

# Andrew Amis

12 Pine Hill Dr.  
Santa Cruz, CA  
95060  
T 1 (831) 423 0895  
E [atamiser@gmail.com](mailto:atamiser@gmail.com)  
[atamis.me](http://atamis.me)  
[github.com/atamis](https://github.com/atamis)

## Education

**Bachelor's in Computer Science, Oberlin.** **2013–2017**

I attended Oberlin College, and obtained a Bachelor's in Computer Science.

## Experience

### *Vocational*

**Symantec, eBusiness Internship.** **2015**

I worked with QA teams on relevant programming projects. I also worked on a reporting dashboard for manual testing in Java, and worked on test automation for the storefront team in Go.

**Townsend & Clark, freelance programmer.** **2014–2016**

I opened a web design firm focusing on small business websites and web applications. As the backend engineer and systems administrator, I developed several Rails applications ranging from the simple campaign sites to complex web applications.

**Oberlin College, Work Study.** **2013–2016**

I filled a variety of roles including Macintosh Desktop Support, Assistant Network Technician, and Web Developer for Oberlin IT and Psychology Departments. I primarily used Ruby and Rails.

**Georgiana Bruce Kirby, Computer Technician.** **2008–2013**

I assisted the IT administrator at my high school in his maintenance of the school's network, comprising approximately 100 Windows machines, routers, switches, and central servers. I additionally worked on the school's homework, event management, and communications Rails application, fixing bugs to drastically reduce downtime.

**Codulous, Web Developer.** **2011**

I started a company with two friends through the Teens in Tech program, a startup incubator aimed at young adults. We made a company that built and operated an online code editor. I handled backend code and system administration for the company. I was the main programmer for the Ruby on Rails web application, assisted the frontend team, and managed the company's server operations, both in house and client facing applications.

### *Smaller Projects*

**AYPSELA, System Administrator.** **2008–2014**

I operated a Linux server for a group of friends that ran VoIP communication and packaged game server applications.

**COSMOS, summer program.** **2012**

In the summer of 2012, I attended the COSMOS program at UCSC for video game design. A colleague and I built a strategy game in Java (using the Processing framework). You can find the game I made at <http://atamis.me/2012/08/05/planet-5/>.

**OpenThought, web application.** **2011–2015**

OpenThought is a Ruby on Rails web application I wrote to hold, sort, and list notes, or thoughts, for myself and others. I use it as a note pad, a bookmarking service, and a place to store important text segments I may need later. You can find it at <http://openthought.herokuapp.com/>

**Prestissimo, web application. 2015**

Prestissimo is a Rails web application created by the Oberlin CS department as an alternative to the college's official course listing website. I worked to improve page load time and performance. You can find it at <https://oprestissimo.com/>. This job represents working with an established code base that at the time lacked tests or thorough documentation.

**Gobcom, web application. 2016**

An unreleased web forum written in Rails with multiple identity levels and associated restrictions.

**CSCI361, Game Design, Oberlin.**

At Oberlin I took a course in practical game design using Unity, and created two games over the course of the semester. The focus of the course was practical software development as part of a team, under deadline, and without strong specifications.

**Unity3D Project, 2D game development. 2017-2018**

Reacquainting myself with the Unity3D engine, updating and testing old games to the new version, and developing the core mechanics of a new game.

**Ironie Space Lisp, Programming language development in Rust. 2017-2018**

In order to meet some particular requirements for a separate project, I began development on a new programming language based on Lisp and Erlang. Some of the requirements were that the entire environment be pausable and serializable, and contain deep hooks and callbacks to allow the host program to dynamically control fundamental aspects of evaluation. The interpreter is currently under development in Rust.

## Languages

- Ruby & Rails
- Python & Java
- C#/Unity
- Go
- Erlang/Elixir
- Scheme
- Clojure
- C
- HTML, Javascript, & CSS

## Computer skills

**Linux:** SSH, Git

**Remote:** OpenVPN, VNC, IRC

**Web:** PostgreSQL, NGINX, Apache

**Virtualization:** Xen, AWS, Docker