

# 李昊喆

+86 133-4581-1880 | lihaozhe013@gmail.com | Portfolio | lihaozhe013 | lihaozhe013

## 教育背景

### University of Toronto

B.A.Sc IN COMPUTER ENGINEERING (计算机工程工学学士)

Toronto, Ontario, Canada

2023 年 9 月 - 2028 年 (预计)

**相关课程:** Algorithms & Data Structures, Computer Hardware, Operating Systems, Introduction to Databases, Computer Organization, Software Design & Communication, Communication Systems, Digital Systems

## 专业技能

编程语言	C, C++, TypeScript, JavaScript, Java, Go, Python, RISC-V Assembly, MATLAB
开发工具	Git, Bash, VS Code, Cursor, Vim, Chrome DevTools, Makefile
框架	Node.js, React.js, Spring Boot, Vite, Tailwind CSS, GTK, Electron Framework
DevOps & Cloud	Nginx, GitHub Actions, Docker, AWS
其他工具	Figma, Microsoft Office, LATEX, Google Workspace
硬件/仿真	Simulink, FPGA board, Verilog, LTSpice, Quartus Prime, ModelSim, DESim

## 工作经历

### 全栈工程师 (兼职)

杭州沐宇风科技有限公司

远程

2025 年 9 月 - 现在

- 开发并维护了扩展开源核心的专有企业模块, 同时实现了服务器端日志记录和警报机制, 用于监控系统健康状况并最大限度减少停机时间。
- 构建了 AWS (Pre-Prod) 和阿里云 ECS (Prod) 的混合云基础设施, 通过 Docker 和 PM2 确保公司核心交易平台的高可用性与资源效率。
- 通过 GitHub Actions 设计并实施了持续集成/持续交付 (CI/CD) 管道, 实现了测试、构建流程及 Docker 镜像发布的自动化, 从而缩短了部署周转时间并优化了发布周期。
- 通过配置 Nginx 作为反向代理并启用自动 SSL 证书续期 (Certbot), 强化系统安全性, 确保企业用户数据传输符合 HTTPS 规范。

### 前端开发工程师 (实习)

杭州亿格云公司

杭州, 浙江, 中国

2025 年 5 月 - 2025 年 8 月

- 在一个 7 人前端团队中使用 TypeScript、React.js 和 Ant Design, 基于 Electron 的企业网络安全桌面应用开发了新的前端功能, 并为管理员 web 控制台实现了 UI 功能。
- 使用 Cursor (AI IDE), 通过应用先进的 Prompt Engineering 优化编码工作流程并提升软件质量。
- 利用高级 Git 工作流程和 GitHub PR 管理代码集成到 CI 流水线, 成功解决了 QA 和预生产环境中的合并冲突。
- 使用 DevOps 和 CI/CD 流水线进行测试, 在 Pre-Production 环境中自测代码以确保可行性和稳定性, 成功为 SaaS 版本贡献功能。

### AI 实验室研究助理 (实习)

深圳大数据研究院

深圳, 广东, 中国

2024 年 6 月 - 2024 年 8 月

- 通过编写脚本实现一次性清理和重新安装 Conda 环境及关键软件包 (PyTorch/TensorFlow) 的自动化研究环境, 减少了设置时间, 使团队能够立即在空闲计算资源上运行新模型。
- 在实验室计算机上重新安装了 Ubuntu 和 Debian 系统, 以修复受损的软件环境, 并通过将 SSH 与研究团队的 Web 控制台集成, 配置了无缝模型部署工作流程, 修复了许多研究团队无法用于实验的计算机。
- 为研究团队开发了综合指南并配置了运行时环境, 以便运行一些用于参考的开源模型, 减少了研究团队在配置上花费的时间。

