Zhiyuan Li

+1-672-991-7903 markzhiyuan.li@mail.utoronto.ca hoshipu.top
Zhiyuan Li markli1hoshipu

Education

- **University of Toronto** — Toronto, Ontario
  Bachelor of Engineering Science & Robotics Specialization; Major GPA: 3.90 — *Sep 2022 – Apr 2027*

Experience

- **Prelude CA** — Toronto, Ontario
  Full-Stack Engineer / Frontend Lead (Part-time) — *June 2025 - Present*
  - Led the frontend architecture of a **multi-tenant SaaS platform** using **Next.js 15 (App Router)**, **React 19**, and **TypeScript**, delivering a highly responsive user experience. Built a modular design system with Radix UI and Tailwind CSS, implemented global state management via **Zustand**, and optimized performance through **client-side state caching** and **cookies**, reducing interaction latency by **40%**.
  - Implemented secure authentication and authorization for workspace-isolated tenants, using **OAuth integration** (Google/Microsoft), **role-based access control**, and team invitation flows. Designed multi-tenant isolation across application and data layers, ensured safe **session persistence**, and established **type-safe API contracts**.
  - Designed an AI-driven pipeline enabling automated, **multi-agent data processing** workflows. Built an intelligent **CSV ingestion system** with semantic column matching using **OpenAI embeddings**, achieving **90%+** auto-mapping accuracy.
  - Designed a scalable **CRM relational schema** using **PostgreSQL**, unifying **customers, leads, and deals** into comprehensive user profiles. Ensured efficient query performance and adherence to best practices through **normalization**, **indexed key columns**, **foreign key constraints across multiple tables**, and **transaction-safe updates** supporting real-time validation.

- **Yuhang International Flight(Dalian) (YIF)CO.LTD** — Dalian, China
  Full-Stack Engineer — *July 2022 - May 2024*
  - Designed and developed a **online information management system** using Python, QtDesigner, SQL, and QML to track and manage IOU payments. The system enabled users to **access, update, edit, and output complex information online** via an SQL database, significantly reducing manual input and improving the efficiency of tracking unpaid IOUs.
  - Utilized the MySQL library in Python to establish a robust connection between the application and an SQL database, leveraging **connection pooling for multiple concurrent users** to enhance performance. Enhanced **user authentication by hashed passwords** with the bcrypt library, enforcing security protocols and preventing unauthorized database access.
  - Implemented advanced **multi-key search capabilities** using SQL JOIN, INDEX, and LIKE statements. Developed a fuzzy search functionality using the **FuzzyWuzzy library to accommodate vague user queries**, enabling partial matches and ranking search results based on relevance scores. Integrated **real-time updates with SQL triggers** and after, insert events to automatically refresh data upon changes.
  - Leveraged **pandas and numpy libraries to connect to EXCEL** and other platforms, enabling users to import IOUs from pre-recorded tables or their primary eTerm system. Additionally, developed a feature to **generate visual summaries through graphs**, depicting IOU amount changes over time, and included user-friendly features like **history view and personalized configurations**.
  - Conducted **rigorous system testing**, including handling over **1 million items** in the database and supporting over **100 concurrent users**. Fixed critical bugs related to overloading that previously led to application crashes. The optimized application efficiently processed a database containing over **100k IOUs**, allowing fast and accurate search within **3-5 seconds**, improved operational efficiency by more than 100%.

- **UofT Robotics Association** — Toronto, Canada
  Software Engineer — *Sept. 2023 - July. 2024*
  - Developed and optimized an algorithm for the real-robot Pacman game using **C++, Python, and PyGame**, focusing on **maximizing surviving time and pathfinding generation**. This involved fine-tuning the algorithm to reduce processing time while ensuring accurate and optimal decisions based on **real-time sensing** for character movement, significantly enhancing the overall gaming performance.
  - Utilized **graph theory** to abstract the Pacman maze and implemented advanced pathfinding techniques such as **backward tracking**, allowing the AI to reverse steps for optimal paths, and **Euclidean distance estimation** to calculate the shortest path in a grid-like environment. Combined with **A* and greedy algorithms for point collection**, these techniques enabled efficient navigation even in complex mazes, greatly improving Pacman's ability to avoid enemies and collect rewards.
  - Focused on integrating sensor data by implementing **discrete and continuous time conversion for real-time sensor data** with pathfinding and dynamic adjustments. Calibrated and fixed parameters between sensor readings and real-world values, and incorporated **collision detection**. The final AI implementation exhibited a more intelligent and responsive behavior, improving Pacman's navigation accuracy by **30%** through effective sensor integration.
  - Applied principles from **game theory**, focusing on **minimizing risk** and **maximizing reward**. Leveraged advanced algorithms like **minimax** with alpha-beta pruning to predict enemy behaviors, significantly enhancing Pacman's decision-making abilities. Developed two strategy sets for the robot: one for survival under pursuit, and another for maximizing points by collecting coins. This led to an increase in survival time by over **20%** and a **50%** increase in total score.

