Marcus Willock

125 Amboy Ave   Tel: 602-341-7413
Metuchen, NJ 08840   crazcalm@gmail.com

Work Experience

Peace Corps – China   6/17 – 1/18
- English Teacher at the Chongqing Vocational Institute of Engineering in Chongqing Jiangjin.

Tablet Inc – New York, NY   12/15 – 6/17
- Maintained and built upon a Selenium suite of regression tests that ran in parallel with the use of Selenium Grid. This was written in Python, and I ported the code from Python 2 to Python 3.
- Created and maintained a IOS regression suite that ran in Xcode using the UI Testing Tools, Xcode server, and Xcode server bots. This work was done in Swift.

Adcade – New York, NY   8/14 – 7/15
- Built a utility that parsed the information within Client Service's Google Drive spread sheets and generated a generic QA testing speed sheet. The QA testing spread sheet contained the needed information from the Client Service's spread sheet and the device list needed for testing, which was dynamically generated. This tool was built in JavaScript and used the Google App Script API to interact with the Google Drive Documents.
- Built a utility that parsed Charles (proxy) XML data for ad ids, session ids, and metrics fired. This data was saved locally using the following file structure: ad_id/session_id(s)/metrics_data.txt. This tool was built in Python 3.4 using just the standard library.
- Designed and built a desktop application that provided a wrapper around the file structure output of the above tool. This application, which resembled a text editor, had a file structure tree on the left that provided easy access and visibility to the files. The bulk of the application was a text field that allowed for viewing, creating, and modifying the metric files. The menu bar contained options that allowed for basic analysis of the metric files. For example, the "ad summary" option provided a summary of all the metrics that were fired in that ad and give the count of how many times each metric was fired over the course of all sessions. This application was built in Python 3.4 using Tkinter and other standard library packages.
- Designed, built, and maintained a tool that, through the use of keyboard and mouse automation, allowed for the systematic automation of desktop applications. The automation of the mouse leveraged image recognition to find the x and y coordinates. This tool was built in Python 2.7 and depended on two third party packages by the name of pyautogui and OpenCV.

FuturesMedia – New York, NY   2/14 – 5/14
- Designed, built, and maintained their Admin website, which allowed for a user with Admin permissions to create and modify both Accounts and Users. It also allowed for the manipulations of the Account to User Mappings, which was the list of Accounts that a specific User had access to.
- Developed email notification prototypes. The first email sent Users a list of all the trades their accounts have been involved in. The second email sent Users a message every time an account of theirs had been invited to an auction. The third email provided Users with an overall summary of their activity on the platform, which included the information from the first two emailing plus the information of the auctions that the User had opened, and a list of all the Accounts that were invited to that auction.
- Environment: Ruby, Sinatra, Linux, Apache, PostrgreSQL, Javascript, and Backbone.js

Github: github.com/crazcalm

Projects:

go-pokemontcgsdk (Golang):
- Released an unofficial Pokemon Trading Card Game SDK for Go. It wraps the pokemontcg.io API, which allows you to search for Pokemon cards at will.
- https://github.com/crazcalm/go-pokemontcgsdk

golang example (Golang):
- Post giving a talk on creating a file server that hides dot files, I later submitted a PR to Golang adding an example of this type of file server to their documentation. This has been merged into the go1.12.
- https://github.com/golang/go/pull/26768

go-rss-reader (Golang):
- A terminal rss reader that caches previously fetched rss feeds in a sqlite database while using an interactive interface built with gocui.
- https://github.com/crazcalm/go-rss-reader

html-to-text (Golang):
- Converts html to well formatted text. I built this so that I could use it to convert the html in rss feeds to text that can be displayed in the terminal for go-rss-reader.
- https://github.com/crazcalm/html-to-text

term-quiz (Golang):
- A terminal quiz application that reads ABCD, true and false, and fill in the blank questions (and answers) from a csv file while rendering an interactive experience via gocui.
- https://github.com/crazcalm/term-quiz

artonge/go-csv-tag contribution (Golang):
- I added the dump (and dump to file) functionality, which allows you to write structs with csv tags to file.
- https://github.com/artonge/go-csv-tag

Conference Talks:
- Wrapping Go in Python (https://www.youtube.com/watch?v=6v6OZXk2E_w)

Blogs
- https://crazcalm.github.io/blog/
- https://ogcrazcalm.blogspot.com/

Education
- Arizona State University:
  - BA in Mathematics and a minor in Chinese.

Github: github.com/crazcalm