

ULYSSE CARION

ulyссе.io ♦ ulyссе@ulyссе.io

GitHub: @ucarion

I always try to use the right tool for the task, and I think trying out new languages and tools is part of a programmer's job. I use `git` extensively and work from the Unix command line.

I love hacking on projects in Ruby, Node, and Haskell.

EDUCATION

University of California, Berkeley

2014 - Present

Pursuing B.S. in Computer Science & Engineering

EXPERIENCE

Curriculum

June 2013 - Present

Founder

- Started my own software product that streamlines the way computer science is taught in high schools.
- Wore many hats—programmer, designer, DevOps, help desk, sales—and loved it.
- Developed a Ruby on Rails application that grades student programs using a DelayedJob worker, which in turn talks to Docker. I used Docker containers as sandboxes to execute submissions in.
- Created a one-step deployment process with Capistrano that deploys to a DigitalOcean droplet. Cron jobs automatically back up the database as well as students' code to an AWS S3 bucket.
- I ran the app within Vagrant locally, used Postgres as a backend, tested with RSpec + Capybara, and wrote CSS using SCSS.

ScoreStream

December 2013 - April 2014

Intern

San Diego, CA

- Made web-scrapers to fill in gaps in ScoreStream's data.
- Created an internal website for sending out push notifications to specific users within a selected circle or U.S. state (I did not write the code to actually send notifications out, though). This was written in Javascript, with an AngularJS frontend and a NodeJS + ExpressJS + MySQL backend, all deployed onto an Amazon Web Services EC2 instance.

RCSB Protein Data Bank

June 2013 - August 2013

Intern

University of California, San Diego

- Created a bioinformatics search tool that, given a rough description of drug-protein interaction, would find instances of matching interactions in the Protein Data Bank (PDB).
- Spent a lot of time making the thing fast; the PDB consists of 90,000+ proteins, each containing thousands of atoms. To keep search times within acceptable limits, I relied on precalculating all drug-protein interactions on the PDB (using the filesystem as a poor man's search tree), so that I never had to analyze a single protein structure on search-time.
- Worked on making searching "smart", so that when a user made a query, all related matches would be returned too. This meant detecting graph isomorphisms as well as chemically-equivalent groups (e.g. hydroxyl groups or resonant structures).
- Learned a lot of chemistry.
- Most of the work was in Java, but I made a few Ruby scripts to help out along the way.
- Made numerous contributions to BioJava, an open-source bioinformatics library.

PROJECTS

The Musical Turk, an open-source audio-visual experiment made with WebGL. It portrays a mechanical machine that performs classical music. Check it out at: ulysse.io/turk (it only works on Chrome, sadly).

Godot, a chess-playing program. I made this as a Junior in high school as a way to learn AI. It uses some pretty fancy techniques (bitboards, alpha-beta pruning, iterative deepening, etc.) and plays a mean game of chess. I hooked it up to Selenium WebDriver and had it play online on chess.com, where it ranked in the 99.89th percentile among humans (it beat a few nationally- titled players along the way, too).

Interlingua, a toy programming language with user-defined keywords. It's got basic object-orientation and all that, but it also lets you rename `true` to `verdad`. The idea was to create a language where the keywords weren't hard-coded to be in English, making it easier for non-Anglophones to learn programming.