

SOO MIN 'JACKY' PARK

217-979-1358 | jackypark9852@gmail.com | jackypark.com | in/soominjackypark/

EDUCATION

University of Pennsylvania

Candidate for Master of Science in Engineering – Computer Graphics and Game Technology

Philadelphia, PA

Aug. 2024 – May 2026

University of Illinois at Urbana-Champaign

Bachelor of Science in Civil Engineering, Minor in Computer Science

Champaign, IL

Aug. 2018 – May 2024

TECHNICAL SKILLS

Programming Languages: C++, GLSL, WGSL, C#, C, Python

Software/Frameworks: OpenGL, CUDA, WebGPU, Unreal Engine, Unity, Qt, Maya, Blender

Developer Tools: Visual Studio, Rider, Git, Jira

EXPERIENCE

Graphics Engine Team – Rendering Algorithm Intern

Jun. 2025 – Aug. 2025

Alibaba Group

Hangzhou, China

- Developed a graphics performance optimization plugin using Unreal C++, automating optimizations (e.g., texture scaling, Nanite activation), enhancing rendering performance by approximately 30% for five virtual production scenes.
- Built and integrated an OpenColorIO-based color management pipeline into a proprietary Hydra Delegate renderer, aligning color consistency with existing ACES animation workflows.
- Implemented direct renderer preview image export, reducing validation time from 30 minutes (via traditional Nuke workflow) to a few seconds, accelerating iteration and debugging.

PROJECTS

DirectX 12 Volumetric Cloud Renderer | C++17, DirectX 12, HLSL, RenderDoc, Pix

Nov. 2025 – Dev. 2025

[Video Demo](#) · [GitHub](#)

- Built a DirectX 12 HLSL compute ray marcher for volumetric clouds with adaptive step sizing, per-pixel jitter, and empty-space skipping to eliminate wasted samples in low-density regions.
- Built a GPU lighting cache that pre-computes per-voxel shadow contribution each frame, replacing per-ray shadow traversal with a texture lookup and cutting render time from 20ms to 2ms.
- Profiled per-pass GPU timings with RenderDoc and PIX to guide step-size and sampling choices, driving the 10× cost delta between near-camera detail and distant clouds.

CUDA Path Tracer | C++, CUDA, OpenGL, NSight Graphics

Sept. 2025 – Oct. 2025

[GitHub](#)

- Implemented GGX microfacet with multiple-scattering compensation and a dielectric BSDF with HDRI and PBR texture maps, producing photorealistic metals, glass, and rough surfaces.
- Built a BVH with surface-area-heuristic construction and stack-based traversal, cutting frame time by ~99.3% (about 131×) on a ~50k-triangle scene.
- Added stream compaction for terminated paths and material-based sorting with per-material kernels, reducing branch divergence and wasted work after bounces.
- Integrated NVIDIA OptiX AI denoiser with albedo/normal AOV guides, improving low-spp image quality while preserving edges and broad color regions.

Forward Plus and Clustered Deferred Renderer | WebGPU, WGSL, TypeScript

Sept. 2025 – Oct. 2025

[Live Demo](#) · [GitHub](#)

- Implemented compute-shader light clustering with exponential depth slicing and per-cluster light culling, which let fragments evaluate only nearby lights instead of the full set.
- Built a clustered deferred pipeline with G-buffers and a fullscreen lighting pass, decoupling geometry from lighting and avoiding overdraw on hidden fragments.
- Profiled GPU passes with timestamp queries and tuned cluster dimensions and buffer layouts, reducing frame time by ~90% in a 1,000-light scene compared to the naive baseline and outperforming Forward+ under heavy light counts.