

Luke Lu

lukelu.dev | luke.yanglu@outlook.com | linkedin.com/in/lukeyanglu | github.com/LukeL0000

EDUCATION

Major in Computer Science, Bachelor of Science – University of British Columbia

Sept 2021 – May 2027

GPA: 3.85 | 83%

- Recipient of Dean's List Award for Fall 2021 and Winter 2022 semesters.

PROFESSIONAL EXPERIENCE

Software Developer Co-Op – D-Wave Quantum Inc.

May 2025 – Present

Tech Stack: Go, Python, TypeScript/JavaScript, Django (Django, Django REST Framework), React, Docker, MySQL, AWS (S3, ECS, ECR, Redis)

- Built the next generation of our Quantum Cloud (LEAP) Usage & Management platform's **API** and **Front-End** with our UI and Admin team using **Agile/Scrum** methodology. Actively developing new features, modernizing API endpoints & front-end components, and squashing bugs.
- Rebuilt legacy **Django API** to use **Django REST Framework** reducing lines of code per endpoint by **>75%**. Worked closely with both front and back-end developers to streamline and standardize requests & responses, reducing incoming requests and server load by **>30%**.
- Modernized legacy **React JavaScript** front-end components, reducing lines-of-code by **>50%** and WCAG accessibility issues by **>95%**. Rebuilt parts of front-end on more modern **MUI React TypeScript Components** implementing a **Material Design 3** and **WCAG 2.1** standard.

Full Stack Software Developer – Canadian Nuclear Laboratories

Sept 2023 – Apr 2025

Tech Stack: .NET (.NET Core, .NET MAUI), C#, JavaScript, HTML/CSS, MS SQL Server, SQL Server Reporting Services

- Reimplemented legacy data collection system as an Android application, supporting **>1,000,000 records** and saving **>\$25,000/yr** on external software costs. Built Android front-end on **.NET MAUI** and a **C# API** service interfacing with a self-hosted **SQL Server Database**.
- Created **ASP.NET Core** CRUD webapps for internal business needs from maintenance to fuel tracking, supporting **>10,000 internal records** and saving **>200hrs/yr** in administrative time. Implemented **MVC** architecture connecting **SQL Server** database, **C#** backend, and **Blazor** frontend.
- Redesigned schemas of self-hosted **SQL Server** databases, eliminated all referential integrity violations, reduced queries-per-report by **>50%** and stored procedure execution times by **>50%**. Normalized tables, created triggers, stored procedures, and SQL Server Agents.

RECENT SOFTWARE PROJECTS

AWS Bedrock Agentic Chatbot

Fall 2025

Tech Stack: AWS (Boto3, Bedrock), Streamlit, Python

- Designed and built a **chatbot** with **multi-agent** and session support for D-Wave internal use. Built back-end service using **Boto3** to access **Bedrock invoke_agent API**, handling user prompts over multiple sessions and agents and displaying it on a **Streamlit** front-end.
- Created multiple agents optimized for specific use cases, customized on various **models**, **agent instructions**, **guardrails**, and internal **knowledge bases** for best possible responses on use cases ranging from professional services to software development.
- Tested custom agent against professional services use cases, reducing time taken for evaluating potential clients from **>4 hours** to **<15 minutes**.

Indie Arcade Physics Game (Project Ramster)

Winter 2025

Tech Stack: C++, OpenGL, Box2D

- Built a fully playable **OpenGL** powered **C++** arcade physics game with **2D Graphics** and realistic physics powered by the **Box2D Physics Library** including gravity, inertia, and collision effects.
- Followed game development principles including a robust **Entity-Component System**, and an **inheritance model** that separates core gameplay systems from rendering to physics to control the behaviour of every aspect of our game.

LLM Code Interpreter

Summer 2024

Tech Stack: React/Express.js, JavaScript, HTML/CSS, SQLite, Docker, Ollama

- Built an **SQLite** and **LLM** driven **React/Express.js** code comprehension **web application** that compares user prompted code to original code.
- Implemented **Express.js API** connecting the user, LLM, and database. Seamlessly passed user prompts to **open-source LLMs** using a **Docker-Containerized Ollama API** to generate working code, storing results to **SQLite** database. Built front-end and dashboards using **React**.
- Deployed using **Docker**, ensuring both device compatibility and **ability to host on AWS ECS**.

Trip Organizer

Summer 2023

Tech Stack: Java, JDBC, Oracle Database, JavaFX

- Developed a **CRUD app** for managing day trips. Built a **Java back-end** connected to **Oracle database** via **JDBC** and displayed on **JavaFX**.

TECHNICAL SKILLS

Languages

- C++, C#, Java, Go, Python, TypeScript/JavaScript, SQL, HTML/CSS, XAML

Platforms, Frameworks, and Architectures

- .NET (.NET MAUI, ASP.NET Core), Django (Django, Django REST Framework), React.js, Node.js, Express.js, REST, MVC, Material Design 3, MUI

Libraries and Technologies

- AWS (Bedrock, ECS, ECR, S3, Redis, R53, Amplify), Boto3, Ollama, Docker, Git, OpenGL, GDB, JUnit, PyTest, Unix