

Alex Xu

👤 me.alex-xu.site — 📞 647-674-3251 — ✉ alex.zihao.xu@gmail.com — 💻 alex-xu-912bb528b — 🌐 github.com/AlexZihaoXu

Education

University of Waterloo

2023 – 2028

Computer Engineering, Honours, Co-operative Program

Skills

Languages: JavaScript, TypeScript, Python, Java, C++

Web: Docker, Socket.IO, Tailwind, Bootstrap

Fullstack: React, NextJS, Django, Flask, Firebase, Nginx

Game Dev: Unity, Three.js, OpenGL

Cloud & DBs: GCP, AWS, OCI, MongoDB, PostgreSQL

Developer Tools: IntelliJ IDEA, PyCharm, VS Code, GitHub

Work Experience

BitGo Inc – Palo Alto, California, USA (Remote)

May 2025 – Aug 2025

Software Engineering Intern

Grafana, Mongoose, TypeORM, Node.js, TypeScript, JavaScript, Jira

CaaS (Crypto as a Service) Team

- Collaborated with 3 interns to refactor major **Wallet Platform** codebase by **converting JS files to TS**, sharing experiences and leveraging an **internally built AI-assisted conversion tool** to accelerate repetitive cases.
- Improved backend performance by **identifying and optimizing slow queries** using **Grafana, Loki, and related tools**; applied fixes such as parallelizing async calls and replacing inefficient algorithms to reduce event-loop bottlenecks.
- Migrated database layer from **Mongoose v6 → v7 → v8**, where build errors exceeded **200+** and CI/CD failures surpassed **50**. Collaborated with 3 colleagues & followed the official migration guides, prepared backward-compatible changes in separate PRs first, then performing version bumps and test fixes incrementally for easier & faster reviews.
- Wrote a script to **generate coverage reports in a specific format**, allowing quick check for the WP detangling task.

Go Accounts Team

- Upgraded **TypeORM and TypeScript versions** with a co-worker, applying the same migration approach: make backward-compatible changes, then introduce version bumps and resolve errors in segmented PRs to ensure stability.

Apollo Intelligence Inc – Toronto, Ontario, Canada (Remote)

Sept 2024 – Dec 2024

Fullstack Developer & Designer

NextJS, Supabase, React, Node, Python, NextUI, MUI, RESTful

Apollo Agents

- Reached the **Y Combinator interview stage**, an achievement with a success rate of **less than 1%**.
- Built a chat interface using **React.js** and **Blockly**, hosted services over **Python**, and **Supabase**.
- Designed a **recording-based workflow system** that translates user's recorded browser actions to **VPL**.
- Integrated and **fine-tuned** models on **OpenAI & Claude's API**, **reducing token output by 50%** with higher stability.

Apollo Learn

- Developed the frontend of an **AI-powered** education platform using **NextJS** and **TS**, integrating **NextUI** and **MUI**.
- Implemented assignment submission, grading, feedback functionalities with **Supabase & AWS** for backend.
- Created an **AI-grading system** that analyzes rubric and provides feedback to student automatically, saving 70% time.
- Finished the entire workflow of an **Auth System**, including but not limited to Sign Up and Password Reset.
- Developed a **replay system** to track assignment edits for anti-plagiarism detection and originality verification.
- Deployed the platform for real-world use by a Germany class in **Queen's University** during **Sept–Dec 2024**.

Projects

Lobbify (Multiplayer Platform)

Sep 2024 – Present

<https://lobbify.games/>

Electron, React, HeroUI, Next.js, Supabase, Fastify, WebRTC, Cloudflare

- Built a **lobby-based multiplayer platform** for **Minecraft** where users can **create public rooms** and others can seamlessly **discover and join** games. Utilizes the **LAN game system** with **peer-to-peer networking** over **WebRTC**, with automatic fallback to **Cloudflare TURN** when direct connections cannot be established.
- Designed and implemented a **custom protocol** to port forward LAN games from hosts to participants, with **buffering** and **Brotli compression** to reduce data size by **around 20%** while keeping CPU and latency overhead negligible.
- Developed the client using **Electron React Boilerplate**, with **React** and **HeroUI** for modern and intuitive UI/UX.
- Built the website with **Next.js** for improved **SEO** and **fast server-side rendering**, temporarily deployed on **Vercel**.
- Used **Supabase** for **authentication** and **database management** (Postgres).
- Implemented a **Fastify backend** with **rate limiting** and **Cloudflare proxy** for security and protection against abuse.
- Tested with 30+ users, receiving consistently **positive feedback** from friends and early users.

Minecraft Remake in Java

Oct 2021 – Jan 2022

https://github.com/AlexZihaoXu/Minecraft_AE

Java, LWJGL, GLFW, OpenGL, GLSL

- Developed a complete remake of **Minecraft** from scratch with **Java & GLFW**. **ALL ASPECTS** of the project were **self-taught** using resources like LearnOpenGL. No external code was copied from YouTube or other GitHub repositories.
- Built a custom rendering engine with **OpenGL** and **GLSL** shaders on **LWJGL**.
- Implemented **terrain generation** using **Perlin noise**, optimized with **multi-threading** for improved performance.
- Added a **lighting system** using **Breadth-First Search (BFS)**.

