Jasjaap Dua

📞 +91 8630045875   ✉ jasjaap.dua@mail.utoronto.ca   🔗 linkedin.com/in/JasjaapDua

Education

University of Toronto   Toronto, ON
Bachelor of Applied Science in Computer Engineering + PEY   *September 2020 - June 2026*
Intended Minor in Artificial Intelligence Engineering

Relevant Coursework: Data Structures & Algorithms, Operating Systems, Introduction to Databases, Software Communication & Design, Introduction to Machine Learning, Computer Networks I, Digital Systems, Computer Organisation, Introduction to Control Systems

Skills

Languages: C/C++, Java, Python, SQL, MATLAB
Platforms/Technologies: Git, PostgreSQL, Unix/Linux, GNU Debugger, NumPy, Sci-kit Learn, PyTorch
Technical: Agile development, object oriented programming, debugging

Experience

GIS Application Team Technical Lead   January 2023 – April 2023
Software Communication & Design   *University of Toronto, ON*
- Led a team as **Technical Lead** to design and implement a Geographic Information System in **C++** by using the **OpenStreetMap Database**.
- Employed **agile methodologies** to deliver milestones, ensuring iterative and incremental progress.
- Developed backend functions for efficient data handling and computation and frontend elements using **Glade, GTK and EZGL** for visualisation of terrain, streets, buildings, and points of interest.
- Implemented efficient pathfinding algorithms **(Dijkstra and A*)**, achieving route calculations in **under 40 milliseconds**.

ARM-A9 FPGA Platforming Game Developer   January 2023 – April 2023
Computer Organisation   *University of Toronto, ON*
- Collaborated in a two-person team, ensuring effective communication and dividing tasks to successfully create an **ARM-A9 FPGA** version of the platforming game Celeste (PICO-8).
- Developed a **C++ parser** to convert **ImageMagick** files to C-arrays, enabling efficient buffering of sprites onto the **VGA display**, optimising graphical performance.
- Integrated support for **PS/2 keyboard** and on-board FPGA buttons using **polling** and **interrupt** methods to control the in-game sprite, ensuring smooth user input and interaction.
- Engineered robust in-game collision detection and handling for walls and obstacles, achieving seamless sprite respawns upon in-game death to maintain continuous gameplay.

File Transfer Application Developer   January 2024 – April 2024
Computer Networks I   *University of Toronto, ON*
- Designed and implemented a **client-server** file transfer application in **C** using **UDP sockets** for communication.
- Developed robust error handling with **ACK and NACK mechanisms**, ensuring **reliable data transmission**.
- Simulated packet loss using random number generation, enhancing the application's resilience to network conditions.

- Employed **GNU debugger** for thorough debugging of both client and server applications, ensuring **stability and performance**.

Library Code Development   September 2023 – December 2023
Operating Systems   *University of Toronto, ON*
- Created a custom implementation of the **ps** command for **Unix** systems, enhancing process management capabilities.
- Developed a **process manager** using **fork** and **exec** system calls, including a **subreaper** for zombie process management.

- Built a **concurrent, thread-safe hash table** using **mutexes**, experimenting with different levels of granularity for performance optimisation and conducting performance analysis across single and multiple-core CPU environments.

Machine Learning Algorithms   January 2024 – April 2024

Introduction to Machine Learning   *University of Toronto, ON*

- Developed and evaluated a robust **linear regression** model and **Perceptron** algorithm using **NumPy** and linear algebra, benchmarked against **Scikit-Learn's** implementation to predict diabetes progression.
- Implemented a **feedforward neural network** using **PyTorch** for image classification on the notMNIST dataset, including custom data loading, model training, and accuracy evaluation.
- Implemented and evaluated **K-means clustering** using **PyTorch** and **Scikit-Learn** on 2D data, involving data preprocessing, model training, and performance comparison.