Tobias Berman

✉ tberman@umich.edu 🌐 tobiasberman.com in @tobias-berman @TobiasBerman

Education

University of Michigan **Ann Arbor, MI**

B.S. in Mathematics and Computer Science *Expected May 2026*

- **GPA:** 3.8/4.0
- **Relevant Coursework:** Data Structures & Algorithms, Computer Organization, Combinatorics & Graph Theory, Probability Theory, Real Analysis, Advanced Linear Algebra, Differential Equations, Classical Mechanics, Electromagnetism, Algorithm Design & Complexity Theory

Skills

Languages: C++, C, Python, Java, GLSL, HTML/CSS/JavaScript, ARM Assembly, LaTeX, Bash
Technologies: Git, Unix/Linux, Docker, Kubernetes, NumPy, Jupyter Notebook, Fusion360, Unity, Unreal Engine, Blender
Interests: Visual Art, Chess, Hiking, Physics

Projects

Financial Market Simulator

- Designed a C++ program to handle efficient order matching, priority-based execution, and real-time trade processing in a simulated stock market.
- Utilized STL containers and algorithms to handle 20,000+ orders/sec and optimize runtime performance.

Agent-based Collective Behavior Model

- Designed a real-time simulation of emergent flocking in JavaScript to model complex swarm behaviors such as separation, alignment, and cohesion.
- Utilized JavaScript libraries to optimize algorithm efficiency, simulating 1000+ autonomous entities concurrently at 60FPS.

Forum Post Classifier

- Designed and implemented a C++ program for automated Piazza post categorization, utilizing a bag-of-words model and a custom Bernoulli Naive Bayes Classifier to achieve 93% classification accuracy on large datasets.
- Optimized data structures and algorithmic efficiency, reducing inference latency by 30%.

Assembly Compiler, Linker, and Simulator

- Built a C-based simulator to execute machine code on a pipelined CPU architecture, incorporating branch prediction and data forwarding.
- Developed a custom assembly compiler in C to translate assembly instructions into object files.
- Implemented a linker to combine multiple object files into a single executable machine code file.

Perlin Noise Generation Algorithm

- Developed a Perlin noise generator in Python, creating procedural textures in under 50ms with optimized matrix operations.
- Utilized NumPy for efficient numerical computations and PIL for image generation and visualization.
- Integrated noise generation with OpenGL/C++ to visualize real-time terrain rendering.

Experience

Robotics Club **San Jose, CA**

Vice President / Engineer *Aug 2019 – Jun 2023*

- Developed and delivered technical training on embedded systems, Arduino programming, circuit design, and Fusion360 CAD, successfully onboarding new members and driving cross-functional collaboration.
- Led the end-to-end development and implementation of robotics systems, including designing and assembling a modular chassis and precision ball-launching mechanism.
- Leveraged iterative development and problem-solving methodologies to refine prototypes, ensuring system robustness under competitive constraints.