

Hudson Nguyen

thehudson.dev

Seattle, WA

hudsonhannguyen@gmail.com

EDUCATION

University of Southern California

- **M.S.** - Computer Science June 2022 - May 2023
- **B.S.** - Computer Engineering and Computer Science Aug 2018 - May 2022
 - **Minor:** Video Game Design and Management

WORK EXPERIENCE

[ProbablyMonsters](#)

Bellevue, WA

Associate Core Tech Engineer

Jan 2024 - Nov 2024

- Created entirety of the performance automation testing pipeline for multiple game projects from in-game profiling, to exporting data in human-readable formats, to data presentation for artists and engineers to use
- Debugged rendering issues including failed texture loads and excessive video memory consumption
- Established minimum and recommended PC specifications for milestone builds, coordinating with graphics engineers, technical artists, and QA

[Industrial Light & Magic \(ILM\)](#)

San Francisco, CA

Graduate Research & Development Intern

June 2023 - Oct 2023

- Designed and implemented UI/UX features for StageCraft virtual production tools to streamline asset pipeline between Unreal Engine and in-house DCC tools
- Optimized acceleration structure generation for real-time path tracer and modified render pass resource system to avoid allocating excess, unused resources per frame, cutting memory costs by 6%
- Implemented soft particle system in renderer for smooth intersections between geometry and translucent sprites

[Playbook XR](#)

Los Angeles, CA

XR Programmer

Jan 2022 - Mar 2023

- Created ReactJS plugin for Figma to export wireframe elements into JSON format and developed 3D viewer to preview designs in XR environment
- Streamed Figma API data into Unity VR application to replicate UI elements from wireframe to in-app
- Developed solutions for runtime instantiation of 3D file formats (i.e. GLTF, OBJ) into an interactable component

PROJECTS

Lunatic Engine - [Demo](#)

Mar 2025 - Present

- Vulkan real-time renderer featuring cascaded shadow maps, indirect lighting with reflective shadow maps, temporal anti-aliasing, screen-space ambient occlusion, and deferred rendering to efficiently handle a larger number of lighting calculations
- Optimize performance by minimizing per-frame draw calls through a GPU-driven system that uses indirect draws with bindless textures, bindless vertex and index buffers, as well as frustum culling through compute shaders
- Link custom engine DLLs to Unreal Engine to easily control placement of objects in external level editor, streaming mesh data into the engine at runtime and updating in render preview
- Enhance visual fidelity of meshes through a physically-based rendering system based on Cook-Torrance BRDF that supports metallic, roughness, normal, and albedo textures for materials

SKILLS AND QUALITIES

Languages: C++, C#, Python, HLSL, GLSL, PowerShell, Java

Tools/Frameworks: Visual Studio, Rider, Unreal Engine, Unity, Git, Perforce, Vulkan, DirectX, RenderDoc, Nsight, PIX