Onuralp S. Sahin

Graphics Programmer

onuralpsahin49@gmail.com | LinkedIn | GitHub | Itch.io

PROFILE

Graphics Programmer with 3+ years of professional experience at d3t Ltd., specialising in Unreal Engine rendering systems, shader programming, and real time graphics optimisation. Delivered engine level rendering work across a shipped AAA title and an upcoming release, including NVIDIA DLSS integration and debugging (Frame Generation, Ray Reconstruction), Virtual Shadow Map optimisations, multi-monitor display systems, and graphics settings pipelines. Strong C++ and graphics API background with hands on experience in Vulkan, OpenGL, HLSL, and GLSL.

PROFESSIONAL EXPERIENCE

Graphics Programmer | d3t Ltd. April 2023 - Present

NVIDIA DLSS Integration, Frame Generation & Ray Reconstruction

- Identify a DLSS Frame Generation fault where the HUD-less buffer was rejected for mismatching backbuffer dimensions whenever secondary screen percentage ran below 100%. The plugin's resource tagging fired before Unreal's spatial upscaler had upscaled the HUD-less buffer. Implemented a custom post-upscale callback hook, subscribed to from the DLSS plugin, re-tagging the buffer at final resolution and restoring Frame Generation under scaled viewports.
- Eliminated speckling artefacts on rough metallic surfaces when DLSS Ray Reconstruction ran alongside downsampled Lumen reflections; DLSS-RR bypasses the stock temporal and denoising passes, and its neural denoiser (trained to resolve randomised noise) preserved the fixed-pattern noise from Lumen's 4-rook upsampling as detail. Replaced the downsample/upsample path with full-resolution sparse tracing, 25% of pixels traced at randomised points, the remainder reconstructed by neighbourhood averaging. This then allowed us to retain the original shader-side saving in exchange for a measured ~40 MB of additional texture memory at native 4K.

Virtual Shadow Maps (VSM) & Lighting

- Extended Unreal Engine's VSM pipeline with a new shadow invalidation behaviour for dynamic shadow casters, auto regulating with camera distance.
- Developed screen size based VSM culling, excluding shadow casters with negligible pixel contribution.
- Fixed self shadowing artefacts on WPO enabled static meshes via per pixel rejection logic comparing contact shadow results against VSM output, generating a rejection mask applied in a clean up pass; shipped engine wide.
- Improved hair lighting fidelity via ray tracing, reducing non shadow casting light bleed and overexposure artefacts.

Rendering Systems & Optimisation

- Maintained and optimised a custom Unreal Engine shadow pass end to end: CPU-side resource creation, pass scheduling, shader development and maintenance, an artist-facing Blueprint node API, and RHI resource management.
- Developed custom shader code in .ush/.usf files; extended the material editor with custom material nodes.
- Worked across deferred, forward, and translucent rendering pipelines; proficient in Unreal's RDG, pass management, and artefact debugging.
- Modified Niagara particle shaders and integrated custom particle behaviours.
- Ported custom rendering code across multiple Unreal Engine version upgrades, fixing the associated rendering bugs and crashes.

Display Systems & Graphics Settings (Shipped on Two Titles)

- Designed and implemented a multi-monitor switching system enabling runtime viewport relocation, resolution synchronisation, fullscreen handling, and DPI correction through Slate, Display Metrics, and Windows Display APIs. This was built before Epic introduced native multi-monitor support in later engine versions.
- Developed graphics settings systems covering front-end UI and back-end engine bindings for scalability profiles, motion blur, depth of field, FOV, HDR, ray tracing, display switching, and resolution.

PERSONAL PROJECTS

Vulkan Graphics Engine | MSc Dissertation | Video

- Custom C++ Vulkan abstraction layer with PBR shaders, model loading, skyboxes, and particle simulation driven by simplex and curl noise; abstractions for descriptor sets, swapchains, PSOs, render passes, and mipmap generation. Open source on GitHub.

Anor Engine | Personal Project | Video

- Custom 3D/2D OpenGL rendering engine featuring normal/specular mapping, Blinn-Phong lighting, shadow mapping for directional and point lights, YAML scene serialisation, and an ImGui-based editor.

Also: winner of Kingston University's 48-hour game jam (Keyboard Warrior); built a 20-player online multiplayer game in Unity with Photon (Bot Breakout).

EDUCATION

- **MSc Games Programming - Distinction** | Kingston University | Sep 2021 - Sep 2023
- **BSc Software Engineering - High Honours, GPA 3.77/4.00** | Atılım University | Oct 2016 - Aug 2020

TECHNICAL SKILLS

Languages: C++ (5+ years), Python (scripting)

Engines: Unreal Engine 4/5 (3+ years, professional), Unity3D (2 years)

Graphics APIs: Vulkan, OpenGL

Upscaling & Denoising: NVIDIA DLSS (Super Resolution, Frame Generation, Ray Reconstruction), Unreal temporal/spatial upscaler pipeline

Shaders: HLSL, GLSL, Unreal shader authoring (.ush/.usf), PBR, Niagara

Rendering: Deferred/Forward Rendering, Lumen Reflections, VSM Optimisation, Lighting Pipelines, Ray Tracing, GPU/CPU Optimisation, Post-Processing, RDG / Render Graph Management

Tools & Profiling: Visual Studio, Git/GitHub, Perforce (p4), RenderDoc, PIX, Nvidia Nsight, PlayStation Razor GPU/CPU, Xbox Debugging Tools, ImGui, Slate, Windows Display API

LANGUAGES

Turkish (native) | **English** (fluent)