# Mitchell Keith Bloch

PhD, Computer Science & Artificial Intelligence

University of Michigan

+1 (734) 926-5042

bazald@gmail.com

bazald.com

github.com/bazald

linkedin.com/in/bazald


## Education

Doctor of Philosophy, Computer Science and Engineering (University of Michigan, August 2018)
Thesis: *Computationally Efficient Relational Reinforcement Learning*
Master of Science in Engineering, Computer Science and Engineering (University of Michigan, 2010)
Bachelor of Science in Engineering, Computer Science, *Summa Cum Laude* (University of Michigan, 2008)

## Work Experience at Amazon

Software Development Engineer (SDE) for Amazon GG (Game Growth, formerly Game Publishing Studios)

- Mid 2021: Collaborated on the design and implementation of a high priority beta key delivery system
- Early 2021: Designed a descriptive file format and implemented a programmatic interface to enable a shim to differ permissions and routes based on caller, endpoint, and data attributes
- Late 2020: Built an Unreal Engine Plugin for the Persona SDK, enabling full functionality of the identity service for both Blueprints and C++ developers
- Mid-Late 2020: Designed and implemented the OAuth 2.0 endpoints for the Persona identity service
- Mid 2020: Implemented the Persona entitlement federation service
- Late 2019 – Early 2020: Implemented the bulk of the Persona privacy automation service to handle customer requests related to right-to-be-forgotten legislation

## Work Experience at the University of Michigan

Instructor for EECS 494 (2016-2017) – Game Design and Implementation

- Reorganized the curriculum to allow 4 sprints for final projects, gave lectures, managed TAs

Assistant Instructor for EECS 494 (2008-2015) – Game Design and Implementation

- Developed and taught zenilib (game engine), helped students with their projects, grading

Assistant Instructor for EECS 381 (Winter 2009) – Object-Oriented and Advanced Programming

- Assisted students, managed and ran the auto grader, hand graded code for student projects

## Projects

- `rete` (2021-) – A Rust crate with the beginnings of a parallel Rete algorithm implementation
- hash-trie (2021-) – A Rust crate that currently provides a HAMT implementation supporting set operations with comparable complexity to that provided by `im`, but with greater flexibility for set operations
- vote.cheap (2019-2020) – A verifiable and tamper-resistant online ranked choice platform implemented in pure Rust and deployed to ECS
- ConcRete (2018-2019) – A lock-free concurrent Rete implementation suitable for rule-based systems
- Carli (2012-2017) – Relational Reinforcement Learning (RRL) architecture, melding Rete and RRL
- zenilib (2006-2014) – A cross-platform game engine that I developed, that I used to teach at U-M camps and student groups, and that was used as the official game engine for EECS 494 for 6 years

## Select Posters and Presentations

Deciding to Specialize and Respecialize a Value Function for RRL (RLDM 2017)
Automatic Value Function Refinement and Unrefinement for RRL (The 36th Soar Workshop)
The Carli Architecture—Efficient Value Function Specialization for RRL (RLDM 2015)

## Hobbies

Argentine Tango | Singing | Burning Man