

# 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 5+ 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

<b>Software Developer</b> <b>PCloud Innovations</b> <ul style="list-style-type: none"><li>Worked as a full stack developer, both frontend and backend, using MySQL, C#, Node.js, Typescript, etc.</li><li>Used WebGPU, TypeScript, and C# to render incredibly large datasets of manufacturing plant data.</li><li>Conformed with engineering and industry standards when reconstructing plant data.</li><li>Used Microsoft Azure DevOps environment to efficiently manage the codebase, work, and features.</li></ul>	<b>July 2023 - Feb 2024</b>
<b>Graphics programmer</b> <b>Wizards of the Coast (Hasbro)</b> <ul style="list-style-type: none"><li>Collaborated with team members and played a crucial role in the production of games under Wizards of the Coast.</li><li>Used C++, C#, and HLSL to develop fundamental rendering pipelines, and external tooling.</li><li>Used Git version control, reviewed pull requests, and managed a production standard codebase.</li><li>Used tools like Asana and Confluence to manage daily tasks, documentation, and project release schedules</li></ul>	<b>Oct 2022 - Oct 2023</b>
<b>Graphics programmer</b> <b>Freelance Contract Employment</b> <ul style="list-style-type: none"><li>Created generative shader art using different ray marching techniques.</li><li>Explored real-time volumetric rendering and many optimization techniques for performance.</li><li>Explored procedural generation, and physically based path tracing algorithms.</li><li>Reverse engineering of complex shaders in different languages.</li></ul>	<b>Sep 2021 - Dec 2021</b>
<b>Project Intern</b> <b>Urban Rez Solutions</b> <ul style="list-style-type: none"><li>Research and Compilation of entertainment medium for the Urban Rez motorhome project.</li><li>Collaborated with colleagues to normalize our research to be inline with project objectives.</li><li>Attended work related seminars on the progress of the Urban Rez motorhome project.</li></ul>	<b>July 2021 - Oct 2021</b>

## PERSONAL PROJECTS

<b>Full scale procedural Universe (C++, GLSL, OpenGL, Vulkan) (WIP)</b> <ul style="list-style-type: none"><li>Use of 64 bit doubles and grid-fract pair vectors for nested coordinate systems for celestial objects.</li><li>Rendering of galaxies, stars, planets, and other celestial objects at real scale distances and sizes.</li><li>Procedural planetary terrain generation using GPU Compute based Icosahedron LOD with custom stack recursion.</li><li>Physically based procedural generation of stellar systems using official NSSDCA planetary data.</li></ul>
<b>The Hexo Engine (C, C++, Python, GLSL, CMake, OpenGL ES, D3D12, Vulkan) (WIP)</b> <ul style="list-style-type: none"><li>Implementation of asynchronous data driven rendering pipelines using a task and worker pool approach.</li><li>Integration of modern graphics programming techniques, such as Fast volume rendering, DXR hybrid rendering (WIP), and data driven Meshlets rendering (WIP).</li><li>Abstraction of low level platform dependent functionality, Filesystem, Memory Paging, Threading, Input, and Audio.</li><li>Abstraction of both old and modern graphics APIs such as D3D12, Vulkan, OpenGL ES 3.1, and WebGPU.</li></ul>
<b>Shader programming (GLSL), Shadertoy @TheNosiriN</b> <ul style="list-style-type: none"><li>Ray marching Signed Distance Functions to create procedural scenery for generative and scalable art</li><li>Signed Distance Function modeling using applied mathematics.</li><li>GPU Ray tracing, rendering light transport and Implementing physically based BSDFs.</li></ul>

**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