

# Alan Yuan

[alan.yuan.jly@gmail.com](mailto:alan.yuan.jly@gmail.com) | [linkedin.com/in/jalnyn](https://www.linkedin.com/in/jalnyn) | [github.com/jalnyn](https://github.com/jalnyn)

## WORK EXPERIENCE

---

### Amazon

June 2023 – Aug 2023

*Software Developer - Intern*

*Toronto, Ontario*

- In progress

### Amazon

May 2022 – Aug 2022

*Software Developer - Intern*

*Vancouver, British Columbia*

- Engineered a microservice in **Java** to send notifications to **100mil+** customer of cashback on select products
- Ensured modularity by designing a plugin system for processing the customer orders for microservice
- Utilize **AWS** webservice such as **Lambda**, **SQS** and **SNS** to ensure scalability of the notification system

### Intel

May 2021 – May 2022

*Software Engineer - Intern*

*Toronto, Ontario*

- Developed flagship product using **C++**, **Python** and **Bash** for speedup by re-routing the compilation
- Developed support software to generate 4000+ of completely random test-cases for edge-case testing
- Optimized support tool's Ram templates to reduce false positives and failing cases by around **70%**
- Implemented various new features and upgrades such as re-scripting tools to utilized new control system, dynamic database size notifier, hierarchy re-router for missing entities, automatic parameter setting aggregator and more
- Migrated all 600+ failing regression test cases to the new compilation flow leading to decreased build failures

### Centivizer

April 2020 - Sept 2020

*Software Developer - Part-time*

*Toronto, Ontario*

- Designed and wrote backend application using **Node.JS** and **SimplePeer** to connect users via video call
- Establish communication between client and backend for video feed using **socket.io**
- Integrated video feature with user database through **RESTful API** using the **Axios** Library

## EDUCATION

---

### University of Toronto

— cGPA

*MScAC*

*Sept 2023 - Jan 2025*

### University of Toronto

3.84 cGPA

*BSc Computer Science Specialist, Major in Mathematics*

*Sep. 2018 – May 2023*

**Relevant Coursework:** Data Structures and Algorithms (A+), Operating Systems (A+), Parallel Programming (A+), Neural Networks and Deep Learning (A+), Intro to AI (A+), Algorithm Design, Analysis & Complexity (A)

## UNDER-REVIEW

---

**J. L. Yuan**, Z. Zhou, K. Darvish, X. Zhou, A. Mandlekar, A. Garg, (Under Review), *CoRL 2023*

## PUBLICATIONS

---

M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, **J. L. Yuan**, R. Singh, Y. Guo, H. Mazhar, A. U. Mandlekar, B. Babich, G. State, M. Hutter, A. Garg, "ORBIT: A Unified Simulation Framework for Interactive Robot Learning Environments", *IEEE Robotics and Automation Letters (RA-L) 2023*

## PROJECTS

---

### PAIR lab assistant | Private repo

Sep 2021 – present

- Collaborate with **Nvidia** to build on top of Isaacsim creating a robot reinforcement learning **benchmark**
- Implementing a task-flow and environment randomizer for **causal reward** and **long-horizon task** based research
- Creating physics scenes and testing **reinforcement learning** algorithms to be used as a benchmark
- Setup and trained a variety of robots including **frankas** and **mobile manipulators**

### Tenant-Landlord Matching App | Links: server-side, client-side

Aug 2020 – June 2021

- Fullstack development of an mobile application to match landlords and tenants

- Constructed front-end using **React Native** and common packages such as **React Navigation** and **axios**
- Features: Authentication, Images upload utilizing **multer**, Tinder-like swiping, instant messaging with **Socket.io**
- Utilized **Node.js**, **GraphQL**, and database **Postgres** to construct backend

**CaNetDa: Deep learning for GeoGuesser in Canada** | Link: [GitHub](#)

Jan 2021 – April 2021

- Utilized a deep learning approach utilizing multiple deep learning techniques to have an AI play GeoGuesser.
- Utilized **ResNet**, **EfficientNet** and **Vision Transformer** to predict the location
- With our approach, a accuracy of **60%** was consistently achieved

**Tron UDP multiplayer** | Link: [GitHub](#)

Sep 2019 – Dec 2019

- Created a four player game for local networks using the **UDP** network protocol and C++
- **Forked** timer from the server to ensure the game runs on time
- Utilize **epoll** for both client and server to monitor the socket as well as the timer (server) and stdin (client)

**BF-interpreter** | Link: [GitHub](#)

Mar 2018 – Nov 2018

- Built **interpreter** that runs BF in C (runs all BF programs from [wikipedia](#))

## TECHNICAL SKILLS

---

**Languages:** Python, C/C++, JavaScript, Java, C#, R

**Tools:** Git, React Native, Node.js, MongoDB, SQL (Postgres), PyTorch, Numpy, Pandas, GDB, GraphQL