# Alexander Mcneilly

(847) 650-5488
alex-mcneilly.github.io
mcneilly [at] mit [dot] edu

*Education:*

**Massachusetts Institute of Technology (MIT)**

B.S. in Mathematics, Minor in Computer Science                     Aug 2022 — Dec 2026

Courses: *Software Performance Engineering, GPU Performance Engineering, Algorithm Engineering, Low-level Programming, Combinatorial Analysis, Analysis of Algorithms, Linear Algebra, Real Analysis, Probability*

*Work:*

**MIT Informatics Tournament (MITIT)** — Various Organizing Roles (Software, Marketing, Logistics)

*Sep 2023 — Dec 2025*

- ❖ Recruited and collaborated with software team handling 80,000+ submissions across 4 contests
- ❖ Created and set strategy for marketing team to support 4,300+ contestants from 90+ countries
- ❖ Secured $50,000+ in sponsorship from top firms, including Jane Street, HRT, Jump, and Citadel
- ❖ Transported and organized supplies for day-of competition, including food, decorations, equipment

**World Labs** — Software Engineering Intern

*May 2025 — Aug 2025*

- ❖ Worked on full-stack features for text-to-3D world model product, improved performance (Marble)
- ❖ Contributed to open-source Gaussian Splatting renderer for the web (SparkJS)
- ❖ Collaborated with Dr. Fei-Fei Li, Dr. Ben Mildenhall, and team on product decisions and UX

**Krea AI** — Software Engineer, Product

*Feb 2025 — May 2025*

- ❖ Launched text-to-3D products generating editable 3D scenes and objects from prompts
- ❖ Integrated GPT Image for scene rendering in text-to-3D scene editor
- ❖ Refactored and optimized real-time AI editor, improving UX during $8M to $13.5M ARR growth
- ❖ Co-led complete UI/UX redesign of core product suite, enhancing accessibility across 20+ million users

**MIT STEP Lab** — Research Assistant

*Jun 2024 — Aug 2024*

- ❖ Optimized shader code to boost render performance in 3D web-based simulation tool
- ❖ Increased supported geometries in simulations tenfold by improving import functionality
- ❖ Conducted cross-browser performance testing and benchmarking for consistent WebGL rendering