

Alexander L. Carter

Generalist Software Developer

916-275-3474

alrunner4@gmail.com

linkedin.com/in/alrunner4

I'm passionate about software correctness, clarity, and expressing specifications as types. I'd like to apply my knowledge of practical functional programming to build impactful and reliable solutions to difficult problems.

LANGUAGES

FLUENT: C++ (10 years), GNU Bash (10 years), Idris (5 years), Haskell (4 years), C (3 years), Java (3 years), Scheme (2 years)

VERY FAMILIAR WITH: Nix, JavaScript, HTML, CSS, Ruby, Lua, PureScript, GLSL, Objective-C, PHP, Puppet, C#

EDUCATION

Digipen Institute of Technology

2006 - 2010

Bachelor of Science, Real-Time Interactive Simulation, Mathematics Minor

- Four year-long team game projects completed from scratch (C++) including installers and uninstallers (Inno Setup)
- Software rendering algorithms from scratch, DirectX, and OpenGL
- Real-time 2D physics simulation
- Win32 API, WinForms, GDI+, GTK, .NET 4.0

ROLES

Private Tutor

APRIL 2020 - PRESENT

My tutoring strategy is goal-oriented and starts by meeting any student at their current level of understanding. Through discussion and exercises, I've had hundreds of hours of successful lessons supporting students with:

- getting started with programming through games at the middle-school level
- introductory through advanced C, C++, and Java language topics
- functional programming with Haskell and Scheme
- masters-level functional programming languages and compilers

Adjunct Lecturer @ DigiPen

MAY 2019 - DECEMBER 2020

As an adjunct, I managed all aspects of several semesters of the core computer science curriculum for freshmen and sophomores, and led summer elective courses in functional programming with curriculum of my own design.

- CS 120: High-Level Programming 1 (C)
- CS 170: High-Level Programming 2 (Introduction to C++)
- CS 185: Introduction to C++ for Game Designers
- CS 225: Advanced C/C++
- CS 399: Introduction to Functional Programming (Idris/Haskell)

I was the #2 individual contributor for the infrastructure team during my tenure at SigFig, and trusted with full operational control over all production and data systems for a million monthly active users and thousands of clients with assets under management.

- Lead developer and operator of production change management systems (Ruby, Bash, Puppet)

The financial institutions with which SigFig interoperates require high levels of uptime with managed windows for all production changes, for which I designed SigFig's automated human-in-the-loop tooling to increase production change confidence even for substantial multi-service deployments.

- Lead developer and operator of continuous integration systems (Jenkins, Ruby, Bash, Git, Puppet)

I was known throughout the product teams as the subject matter expert for all things build-and-deploy for all of our Java, Scala, and Angular projects across development, QA, and production environments.

- On-premises and hosted virtualization provisioning, networking, and management (VMware vSphere, AWS EC2)

I was responsible for all aspects of host and network provisioning in our virtual environments, including both Linux and AWS network firewalls and corporate authorization controls.

- Containerization of JVM and NodeJS services and build pipelines (Docker)

I spearheaded SigFig's transition to containerization as the team was diversifying into Java microservices alongside a PHP legacy. Containerizing the build system for new services as the initial experiment proved to increase confidence in reproducibility, and paved the way for an eventual decision to transition to Kubernetes as the production backend substrate.

- Logging and monitoring (Logstash, Nagios, Kibana)

In addition to system monitoring and firefighting with Nagios, I encouraged service teams to take advantage of visualization of logs-based metrics made available by our Kibana installation.

- Database administration (MySQL)

In addition to running point on all production changes, including database migrations, I provisioned and managed backups and disaster recovery, and collaborated with service developers about potential impacts of database changes ahead of production change events.

- Corporate and datacenter VPN administration (PfSense)

I was the subject matter expert and on-call for all corporate network troubleshooting, and owned user certificate distribution and recovery during the years before SigFig had a dedicated IT team.

- Key contact for all technical SOC auditing requirements

I was responsible for rectifying auditor-discovered gaps and producing proof of numerous security controls and audit systems for SigFig's successful SOC-2 compliance certification.

Mobile Game Development @ TinyCo

AUGUST 2011 - OCTOBER 2013

My contributions to Tiny Pets and [Tiny Monsters](#) earned me a vote of confidence from the studio's senior developers to participate in our cross-platform game-engine steering committee, and my efforts were key to enabling several of TinyCo's 2D games to dramatically increase animation detail and variety beyond most competitors in the space, especially evidenced in [Family Guy: The Quest for Stuff](#).

- Proposed, designed, implemented, and led ongoing engineering for an overhaul of the team's core 2D animation system, including client code, asset pipeline, viewer, and performance profiler, dramatically decreasing animation storage and runtime memory requirements (90-95%) without disrupting animators' production workflow (C++, OpenGL, Flash ActionScript)
- Led a team of 5 engineers on Android and iOS development of *Super Slots*, for which I pushed our Lua scripting system to new levels of flexibility to enable content creators a high degree of freedom to add content without game rebuilds

Graphics Hardware Debug @ Intel

AUGUST 2010 - JULY 2011

I supported the Graphics Debug Team's efforts in reproducing and analyzing issues discovered by the Test Team and reported by customers, primarily through the SandyBridge platform's support for limited programmable on-die logic probes observable with the lab's logic analyzers.

- After studying many sporadic reproductions of a particularly elusive hardware failure over the course of 9 months, I managed to prove the existence of a transistor voltage leak on 1% of SandyBridge parts that single-handedly created the E stepping.