

Omri Bornstein

Software Engineer

Greater Melbourne Area
Australia

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

Updated on **2023-04-21**

Education

- 2020 **Bachelor of Computer Science**, [Monash University](#), Melbourne
In Progress
- 2017 **South Australian Certificate of Education**, [Australian Science & Mathematics School \(ASMS\)](#),
2019 Adelaide

Experience

Volunteering

- 2023 **President**, [Monash Cyber Security Club \(MonSec\)](#), Melbourne
- Coordinated collaboration with the university's [Faculty of Information Technology](#) for purposes of events and advertising.
 - Added to the [club's website](#) a footer with social media links and a [tree](#)-like content organisation system.
 - Added more Azure infrastructure and an automation pipeline to the [club's CTFd server](#) with [Terraform](#) and [GitHub Actions](#).
 - Successfully registered the club for [Canva](#) and [GitHub](#) non-profit licenses.
- June 2022 **Vice President**, [Monash Cyber Security Club \(MonSec\)](#), Melbourne
Present
- Coordinated collaboration with the university's [Faculty of Information Technology](#) for purposes of events and advertising.
 - Organised and ran an introductory-level workshop about [steganography](#) (a recording is available on [YouTube](#)).
 - Represented the club at Faculty of IT's [Take CTRL](#) and [Munch & Mingle](#).
 - [Capture the Flag](#) (CTF) participation: [The University of Adelaide's CTF](#), [SHELL CTF](#), [DownUnderCTF](#)
- January 2022 **Secretary**, [Monash Cyber Security Club \(MonSec\)](#), Melbourne
June 2022
- Organised and recorded official committee and club meetings.
 - Represented the club during the orientation week of 2022's 1st semester.
 - Organised and ran an introductory-level binary reverse-engineering workshop (a recording is available on [YouTube](#)).
 - Started a section on the [resources page](#) of the club's website, with a detailed section with a guide on how to easily install and set-up a [Kali Linux](#) virtual machine.
 - Wrote a [Python](#) data visualisation program in order to analyse membership data, such that it would be easier for future committees to compile reports, to understand registration trends and to apply for student association grants.
- May 2021 **Assistant Member Training Officer**, [Monash Cyber Security Club \(MonSec\)](#), Melbourne
January 2022
- Helped to organise and ran a workshop about brute-forcing tools used for penetration testing.
 - Assisted in the club's management and operations.
 - Participated in [ångstromCTF](#).

Freelancing

- June 2021 **Software Engineer**, *Contract*, Melbourne
December 2021
- Implemented a fault-tolerant file back-up system that enables the continuation of file transferring from a variably-approximate point in time before the back-up disruption.
 - Built with [Go](#).

Projects

Open-Source

- May 2022 **raker**, [AppleGamer22/raker on GitHub](#)
Present
- 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.
 - Server-side is built with [Go](#), [MongoDB](#), [JSON Web Tokens](#) (JWTs) and [Docker](#).
 - Client-side is built with HTML/CSS ([Bootstrap](#)).
 - The companion CLI utility and configuration are built with [Cobra](#) and [Viper](#).
- May 2022 **stalk**, [AppleGamer22/stalk on GitHub](#)
Present
- A cross-platform file-watcher that can run a command after each file-system operation on a given set of files or simply wait once until a file is changed.
 - Built with [Go](#), [Cobra](#) and [FSnotify](#).

- January 2022 **cocainate**, [AppleGamer22/cocainate on GitHub](#)
- Present
 - 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.
 - Built with [Go](#) and [Cobra](#).
- June 2020 **sp**, [AppleGamer22/sp on GitHub](#)
- January 2021
 - 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.
 - Built with [Kotlin](#).
- April 2019 **scr-cli/scr-web**, [AppleGamer22/scr-cli on GitHub](#)/[AppleGamer22/scr-web on GitHub](#)
- May 2022
 - My previous attempt at building a full-stack (and a CLI) social media scraper with a single-page website framework and a RESTful server.
 - Server-side is built with [TypeScript](#) & [Nest](#) (with a [Node.js](#) runtime) [MongoDB](#), [JSON Web Tokens](#) (JWTs) and [Docker](#).
 - Client-side is built with [Angular](#) and [Ionic](#).
 - The full-stack package is bundled with [Nx](#).
 - The CLI is built with [OCLIF](#)

Research

- 2023 **Software Contributor**, Monash University's [FIT3144 unit](#), Melbourne
- in progress
 - Extending a browser-based tool ([Wagner et al., 2023](#)) used for building evolutionary algorithms in educational settings.
- 2021 Semester 2 **Software Contributor**, Monash University's [FIT2082 unit](#), Melbourne
 - 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.
 - I modified 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.
 - Built with C/C++ and [Python](#) on top of Linux.

Skills

Computer Programming Languages

Go	raker , stalk & cocainate	<i>server-side & command-line interfaces (CLIs)</i>
TypeScript	scr-cli/scr-web	<i>full-stack</i>
Python	FIT2082 research project	<i>data analysis</i>
Kotlin/Java	sp	<i>Minecraft plugins</i>
C/C++	FIT2082 research project	

Document Markup Languages

HTML & CSS	<i>client-side UI on web browsers</i>
T_EX/L_AT_EX	<i>PDF document typesetting</i>
Markdown	<i>technical documentation and communication</i>

Tools

Git	open-source projects	<i>source code version control</i>
GitHub/GitLab	open-source projects	<i>collaboration & CI/CD</i>
MongoDB	raker & scr-cli/scr-web	<i>document non-relational database</i>
SQL		<i>relational database querying</i>
Docker	raker & scr-cli/scr-web	<i>container-style packaging</i>
Kubernetes		<i>container orchestration</i>
Vagrant		<i>virtual machine (VM) management</i>
Terraform		<i>infrastructure definitions as code</i>

Other

- **Platforms:** Linux, Cloud Native, web servers/browsers, macOS, Windows