KARIM MOHAMED
C++ SOFTWARE ENGINEER

Riyadh, Saudi Arabia | +966573166308 | Email | Portfolio Website | GitHub Account | LinkedIn Account

EXPERIENCE

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

- Maintained and optimized a cross-platform high-performance framework for PlayStation, Xbox, Switch, and other platforms.
- Refactored and modularized core systems, leveraging advanced C/C++ features to improve maintainability and performance.
- Optimized ray tracing by using GPU ray tracing cores reducing frametimes from 2.8ms to 0.7ms on an RTX 3080.
- Enhanced testing pipelines by implementing automated validation and debugging tools with C++.

Graphics Engineer, Sensor Foundries Inc. (Remote) — May 2022 - April 2024

- Refactored engine to ensure seamless integration between different graphics APIs, employing more modern C++ design principles.
- Optimized performance through multi-threading, efficient resource management, and extensive profiling with Intel VTune Profiler.
- Designed modular systems for resource handling and rendering workflows, improving scalability and maintainability.
- Used debugging and profiling tools to optimize performance and improve stability.

Graphics Engineer Contributor, Studio Cherno (Remote) — March 2021 - April 2022

- Improved rendering workflows by adopting modern C++ practices, enhancing modularity and performance.
- Implemented efficient algorithms for real-time rendering, improving the quality and performance of visual effects.
- Designed and implemented a shader pre-processor similar to C/C++'s preprocessors for better portability and less code duplication.

PERSONAL PROJECT

Beyond Engine, a private fork of The Cherno's Hazel Engine — Source code here

- Architected and optimized core subsystems with Vulkan API, focusing on scalability and efficiency in C++.
- Optimized CPU code using multiple techniques, including shader metadata caching and reading, reducing load times from 8s to 5s.
- Achieved significant memory and performance optimizations using custom allocators and efficient resource management.
- Optimized resource descriptor processing by implementing a bit-manipulated hashmap, reducing times from 1ms to 0.02ms.
- Optimized compilation times by reducing header parsing counts, cutting compilation time from 75s to 60s on an i5 13600KF.

CUDA Ray Tracing in One Weekend: High-performance GPU path tracer. — Source code here

- Built a real-time path tracer in CUDA, achieving sub-9 ms frame times (RTX 3080) and ~350 ms on CPU (i5-13600KF).
- Optimized memory access with SoA layouts and cache-fit working sets, reaching ~99% L1 hit rates and minimal VRAM usage.
- Replaced virtual dispatch with data-oriented design and branchless logic, improving warp efficiency and control flow.
- Tuned resource usage to maximize SM occupancy and throughput without relying on dedicated ray tracing cores.

SKILLS AND EXPERTISE

- **Languages:** C++, x86 Assembly, Python, Java, GLSL, HLSL, JavaScript.
- **Graphics & Compute:** Vulkan, OpenGL, CUDA, RenderDoc, NVIDIA Nsight, Intel VTune.
- **Build & Dev Tools:** Premake, CMake, Git, Jenkins, MSVC, Clang, GCC, VCPKG.
- **Programming Concepts:** Modern C++ (concepts, ranges), multi-threading, SIMD, object- & data-oriented design.
- **Optimization:** Cache-aware algorithms, memory management, bottleneck analysis, modular systems.
- **Systems & Architecture:** Custom data structures, memory allocators, CPU/GPU internals, memory hierarchies.

EDUCATION

Multimedia University in Malaysia — July 2018 - July 2021

Bachelor's degree in Computer Science (Hons.) - Software Engineering Specialisation.
GPA: 3.11

CERTIFICATES

- Data Structures and Performance (Coursera): Certificate Link
- Object Oriented Programming in Java (Coursera): Certificate Link