Kurt Revis

krevis@snoize.com | http://www.snoize.com
LinkedIn | GitHub | Stack Overflow

SUMMARY

Staff/lead software engineer, specializing in iOS and Mac, but curious about everything.

EXPERIENCE AND HISTORY

Software Engineer, Figma
April 2022 — August 2024

Improved the Figma and FigJam experience on iPad, iPhone, and other platforms.

- Added multitouch gesture recognition to Figma's core canvas platform, and used that to build better micro-interactions like scrolling, selection, etc. This gets web events via JS/DOM and processes them in the C++/WebAssembly/TypeScript engine. It's used on iPhone/iPad and on some digital whiteboards.
- The core was written for traditional web on desktops, and assumed the only input devices were a single mouse and keyboard. I made it handle touch, stylus (Apple Pencil), and trackpad, and allowed for event handling code to be specialized for each.
- Added an iOS/iPadOS "native layer", which shows native views (e.g. toolbars) on top of Figma's web content. This lets us use native UI elements to provide a better user experience, while still allowing the web side to be in control.
- Fixed other tricky bugs requiring knowledge of the whole stack: from our web content, to the WebKit code that renders it, to the UIKit code underneath that.

Software Engineer (ICT5), Apple
August 2014 — November 2020

Engineer on UIKit, the UI framework that millions of iPhone, iPad, Apple Watch, Apple TV, and Mac Catalyst apps are built on. It also powers the system UI for iOS, iPadOS, watchOS, tvOS, and CarPlay.

Lead engineer (aka DRI, "Directly Responsible Individual") for Dark Mode in iOS 13. This design refresh affected not just apps, but nearly all UI in iOS and iPadOS. Built prototypes, developed the real implementation, and designed the API. Worked with hundreds of people across the whole organization. Presented it to the world at WWDC 2019.

Also responsible for:

- Drag and drop on iOS: wrote the core infrastructure in apps, and a coordinating daemon. Developed the API, and helped apps adopt it.
- Pointer support on iPad (mouse and trackpad): Made scrolling work with trackpad gestures and mouse wheels, including some subtle but important interactions.
- Maintainer for scrolling, the most famous and ubiquitous interaction in iOS.
- Navigation and "split view" features in apps. And lots of other bits of UIKit.

Developer conference talks: *Implementing Dark Mode on iOS*, *Introducing Drag and Drop*, *Getting Started with Multitasking on iPad in iOS 9*, *Making Apps Adaptive*, and *Build for iPad*.

Staff Software Engineer, Twitter
July 2012 — August 2014

Lead engineer on the Twitter Cards canvas team.

Cards were a way of attaching dynamic interactive content to tweets, based on a custom markup language, with renderers integrated in the iOS, Android, and Web Twitter clients. Cards could:

- Present content like article summaries, photo galleries, and video players
- Show customized ads to promote mobile apps, generate leads, and promote

products

- Handle app "deep links", App Store integration, and install tracking
- Present custom experiences for events like World Cup scores and Oscars polls
- Allow third parties to integrate into Twitter

I started by implementing the cards renderer for iOS, then also took responsibility for the markup language compiler in the backend, written in Scala.

Coordinated with the teams running the cards backend and publishing tools, and assisted internal and third-party teams using and creating cards. Helped set our feature roadmap and quarterly goals, planned new features, and wrote architecture documents for engineers and execs.

Senior Software Engineer, Avatron Software
October 2011 — July 2012

Developed a new iOS / Mac remote access product, Air Login. On the Mac, implemented a system-level coordination daemon, per-user screen capture agent, and controller UI, all communicating via XPC. On iOS, prototyped the client app.

Senior Software Engineer, Apple
January 2003 — April 2010

Shipped several releases of Keynote, Pages, Numbers, and iWeb, on both Mac and iPad.

For iWork on iPad: Designed and implemented a Core Animation-based graphics and interaction framework. On a tight schedule, we built a revolutionary touch-based UI, pushing the limits of an untried OS and hardware, and shipped all three apps on time with the first iPad. Led a team of three engineers.

For iWork and iWeb on Mac: Lead engineer for our graphics framework, built on OpenGL and Quartz. Designed and implemented new features, from high-level UI to low-level details, across the whole suite. While we were primarily responsible for graphics and interaction, I also had a hand in text layout and rendering, document storage, import/export, performance, and HTML widgets.

Senior Software Engineer, The Omni Group
April 1998 — July 2001

Developer on a variety of Mac OS X and WebObjects projects, both for consulting clients and for Omni itself.

Software Engineer, Edmark
January 1996 — April 1998

Worked on 3 original multimedia education titles for Mac and PC, using C++ and in-house cross-platform libraries.

SKILLS

Expert in:

- iOS, iPadOS, and Mac OS X app development
- UIKit, Cocoa, AppKit, Xcode, Combine, Quartz (Core Graphics), Core Animation, Grand Central Dispatch (GCD), XPC
- Swift, Objective-C, and C languages
- Mac OS X and iOS performance tuning

Proficient in:

- C++, TypeScript, JavaScript, HTML, CSS, XML, Protocol Buffers, Thrift
- OpenGL, launchd, Core Text, CoreAudio, CoreMIDI
- UNIX programming and scripting

Familiar with:

- SwiftUI
- Scala, Haskell, Java, Python, Ruby, SQL

PATENTS

Drag and drop for touchscreen devices, US Patents 10,444,976, 10,705,713, 10,860,200, 10,884,604
Web Widgets, US Patent 8,667,415
Image mask interface, US Patent 8,209,632

EDUCATION

California Institute of Technology (Caltech)
B.S., Engineering and Applied Science, concentration in Computer Science
June 1995

PERSONAL PROJECTS

MIDI Monitor and SysEx Librarian

Two open-source MIDI utility applications for Mac OS X. Useful for working with external MIDI devices (like synthesizers), troubleshooting, and testing apps and drivers. Have maintained these apps for 20+ years.