Jim Eckerlein
Software Engineer

Münster, Germany
Email: jim.eckerlein@gmail.com
Web: jimec.dev
GitHub: github.com/jim-ec

EXPERIENCE

GPU Software Engineer, UX3D GmbH
January 2019 — Present

- Development on Gestaltor, the company's product, using Qt and C++
- Engine and middleware development in C++, Vulkan, and OpenGL
- Contribution to the official glTF Sample Viewer by the Khronos Group
- Extending the official Blender glTF Importer and Exporter to support Draco mesh compression
- Implementation of a Full-stack SvelteKit App
- Development on a React App
- Forking and customization of the render code of the Unity High-Fidelity Render Pipeline (HDRP)
- Various prototypes for several use-cases, mostly in Rust

Software Developer,
MBS Electronic Systems GmbH & Co. KG
June 2017 — December 2018

- Implementation of a PDF rendering widget in C++, QML, Qt Quick targeting an embedded device

Reference

Trainee, ESR Labs GmbH
September 2015 — July 2016

- Implementation of CAN message sender and receiver on an Arduino device
- Construction of Hardware on which the software implementation is supposed to run on

Reference

EDUCATION

Technical University of Munich
October 2018 — August 2023

Bachelor of Science in Computer Science
Seminar work: The Evolution of the C++ Memory Model

Thesis: An Environment for Continuous Integration and Software Testing for sys-sage

SKILLS

C++20, Rust
JavaScript, Svelte, React
Java, Kotlin
Git, GitHub
WebGPU, Vulkan, Metal, OpenGL

Familiar: Swift, Android, Haskell

Languages: German, Czech, English C1, Spanish

Personal Interests: Realtime rendering, mathematics, Geometric Algebra, programming close to hardware, being productive

PERSONAL PROJECTS

4D Geometry Renderer
Implementing a Flutter App rendering a spatial slice of a draggable 4-D geometry. Features interactive rotation on the X-W plane, the resulting 3-D slice is computed and rendered in real time. Source code

JavaScript mini IDE
Implementing an Android App featuring a JavaScript editor with syntax highlighting. The code is parsed in C++, the result passed back through the JNI. Features a built-in file explorer to persistently store scripts. Source code