

Sean L. Snaider

sean@snaider.com | (516)-587-1551 | linkedin.com/in/seansnaider/ | github.com/SeanSnaider | seansnaider.com

EDUCATION

Northeastern University

B.S. in Computer Science

Boston, Massachusetts

Expected Graduation, May 2028

- o **Concentration:** Software Engineering
- o **GPA:** 3.93/4.00, *Dean's List*
- o **Related Coursework:** Computer Systems, Algorithms and Data Structures, Fundamentals of Computer Science 1, Fundamentals of Computer Science 2, Object-Oriented Design, Introduction to Databases, Theory of Computation

SKILLS

Languages: Python, Java, C, Assembly, SQL, JavaScript, TypeScript, Racket

Frameworks & Libraries: React, Tailwind, Next.js, Vite, Pandas, NumPy, Matplotlib, Pygame, Java Swing

Tools & Platforms: Git, WSL, VSCode, IntelliJ IDEA, GDB, Unittest, JUnit, Neovim

EXPERIENCE

Khoury College of Computer Sciences, Northeastern University

Boston, Massachusetts

Sep 2025 - Present

TA for Program Design and Implementation 1 (Fall '25) & 2 (Spring '26)

- Used Python and Java to lead weekly office hours and lab sections for 30+ students, debugging runtime errors, logic issues, and algorithmic inefficiencies to improve student comprehension and assignment completion rates
- Used Python's **Unittest** and **JUnit** to individually tutor 20+ students weekly on testing methodologies, teaching them to identify flaws in their code and systematically debug errors across Object-Oriented Java and Python data analytics projects

Disrupt, Northeastern University

Boston, Massachusetts

Sep 2025 - Present

Software Engineer

- Used **Next.js**, **Tailwind**, and **JavaScript** to implement UI/UX improvements to an existing codebase, collaborating via **Git** branches to enhance the responsiveness and visual consistency of the club's website

PROJECTS

Rubik's Cube Solver and Teaching Tool - Python, PyGame

Oct 2025 - Jan 2026

- Used **Python** and **Pygame** to engineer an interactive 3D Rubik's Cube simulator, enabling users to manually scramble and solve the cube via keyboard controls with real-time visualization of 18 face rotations
- Used the Kociemba two-phase algorithm to generate near-optimal solutions (≤ 20 moves) in under 1 second, displaying step-by-step move sequences for users to follow
- Used state validation logic in **Python** to build a step-by-step tutorial mode, requiring users to complete each solving stage before advancing to reinforce learning progression

Guitar Learning Tool - React, TypeScript, Tailwind, Python, SQLite, Vite, Pydantic

Dec 2025 - Present

- Used **React** with **Vite** and **TypeScript** to develop an interactive fretboard visualization, allowing users to explore scales, view tablature, and mark positions as learned through a responsive UI
- Used **FastAPI** with **SQLAlchemy** and **Pydantic** validation to design a RESTful API implementing CRUD endpoints with **SQLite** persistence, enabling user progress tracking across sessions

Sanguine - Java, JUnit, Java Swing

Nov 2025 - Dec 2025

- Used **Java Swing** with a pub-sub pattern to implement an interactive UI, synchronizing both players' views on each move for seamless gameplay
- Used **Java** interfaces to design pluggable AI strategies selectable at runtime, enabling automated opponent move generation with interchangeable difficulty levels
- Used **JUnit** with mock objects for controller, strategy, and model layers to isolate unit tests, achieving 90%+ code coverage on model and strategy components

File System - C, FUSE

Nov 2025 - Dec 2025

- Used **C** and **FUSE** to build a filesystem driver mounting a 1MB disk image as a Unix-style filesystem, supporting 128 files/directories with persistence across mount cycles
- Used mmap to implement block-based storage with inodes, free-space bitmaps, and POSIX syscalls (read, write, mkdir, unlink, truncate), achieving compatibility with standard Unix commands