

Nathan Hutton

YouTube : www.youtube.com/@nathanhutton3895

LinkedIn : www.linkedin.com/in/nathanhutton/

GitHub : github.com/Nathan-Hutton

Phone : 801-560-3611

Email : nathan.d.hutton@proton.me

EDUCATION

University of Utah

Master of Computer Graphics and Data Visualization — GPA: 4.0

SLC, UT

Dec. 2025

Westminster University

Bachelor of Computer Science — GPA: 4.0

SLC, UT

May. 2024

- Applied mathematics minor.

- Received the outstanding computer science student of the year award.

Salt Lake Community College

Associate of Science — GPA: 4.0

SLC, UT

May. 2021

- Earned degree while in high school.

EXPERIENCE

Flight Safety International

Software Engineer Intern

Broken Arrow, OK

May. 2025 - Aug. 2025

- Helped integrate Phidgets API with RACKtangle hardware for real-time flight simulation I/O in C++.

- Worked exclusively on Windows & Visual Studio.

Idaho National Laboratory

Software Engineer Intern

SLC, UT

Feb. 2023 - May. 2024

- Debug and reformat C++, C, and Python code.

- Convert Matlab test harness to Python.

- Configure CI/CD pipelines.

- Implement, refactor, and debug GUIs.

- Worked exclusively on Linux.

Westminster University

Computer Science Tutor/TA

SLC, UT

Aug. 2022 - Dec. 2023

Westminster University

IT Technician

SLC, UT

Aug. 2021 - May. 2024

PROJECTS

- **Boids Flocking Simulation** [[Demo Video](#)]: OpenGL flocking simulation in C++. A simple ruleset by Craig Reynolds leads to complex flocking behavior. Uses ImGui, color blending in the HSB colorspace, and obstacle avoidance.
- **Solar System** [[Demo Video](#)]: Solar system simulation with dynamic physics in OpenGL/C++. Utilizes shadow mapping, HDR, environment mapping, verlet numerical integration, bloom, and more.
- **AQI Visualization** [[Demo Video](#)]: Interactive visualization for Utah's air quality index using Javascript and D3. I made 68% of all commits in this group project.
- **Volume Renderer** [[Demo Video](#)]: OpenGL/C++ volume renderer with modifiable transfer functions.
- **Squibbles**: 4-person team Unity project using AGILE. I made 54% of all commits and created the main gameplay loop, movement, online leaderboards, audio, AI, UI, menus, and more.
- **Ray Tracer**: Python Ray Tracer made with no reliance on third-party libraries.
- **Minijava Compiler**: Includes type-checking, lexing and parsing with ANTLR4, JVM assembly code generation, AST generation, flow control, package inclusion, and method calls.
- **Java Chatroom**: Terminal based chatroom utilizing java.net. Features include broadcast messages, private messages, user list retrieval, and notifications when users join or leave.

PROGRAMMING SKILLS

Languages: C++, Python, C, Java, C#, Javascript

Tech: OpenGL, Linux, Docker, Gitlab CI/CD, D3