Caroline Cullen

Senior Software Engineer at Mozilla
carolinecullen.com
Github: carolinecullen
Linkedin: carolineetcullen

1840 Turk St, #10
San Francisco, CA 94115
(707) 315 5906
cullen.caroline@gmail.com

EXPERIENCE

Mozilla, San Francisco — *Senior Software Engineer*

August 2019 - PRESENT

Working on Firefox's JavaScript engine, SpiderMonkey. Contributed to a variety of projects and parts of the engine. Including, but not limited to:

- Implementing the new JavaScript language feature, Atomics.waitAsync. (In progress)
- Participating in SpiderMonkey's relationship with JavaScript's standards body, TC39. Attending TC39 meetings, SES meetings, and assisting in leading team proposal evaluations.
- Auto generating the instructions for the engine's mid and low tier intermediate representation.
- Contributed to SpiderMonkey's newest just in time compiler, WarpMonkey. Including building out the tooling for this new JIT, Cache IR Health Report.

Mozilla, *WASM Compiler Engineer Intern*

June 2018 - September 2018

Contributed to Cranelift, a WebAssembly code generator written in Rust language. Cranelift was, at the time, maintained by Mozilla and is now predominantly maintained by Fastly. Worked with Cranelift's IR and helped facilitate a better contributor experience on the open source project.

Veeva Systems, *Automation Engineer Intern*

June 2017 - September 2017

PROJECTS

LANGUAGE DESIGN

Mini Compiler — *Java*

Compiler for Mini language. Utilizes LLVM IR.

EDUCATION

Cal Poly, San Luis Obispo
Bachelors of Science in Computer Science
June 2019

Diablo Valley College
June 2014 - August 2016

SKILLS

Compilers and JITs

Open Source Software

Language Design

Graphics

Communication

Contributing to a positive team environment

LANGUAGES

C++, C, Python, Rust, Java

COURSEWORK SAMPLING

Programming Languages I, II (Compiler and Interpreter Design), Computer Graphics, Advanced Rendering Techniques, Game Engine Design, Theory of Computation

Nexus Interpreter — *SML*

Interpreter including: lexing, parsing, evaluation, objects, and static type checking

GRAPHICS

Game Engine — *C++*

Exploration and collection game called obtain., focused on aesthetics and graphics technology exploration. Includes a variety of technologies such as normal mapping, spring implemented camera motion, collision detection, bloom, particle system with billboarding, soft shadows, and view frustum culling.

RayTracer — *C++*

Fully implemented raytracer including Blinn-Phong BRDF, refractions and reflections, super sampling, object transforms, bounding volume hierarchies, Monte Carlo global illumination, depth of field, and generated animations with python scripting.

Blender the Video Game — *C++, OpenGL*

Video game that incorporates collision detection, randomly generated map layouts, skysphere with layered textures that are interpolated with a randomly generated sun position that makes for a cool effect, as well as a particle system with billboarding.