distributed Systems

Experience

Software Engineer - Commure August 2019 — Current

- Developed Rust procedural macro for strongly-typed metrics
- Implemented HL7v2 lexer with copy-on-write escape sequence support
- Augmented FHIR mapping language engine with CSV, JSON, XML support

Research Assistant - Nate Foster, Cornell Engineering May 2018 — May 2019

- Designed type system for the P4 network programming language
- Translated informal P4–16 specification into OCaml code
- Discovered bugs in p4c compiler and wrote minimal test cases

Teaching Assistant - Functional Programming and Data Structures Jan 2018 — May 2019

- Lead semiweekly lecture and exercise-based recitation of 30 students
- Created review exercises on concepts like monads, interpreters, and streams
- Received average rating of 4.7/5.0 across 19 metrics and 21 student evaluations

Teaching Assistant - Honors Object-Oriented Programming Aug 2017 — Dec 2017

- Held office hours for 10–20 students, one and a half hours per week
- Taught lab with four other consultants for 25–35 students, one hour per week
- Wrote automated submission format checker

Projects

tigerc - Tiger Programming Language Compiler Jun 2018 — Aug 2018

- Compiles high-level Tiger language down to x86–64 assembly using Rust
- Performs type-checking, IR translation, naive register allocation, etc.
- Applies macro-based metaprogramming for test boilerplate generation

paxos - Paxos Distributed Consensus Protocol Nov 2018 — Dec 2018

- Implements a generic replicated state machine library backed by Multi-Paxos
- Verifies correctness with a JSON DSL-based test harness and extensive logging
- Includes an example chatroom state machine with runnable server and client

gnocchi - Basic Procedurally Generated World Nov 2018 — Dec 2018

- Renders with fog of war and directional shading using JavaScript and WebGL
- Supports multiple concurrent players using a client-server architecture
- Optimizes performance with chunking, bitflags, and occlusion culling

Skills

- **Languages:** Rust, Java, OCaml, Python, Javascript, C
- **Software:** Git, LaTeX, Bash, Unix, Vim, tmux
- **Interests:** Violin, guitar, bouldering, volleyball, cooking, reading