Stuart Rankin

Inahocho, Higashi 2-chome, 5-8, Kitahiroshima-shi, Hokkaido, 061-1125, Japan
Home: +81 011-375-6145 | Mobile: +81 080-8295-2684 | Email: sjrankin@gmail.com

Objective

A position that allows me to use my programming and people management skills to create and sustain products on a variety of platforms and devices.

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 three years experience creating graphics-related applications for iOS/iPadOS and macOS using Swift.
- 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

Kitahiroshima, Hokkaido, Japan and Yubari, 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 (specifically, images from the Curiosity Mars Rover) that attracted positive online attention.
- iOS/iPadOS Applications
    - Working on a day/night visualization program for iOS/macOS 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 (whether stills or from the camera's video stream) to 3D images using simple image analysis and Apple's SDKs. The macOS version converts video streams at almost real-time. (Swift)
    - 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. (Swift)
    - Wrote a simple GPS track as a tool for another program to track local commute and express trains. (Swift)
    - Filed over 100 bug reports on Apple's Feedback Assistant tool.

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.
    - Validation program manager for Intel's first low-power Xeon CPU. This included bringing up 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.
    - Was lab owner for the validation team in Folsom for over three years with responsibility for the safety and security of the personnel and lab.
    - 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.
    - 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 54-bit processors.
    - Team earned excellent feedback for testing virtualization technology.
    - Team also validated chipsets 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.
    - I was graphics program manager for the validation of several versions of Intel's graphics driver for the i810.
    - Validation program manager for the Timna CPU graphics driver.

- Validation program manager for the i830M graphics driver – team found and reported a very large number of defects that were not expected.

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.

Baxter MicroScan, March 1985 – March 1991

West Sacramento, California

- I developed, implemented and ran a federally mandated software quality assurance program for biomedical devices and software.
- Validated over seven major versions of software for data management of patient results and diagnoses.
- No major escapes from software I validated.
- My software, documentation, and processes passed 100% of all federal and corporate audits over the course of seven years.
- Wrote tools (in Pascal) for the validation team on an as-needed basis.

Education

- California State University, Sacramento, May 1984
- Bachelor of Science, Computer Science, honors from the School of Electrical Engineering and Computer Science, 3.04/4.00
- Task oriented classes taken at Intel Corporation, including multiple management classes, safety and regulatory classes, and product-specific technological classes.

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. https://github.com/sjrankin/FlatlandView and https://github.com/sjrankin/Flatland
- 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.
- 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. 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.
- ColorBlend. Windows program that was a test bed for software image and gradient blending as well as graphics image manipulations, including Bayer decoding. Written mostly in C# with C code for lower-level image processing and uses WPF for the user interface. https://github.com/sjrankin/ColorBlend.
- Color Space Flyer. Program that generates images that when used to create a video, makes it appear as if you are flying through color spaces. Written in C#. https://github.com/sjrankin/ColorSpaceFlyer.