

NIELS BRUNEKREEF

Senior graphics programmer

ABOUT

📍 Netherlands · nielsbishere.com ·
github.com/Nielsbishere · n@osomi.net

I'm a low-level graphics programmer with over 6 years of experience focused on real-time rendering, D3D12/DXR engine development, shader tooling, and cross-platform graphics.

At Lumion, I led large parts of the migration from a legacy Effects11 codebase to modern **DXR**, built custom shader tooling and performance systems, and shipped multiple DXR features such as **RT glass**, **grass**, **water**, **GI**, **shadows** and **reflections**.

Outside of work, I build my own Vulkan/D3D12 framework (running on Android, Linux, macOS, and Windows), experiment with emulation, and contribute to open-source projects.

INTERESTS

Raytracing, shader compilers, graphics R&D, emulation, baking, gardening, building PCs, modding/playing games.

SKILLS

Graphics APIs

D3D12, DXR, VK, OGL, D3D11, D3D9

Programming languages

C, C++, HLSL, C#, GLSL, ARM9 ASM

Tools

Git(Hub), VS(Code), Trello, Jira, Pix, SVN, CMake, NSight, RenderDoc, P4V

Engines

Unreal, Unity, Quest3D (Lumion)

EARLIER (<2018)

- **Mineblowers (Team Moose)**: Built spline tools and gameplay logic for a custom-controller Unreal game (study).
- **Rail Recon (Triggered studios)**: Prototyped mechanics, and VFX for a Steam-released Unreal project (study).
- **Rom hacking & Gameboy (Advance) emulation**: Tooling and emulation in C++; parsing NDS formats and exploring emulating ARM7TDMI & Z80.
- **Minecraft modding & plugins (2012 – 2016)**: Created gameplay mechanics, and server plugins in Java using Bukkit/Forge.

EDUCATION

Breda University of Applied Sciences

BSc. CMGT International Game Architecture and Design (2016 – 2020)

EXPERIENCE

Lumion / Quest3D – Archviz Engine

Intern → Junior → Senior Graphics Programmer (2019–Present)

Architecture visualization software used by hundreds of thousands of architects worldwide. Worked on the modernization of Lumion's renderer (**FX11**→**DXR**). Focused on raytracing, shaders, GPU based systems and performance.

DXR 1.1 & D3D12 Integration:

- DXR 1.1: Built support for **BLAS/TLAS**, **shader binding tables**, **raytracing state objects**, **inline raytracing** and **raytracing collections**.
- Unify RT + Raster: **bindless rendering**, **visibility buffer** and **instance management**. Moved from CPU to GPU due to heavy visual scripting overhead.
- Denoising via **NRD** (ReLAX, ReBLUR, Sigma) and **OIDN/OIDN2**.
- Integrated **streamout** for migrating tessellated trees into static geometry.
- Ported D3D9 techniques to D3D12 using pipeline state factories and by managing descriptor sets & constant buffers.

R&D: Advanced Raytracing Features

- Prototyped **GPU-based kD tree builds** for **raytraced caustics** (photon mapping).
- Developed support for effects such as **raytraced water**, **grass**, **glass**, **transmission**, **soft shadows**, **GI**, **reflections**, **clip planes** and **colored shadows** (fake caustics).
- Experimented with **improved raytraced grass**, **camera types (VR, 360, ortho, etc.)**, **DoF** and **animated characters** with dynamic BLAS updates.

Shader Infrastructure & Tooling

- Developed an **HLSL parser** to convert 600+ legacy shaders (SM4/5) to **SM6+**.
- Improved the **shader system** by adding includes, binaries on disk, and maintained a newer parser for missing DXC reflection.
- Faster **offline shader compilation** (4.3x hit, 2.7x miss, 3.2x raygen and 1.6x gfx shader speedups) and **multi-threaded state object creation**.
- Isolated DXC into a separate process to prevent editor crashes.

Performance, Debugging & Maintenance

- Enabled Shader Execution Reordering (**SER**) & Moved code from raygen to hit shaders → ~5x on RTX 40xx, ~3x on AMD, ~2x on previous-gen Nvidia.
- Fixed large-scene transition bottlenecks (0–1 FPS → 50+ FPS for ~2k mesh scenes).
- **FSR2** and motion vectors for TAA and/or upscaling.
- **GPU memory inspection**, **timing/debug markers**, **crash diagnostics**.
- Reported and debugged GPU driver issues across **Nvidia**, **AMD**, and **Intel**.

Open source contributions:

- **microsoft/DirectXShaderCompiler**: Minor bug fixes (SV_PrimitiveID, -fvk-invert-y in lib), fix compile with new DirectX-Headers, expose DXIL reflection info.
- **microsoft/DirectX-Headers**: Extended ID3D12(Shader/Function)Reflection1.
- **tgjones/HslTools**: 16-bit and 64-bit types (scalar, vector, matrix).
- **MicrosoftDocs/win32**, **KhronosGroup/SPIRV-Tools**: Minor typos.

Project Wisp – real-time raytracing | Graphics & optimization

October 2018 – December 2019

Student project: Real-time raytracing for Maya viewport from the ground up. Responsible for: rt reflections, parenting, culling, deferred rendering, fixes.

Voxel engine | Graphics & optimization (May – June 2018)

Student project: built a GPU-driven voxel particle system and tool for the Switch; implemented culling in compute and multi-draw indirect to reduce overhead.