Jackie Le

604-318-4795 jackiele.ca jddle@uwaterloo.ca github.com/jackiele123

Technical Skills

Languages: C++, C, C#, JavaScript, TypeScript, Python, HTML, CSS, Dart
Frameworks/Technologies: React.js, Node.js, Three.js, Vue, Flutter, Bootstrap, Docker, GitHub, PostgreSQL, TensorFlow, MongoDB
Software/Tools: VSCode, SolidWorks, Inventor, Ultimaker Cura, Jira Confluence, SourceTree, Linux

Experience

Software/Robotics Engineer **Jan. 2024 – Apr. 2024**
Lincoln Electric *Waterloo, ON*

- Developed a virtual environment for a robotic arm using **Vue** and **Babylon.js** to improve marketing and reduce testing times by 50%.
- Independently researched and designed the software architecture for the virtual environment, using **SocketIO** and **Express.js** to mimic the robot firmware and include support for projects.
- Initiated tasks to improve the company's inverse kinematic algorithms, reducing processing times by 13%.

Unity VR Developer **Apr. 2023 – Aug. 2023**
EXO Insights *Waterloo, ON*

- Collaborated with the Art and UX team to develop a full-scale, functioning game simulation of a power plant for **VR/PC** with over 100 unique interactive objects, enhancing the development of employee training modules.
- Spearheaded the creation of a VR training program for employee procedures, programming features for task progress, hints and dynamic object interactions using **C#**.
- Utilized MirrorNetworking to develop a collaborative **multiplayer** platform, enabling users to jointly spectate and engage, thus simulating more lifelike training scenarios.
- Engaged in bi-weekly client meetings, presenting milestones and assimilating feedback to improve the project.

Software/Robotics Engineer **Sept. 2022 – Dec. 2022**
Mission Control Space Services *Ottawa, ON*

- Collaborated with the **AI** team to develop methods of collecting data and testing for training computer vision **Machine Learning** models.
- Established an authentication system using Ory and **REST API** to safely deploy a web application for client use.
- Designed a front-end GUI that controls robotic arms using **C++, HTML, SCSS, Typescript** and **Solid.js** to offer students an engaging educational experience.
- Integrated **SocketIO** to handle communications between the GUI, Robotic Arms, and other company technologies.

Mechanical/Software Design Lead **Sept. 2017 – June 2021**
Vex Robotics Competition *Surrey, BC*

- Achieved **1st in the World** for programming.
- Programmed in **C++** an autonomous robot that used Odometry with PID Loops and Encoders to track orientation within 1° and position within $\frac{1}{2}$ inch.

Projects

My Personal Website | https://jackiele.ca | *React.js, CSS, JavaScript, HTML, GitHub*

- Designed a dynamic application using **React.js** and **CSS** to provide an overview of my experience and skillset.
- Utilized **React Bootstrap** to create a website that easily connects people to my projects, skills, and accomplishments.

6DOF Robotic Arm | https://project.jackiele.ca | *MLP, SolidWorks, Three.js, Javascript, HTML, CSS, C++*

- Created an interactive 3D portfolio website using **Three.js**, designing GUI control features and animations to showcase the design and capabilities of the robotic arm.
- Developed classes and methods to import CAD models from **SolidWorks** for use in animating individual components.
- Utilized **Three.js** and **Javascript** to simulate the robot arm, for developing the inverse kinematic algorithm.
- Programmed from scratch in **Python** a **Multi-Layer Perceptron**, to train a **Machine learning** model with data collected from the website; calculates inverse kinematics with 92% accuracy.
- Tinkered with 3D Printer firmware (Marlin) using **C++** to program a web interface for precise stepper motor controls, streamlining debugging processes.

Sign Language Translator | *Machine Learning, TensorFlow, Keras, MediaPipe, Python*

- Developed a **Machine Learning** model using **TensorFlow** to translate sign language through a live video feed.
- Designed scripts in **Python** to collect, analyze and display data to produce datasets that increased accuracy by 12%.
- Integrated **Computer Vision** using **OpenCV** to create a program that compares and displays the live video performance of multiple models, producing optimal models faster.