adenine.dev@gmail.com
www.linkedin.com/in/adenine-dev
www.adenine.dev

Adenine Griffin

EDUCATION

Lexington, KY **University of Kentucky** **Aug 2019 – May 2022**

- **Accreditation:** BS in Computer Science, Mathematics minor (cumulative GPA: 3.82)
- **Notable Courses:** Computer Graphics, Linear Algebra/3D math, Systems Programming, Algorithms and Data Structures, Statistics, Applied Artificial Intelligence, Applicable Algebra, Cryptography
- **Honors:** Dean's List (2020-2022), Summa Cum Laude
- **Skills:** C++, C, ECMAScript/JavaScript, Rust, OpenGL, Vulkan, GLSL, 2D Graphics, 3D Graphics, Artificial Intelligence, HTML, CSS, SASS/SCSS, React, Vue, Angular, JQuery, Python, Agile Processes, Kanban, Scrum, Visual Studio

EMPLOYMENT

Software Developer I **Evident Scientific (formerly Olympus)** **Jul 2022 – Nov 2022**

- Maintained cross platform XRF applications including embedded device, desktop, and web applications using QT, C++, and Node.js with Angular and typescript
- Redesigned web application UI to be more in line with existing native applications
- Owned large scale Angular dependency upgrade (9.1 to 14.2.1)

Full Stack Web Developer **Jessamine County Schools** **Oct 2018 – May 2019**

- Designed UI/UX in Adobe Experience Design (Adobe XD) and communicated with client to iterate on it and come to a shared design
- Built a web application with over 10 thousand active users using Ruby on Rails
- Assisted the maintenance of a large, multi-building network for the school system
- Accelerated hardware and software troubleshooting for various network issues

RESEARCH

Lead Developer **University of Kentucky** **Jan 2022 – May 2022**

- Led development on an entity component architecture
- Owned creator tools for authoring codeless simulations in custom environments
- Optimized performance for the architecture to perform well with several thousand object scenes using tools such as Valgrind, Very Sleepy, and Google Benchmark
- Used Test Driven Development to ensure correctness

PROJECTS

Ray Tracer:

- Created a monte carlo multi-threaded CPU ray tracer in rust with importance sampling
- Wrote a custom vector math module tailored to the needs of ray tracing
- Optimized performance to be semi-interactable for simple scenes using acceleration structures such as bounding volume hierarchies
- Implemented importance sampling for faster convergence with fewer samples
- Supported model loading, skyboxes, and various primitives including SDFs for complex scenes

Vulkan Graphics Engine:

- Created a real time physically based rendering application engine in C++ using Vulkan
- Managed packaging and build automation of the project through CMake
- Wrote comprehensive documentation rendered using Doxygen
- Profiled optimizations and performance using several tools including RenderDoc, Intel Graphics Performance Analyzers, and a custom profiler