Self-Balancing Robot

Aug 2025

Personal Project

ESP32, TMC2208, NEMA 17, MPU6050, PID, UDP, Sensor Fusion, Fusion 360, 3D Printing

- Built a **two-wheeled self-balancing robot** powered by an **ESP32**, dual **TMC2208 drivers**, and **NEMA 17 stepper motors**, with motion sensing from an **MPU6050**.
- Implemented a **two-layer PID controller**: one for precise **angle stabilization**, another for **velocity control**, enabling fast recovery (<2s) to its original position after strong pushes.
- Added **sensor fusion** with **Kalman** and **complementary filters** for accurate orientation estimation.
- Developed custom **UDP protocol** for real-time **Wi-Fi remote control**.
- Integrated **fall and collision detection** with automatic motor cutoff to ensure safety and prevent hardware damage.
- **3D modeled and printed** the robot frame using **Fusion 360**, and **soldered** all electronic components to create a fully custom hardware assembly.

Minecraft Remake in C++

Oct 2022 – Oct 2023

<https://github.com/AlexZihaoXu/minecraft.cpp>

CMake, C++, GLSL, STB, OpenGL, GLFW

- Built the entire rendering engine with **OpenGL** and **GLSL** shaders on the **GLFW** library.
- Implemented **meshing system** that only adds visible faces to the geometry, greatly **reduces VRAM usage**.
- Tested using **Vanguard**, ensuring **no memory leaks** during runtime.

Ice Knight (Game)

Feb 2025

<https://me.alex-xu.site/unity/ice-knight/>

Unity, C#

- Collaborated with **2 other developers** during UWaterloo Game Dev Club W2025 Game Jam in 3 days.
- Implemented multiple core game features including but not limited to **player control**, **enemy AI**, and **projectiles**.
- Created and worked on various **VFX** such as **camera shake**, **particles**, **shadow casting**, **2D lighting** and some post-processing effect such as **glowing**. **Drew** and **animated ALL entities** in the game.

Party Physics (Game) (Grade 12 Computer Science Final Project)

Oct 2022 – Jan 2023

<https://github.com/AlexZihaoXu/PartyPhysics>

Java, Java2D, Dyn4J, Socket, TCP

- Developed a **multiplayer game** similar to **Stick Fight** from scratch, fully cross-platform.
- Engineered a **custom active rendering framework** in **Java Swing** with **anti-aliasing**, **particle system**, and **interpolated animations** for smooth physics at a lower tick rate.
- Built multiplayer functionality on a **custom TCP networking framework** with an abstraction layer for efficient **serialization** and **deserialization**, supporting **300+ moving objects** with **sub-10 ms latency**.
- Integrated real-time physics simulation with the **Dyn4J** library, **minimizing network latency conflicts**.

SoulKnight Clone

Feb 2020 – Mar 2020

Personal Project

Python, Pygame

- Created a tool to **edit animations** and **adjust frame offsets** for extracted sprites.
- Developed a **SoulKnight remake** in **Pygame**, optimizing rendering with a **caching system** for **5x more framerate**.
- Designed and integrated **randomly generated dungeons** and **enemy AI**.

Robotic Arm Project

Sept 2023 – Nov 2023

University of Waterloo – ECE 1A – ECE 198

- Developed a robotic arm from scratch for the **ECE 198** project using the **STM32 Nucleo** board, programming the system and UI in **C** for real-time control. **Achieved a final course grade of 94%**.
- Designed and refined the robotic arm using **Fusion 360**, optimizing the model through **5 iterations**.
- Created a custom gear set to enhance torque transfer, improving **stepper motor** efficiency. Designed a box-shaped control unit with an integrated wind tunnel for **improved cooling**.
- Implemented an **asynchronous I2C LCD driver** and configured an **ADC DMA joystick**, enhancing system responsiveness and preventing process blocking.
- Utilized **spring physics** to **smooth** arm movement, reducing vibration and instability during operation.
- Implemented a **reverse-kinematics** system, abstracting a **coordinate system** to achieve easier control.

D2L Backup Utilities

Mar 2023 – Apr 2023

Personal Project

Python, Django, JavaScript

- Leveraged **Django's ORM** to model the course structure on **D2L**, utilizing **Django's Admin features** to manage and filter course content effectively, including sections and files.
- Developed a **GraphQL API** using **graphene-django** to enable advanced and flexible data queries and mutations.
- Designed a **JavaScript** program that autonomously analyzes the structure of a D2L course from the Content page, outputting the structure with downloadable links in a nested **JSON** format.
- Created a **Python script** that utilizes nested **JSON** data generated by the JavaScript tool to download and populate all course data into the **database** via **Django's ORM**.