# 个人项目

---

## 贸易流管理系统

个人项目

TypeScript, Node.js, SQLite, React.js, Vite

2025 年 7 月 - 现在

- 构建了采用模块化单仓库结构的全栈式交易管理解决方案, 具备基于 JWT 的 RBAC 认证机制及 i18n 支持, 以实现订单与财务管理的无缝衔接。
- 采用 **Node.js** 和 **Express** 构建高性能后端系统, 运用 **Decimal.js** 实现精准金融计算, 并借助 **ESBuild** 优化构建产物。
- 采用 **React** 和 **Ant Design** 开发响应式前端, 采用组件驱动架构以加速功能交付并保持代码复用性。

## GIS Route Optimization Application

UNIVERSITY OF TORONTO

C++, GTK, Git, A\*, Dijkstra

2025 年 1 月 - 2025 年 4 月

- 在课程项目的三人小组中, 使用 **C++** 和 **GTK** 在 Mate 桌面环境下开发了一个 GIS(地图) 桌面应用, 实现了地图渲染、地名搜索、最短路径和多停点路径查找功能。
- 利用 **Multithreading** 预处理 4GB 原始坐标数据, 将其转换为结构化的点、线和多边形格式, 以便在 16 核测试机器上在 50 秒内完成从数据解析到画布渲染的全部操作。
- 实现了包括 **A\*** 的最短路径计算算法, 以及 **Dijkstra**、**Multi-start Greedy** 和 **Simulated Annealing** for Multi-Stop Route Optimization, 实现了技术评估分数的 90%。
- 通过使用 TomTom API 并整合 **libcurl** HTTP 模块, 实现了实时交通可视化, 在导航期间动态显示交通拥堵, 提升路线规划效果。
- 维持干净的 **Git** 工作流, 采用功能分支, 并使用基于 Makefile 的构建脚本来简化协作, 确保团队高效开发。

## StreamFile Server

个人项目

Go, Gin, TypeScript, Video.js

2024 年 6 月

- 开发了一个基于 Go、无数据库的静态资源托管服务器, 提供私密链接生成、文件上传、HTTP Range 支持和文件搜索功能。
- 实现了前端功能, 如 Markdown 渲染、视频/音频播放、静态网页托管。.
- 使用仅 **Tailwind CSS** 和原生 HTML DOM 优化大部分前端组件, 使用 **ESBuild** 进行 JavaScript 压缩, 打造轻量级前端, 大幅减少低配置设备的页面加载时间。

## Runner Game (FPGA Board Game)

UNIVERSITY OF TORONTO

C, RISC-V Assembly, CPUlator

2025 年 5 月

- 在 **DE1-SoC FPGA Board** 上使用 **Verilog** 开发了一个 2D Runner Game, 实现核心游戏逻辑、PS/2 Keyboard、音频组件和 VGA 组件, 并在三周内交付了可玩的游戏。
- 集成了 PS/2 Keyboard 以实现实时用户控制, 利用 **CPU Interrupt** 实现低延迟输入处理。
- 构建了一个轻量、无 OS 的 2D 引擎, 以 60 帧每秒渲染彩色图像、动画和文字。
- 开发了一个基本的 Square Wave Synthesizer, 在游戏事件 (如得分和完成游戏) 中实现动态音效。

## Greedy Mouse Game (FPGA Board Game)

UNIVERSITY OF TORONTO

Verilog, FPGA Board, ModelSim

2024 年 11 月

- 在 **DE1-SoC FPGA Board** 上用 **Verilog** 设计并实现了一个 2D Greedy Mouse Game, 具备 PS/2 Keyboard 输入和 VGA 图形输出功能。
- 开发了一个 PS/2 Keyboard Controller 以实现实时用户输入, 并将其与有限状态机 (FSM) 和片上存储模块集成, 用于驱动 VGA 输出, 实现位图动画的流畅显示并增强游戏视觉效果。
- 构建了一个轻量、无 OS 的 2D 引擎, 以 60 帧每秒渲染彩色图像、动画和文字。
- 使用 **ModelSim** 和 **DESim** 进行早期阶段的仿真和调试, 最终使用 Quartus Prime 完成综合并部署到 FPGA Board, 并进行了反复优化。