E-Mail lalishansh@gmail.com
Phone +91-7805890067
Website lalishansh.github.io/about
Profile linkedin.com/in/lalishansh
GitHub github.com/lalishansh

Ishansh Lal

Graphics Software Engineer



Summary

C++ software & graphics engineer with 4+ years building real-time 3D rendering pipelines, performance-critical applications, & developer tooling. Strong in multithreading, profiling, & optimization across **OpenGL**, **DirectX 12**, & **Vulkan**, with shipped work on **Skull & Bones** & the **AP Monorepo** codebase, and **Qt**-based desktop software with CI/CD-ready builds.

Education

B.Tech, Computer Science
Jaypee Institute of Information Technology, Noida, India

Languages & Frameworks

C++17/20 | HLSL / GLSL / Slang / OSL | C | x86-64 Assembly (SSE, AVX, AVX2) | Python | C# | Jai | Qt 5/6 | ImGUI

APIs

OpenGL 4 | DirectX 12 | Vulkan | OpenGL ES 3 | WebGPU

Concurrency & Compute

Multithreading & Synchronization | Job Systems | oneTBB (hobby) | CUDA / OpenCL (hobby)

Profiling & Debugging

Visual Studio Debugger | RenderDoc | PIX (Xbox / Windows) | Nsight | Compiler Explorer

Build, Test & CI

CMake | Jenkins | Boost.Test | GTest

Workflow

Git | Perforce | Anvil (C#/C++, Ubisoft)

Concepts

Software Design & OOP | Render Graphs | Multithreading & Memory Optimization | Spatial & Geometric Algorithms | SIMD | Acceleration Structures | Point Clouds | Rendering Pipelines | Unit Testing & CI | Performance Profiling & Bottleneck Triage | Mesh Processing | Shader Debugging | Cross-Platform Development | Developer Tooling & Automation | CPU/GPU Architecture | Ray Tracing

Spoken Languages

English | Hindi | Japanese (JLPT N5)

Experience

Expleo Solutions

- **ADAS Software Engineer** (Aug 2025 - Present · On-Site)
  - Develop **free-view camera** algorithms & ADAS 3D visualizations for client **Valeo**.
  - Profile the rendering pipeline & cut frame time by 10-15% by reusing shaders, caching textures, & minimizing texture swaps (bringing it within the 30fps budget).
  - Expanded the set of tunable runtime parameters & fixed 50+ production bugs.

Ubisoft India Studios

- **3D Programmer** (Jun 2023 - Aug 2025 · On-Site)
  - Contributed to the **New AP monorepo** shared by **AC Shadows** & upcoming Anvil/Scimitar titles; documented & presented pipeline internals to other teams.
  - Built the **2D Living Map** pipeline for an unannounced title from scratch with offline rendering of terrain, road/rail routes, building silhouettes, forest, water, OOB area etc.
  - Built **artist-facing tools & automation** across Ubisoft's internal C# editor, C++ engine, & UI stack, including automated map generation that cut iteration time from hours by hand to seconds.
  - Profiled & optimized rendering, saving ~3ms per frame when rendering the map.
  - Debugged rendering issues on **PS5 & Xbox** by reading **shader instructions** directly in PIX & Razor to locate costly or incorrect passes in **Skull & Bones**.
  - Maintain graphics systems & fix rendering bugs for **Skull & Bones** as part of the quality pod, like light bleeds, GI etc.

Viga Entertainment Studio

- **Graphics Engineer** (Oct 2021 - May 2023 · Intern, then Remote)
  - Made builds single-step & CI/CD-ready (CMake, vcpkg, **Qt 5/6**, automatic dependency resolution); planned the ScanStage project architecture.
  - Sped up ScanStage tools such as Face Capture by ~120x (2x from hot-path optimization, 64x from **multithreading**) & reduced memory use by ~15% (observed).
  - Wrote unit tests with **Boost.Test** & **Google Test** for ScanStage & integrated them into **Jenkins CI**.
  - Built a pipeline converting stereo **ZED** depth video into 3D meshes & point clouds for volumetric streaming (Dubai Expo 2023, ScanStage).
  - Optimized dense image-derived point-cloud meshes by removing redundant vertices, downsampling, & restoring detail via mean curvature (multithreaded, real-time editable, disk-cached).
  - Implemented AprilTag camera calibration for face reconstruction & wand calibration for motion capture.

Projects

- **SETU Engine**: solo cross-platform engine (Windows, Linux, Android; OpenGL 4, GLES 3, DX12) with ECS & job system. C++, CMake.
- **GPU Ray Tracer**: GPGPU-accelerated implementation of the *Ray Tracing in One Weekend* series.
- **GPU-RHI**: general-purpose graphics API abstraction (Vulkan, WebGPU) in **Jai** (private beta); builds the render graph from shaders using compile-time metaprogramming.
- **x86-64 SIMD Study**: example project using SSE, AVX, & AVX2, with generated code reviewed in Compiler Explorer.
- **NotPing-OnlyPong**: game with a custom software rasterizer, GUI, VFX, SFX, & tutorial system, using only Win32 & C++.
- **Particle Swarm Optimization Visualizer**: Vulkan app with MSDF text rendering & custom GUI.
- **Other**: CUDA, OpenCL, & oneTBB experiments; Hazel & Kohi engine studies; procedural terrain in Unity; scene perception for the visually impaired (computer vision).

Awards:

- Won µCR, Microcontroller System & Robotics hub event Eximietas, 2019