

## Professional Summary

Graphics and game development programmer with ~3 years of experience focusing on low-level game engine programming. Background in physics competitions

---

## Programming Experience

### *Lum* Game Engine

(mostly Vulkan renderer)

- **Sparse voxel block grid real-time path tracer** (part of *Lum* engine) Implemented without VK\_KHR\_ray\_tracing\_pipeline, using only compute shaders
- **Spatiotemporal A-trous real-time filter** (part of *Lum* engine) Developed a denoiser and accumulator (through motion vectors)
- **Profile-guided upscaler** (part of *Lum* engine) Created an upscaler that performs better than FSR (for voxels)
- **Distance field Builder and path tracer** (initially part of *Lum* engine) Created linear time sdf builder and pretty efficient sdf marcher

### Assembler language

- Designed and implemented CPU emulator with custom ASM language - compiler and interpreter

### WebGL2 fractal raymarcher

- Implemented Julia set ("3D fractal bulb") renderer developed as the background for my portfolio [site](#) on GitHub

### OpenGL path tracer (*true\_mmo*)

- Developed a path tracer in OpenGL (for textured-blocks-only) with simple physics engine (convex shape intersection + Newton laws of motion) and server side (*true\_mmo-server*)

### minor experience:

- Libraries ranging from a "safe stack" to a macro's monster for GLSL-style linear algebra in pure C
- 

## Education

### Incomplete Degree in applied mathematics and physic

### Moscow Institute of Physics and Technology (MIPT)

---

# Skills

- Primary
  - **C & C++ core**
  - **C & C++ Build systems & toolchains**
  - **ASM and hardware relationships with languages**
  - **Vulkan**
  - **OpenGL/WebGL**
  - **ray-tracing**
    - physically correct rendering
    - brdf
  - **in-depth Physics knowledge**
- Secondary
  - **GLSL**
    - subgroup operations
  - **shader effects**
    - color mapping, noise, dithering, bloom and others
  - **Python**
    - network data scrapping and analytics, "everyday" scripts
  - **C/C++ Code Generation (Macros)**
    - C macro system
  - **Networking**
    - KCP / TCP / UDP
  - **Git**
  - **CSS/HTML**
    - also RmlUi (Qt but faster) markup & styling languages, they are pretty similar