

Joshua Ye

CONTACT

joshua-ye.com

[linkedin.com/in/j-ye](https://www.linkedin.com/in/j-ye)

github.com/joshua08ye

EDUCATION

University of Michigan **Ann Arbor, MI**
BSE. in Computer Science
GPA: 3.7 *April 2023*

Awards:

Dean's Honor List, William J. Branstrom Freshman Prize, James B. Angell Scholar

Coursework:

- Software Engineering
- Computer Vision
- Human Centered Machine Learning
- Artificial Intelligence
- Computer Organization
- Data Structures and Algorithms
- Practical Programming in Java
- Practical Data Science for Engineers
- User Interface Development

Activities:

- **Eta Kappa Nu**
Computer Science and Electrical Engineering Honors Society
- **UMich Solar Car Team**
Strategy/Software division member

SKILLS

Technical

C++, Java, Rails, Python, HTML, CSS, Android, JavaScript, SQL, MATLAB

Industry

Git, Docker, Project management, Object Oriented Programming (OOP)

VOLUNTEERING

HuaXia Chinese School *Spring 2020*
Mathematics Teacher

- Mathematics teacher for middle-school students helping to develop problem-solving skills via an inquisition-based learning style.
- Planned and presented lessons, proofs, and derivations on advanced topics like combinatorics, trigonometry, and calculus for class of approx. 30 students.
- Assigned and graded weekly written homework assignments.
- Created remotely accessible, asynchronous curriculum during the COVID-19 pandemic.

EXPERIENCE

Software Engineering Intern

May–August 2021

IPVM

- Implemented and maintained core features of company website as a full stack developer on the front end engineering team.
- Developed authoring, administration, and logging features using Rails, HTML, and Javascript and integrated with industry-scale legacy codebase.
- Gained insight into coding paradigms such as MVC as well as industry workflows used to triage and address issues, bugs, and feature additions.
- Managed backend database using SQL and Rails Active Migrations.
- Drove end to end feature development from UI/UX to engineering execution to create a seamless experience for both users and administrators.

Strategy and Software Programmer

September 2020–January 2021

University of Michigan Solar Car Team

- Led research into solar irradiance calculations and its components to develop constant time algorithm calculating solar irradiance from road conditions, weather data, and car orientation.
- Developed array model, testing, and data visualization suite in MATLAB and integrated solar array model with legacy race simulator in C++.
- Collaborated across divisions and connected endpoints with shading model to develop holistic algorithm to calculate total solar flux received by array.
- Developed a race simulator in C++ to optimize cruising speed using binary search.

PROJECTS

Pac-Man Reinforcement Learning AI

December 2022

- Wrote an AI in Python to play a simplified, non-deterministic version of Pac-Man by implementing Q-Learning and SARSA algorithms.
- Modeled iterative learning by storing calculated reward values in memory and updating estimated rewards and optimal strategies after each iteration.
- Used stochastic strategy selection during training to avoid convergence at local minima while updating reward estimations.

Source Code Mutator

October 2022

- Developed fuzzer in Python using abstract syntax tree parsing to create mutations for an input source code file in Python by negating some comparison operators, swapping binary operators, and deleting function calls and variable assumptions.
- Generated suite of mutated source files to validate strength of test suites by assessing quantity of mutated implementations caught by test suite.
- Demonstrated correlation between test suite mutation score and line coverage by ordering nested test suite subsets through mutation scores.

Chess Minimax/DNN AI

September–October 2022

- Wrote chess engine with variable move depth in Python using a minimax algorithm with variable depth and Alpha-Beta pruning
- Created a Deep Neural Network chess engine using Tensorflow by training and testing on board position evaluations generated by Alpha-Beta pruning
- Analyzed the difference in runtime and performance of Alpha-Beta pruning and DNN and concluded that while minimax is guaranteed to find the board position with the strongest evaluated heuristic value, DNN evaluations take significantly less time

Test Suite Minimizer/Fault Localizer

October–November 2022

- Implemented and applied delta debugging algorithm in Python to reduce test suite to smaller subset with equivalent line coverage.
- Created Turing reduction between test suite minimization and vertex-cover problem to conclude that delta debugging reduces test suite size but that resulting subset is not guaranteed to be minimal.
- Developed program in Python to localize faults given test suite gcov files.
- Studied and implemented Ochiai coverage-based fault localization algorithm to determine most suspicious lines of faulty source code.