

# Vincent Chang

vntchang@gmail.com | +886 979-723-415 | [github.com/niuee](https://github.com/niuee) | [Personal Website](#) | Taipei, Taiwan

## Work Experience

---

### Jubo - Frontend Engineer

Jan. 2024 - Present

React/TypeScript/Golang/PostgreSQL/ASP.NET

5 Months Taipei, Taiwan

- Used **React**, **TypeScript**, and **Next.js** to maintain a consumer-facing elderly care portal website.
- Collaborated with the backend team to replace the phased out Azure Media Player on the education platform; used the open-source library **shaka-player** to replace the video playback feature.
- Collaborated with the mobile team to develop a screenshot feature that works for both the web app and the mobile webview
- Collaborated with the backend and data engineering team to develop a feature that displays different data based on the user's location.
- Used **Golang** to support the backend team in developing basic APIs.

### Droxo Tech - Software Developer

Jan. 2021 - Jul. 2022

Golang/TypeScript/MongoDB/React/ROS

1 year, 7 Months Tainan, Taiwan

#### Full Stack Development: Fluid Storage Tank Inspection Robot [Detail Info](#)

- Built a responsive and cross-platform GUI application using **React** and **Electron** to monitor vital robot states.
- Designed the backend data flow; combine and stream critical robot information and control commands through a **WebSocket** server to and from the monitoring GUI application.
- Used **MongoDB** to handle the storage of measurement data for future analysis.
- Built a GUI app for visualizing the inspection result and corrosion analysis using **threejs**, **React**, **Electron**.
- Used external **RESTful API** to connect and integrate measurement data from the instrument with the core robot system.

#### Agricultural UAVs: using ROS(robot operating system) to integrate the newly added sensors with the flight control system.

- Utilized the GPS information to adjust the pesticide spraying pattern minimizing waste. (compare the geolocation with the hotspot of certain weeds)
- Implemented a precision landing system using the onboard camera with **arUco Tags** to minimize the margin of error when landing a UAV at specific locations.
- Automated mundane tasks by developing an application using **Node.js** for parsing binary flight logs from third-party UAVs into JSON format and uploading them to government-hosted APIs.

### Mandatory Military Service

Jan. 2020 - May 2020

Tainan, Taiwan

### Magpie Tech Corp. - Support Technician

Feb. 2019 - Sept. 2019

Los Angeles, CA

## Education

---

### Purdue University - B.S. Mechanical Engineering

Aug. 2014 - Dec. 2018

West Lafayette, IN

Started taking courses within the CS curriculum as technical electives. Followed the track for CS minor including data structure and algorithms, discrete math, etc. This is where I built a solid foundation for future endeavor in the CS field.

# Technical Skill

---

## Programming Related

- Language: Java, C/C++, JavaScript/TypeScript, Golang, Python, PHP, HTML, CSS
- Library/Framework: Express.js, Django, React, Electron.js, GraphQL
- Version Control: Git
- Unit Test: Mocha, JUnit, Jest
- Database: PostgreSQL, MongoDB

## Language

- Chinese/Mandarin: Native
- English: Fluent

# Personal Side Project

---

## board

This is a library that transform HTML canvas into infinite canvas. [GitHub Link](#). It has the following highlights:

- It does not rely on any external libraries. Built with vanilla JavaScript and TypeScript.
- Out of the box support for panning, zooming, and rotating the canvas.
- Supports different inputs: keyboard-mouse, trackpad, and touch control.
- Bundled using Rollup, testing with Jest, and GitHub Actions for CI and automating publishing to npmjs.
- Some example applications of board can be found at [board testground](#).

## Below are some other projects that I have worked on:

- bend: a bezier curve computation library [GitHub Link](#)
- bolt: a rudimental physics simulation that utilize a quad tree for quick collision detection [GitHub Link](#)
- bounce: a animation library [GitHub Link](#)

## Web Based Horse Racing Simulation Game

This is a series of projects that will ultimately come together as a horse racing simulation game.

- [HR Physics Simulation](#): A simplified physics simulation implemented in python.
- [HR GraphQL Server](#): A GraphQL server implemented in golang using gqlgen library. This graphql server serves as an API endpoint for querying complex data regarding race horses. (e.g. the pedigree of a race horse) [Live Demo](#)
- [HR Crawler](#): A simple crawler implemented in python using BeautifulSoup to parse the HTML pages; the crawler scrapes real world race horses data from a Japanese website netkeiba.com. Some portion of the crawled data is presented in the GraphQL server mentioned above.
- [HR Racetrack Maker](#): An editor used to create race tracks that can be used in the physics engine mentioned above to simulate horse races. This is essentially a bezier curve editor. The logic behind different operations is inspired by the 3D modeling software blender. [Live Demo](#)
- [point2point](#): A simple TypeScript library used for vector calculations. This is mostly used in the race track maker.