

# CHINOMSO NOSIRI

Graphics Programming,  
Full-stack Developer,  
C++ Programming.

Email: [thenosirin@gmail.com](mailto:thenosirin@gmail.com)  
GitHub: [TheNosiriN \(Chinomso Nosiri\)](#)  
Linkedin: [Chinomso Nosiri](#)

## OVERVIEW

Experienced software engineer skilled in real-time 3D application, game, and web development with 4+ years of expertise. Proficient in C/C++, JavaScript, TypeScript, Node.js, Shaders. Specialized in frontend and backend development, adept at leveraging modern technologies like WebGPU, Vulkan and D3D12. Proven track record of collaboration in multidisciplinary teams, adhering to industry standards, and utilizing agile methodologies for project success.

## WORK EXPERIENCE

**Graphics Developer** August 2024 - Present  
**Cosmographics LLC**

**Software Developer** July 2023 - Feb 2024  
**PCloud Innovations**

- Developed full stack applications, handling both frontend and backend tasks using MySQL, C#, Node.js, and TypeScript.
- Leveraged WebGPU, TypeScript, and C# to render extensive datasets from manufacturing plants.
- Adhered to engineering and industry standards while reconstructing plant data.
- Managed the codebase, tasks, and features efficiently within the Microsoft Azure DevOps environment.

**Graphics programmer** Oct 2022 - Oct 2023  
**Wizards of the Coast (Hasbro)**

- Collaborated closely with team members in the production of games for Wizards of the Coast.
- Utilized C++, C#, and HLSL to develop core rendering pipelines and external tools.
- Employed Git for version control, reviewed pull requests, and maintained a production-standard codebase.
- Managed daily tasks, documentation, and project release schedules using Asana and Confluence.

**Graphics programmer** Sep 2021 - Dec 2021  
**Freelance Contract Employment**

- Created generative shader art using advanced ray marching techniques.
- Investigated real-time volumetric rendering and optimized performance.
- Explored procedural generation and physically based path tracing algorithms.
- Reverse-engineered complex shaders across various languages.

## PROJECTS

**Full scale procedural Universe (C++, GLSL, OpenGL, Vulkan)**

- Use of 64 bit doubles and grid-fract pair vectors for nested coordinate systems for celestial objects.
- Rendering of galaxies, stars, planets, and other celestial objects at real scale distances and sizes.
- Procedural planetary terrain generation using GPU Compute based Icosahedron LOD with custom stack recursion.
- Physically based procedural generation of stellar systems using official NSSDCA planetary data.

**The Hexo Engine (C, C++, Python, GLSL, CMake, OpenGL ES, DirectX 12, Vulkan) (WIP)**

- Implementation of asynchronous data driven rendering pipelines using a task and worker pool approach.
- Integration of modern graphics programming techniques, such as Fast volume rendering, DXR hybrid rendering (WIP), and data driven Meshlets rendering (WIP).
- Abstraction of low level platform dependent functionality, Filesystem, Memory Paging, Threading, Input, and Audio.
- Abstraction of both old and modern graphics APIs such as D3D12, Vulkan, OpenGL ES 3.1, and WebGPU.

**Shader programming (GLSL), Shadertoy @TheNosiriN**

- Ray marching Signed Distance Functions to create procedural scenery for generative and scalable art
- Signed Distance Function modeling using applied mathematics.
- GPU Ray tracing, rendering light transport and Implementing physically based BSDFs.

**BabylonJS Planets (JavaScript, WebGL 2, GLSL)**

- Large-scale planet rendering performed in WebGL 2, Babylon.JS and GLSL.
- Efficient use of Web Workers to generate large scale Icosahedron LOD trees from implicit data.
- Use of WebGL 2's transform feedback in appropriate places of the generation pipeline.
- Real time implementation of atmosphere rendering models.

**Iso-surface Extraction (JavaScript, GLSL, C#, HLSL, C++, Unity)**

- Fast implementation of the Marching Cubes algorithm in Babylon.JS, Unity and C++ (closed source).
- Use of Web Workers in JavaScript to polygonise Iso-surface density fields.
- Use of GPU Compute (C++) and Transform Feedback (WebGL 2) to generate density fields.

**EDUCATION**

---

**Ontario Tech University (UOIT). Oshawa, ON, Canada.**

Degree: Bachelor of Engineering (Honors) in Software Engineering