Education

- 1. 2017 2019: secondary education at Australian Science & Mathematics School, Adelaide, South Australia
- 2. 2020 Present: Bachelor of Computer Science at Monash University, Melbourne, Victoria

Skills

- Computer Programming Languages: Go, TypeScript/JavaScript, Pyhton, Kotlin/Java, C/C++
- Document Markup Languages: HTML, CSS, TFX, Markdown
- Natural Languages: English, Hebrew
- Tools: Git, GitHub, Docker, Kubernetes, GoReleaser, GitHub Actions
- Platforms: Linux, Cloud Native, Web
- Soft Skills: Writing, Presenting, Collaboration

Experience

- 1. May 2021 January 2022: General Representative at Monash Cyber Security Club.
- 2. January 2022 June 2022: Secretary at Monash Cyber Security Club.
- 3. June 2022 Present: Vice President at Monash Cyber Security Club.

Projects

Open-Source

- cocainate is a cross-platform re-implementation of the macOS utility caffeinate that keeps the screen turned on either until stopped, for a set duration of time or while another process still runs.
- stalk is a cross-platform file-watcher that can run a command after each file-system operation on a given file(s) or simply wait once until a file is changed.
- rake is a social media scraper that is interfaced via a server-side rendered HTML user interface (or a CLI), and is managed by a REST API and a NoSQL database.
- scr-web (and its scr-cli counterpart) is my previous attempt at building a full-stack social media scraper, which was abandoned due to the excessive number of dependencies and the rather large build-size.
- sp is my first attempt at building a Minecraft server plugin. This plugin adds the requirement that the player supplies the password (via a server command) before proper server interaction is allowed. Until as password is provided, the currently-unauthorized player is blinded and immobile.

Research

As part of the FIT2082 unit, I contributed to an existing codebase, based on prior research by (Gange, Harabor and Stuckey, 2021) about $Lazy\ CBS$, their Multi-Agent Path Finding (MAPF) algorithm. The MAPF problem is a subset of the path finding research field, which presents the additional requirements of multiple agents, each with a unique pair of a source and a target, such that the path between them does not intersect with another path during the same point in time. My task was to modify the $Lazy\ CBS$ codebase such that the algorithm also outputs the final set of constraints that is used to rule out possible paths, such that the Lazy is formally an Explainable Multi-Agent Path Finding (XMAPF) algorithm. In addition, I added Python-to-C++ bindings, such that the compiled $Lazy\ CBS$ codebase can be used as a Python-facing library for future projects.

Freelancing