# Paul Kang

solipsistic@berkeley.edu
GITHUB: sollipse

Pleasant Hill, CA 94709
(925) 640-0310

## Education

Senior, University of California, Berkeley
Currently completing B.A. Computer Science — Graduation: Fall 2014

## Work Experience

Fullstack Software Intern, Soldsie Inc.

- Built out several key site components, from search bars to modal option menus, merging backend Rails database hooks with frontend sass or coffeescript functions.
- Responsible for coordination between database and sales/ops analytics, piping data into Salesforce and Mixpanel for better analysis.
- Built out various API-dependent interfaces using Shopify and Instagram that extended site functionality to other popular apps.

**For information regarding this internship, please contact Soldsie's lead engineer, Matt Hui at matt@soldsie.com**

## Extra Curriculars

Officer, Computer Science Undergraduate Association.

- Direct coordinator for Berkeley's representatives to the Cloud Computing summit at the Computer History Museum, San Jose.

## Projects Completed

Projects can be found on github.com/sollipse

- **Android Test Applications**: Trivial mobile android applications made to explore various development environments. While the apps themselves display only simple text such as "hello world" or interact very basically with facebook OAUTH tokens, they were a useful way to familiarize myself with Eclipse and IntelliJ plugin environments for Android development.
- **Graph-Based Mapreduce**: A java-based framework that completes a parallelizable task by forking different tasks (mapper), and joining them (reduce). This implementation converts nodes in a graph to arrays of writable objects, then returns a histogram of their average distances.
- **Engima Simulator**: Java package of Rotors, Cylinders, and master class Machine. Rotors and Cylinders are instantiated with historically accurate scrambled alphabet keys. Rotors are then run in randomized value by Machine class to simulate a polyalphabetic substitution cipher.
- **Model Operating System**: The NachOS model operating system is a test environment that deals with how to allocate resources so that a machine can run many different programs and give the illusion of concurrency. The model OS is separated into four primary PROJECTS that deal with networking, thread allocation, memory allocation, and kernel management.

## Skills

| | |
|---|---|
| **Languages (by recent use)** | Python, Ruby, ERB, C, C++, Scheme, Java, HTML, CSS |
| **Version Control Systems** | git |
| **Virtualization Platforms** | VMWare, VirtualBox |
| **Web Development Frameworks** | Ruby on Rails, PosGRES, Spork |
| **Operating Systems** | Windows, UNIX/Linux, Android |