Personal Projects

- **Desktop mechatronic robotic pet**:
  - Developed a mechatronic robotic pet, capable of **multi-modal interaction** through **voice commands, visual recognition, and electronic wireless signals**. The design combined mechanical engineering, embedded systems, and software design to create a responsive and interactive pet as an entertainment for the user.
  - Developed the **embedded system** using **ESP32-WROOM** and **Arduino IDE** to implement **C++**-based control algorithms for multi-state behavior management. Achieved **asynchronous event-driven reactions** by integrating a **multi-threaded** design, ensuring seamless responsiveness to multiple inputs. Extended functionality through a **Python API** for website integration, enabling remote control and monitoring.
  - Engineered the hardware system for **power distribution**, ensuring optimal operation of motors, speakers, LEDs, and micro-controllers via circuit design. Designed the **CAD model** for the robot's outer shell, focusing on structural integrity and aesthetics to deliver a compact and visually appealing design.
  - Optimized the system's performance by implementing **low-power operation modes** and refining the interaction experience, providing a robust proof of concept for consumer-grade robotic pets with interactive features.
- **Delivery turtle-bot**:
  - Designed and developed a **delivery robot** prototype capable of navigating a test environment, recognizing its current position using a **color camera**, and following a pre-defined floor path to stop at designated locations for delivery tasks. The entire system is developed with turtle-bot and Linux system, using a combination of **ROS and Python**.
  - Implemented **PID control** for precise path-following using white tape as a visual guide on the floor. Conducted rigorous testing and parameter tuning to minimize **overshoot** and **steady-state error**, ensuring smooth and reliable navigation in various test scenarios.
  - Integrated **Bayesian localization** to enable self-localization within the test environment using visual landmarks captured by the robot's camera. This allowed the robot to **reorient** itself and return to its designated path after starting from any position or encountering deviations.
  - Delivered a fully functional prototype that demonstrated robust path-following and delivery task completion, showcasing adaptability and reliability in simulated real-world conditions.
- **Research Data Process software**:
  - Developed an application to process scientific data and **evaluate statistics for research analysis** needs, incorporating an **AI-driven summary feature** that generates insightful responses based on the dataset characteristics.
  - Implemented the frontend using libraries such as **SciPy, NumPy, and pandas** to handle complex data processing tasks. The application evaluates key statistical metrics, including **standard deviation, covariance, and gradient graphs**, allowing researchers to visualize data trends and relationships effectively.
  - Connected the backend to the **ChatGPT API** to enable automated generation of insightful responses and analyses based on user queries related to the dataset. Employed **robust error handling and API management techniques** to ensure seamless communication between the application and the AI service.
- **HTTP Testing Software**:
  - Designed and implemented an **HTTP testing application** using PySide6 for the GUI and the requests library. The application enables users to **configure and simulate HTTP requests** (GET, POST, PUT, DELETE) while managing custom headers dynamically.
  - Utilized **Qt Designer** to create a user-friendly interface, allowing users to input the request URL, method, headers, and body. Key components include a **method selection box**, an editable URL field, and a table for managing headers, facilitating easy request configuration.
  - Implemented robust request handling with efficient connection management via **requests.Session**. Integrated error handling using **traceback** to provide detailed feedback on request failures.
  - Added **automation testing capabilities**, enabling users to execute predefined test cases automatically. Users can save configurations for batch processing of requests, **streamlining the testing and debugging of web services**.

Awards

- **Putnam Math Contest**: Personal score 42 and Rranking **Top 154**. **Top 3** member of Top 5 Winning Team.
- **CEMC Math Contests**: Euclid 2-time Honor Roll (Rank 77), Hypatia, Fermat, CSMC Honor Roll (Rank 100)
- **AMC/AIME Math Contest**: AMC 2-time Honour Roll(Top 0.1%), AIME 2-time Honour Roll (Top 0.1%)
- **Canadian Mathematics Olympiad**: Invited participant.
- **University of Toronto Undergrad Mathematics Contest**: **Top 1** among all participants from all years, all campus.

Programming Skills

- **Languages/Frameworks**: C, C++, Python, ROS/ROS2, Java, C#, Javascript, Three.js, React.js, Node.js, OpenCV, Langchain, Tensorflow, PyTorch, Keras, NumPy, scikit-learn, OpenAI, Unity, HTML, CSS/SCSS, Swift, MATLAB, R, Shell Scripting, LaTeX
- **Development Toolkit**: Linux Shell, Git, Docker, Vim, Jupyter Notebook, XCode, Compilers