Tyler Forgione

Montreal, Quebec | tyler.forgione26@gmail.com | (514) 594-4065 | tylerforgione.github.io
linkedin.com/in/tyler-forgione | github.com/tylerforgione

Education

McGill University, BSc in Computer Science — Sept 2025 – May 2027 (Expected)

McGill University, BSc in Honours Psychology, Minor in Computer Science — Sept 2022 – May 2025

- GPA: 3.97/4.00

Experience

Counsellor, Summit School – Saint-Laurent, QC, CA — July 2023 – August 2023

- Worked with special needs children of age 10 to 13
- Designed games and activities to keep children occupied for the whole day

Research Assistant, Otto Lab – McGill University — Sept 2023 – Sept 2024

- Designed and implemented a web-based psychological paradigm using the MERN stack, enabling efficient measurement of time-on-task effects on affect and performance.
- Conducted behavioral data collection and analysis for a strenuous-task experiment, contributing to an honours prospectus examining affective and performance changes over time.

Projects

Latent Semantic Analysis — github.com/LSA

- Implemented my own version of Latent Semantic Analysis in Python (writeup on GitHub)
- Trained on Wikitext-103, a dataset of over 100 million tokens
- Tools Used: Python, Sklearn, Scipy

Spaceship Paradigm — github.com/js_spaceship

- Built a full-stack MERN application for a McGill psychology lab, enabling automated measurement of cognitive fatigue and affect across extended time-on-task sessions.
- Used MongoDB to store behavioral and affective data for pilot participants.
- Tools Used: MERN full-stack

License Plate Recognition & Reading — github.com/ALPR

- Trained a YOLOv11 object detection model achieving **95% recall** on cluttered real-world license plate images, enabling robust downstream OCR.
- Trained a Bidirectional CRNN to read license plate characters with **80% character accuracy**.
- Tools Used: Python, Sklearn, PyTorch, cv2

Connect 4 Minimax AI — github.com/connect4

- Implemented a full adversarial search agent using Minimax with heuristic board evaluation to choose optimal moves.
- Added pruning and depth limits to reduce exponential branching, achieving real-time play with a transposition table to decrease move time.
- Tools Used: Python, Pygame

Skills

Languages: C, C++, Java, Python, JavaScript, Lua, SQL, OCaml, Bash, MIPS Assembly, HTML/CSS

Frameworks & Libraries: PyTorch, Transformers, React, Node.js, Unity, LÖVE2D

Technologies: MongoDB, Express, Git, Linux