

## 3D graphics programmer

---

### Experience

- **Lum Engine** — [Repository](#) (C++17)
  - Created a Vulkan C++ voxel engine from scratch, focusing on performance, with fully dynamic global illumination, ray-traced reflections, and modern rendering pipelines
  - **Software Voxel Raytracer (RaVE)** — [Repository](#) (C99)
    - Developed a CPU multithreaded SIMD raytracer, designed for simple integration in any voxel engine
  - **Deferred Renderer**
    - Implemented a subpass-based deferred renderer, achieving high performance on Tile-Based GPUs through subpass management and custom compression algorithms.
  - **Radiance Field GI**
    - Created a real-time, fully dynamic global illumination system for low-frequency light using custom ray-tracing algorithm and acceleration structure
  - **Reflections**
    - Developed a real-time raytraced reflections system for glossy surfaces
  - **Volumetrics Renderer**
    - Designed a high-performance screen-space volumetric renderer based on Lambert's law and 3D Perlin noise for realistic effects.
  - **Foliage Renderer**
    - Engineered a GPU-driven foliage renderer, capable of rendering hundreds of thousands of grass blades in hundreds of microseconds".
  - **Realtime Denoiser** (Currently unused)
    - Developed an edge-avoiding Å-trous wavelet-based spatial filtering algorithm for efficient low spp path-traced global illumination denoising.
- **Lum-al** — [Repository](#) (C++ Vulkan)
  - Designed a low-level Vulkan library optimized for high-performance applications with a simple and efficient architecture.
  - **Vulkan Resources Management**
    - Reduced complexity by cutting useless for game engines features, enabling automated resource management for streamlined performance.
  - **CPU Synchronization**
    - Implemented Frames In Flight, utilizing ring buffers for every CPU-GPU resource to enhance performance while keeping synchronization easy

- **Mangaka** — [Repository](#) (C++ Vulkan)
  - Developed a manga-style renderer using Lum-aI, achieving fast, high-quality rendering of stylized content.
  - **Outline Rendering**
    - Sobel-filtered normal & depth buffers for efficient outline rendering via discontinuity detection.
  - **Ben-Day Dots**
    - Designed a math-driven, software multi-sampled dot rendering algorithm for Manga shading effects.
  - **GLTF**
    - Implemented GLTF file support.
- **Assembler** — [Repository](#) (C99)
  - Created a CPU emulator with a custom instruction set, registers, memory and visual output to learn more about real-world CPUs
- **SL-Vec** — [Repository](#) (C23)
  - Designed a macro library for GLSL vector types, casts, and functions in C23.
- **Fractal Raymarcher** — [Repository](#) | [Live Demo: click button in bottom-right](#) (JavaScript)
  - Implemented a WebGL 4D Julia set (fractal) renderer for raymarching

## Awards & Honors

**Gold Medalist** — International Al-Farghani Physics Olympiad (IAFPhO), 2021

## Education

**Moscow Institute of Physics and Technology (MIPT)** — Applied Mathematics and Physics  
 2022 - 2023 (*completed 1 year*)