SETH KINGSLEY

sethkingsley@icloud.com
510-306-8649
https://sethk.github.io
Berkeley, CA

Summary

Deep generalist game engine programmer. With a recent focus on gamedev, I bring 13+ years professional experience shipping apps for desktop, mobile, and cloud.

Areas of expertise

- Modern, multi-core, and cross-platform game engine architecture
- Real-time rendering
- Real-time and offline audio processing
- Native GUI frameworks on desktop and mobile
- Low-level debugging—with or without source code
- Design patterns, refactoring, and static analysis
- Open-source components
- Unix and Unix-like operating systems
- Build systems, shell scripts, unit testing, and CI/CD pipelines

Technical Proficiencies

Languages: C++, C, Objective-C, Ruby, HLSL, GLSL, Java, Python, Various flavors of assembly, JavaScript, Go
Platforms: Windows, macOS, PS4, iOS, GNU/Linux, FreeBSD
Graphics tech: OpenGL (4, ES), Direct3D/DirectCompute 11, RenderDoc, Three.js, SDL, Glfw, Quartz 2D
Gamedev middleware: PhysX, Wwise, Dear ImGui, FMOD
Build systems: Make/GNU Make, CMake, MSBuild, IncrediBuild, Maven, Jenkins
Source control: Git, Subversion, CVS

Career at a Glance

Years	Title	Company	Industry
2019–2020	Staff Software Engineer	Cryptic Studios	MMORPGs
2018–Present	Technical Consultant	(Language and Cognition Research)	Academia
2018	Real-time Graphics Self-study	—	—
2017	Full Stack Engineer	Thumbtack	SaaS
2014–2017	Founder	Filmosaur	SaaS
2011-2015	Contract & Indie. iOS Developer	(Various)	Mobile Apps
2006-2011	Software Engineer	Apple	Music Software
2006	Contract Software Engineer	(Various)	—
2005–2006	Software Engineer	iXsystems	Server Hardware
2004–2005	Network Programmer	Simple Telecom	VoIP
2004	Contract Software Engineer	(Various)	—
2004	Sysadmin	Egation Comm.	Wireless ISP
2002–2003	Software Engineer	OffMyServer	Server Hardware
2001	Associate Engineer	Wind River Systems	Embedded Software
2000–2001	Junior Sysadmin	Berkeley Software Design	Operating Systems
1999-2000	IT Assistant	Delphi Info. Sciences	Mortgage Banking

Selected Work History

Staff Software Engineer, Core Game Systems, Cryptic Studios, Inc.   January 2019–February 2020
Los Gatos, CA   (1 year, 2 months)

Cryptic are the developers of several MMORPG titles for PC and console including *Star Trek Online*, *Neverwinter*, and the upcoming *Magic: Legends*. In the Core Game Systems group, I was able to have an impact on many subsystems of Cryptic's in-house game engine, each having its own editors, asset pipelines, and debug tools. In addition to feature work, I also fixed many bugs in the live games and upgraded vendored middleware components like Wwise.

The primary project going on in my group was building out the new particle effects system, based on arbitrary operation stacks that are mostly executed on the GPU via DirectCompute and PSSL. My contributions included:
- Applied various techniques to improve performance: better tooling for the artists, a new LOD system, and enhancements throughout the pipeline to reduce overdraw.
- Revamped the architecture of the debug tool to make it easier to maintain across PC and console.

Other notable projects:
- Prototyped some new ragdoll physics behaviors so the game designers could experiment with blending back and forth between simulation and animation.
- Designed and built a system for one-shot sound triggers coming from other game systems, so the designers wouldn't need to create dummy particle effect instances for this.
- Got a small amount of console experience fixing video stuttering issues on PS4.

