

# MARTIN KLASNINOVSKI

☎ 647-891-5056 — ✉ mklasninovski@gmail.com

🐙 github.com/mklaso — in linkedin.com/in/martin-klasninovski

## EDUCATION

---

### Wilfrid Laurier University

Honours Bachelor of Science - *Computer Science*

*Sep 2020 - Ongoing*

### University of Toronto

Honours Bachelor of Science - *Statistics, Computer Science*

*Sep 2018 - Apr 2020*

## TECHNICAL SKILLS

---

### Languages

Python · Java · C · HTML5 · CSS3 · Javascript

### Frameworks & Tools

React.js · Express.js · Node.js · SCSS · Git · Linux · Postman

### Databases

SQLite · MongoDB · PostgreSQL

## EXPERIENCE

---

### Dalvina Canada | Office Assistant

*Jun 2017 - Aug 2019*

- \* Updating inventory, receiving and processing order forms with Excel.
- \* Organizing and preparing cases for shipping according to delivery schedule.
- \* Communicated with clients regarding orders, samples, and general inquiry.
- \* Maintaining front-end of the company website.

## PROJECTS

---

### Weather App | Java, SQLite, JavaFX, CSS



- \* Utilizes the OpenWeatherMap API to retrieve accurate location information.
- \* Developed a GUI to display components of the weather, with matching images to the weather results.
- \* Incorporated a current weather forecast, as well as a 5 day forecast for the desired search location.
- \* Utilizes a local database with **SQLite** to store up to 10 locations to optimize the speed at which the weather data is displayed.

### Puzzle Solver | Python

- \* Implemented four classic puzzle games, each with single player functionality.
- \* Developed efficient puzzle solvers that utilize a **DFS** or **BFS** approach depending on the puzzle at hand, with the capability of solving most puzzles under 2s, and upwards of 10s for the most difficult puzzles.

### Othello | Java, JavaFX

- \* Recreated the game of Othello within a group of 4. Some features include: single player, multiplayer, player vs bot, bot vs bot functionalities, along with the ability to adjust the difficulty of the bots.
- \* Followed an **agile** approach of development, using **scrum** to implement the game and fix bugs efficiently.
- \* Implemented a working GUI for the game; utilizing **MVC** architecture, and various **design patterns**.

### Face Detection | React.js, Node.js, Express.js, PostgreSQL



- \* Set up a website to help detect faces from user uploaded images by using Clarifai's face detection API.
- \* Created a user registration system, as well as a ranking system that keeps track of the number of uploaded images by each registered user within the site.