(201) 247-0282
wgoodall01@gmail.com
github.com/wgoodall01
williamgoodall.com

William Goodall

Education

2019-2023 (est.) **Georgia Institute of Technology,** College of Computing — Atlanta, GA
Bachelor of Science, major in Computer Science, 3rd-year, GPA 3.81.

Relevant coursework: **CS 1332** (Data Structures/Algorithms), **CS 2110** (Systems Architecture), **CS 3220** (Processor Design), **CS 3210** (Operating Systems), **CS 3251** (Computer Networking)

History

2019-pres. **Mesmer Eyes, Inc** — *Remote*
Developer (full-time summer 2021; part-time otherwise fall 2019-present), Summer Intern (2019)
Worked to automate mobile app testing and WCAG accessibility auditing. Spearheaded several projects:

Mesmer Gherkin (a compiler for English-language descriptions of app tests)

- Empowered customers to precisely define app tests across Android and iOS in a plain-English domain-specific language, with a parser, semantic analyzer, and AST built in Rust and WebAssembly
- Worked to unify many different internal representations of test logic into one common, well-defined AST
- Collaborated with the Customer Success Team, writing Gherkin snippets to rapidly solve client problems
- Employed coverage-guided fuzzing and extensive snapshot testing to find new bugs and stop regressions

The Mesmer Sidecar (beta release, cli.mesmerhq.com/sidecar)

- Built system to audit Android apps for accessibility issues, attaching to target apps at runtime and evaluating a proprietary ruleset against data extracted from Android APIs and in-house heuristics
- Created WCAG 2.0 rules engine, giving developers fast, explainable accessibility feedback
- Designed marketing materials, website, and pitch video for release on Product Hunt (#3 App of the Day)

The Mesmer CLI (cli.mesmerhq.com)

- Integrated the Mesmer platform with developer workflows as part of Mesmer's larger shift-left strategy
- Built connections with unstable internal APIs, maintaining a stable command-line interface for client scripts, integrations, and automation
- Implemented CI/CD and a comprehensive test suite, auto-importing user error reports into failing tests, and deploying fixes to production across Mac, Windows, and Linux auto-updaters in 20min
- Provided frontline support, dealing directly with five Fortune 500 clients

2019 **Wolfram Research** — *Waltham, Massachusetts*
Student, Wolfram Summer Camp
Researched both traditional and machine learning approaches to algorithmically determine the physical scale of satellite images. Final model achieved $r^2=0.73$ using feature extraction and a small dense neural net.

2017-2018 **Transparensee Systems** — *New York, NY*
Summer Intern, Developer
Created a client-facing customization tool for web widgets. This project was deployed to more than 60 local newspapers across the US, allowing non-technical users to integrate real estate platform enclosure.io and agricultural classifieds site agrisearch.com with their own web properties.

Responsibilities included:

- Designing and building full-stack web applications (React/Redux frontend, Express.js backend)
- Creating an automated CI/CD pipeline to build, test and deploy the application to Kubernetes.

Skills and Projects

Languages	Rust, Java, C, C++, JavaScript/TypeScript, Go, Python, Wolfram Language, VHDL, Verilog
Frameworks/Tools	Git, CI/CD (Jenkins, GH Actions), Docker, GraphQL, React (+Native), Next.JS, WebAssembly
Platforms	Android, Google Cloud, Firebase, Kubernetes, Heroku, Wolfram Cloud, AWS

Motor Controller (2021) FPGA-based closed-loop proportional servo controller. Used feedback from a quadrature encoder to position an output shaft with sub-degree precision. *Designed in VHDL, coursework for ECE 2031, Digital Design Lab.*

Boolean (2020) Symbolic computer algebra system for rule-based simplification of Boolean expressions. Contains a rudimentary SAT solver. *Written in Rust, compiled to WebAssembly, runs in-browser at* boolean.w01.dev

Cookie (2019) Dynamic cookbook. Based on the user's cooking preferences, construct a set of recipes as dependency graphs, simplify them by merging similar steps, and sequence them in parallel as the user cooks the meal. *Android/iOS app implemented with React-Native and TypeScript. Released to Google Play*: bit.ly/2muVZIY