Raymond W. Ko

⌂ 911 Spring St, Philadelphia, PA 19107
☎ +1 215 8057074
✉ raymond.w.ko@gmail.com

Work Experience

Current, from Jan 2011

Syandus Inc., Exton, PA

Software Engineer

Created medical simulations, games, web sites, and metric reports. Maintained internal game engine code base and servers.

Accomplishments

- Served as main programming lead in several medical simulations. Responsible for encoding medical knowledge from various scientists and doctors into simulations.
- Programmed graphics and unit logic components of ImmuneQuest, a 3D game designed to teach students the basics of the human immune system.
- Ported internal Windows only 3D game engine to run on Mac OS X, iOS, Android, and Linux.
- Integrated Lua scripting language support into internal gaming engine to increase developer productivity compared to when C++ was the only available choice.
- Created an internal software framework for informational group sessions with virtual patients.
- Remade syandus.com website from a previously Flash only one to support mobile devices.

Sept. 2009 – Dec. 2009

University of Pennsylvania

Teaching Assistant, Operating Systems

Marked test papers, graded semester projects of Linux shells written in C, and held office hours for answering student questions.

Sept 2006 – May 2010

Penn College House Computing

Senior Information Technology Assistant

Provided technical support for student computers. Troubleshot cases that involved data recovery, full operating system reinstalls, hardware installation and replacement, malware removal, and 802.1X wireless network setup for new students.

Education

2006 – 2010 **Bachelor of Science in Engineering**
Computer and Information Science
University of Pennsylvania

Computer Skills

Working Knowledge	C++, Lua
Basic Knowledge	Python, Clojure, Java,
	Bash shell scripting, Windows BAT scripting,
	HTML, PHP, JavaScript, CSS,
	LaTeX

Side Projects

A full list of personal side projects available at github.com/raymond-w-ko and bitbucket.org/rko.

omegacomplete A keyword completion plugin for the Vim text editor. Uses Python bindings to communicate with Vim and C++ to implement the algorithmic portions. Designed to be multithreaded with background processing to scale with modern CPUs and avoid blocking. Supports various completion types, such as: prefix, title case, camel case, subsequence, closest edit distance, and keywords with underscore and hyphen separators.

tsukuyomi A Lisp heavily inspired by Clojure which is compiled down to Lua.

JVM bindings for Vim A highly experimental fork of the Vim text editor with JVM (Java Virtual Machine) bindings. This allows JVM hosted languages such as Clojure, Scala, and Groovy to be used inside Vim.