

Stuart Rankin

6351 Embarcadero Drive
Stockton, California 95219

Cell: 530-306-8328

Email: sjrankin@gmail.com

LinkedIn: <https://www.linkedin.com/in/stuart-rankin-4175555/>

GitHub: <https://github.com/sjrankin>

I am an experienced engineer with extensive involvement in product management, programming, and leading teams at large companies, with a wide range of skills. Currently open for exciting and challenging opportunities in technical management, software engineering, or programming.

Skills Profile

- Extensive experience developing with C/C++ and C# on Windows (from Windows 3.0 to Windows 10) on Win32 and WPF/.Net.
- Over four years experience creating graphics-related applications for iOS/iPadOS and macOS using Swift and SwiftUI along with some Objective-C experience.
- Over 35 years of software/hardware validation at a variety of companies and over 10 years as a manager at Intel Corporation.
- Speak conversational Japanese.

Employment History

Self-Employed, June 2016 – current

Stockton, California and Hokkaido, Japan

- Graphics Programming: Wrote a program in C# for graphics image manipulation utilizing lower-level C routines for performance optimization. Created a filter to decode Bayer-encoded images that attracted positive online attention.
- iOS/iPadOS Applications
 - Working on a day/night visualization program for iOS to assist those working remotely. Added real-time earthquake data and working to add near real-time NASA imagery.
 - I wrote a program to convert 2D images to 3D images using simple image analysis and Apple's SDKs. The macOS version converts video streams at almost real-time.
 - I wrote a camera program that makes use of extensive filters, some stock from Apple's SDK, and some I wrote myself to alter images' colors.
 - Wrote a simple GPS track as a tool for another program to track local commute and express trains.
 - Filed over 200 bug reports on Apple's Feedback Assistant tool.

- macOS Applications
 - I wrote a macOS version of the day/night visualization program for iOS. Includes getting real-time data from third-party servers, such as real-time earthquake information and public satellite imagery from NASA.

Intel Corporation, July 1999 – June 2016

Folsom, California

- Graphics System Validation, 2012 – 2016
 - Added power management validation test features for graphics testing program for a consumer-level CPU. Provided rapid turn-around time for emergency feature requests. (C#)
 - Increased robustness and capability of test suite with the addition of a scripting language and bug clean-up (C#). Added missing documentation.
 - Ran task force to drive communication improvement due to an issue that caused a significant schedule delay. Successfully implemented fix.
- Chipset System Validation, 2010 – 2012
 - Created USB 3 error injection testing for a system validation test suite. Code was written in C and ran on Linux and tested how the drive/hardware reacted to known errors injected during testing.
 - I wrote a program to convert proprietary scripts from one version to another.
- Graphics Validation and Management, 2001 – 2010
 - I helped to start an embedded validation group for processors and graphics. This included identifying and creating a test lab, hiring staff in both California and Malaysia, and writing test plans. As validation program manager for Intel's first low-power Xeon CPU I started a team from scratch that performed circuit marginality validation (testing of speed paths under heat stress) as well as leading the entire validation team, located in two different states. Our testing was completed on schedule and budget.
 - Lab owner for the validation team in Folsom for over three years with responsibility for the safety and security of the personnel and lab. Responded to safety incidents to keep lab personnel safe.
 - Set up generic test plans for testing low power Xeon CPUs for future use by my team and other teams.
 - Moved to embedded software validation and lead a team of engineers in three geographical locations: Folsom, California, Chandler, Arizona, and Penang, Malaysia. No defects escaped from any team for over three years.
 - Became manager of Compatibility Validation for embedded devices and led a team of 13 engineers. All tests completed on time and budget.
 - Brought up a second team in Penang, Malaysia to perform test execution.
 - Team was first at Intel to fully test reliability features for low-power 64-bit processors.
 - Team earned excellent feedback for testing virtualization technology in a low-power context.
 - Periodically covered for direct manager and supervised up to 100 engineers.
 - Moved to individual contributor role and wrote several internal tools in C and C# for lab use. Earned two Group Recognition Awards for the tools.

- Graphics Driver Validation, 1999 – 2001
 - Lead team for validation of 3D graphics on early Intel graphics chips. The team grew from four (two engineers and two technicians) to over 20 (mix of engineers and technicians). Team successfully completed all validation activities on time and budget.
 - I was graphics program manager for the validation of several versions of Intel's graphics driver for the i810. In this role, I led not only my team, but other teams for the validation of the entire graphics driver.
 - I was also the validation program manager for other graphics hardware, such as the i830M and an unreleased product. Ran meetings to assess validation progress and reported out high-level data. Drove teams to meet schedules and suggest schedule updates to the overall program manager. The i830M effort involved a large amount of data due to the extensive nature of testing we performed.

Hudson Soft, Ltd, March 1993 – June 1999

Sapporo, Hokkaido, Japan

- Tools Development, 1998 - 1999
 - I developed the host-side device driver interface and client-level debug interface for the Nintendo-64 game console.
 - I worked closely with a third party (Metrowerks) and Nintendo to ensure our customers were satisfied with not only my software but with any other issues that came up.
 - Wrote my own debugger to validate the code I wrote for our customers. This debugger was sufficiently robust to allow me to create a Tetris-like game as test code that ran on the Nintendo-64.
- Game Development
 - I was part of team that developed games for the PC-Engine, an unreleased game console, and the Nintendo-64.
 - Wrote a movie editor use a proprietary format that allowed frame editing of videos for 8-bit games.

International Academy for Youth English Language School, March 1991 – March 1993

Sapporo, Hokkaido, Japan

- English language instructor for conversational English. Responsible for teaching English to people from two years old to over eighty years old. IAY's only technical English teacher.

Education

California State University, Sacramento

Bachelor of Science, Computer Science, honors from the School of Electrical Engineering and Computer Science

GitHub Repositories

- **Flatland.** Program (for both iOS/iPadOS and macOS) that display the world with various embellishments to help workers understand at a glance the time and conditions of remote locations and/or workers. Desktop: <https://github.com/sjrankin/FlatlandView>, and Mobile: <https://github.com/sjrankin/FlatlandMobileUI>.
- **BlockCam.** Program that takes saved images or frames from the video stream of an iOS (or iPadOS) device and converts them from 2D flat images to 3D images using simple analysis and Apple's various SDKs. I also am writing a desktop version for testing algorithms for new shapes that includes a near-real time video stream conversion from 2D to 3D. Written in Swift. <https://github.com/sjrankin/BlockCam> and <https://github.com/sjrankin/DeskBlockCam>. Currently rewriting the program in SwiftUI.
- **GPS Log.** Simple GPS logger intended to be used as a data gathering device for a different program. Stored logged data in the local SQLite database. Written for iOS in Swift. <https://github.com/sjrankin/GPS-Log>.
- **Fouris.** Tetris-like game that runs on iPadOS that features a rotating bucket. Eventually will include rotations in three dimensions. <https://github.com/sjrankin/Fouris>.
- **BumpCam.** iOS camera program that uses filters from Apple's SDK as well as filters I developed (mostly written as Metal filters) for the program. The bulk of the code was written in Swift, with the Metal filters in a dialect of C. Rewriting in SwiftUI. <https://github.com/sjrankin/BumpCamera>.
- **TDDebug.** Remote logging tool for iOS and macOS. Written in Swift. <https://github.com/sjrankin/TDDebug>.
- **BarcodeClock.** Clock program of iOS that displays time in various ways, mainly barcodes. Written in Swift. <https://github.com/sjrankin/BarcodeClock>.