

Choi Hongsu

Technical Artist

Phone 010-7538-8268 | Email hongsulovey@gmail.com | Portfolio hongsu.dev | GitHub github.com/hongsulovey

PROFILE

Technical Artist with experience in shader development, GPU/CPU profiling, rendering optimization, and art pipeline improvements for Unity URP and Unreal Engine-based projects.

With a background in 3D environment art, I have hands-on experience in modeling, material creation, and engine implementation. My current strengths include mobile rendering optimization, custom HLSL shader development, ScriptableRendererFeature-based rendering systems, and production-oriented technical documentation.

I focus on designing maintainable structures for real production environments, and validate problems through measurable data and profiling tools rather than simply adding complex features.

CORE SKILLS

Rendering / Shader

Unity URP, Unreal Engine, HLSL, GLSL, Shader Graph, Unreal Material Editor, Custom Render Pass, ScriptableRendererFeature, Matcap, Toon Shading, UI Shader, Particle Shader, Dissolve, Outline, Stencil, Depth, Texture2DArray

Optimization / Profiling

RenderDoc, Android GPU Inspector, Unity Profiler, Unity Memory Profiler, Frame Debugger, CPU/GPU Bottleneck Analysis, Mobile / PC Rendering Optimization, Overdraw Optimization, Draw Call Optimization, SRP Batcher, GPU Instancing, LOD Optimization

Art / Pipeline Tools

3ds Max, Blender, ZBrush, Substance Painter, Substance Designer, Photoshop, Git, Vertex Normal Workflow, FBX Import Settings, VFX Material Setup, Shader / VFX Guideline, Production Workflow Improvement

EXPERIENCE

Devsisters | Technical Artist, CookieRun: OvenSmash

2025.11 – present

- Worked on character rendering, shader development, and rendering optimization for a mobile PvP project built with Unity 6 / URP 17.
- Analyzed rendering bottlenecks using Android GPU Inspector and Memory Profiler, improving DrawOpaqueObjects GPU Time on Galaxy S21 from 46ms to 27.1ms.
- Converted a Compute Shader-based Bloom system to a Fragment Shader-based structure, reducing COSPostProcessing GPU Time from 8.37ms to 1.26ms and optimizing Bloom RT memory from 16.6MB to 1.4MB.
- Tracked a large volume of URP Blitter duplicate initialization errors on Android through hypothesis-based debugging, reducing the log count from 137,080 to 2,978 after the first fix.
- Organized a Vertex Normal / Custom Normal Workflow to replace normal-map-dependent character highlight correction, reducing texture dependency and documenting production standards suitable for LOD optimization.
- Created technical guidelines for production character assets, covering LOD, BlendShape, Skin Weight, Mesh Compression, and FBX Import Settings.

KongStudios | 3D Artist & Tech, FlashGambit

2024.03 – 2025.11

- Developed custom shaders for units, UI, VFX, particles, dissolve effects, Matcap lighting, and team-color workflows for a mobile RTS project built with Unity URP.
- Implemented Fog of War shader integration for a top-view RTS project using Render Textures and global shader textures.
- Organized texture sets for the SH_Unit_Matcap shader and documented production rules for managing Mask / Normal / Emissive data in a single packed texture.
- Created UI optimization guidelines focused on separating Static UI and Dynamic UI, and analyzed Canvas Rebuild costs in a structure containing approximately 12,979 RectTransforms.
- Created optimization guidelines to reduce overdraw, duplicated materials, shader fragmentation, and scattered texture usage in the VFX production workflow.

SuperStormStudio | 3D Artist, StormStriker

2022.12 – 2024.02

- Created 3D environment assets and materials for the StormStriker project.
- Handled the full environment asset production process, including modeling, UVs, texture setup, material creation, and engine implementation.
- Supported level art and look-development tasks by balancing asset quality with optimization requirements for real-time rendering environments.
- Resolved technical issues during asset production, including collision setup, scene organization, material setup, and optimization checks.

Allm | 3D Artist, Kritika2

2022.05 – 2022.12

- Created environment props, set up materials, supported lighting, and contributed to look development for Kritika2, a hyper-action TPS project.
- Supported Unreal Engine rendering-related workflows, including Level Merge, Culling, RVT workflows, and dither-based visual setup.
- Bridged art production and engine implementation by resolving technical issues across modeling, materials, and scene setup.

SELECTED PROJECTS

CookieRun: OvenSmash — Device-Specific GPU Profiling

Analyzed rendering bottlenecks across mobile devices using Android GPU Inspector. Identified that background DrawOpaqueObjects was consuming most of the frame budget, and improved the rendering direction through normal map cleanup, shader simplification, and geometry optimization.

CookieRun: OvenSmash — Post-Processing Bloom Optimization

Converted a Compute Shader-based Bloom system into a Fragment Shader-based structure. Significantly reduced GPU Time and RenderTexture memory usage by removing History RTs, reducing RT format/resolution cost, lowering the number of samples, and eliminating GC allocations.

CookieRun: OvenSmash — Vertex Normal Workflow

Established a production workflow for controlling character highlights by directly editing vertex normals without relying on normal maps. Documented 3ds Max workflows based on Weighted Normal, Edit Normal, Edge Split, and Largest Face methods so artists could create assets with consistent standards.

CookieRun: OvenSmash — URP Blitter Duplicate Initialization Error Fix

Tracked a large volume of URP Blitter duplicate initialization errors on Android through log-based debugging. Identified that URP Asset property setters were being called every frame, causing RenderPipeline recreation, and fixed the issue by calling setters only when changes were detected.

FlashGambit — Shader & Texture Set Pipeline

Defined texture set rules for mobile RTS unit production by managing Matcap, Mask, Normal, and Emissive data in packed textures. Improved production consistency by documenting texture naming rules, channel usage, and in-game texture size standards.

FlashGambit — UI / FX Optimization

Optimized the UI/FX production structure by separating Static UI and Dynamic UI, analyzing Canvas Rebuild costs, reducing VFX overdraw, removing duplicated materials, and using MPB-based color controls.

EDUCATION & ADDITIONAL

Education: Mokwon University, Department of Western Painting — Degree not completed

Military Service: Republic of Korea Army, Sergeant / Driver, 2018.02 – 2019.09