Sean Siddens

seansiddens@gmail.com · (805) 390-0976 · seansiddens.github.io

Education

University of California, Santa Cruz — Santa Cruz, CA
BS Computer Science *GPA: **3.97*** — Sep 2020 - Aug 2023
Relevant coursework: Parallel and Concurrent Programming, Fundamentals of Compiler Design, Computer Architecture, Analysis of Algorithms, Full Stack Web Development, Database Systems

Skills

Programming Languages:	C/C++, Python, Javascript, SQL, OpenCL, GLSL, WGSL, Rust, HTML/CSS, Haskell, Bash
Tools/Frameworks:	Unix, Git, Vim, React, AWS, Vulkan, OpenGL, WebGPU, CUDA, Make, CMake, MaterialUI, Express.js, Node.js, PostgreSQL

Experience

University of California, Santa Cruz — Santa Cruz, CA
Concurrency and Heterogeneous Programming Lab (CHPL) Research Assistant — Apr 2023 - Present

- Designed and implemented cross-platform benchmarks for evaluating performance of fine-grained synchronization and dynamic work allocation on GPUs
- Analyzed and presented benchmark results leading to the identification of novel performance models on a wide variety of GPUs across multiple vendors.
- Significantly contributed to a Vulkan compute library, implementing GPU latency measuring capabilities and optimizing GPU resource usage.

Undergrad Researcher — Apr 2023 - June 2023

- Team lead for undergrad research group investigating techniques for automatically collecting annotated data for image classification tasks.
- Delegated tasks to the three other members in my group, organized meetings and research data, responsible for research direction and story.
- Designed and implemented an ML model for facial attribute classification using VGG16 fine-tuned on the CelebA dataset.

Projects

Epiphron *C++, Vulkan, OpenCL*

- Microbenchmark suite targeting the performance of fine-grained synchronization and dynamic work allocation on GPUs.
- Benchmark kernels written in OpenCL and benchmark framework and compute library written in C++ using the Vulkan API.
- Implemented kernel launch, barrier, graph application, and path tracing benchmarks.

Rendering Engine *C++, Vulkan*

- Built a 3D rendering engine using the Vulkan graphics API.
- OBJ model loading and asset management
- arcball and FPS camera systems
- PBR and Blinn-Phong material system
- point, directional, and cubemap lighting

Multithreaded HTTP Server *C, Bash*

- Thread-pool design to concurrently handle multiple clients over sockets.
- Worker threads concurrently fetch from work queue, protected via a mutex
- Preserve server-side coherency and atomization using multiple-reader single-writer semantics implemented via file locks
- Use of bash scripts for integration testing.

Gmail Clone *Javascript/React, HTML/CSS, SQL*

- Features user accounts, sending and deleting emails between users, managing emails via mailboxes.
- Frontend built with React and MaterialUI.
- API backend built using Express.js