Platon Vinnichek

platonvin@gmail.com

3D graphics programmer

- **Lum Engine** — Repository (C++23)
  - Developed a high-performance voxel renderer using Vulkan, delivering fully ray-traced real-time dynamic global illumination (GI)
  - Engineered a SIMD-optimized, multithreaded CPU raytracer for voxel scenes, designed for seamless integration into graphics engines — Repository (C99)
  - Implemented a subpass-based deferred rendering system, optimized for Tile-Based GPUs with advanced compression techniques, achieving significant performance gains in complex scenes in comparison with common methods
  - Designed a real-time GI system utilizing a custom ray-tracing algorithm and acceleration structure, providing dynamic low-frequency light simulation
  - Integrated full-res ray-traced reflections for real-time rendering of glossy surfaces
  - Developed a dynamic quality screen-space volumetric renderer incorporating Lambert's law and 3D Perlin noise, resulting in realistic volumetric lighting effects with constant runtime
  - Created a GPU-driven foliage rendering system, capable of efficiently rendering hundreds of thouthands grass blades in hundreds of microseconds
  - Implemented a state-of-art A-trous spatio-temporal denoising algorithm for filtering GI, achieving noise reduction in <1 spp path-traced scenes

- **Lum-al** — Repository (C++ Vulkan)
  - Architected a high-performance Vulkan framework, targeting init-time resource definitions
  - Simplified resource management by applying specific usecase restrictions, resulting in an simple and lightweight system satisfying Lum requirements
  - Implemented a generic CPU-GPU resource synchronization system, allowing preventing GPU stalls (aka parallelism)

- **Circuli-Bellum** — Repository (C++ Vulkan)
  - Developed ROUNDS clone in C++ Vulkan (with Lum-al), outperforming original game by an order of magnitude
  - Engineered low-overdraw primitive shape rasterization algorithm with infinite antialiasing quality
  - Designed fully GPU-driven precise 1D shadow technique while avoiding data duplication
  - Implemented high-perfomance bloom and Chromatic abberation effects for better visuals

- **Mangaka** — Repository (C++ Vulkan)
    - Developed a manga-style renderer utilizing Lum-al, capable of fast, high-quality stylized graphics suitable for manga-comics-style and animation
    - Implemented outline rendering using Sobel-filter for normal and depth buffers, enabling accurate edge detection and stylized effects
    - Engineered a mathematically-driven, multi-sampled dot and hatches rendering algorithms for traditional Manga shading look
    - Created GLTF loader for easy integration with modern 3D workflows
- **Might and Chess** — Repository (Unity Game Engine)
    - Designed demo for mix of powerup-card-based game and classic chess
    - Implemented scalable chess engine
- **Assembler** — Repository (C99)
    - Developed a CPU emulator (Interpreter + compiler) with a custom instruction set, register architecture, and DOS-like drawing capabilities
- **Fractal Raymarcher** — Repository | Live Demo: click chevron on "Fractal Raymarcher" (JavaScript)
    - Created a WebGL-based renderer for 4D Julia set fractals, utilizing different math-based techniques for distance field estimation, coloring and normals

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)