Technical Consultant, Language and Cognition Research   Spring 2018–Present

I consulted on diverse research projects, removing specific roadblocks and by designing accessible and sustainable systems for academic users with a range of research goals and technical know-how. Specific accomplishments:
- Built a transparent pre-processing pipeline for NLP analyses of large-scale natural language corpora (these notebooks have not been published yet; contact me for access).
- Customized Docker images for JupyterLab (IPython and IR kernels), documented for others to run their own containers.
- Fixed blocking bugs in Datavyu—an open-source Java application for annotating video [merged with upstream].
- Wrapped INDCLUS—a statistics package published in 1979 and taking input from punched cards—in an easy-to-use API using R's native FORTRAN interface.
- Customized the childes-db schema and import pipeline for novel research needs.

Real-time Graphics Self-study   January 2018–December 2018 (1 year)

I decided to take a year to myself to study computer graphics, specifically real-time rendering techniques, in preparation for switching industries.

Primary areas of study: Linear algebra, Computational geometry, Physically-based shading, Skeletal animation, Immediate-mode UIs, C++14/17, Rasterization (historical perspective), Curved surfaces, Subdivision surfaces, BSP trees, Three.js

My sources included books, academic papers, conference presentations, open-source game engine and emulator code.

I completed two sample projects and made a video presenting one of them [https://sethk.github.io/gfx.html].

Selected Work History, Continued

Full Stack Software Engineer, Thumbtack   June 2017–December 2017
San Francisco, CA   (7 months)
Thumbtack is an online service that matches customers with local professionals. I started out on Thumbtack's Growth team, collaborating with designers and other engineers to build mainly frontend features in PHP and JavaScript. Later I was part of the Marketplace team, where I worked on the two-sided matching and pricing service in Go.

Throughout my time there, I analyzed the impact of my feature changes with A/B tests, instrumented their functioning in Grafana and Mode Analytics, and set up alerts.

Founder, Filmosaur   August 2014–April 2017
San Francisco, CA   (2 years, 9 months)
Set out to build a film collecting website focused on streaming media. To begin with, I created a search engine for streaming movies. I assembled a database of ≈110K feature films from sources like EIDR, Freebase, TMDB, and Wikipedia/Wikidata. Then I built a pipeline to continuously index ≈330K streams from the major providers: Netflix, iTunes, Amazon, Hulu, and Google. Deployed to Heroku and AWS (S3, EC2, CloudFront), Cloudinary.

Independent iOS App Developer   March 2012–March 2015 (3 years)
In addition to building iOS apps on contract, I also released a few of my own during this time. For one app, I deployed my own backend and push notification server written in Ruby and PostgreSQL to Amazon EC2.

Contract iOS Developer, Monkey Republic Design   June 2012–July 2014
San Francisco, CA   (2 years, 1 month)
Over a few years I collaborated with the designers at MRD to build iOS apps for their clients. Most of the apps we developed together involved heavy customization of Apple's mobile UI and communicated with REST backends.

Contract iOS Developer, Slice (now part of Rakuten)   July 2011–October 2011
Palo Alto, CA   (4 months)
I did a short contract developing the first version of Slice's iPhone app, mentoring one of their engineers in the process. The next year, Business Insider Magazine named the app one of the "Best Productivity Apps of 2012."

Software Engineer, MusicApps, Apple   April 2006–June 2011
Cupertino, CA   (5 years, 2 months)
I was hired by Apple to be part of a small team whose goal was to bring the software instruments and effects from Logic Pro out of the music studio and into a live performance setting. This task presented some interesting challenges in terms of reliably delivering real-time audio processing with minimal latency on a general-purpose computer running a multi-tasking OS.

The app we built is MainStage, and it has been used by several high-profile musicians on tour, including Nine Inch Nails, The Crystal Method, and Goldfrapp. Video: Trent Reznor of Nine Inch Nails discusses MainStage [https://www.youtube.com/watch?v=4kU0skUZTIw].

Contract Software Engineer, Vidiator   August 2006
Mountain View, CA   (1 month)
I diagnosed and patched a slow memory leak in a closed-source C++ DLL so it could be used within the JVM for weeks at a time without crashing.