Daniel Rebelsky

https://d.rebelsky.com

EDUCATION

Stanford University MS in CS
GPA 4.157, September 2024

Stanford University BS in CS, with honors; minor in music
GPA 4.048, September 2024

Grinnell College Classes (at large as a high school student)
GPA 4.0, 2016–18

Relevant coursework: Calculus II (Multivariable Calculus), Linear Algebra, Differential Equations, Discrete Bridges to Advanced Mathematics: Graph Theory, Thinking in C and Unix, Logic

WORK EXPERIENCE

Algo Works LLC Consultant
September 2023–Present

I worked on software for Algo Works, a software development, consulting, and training company.

Stanford CS 143 (Compilers) CA
April 2024–June 2024

Stanford CS 140E (Operating Systems Design and Implementation) CA
January 2024–March 2024

Stanford CS 107 (Computer Organization and Systems) Head CA
June 2023–December 2023

Stanford CS 107 (Computer Organization and Systems) CA
September 2022–December 2023

I worked as a course assistant, which consisted of hosting regular review sections ad office hours, grading and creating assignments and exams, and teaching lecture. As head CA, I also worked to manage the other CAs, e.g., assigning grading ranges, overseeing grading, etc...

CURIS Research Intern (at Stanford)
June 2022–August 2022

Working with Stanford's Empirical Security Research Group (https://esrg.stanford.edu), I worked on creating an SSH replacement written in Go.

Stanford CS 107 Course Material Helper
March 2022–June 2022

I worked with two of the lecturers for CS 107 to create additional resources for students in the course.

CURIS Research Intern (at Stanford)
June 2021–August 2021

Advised by Moses Charikar and Li-Yang Tan, I researched double-elimination tournaments' efficacy in approximating the Copeland winner.

Stanford CS 106 Section Leader
January 2021–August 2023

As a section leader, I lead a weekly discussion section for approximately 10 students in Stanford's introductory computer science courses (CS 106A/B). I also grade assignments and exams and assist students in debugging.

Craft of Code Camp Counselor (at Grinnell College)
Summer 2018

As a camp counselor, I helped develop curriculum, worked with elementary and middle-school students (individually and in small groups), and supervised activities.

CURATED PROJECTS

FuncK: A Functional Supserset of ChucK (https://ccrma.stanford.edu/ rebelsky/220a/final/)

For my final project in Music 220A (Fundamentals of Computer-Generated Sound), I created FuncK as a superset of ChucK that supported various functional idioms (namely, lambda functions, first-class functions, map, reduce, filter, and foreach).

A JIT Compiler for Whitespace (https://github.com/drebelsky/whitespace-jit)

I wrote a C++ program that will interpret or compile a Whitespace program into x86-64 machine code *and then execute it. The interpreter runs over 30x faster than the original Whitespace interpreter and the JIT compiler runs an additional over 3x faster than that (over 100x faster than the original Whitespace interpreter).*

A DSL/Library for Common Practice Music

For my final project in CS 343D (Domain-Specific Programming Models and Compilers), a partner and I implemented a DSL/library for common practice music. Among other capabilities, it is able to realize a chord progression for four voices according to standard common practice rules.

CoCode: A Website for Introductory CS Collaboration

(https://hci.stanford.edu/courses/cs147/2021/wi/projects/EducationTechnology/COCODE/)

As a group project for CS 147 (Introduction to Human-Computer Interaction Design), I worked on creating a prototype of a website (CoCode) to enable students in introductory CS students to collaborate on simple problems. I was the primary programmer for this project. Out of around 50 groups in the class, my group tied with two other groups for Best Overall Project and Best Demo.

Monte Carlo Tree Search Ultimate Tic-Tac-Toe

As an optional project for CS 109 (Introduction to Probability for Computer Scientists), I implemented MCTS in Python and examined its efficacy on both Tic-Tac-Toe and Ultimate Tic-Tac-Toe. I included interactive modes in which a user could play either game against the algorithm via the terminal or via their browser (with a React frontend and Python backend). The browser version also allowed the user to select who each player should be (MCTS, human, or minimax in regular Tic-Tac-Toe) and provided a visualization of the inner weights the algorithm was using.

PUBLICATIONS

Coauthor on two computer science publications 2015

- *Developing Computational Thinking Through Image Making and Constructionist Learning (Abstract Only). SIGCSE 2015: 690*; *MIST, The Mathematical Image Synthesis Toolkit. SIGCSE 2015: 599*

I worked on both the front end and the (Node.js) back end during the summer of 2014. In the summer of 2020, I helped to update it to a more modern, React single-page application. Among other things, I worked on compiling our textual image format to a WebGL fragment shader.

TECHNOLOGY

Languages: C, C++, Go, JavaScript, Python, Vimscript

Frameworks: Express, Node.js, React

Other: Basic Assembly, Basic Haskell, Basic Racket, Basic Rust, Basic Shell Scripting, Blender, CSS3, HTML5, LaTeX, Linux

AWARDS

Phi Beta Kappa June 2024

STUDENT ACTIVITIES

Stanford Jazz Orchestra September 2019–June 2023

I played baritone saxophone and occasionally, bass clarinet. During my time, we played with a number of notable jazz musicians including Wycliffe Gordon, Sal Cracchiolo, Eric Marienthal, Clairdee, and Wayne Bergeron, among others.