Sean L. Snaider

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

EDUCATION

Northeastern University **Boston, Massachusetts**

B.S. in Computer Science *Expected Graduation, May 2028*

- **Concentration:** Software Engineering
- **GPA:** 3.93/4.00, *Dean's List*
- **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
Frameworks & Libraries: React, Tailwind, Pandas, NumPy, Matplotlib, Zustand, Web Audio API, Fuse, Pygame, Java Swing
Tools & Platforms: Git, WSL, Docker, VSCode, IntelliJ IDEA, GDB, Vite, Unittest, JUnit

EXPERIENCE

Khoury College of Computer Sciences, Northeastern University **Boston, Massachusetts**

TA for Program Design and Implementation 1 (Fall '25) & 2 (Spring '26) *Sep 2025 - Present*

- Debugged **Python** and **Java** code for **30+** students weekly through office hours and lab sessions, diagnosing issues in list comprehensions, class hierarchies, and data structure implementations across two **400+** student courses
- Co-developing curriculum integrating agentic AI tools into coursework, conducting research on how AI-assisted coding impacts student learning outcomes and debugging skills
- Taught systematic debugging and testing practices including breakpoint usage, test isolation, and fault localization, helping students resolve issues independently rather than relying on line-by-line code review

PROJECTS

Guitar Learning Tool - *React, TypeScript, Zustand, Web Audio API, Tailwind* *Dec 2025 - Present*

- Engineered real-time audio synthesis using **Web Audio API** oscillators with ADSR envelope shaping, producing responsive note playback across a 24-fret interactive fretboard
- Built a scale library covering **40+** patterns (modes, pentatonics, jazz, exotic) with CAGED and 3-note-per-string position systems, persisting user progress and preferences via **Zustand** with localStorage
- Designed three practice modes (note identification, sequence drills, interval training) that scale from open position to full neck, drilling pattern recognition through randomized exercises
- Developed an interactive fretboard component in React rendering **144** clickable note positions with dynamic styling based on scale selection, position system, and display mode (notes, intervals, or dots)

Sanguine - *Java, JUnit, Java Swing* *Nov 2025 - Dec 2025*

- Programmed a pub-sub architecture in **Java Swing** that synchronizes two players in real time, triggering UI updates on each move
- Enabled runtime-selectable AI difficulty by designing 2 pluggable strategies using **Java** interfaces, swapping opponent behaviors without code modification
- Maintained **90%+** code coverage on model and strategy components by isolating unit tests with **JUnit** mock objects, catching edge cases in the controller, strategy, and model layers

Rubik's Cube Solver and Teaching Tool - *Python, PyGame* *Oct 2025 - Jan 2026*

- Implemented the Kociemba two-phase algorithm to compute near-optimal solutions (≤20 moves) in under 1 second, rendered in a real-time 3D simulator built with **Python** and **Pygame**
- Created a step-by-step teaching mode covering beginner method stages (cross, corners, second layer, top face, permutation), with state validation that blocks progression until each step is completed correctly

File System - *C, FUSE* *Nov 2025 - Dec 2025*

- Developed a **FUSE** driver in **C** that mounts a 1MB disk image as a persistent Unix-style filesystem supporting 128 files/directories
- Achieved full compatibility with standard Unix commands (ls, cat, rm, mkdir) by implementing block-based storage with inodes, free-space bitmaps, and POSIX syscalls via memory-mapped I/O