

# Connor Taffe

(501) 606-1807 · [cpaynetaffe@gmail.com](mailto:cpaynetaffe@gmail.com) · [byteflame.org](http://byteflame.org)

## EXPERIENCE

---

### **Ensono**

*Entry Applications Developer*

October 2015, current

*Conway, AR*

- Wrote and maintained software in C# for the .NET platform.
- Versioned software in Git and maintained repositories in Bitbucket and Github.
- Used Visual Studio to write and develop programs on Windows

### **UALR EIT IT**

*Student Worker*

August - October 2015

*Little Rock, AR*

- Resolved technology related issues for staff and students.
- Tracked issues in ticketing system.
- Installed and Setup Microsoft Windows, including joining to Active Directory domain.

### **EIT SUPER, UALR**

*SUPER Scholar*

Summer 2015

*Little Rock, AR*

- Summer Undergraduate Program of Entrepreneurship and Research (SUPER).
- Creation of an [Android application](#), written in Scala, to interface with a vehicle's OBD-II system via Bluetooth transmitter.

### **Emerging Analytics Center (EAC), UALR**

*Software Engineering Intern*

October 2014 - May 2015

*Little Rock, AR*

- Presented at IEEE VR 2015 on the integration of Unity 3D, Qualcomm's Vuforia, and Intel's OpenCV to create interactive Augmented Reality (AR) applications ([presentation](#)).
- Used Unity 3D (scripting with C#) for 3D programming and model manipulation.

### **UALR Bioinformatics**

*Research Assistant*

August - October 2014

*Little Rock, AR*

- Preliminary work on refactoring a genetic algorithm codebase with an emphasis on common stylistic guidelines and concurrency.

### **CentOS Server Wordpress Installation**

*Freelance Developer*

August 2014

- Installed Wordpress on a pre-imaged CentOS server.
- Set up a mysql database and edited Wordpress configs.

## EDUCATION

---

### **University of Arkansas, Little Rock**

*B.S. in Computer Science*

June 2018

- Substantail Completed Courses: Data Structures and Algorithms, Computer Systems and Assembly Language, Calculus II, Operating Systems, Databases, and Discrete Math.

## PERSONAL PROJECTS

---

### Lispy

September - November 2014

- Lisp-like interpreter written in Python
- Producer-consumer threading optimizes stages
- Lazy-evaluation of defined variables
- lambda functions and recursive lambdas
- EBNF formal definition of the language

### utf8 / utf8plus

February 2015 - Present

- utf8 parser written in C, and C++ wrapper.
- RFC 3629 compliant.
- Quick encoding/decoding of runes, rune validation, and string parsing.
- Reports errors within the rune and codepoint (`int32_t`) types using appropriate non-valid values in C.
- Use of the `utf8::rune::exception` class to report errors in C++, or appropriate C++ standard exception.

### bf

November 2014

- Optimizing BrainFuck just-in-time compiler
- Emits x86\_64 instructions to executable mmap'd pages
- Producer-Consumer concurrent architecture with a REPL interface

## TECHNICAL STRENGTHS

---

### General Programming

Languages:	C#, C++, C, Go, Rust, JavaScript, Java, Scala, Python
Libraries:	Android SDK
Version Control Systems:	git
Technologies:	Docker
Build Systems:	Make, CMake, Gradle

### Operating Systems

Android:	Familiarity with Android, the Android build toolchain, debugging, etc.
BSDs:	OpenBSD, FreeBSD, Dragonfly BSD, OS X
Linux:	CentOS, Debian, Ubuntu, Fedora, Red Hat Linux

### Unix/Linux Systems

Longtime Linux user. Experienced with system utilities, system administration, userspace programming, and light kernel programming.

### Web Programming

Server side:	Python, Go, Rust, Node.js, Php
Databases:	MySQL, PostgreSQL, CockroachDB
Client side:	Typescript (with Angular 2), JavaScript (jQuery, THREE.js, etc.), Bootstrap (3, Angular 2 + Bootstrap 4), CSS, (X)HTML