Danny Le

+1 (xxx)-xxx-xxx | dannyle4.contact@gmail.com | linkedin.com/in/dannyle4/ | github.com/danny-tle

Education

University of Utah — Salt Lake City, UT

Bachelor of Arts in Computer Science — *Expected Fall 2026*

- **Relevant Coursework:** Data Structures, Algorithms, Data Wrangling, Computer Vision, Human Center Design, Engineering Probability & Statistics, Linear Algebra, Discrete Mathematics, Calculus II & III
- **Activities:** Technical Coding Club, Software Development Club

Experience

Web Developer — Spring 2026

Soup & Sip — *Remote*

- Collaborated with a new restaurant to design and develop a promotional website for their business opening using Next.js and Tailwind CSS
- Implemented responsive layouts using the Next.js App Router with reusable components, global styling, and modern navigation structure

Selected Projects

Manual Driving Simulator | *C++, Qt, Qt Creator*

- Developed a desktop-based manual driving simulator using Qt Creator, allowing users to control a vehicle in a simulated environment with real-time visual updates
- Implemented event-driven input handling through Qt's signal–slot system to process steering, acceleration, and braking interactions
- Structured the simulator using a model–view architecture to separate physics logic, rendering, and user input for maintainability
- Focused on realism and usability by refining motion behavior, screen updates, and control responsiveness

Snake Client & Score Server | *C#, .NET, MVC, SQL Server, TCP/IP, HTTP*

- Developed a networked Snake client that records live game and player data to a SQL Server database, designing relational schemas for games, players, timestamps, and max scores
- Built a lightweight HTTP web server from scratch to dynamically render game history and per-game statistics, implementing proper HTTP/1.1 headers, content length, and UTF-8 encoding
- Applied MVC architecture to cleanly separate networking, data models, database access, and presentation logic, improving maintainability and scalability

SpriteEditor | *C++, Qt, Qt Creator, qmake*

- Designed and implemented a desktop sprite editor supporting multi-layer editing, per-pixel drawing tools, onion skinning, frame-based animation, and animated sprite export
- Built an event-driven GUI using Qt's signal–slot architecture to manage real-time user interactions between tools, layers, canvas updates, and playback controls
- Structured the application using a model–view design, separating rendering logic, editor state, and UI components to improve maintainability and extensibility
- Configured and managed the build system with `qmake`, handling project configuration, resource files, and cross-platform compilation within Qt Creator

Skills

Languages: C/C++, Python, Java
Developer Tools: Git, Docker
Concepts: Software Engineering, Data Wrangling, Computer Vision