KARIM SAYED

Riyadh, Saudi Arabia | +966573166308 | karimsayedre@gmail.com | Portfolio | GitHub | LinkedIn

Experience

The Forge Interactive Inc. | Graphics Programmer | (Remote) April 2024 - July 2024

- Maintained and optimized a cross-platform framework for PlayStation, Xbox, Switch, and others.
- Refactored core systems using advanced C/C++ techniques, improving maintainability.
- Reduced ray tracing frametimes from 2.8ms to 0.7ms on RTX 3080 by utilizing GPU RT cores.
- Improved validation/debugging by building automated tools in C++ and Python.

Sensor Foundries Inc. | Graphics Programmer (Remote)

• Transitioned rendering features from Vulkan-based engine to OpenGL, maintaining performance

May 2022 - April 2024

- Restructured core rendering and asset management systems improving maintainability
- Improved rendering pipeline by optimizing shaders and reducing draw calls.
- Played a key role in optimizing animation rendering to ensure smooth real-time playback.

Studio Cherno | Rendering Engineer Contributor (Remote) March 2021 - April 2022

- Modernized rendering workflow with modern C++ for cleaner, maintainable code.
- Developed efficient real-time rendering algorithms for improved visuals and performance.
- Built a shader pre-processor to reduce duplication and improve portability.

Personal Projects

Beyond Engine: Custom fork of Hazel Engine with an advanced renderer

- Architected and optimized core subsystems with Vulkan and modern C++17/20.
- Reduced load times from 8s to 5s with shader metadata caching and preprocessing.
- Enhanced memory and performance via custom allocators and optimized resource management.
- Cut descriptor processing time from 1ms to 0.02ms using a high-performance bitwise hashmap.
- Reduced compilation time by 20% by optimizing header parsing and reducing dependencies.

CUDA Ray Tracing in One Weekend: High-performance GPU path tracer

[GitHub Repository] [Project Article]

- Built a CUDA path tracer achieving sub-9ms frame times on RTX 3080 and 350ms on i5-13600KF
- Minimized VRAM usage by fitting data into L1 cache via SoA layouts, yielding 99% L1 hit rate
- Removed virtual function calls using data-oriented design for warp coherence achieving 92% branch efficiency
- Tuned SM occupancy and register pressure maintaining high SM throughput and low latency

Skills and Expertise

Languages: C, C++, CUDA, GLSL, HLSL, Slang, Intel x86 Assembly, Python, Java, JavaScript Graphics & Compute: Vulkan, OpenGL, CUDA, RenderDoc, Nsight, VTune Build Tools: Premake, CMake, Git, Jenkins, MSVC, Clang, GCC, VCPKG Concepts: Modern C++20/23 features, Multi-threading, SIMD/SIMT, DOD/OOP Optimization: Cache-aware algorithms, memory management, bottleneck analysis, profiling tools

EDUCATION

B.Sc. (Hons.) in Computer Science	Multimedia University, Malaysia	July 2018 – July 2021
Specialization: Software Engineering /	CGPA: 3.11 / 4.00	

Certificates

Data Structures and Performance | Coursera – Certificate Link Object Oriented Programming in Java | Coursera – Certificate Link