

Omri Bornstein

CV

✉ omribor@gmail.com
🌐 applegamer22.github.io
in [omri-bornstein](#)
🎮 [AppleGamer22](#)

Education

1. **2017 — 2019**: secondary education at [Australian Science & Mathematics School](#) (ASMS), Adelaide, South Australia
2. **2020 — Present**: Bachelor of Computer Science at [Monash University](#), Melbourne, Victoria

Skills

- **Computer Programming Languages**: [Go](#), [TypeScript](#)/JavaScript, [Python](#), [Kotlin](#)/Java, C/C++
- **Document Markup Languages**: HTML/CSS, \LaTeX , Markdown
- **Databases**: [MongoDB](#)
- **Tools**: [Git](#), [GitHub](#)/[GitLab](#), [Docker](#), [Kubernetes](#), CI/CD
- **Platforms**: Linux, Cloud Native, web servers/browsers, macOS, Windows
- **Soft Skills**: technical writing, presenting/public speaking, research, troubleshooting/debugging, explaining, collaboration/teamwork

Leadership Experience

1. **May 2021 — January 2022**: General Representative at [Monash University's Cyber Security Club](#) (MonSec)
2. **January 2022 — June 2022**: Secretary at [Monash University's Cyber Security Club](#) (MonSec)
3. **June 2022 — Present**: Vice President at [Monash University's Cyber Security Club](#) (MonSec)

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 files or simply wait once until a file is changed.
- [raker](#) 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 with [Angular](#) on the front-end, and [Nest](#) on the back-end.
- [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, and as long as the password isn't 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.
 - 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 paths, such that *Lazy CBS* is formally an Explainable Multi-Agent Path Finding (XMAPF) algorithm.
 - I learned how to enable [Python-to-C++ bindings](#), such that the compiled *Lazy CBS* codebase can be used as a Python-facing library for future projects.

Freelancing

- I implemented a fault-tolerant file back-up system that enables the continuation of file transferring from an variably-approximate point in time before the back-up disruption.