

Tony Yiding Tian

Philadelphia, PA | (267)249-1202 | tonygt@seas.upenn.edu | tonyxtian.com

EDUCATION

University of Pennsylvania - School of Engineering and Applied Sciences Philadelphia, PA
B.S.E. in Computer Engineering, Accelerated M.S.E. in Computer Graphics and Game Technology May 2027
• GPA: 3.74 — Relevant Courses: GPU Programming, Advanced Rendering, Interactive Computer Graphics, Computer Animation, Operating Systems, Data Structures & Algorithms, Computer Architecture

PROJECTS

CUDA Path Tracer | *CUDA, C++* Sep 2025 - Oct 2025
• Monte Carlo path tracer capable of rendering complex 3D scenes with custom 3D models and environment maps
• Implemented shading BSDF kernel supporting global illumination, multiple importance sampling, anti-aliasing, sub-surface scattering, capable of rendering various PBR material types with albedo and texture maps
• Integrated third-party libraries of tinyGLTF to support glTF 2.0 mesh loading and Nvidia OptiX for denoising
• Utilized various techniques to boost performance: material sorting (+5%), Russian Roulette (+6% - 24%), stream compaction (+24% - 67%), and Bounding Volume Hierachy ($3\times$ - $160\times$ framerate in complex scenes)
• Project Repo and Demo: github.com/tonygt/CUDA-Path-Tracer. A previous standalone performance focused stream compaction project with detailed analysis in Nsight: github.com/tonygt/Project2-Stream-Compaction

WebGPU Renderer | *WebGPU, TypeScript, WGSL* Sep 2025 – Oct 2025
• Implemented three advanced rendering techniques for real-time lighting of 5000+ dynamic point lights: Naive Forward, Forward+, and Clustered Deferred rendering using WebGPU compute and graphics pipelines
• Engineered screen-space light clustering system using compute shaders, subdividing view frustum into $16\times 9\times 24$ grid with exponential depth slicing and sphere-AABB intersection testing for efficient light culling
• Developed G-buffer architecture with 3 render targets (position, normal, albedo) enabling two-pass deferred rendering that decouples geometry complexity from lighting calculations
• Achieved 53x performance improvement over naive rendering (497ms to 9.3ms at 5000 lights) and 3.5x speedup vs Forward+ through overdraw elimination and optimized memory access patterns
• Built automated performance testing system collecting statistical frame time data across 30 configurations (3 renderers \times 10 light counts), generating CSV analysis for rigorous benchmarking
• Live demo deployed at webgpu.tonyxtian.com, rendering Sponza scene with real-time performance metrics and interactive controls. GitHub repo: github.com/tonygt/Project4-WebGPU-Forward-Plus-and-Clustered-Deferred

Mini Minecraft - Voxel-based 3D Game | *C++, GLSL, Qt* Oct 2024 – Dec 2024
• Collaborated in team of 3 to develop fully-featured voxel game engine in C++ using OpenGL, generating infinite worlds with 1M+ blocks and maintaining 60+ FPS performance
• Engineered procedural terrain generation system using layered 2D/3D Perlin noise algorithms, creating 5 distinct biomes (Grassland, Mountain, Desert, Islands, Caves) with biome-specific block distributions and procedurally placed vegetation assets
• Implemented post-processing rendering pipeline with custom GLSL fragment shaders, featuring dynamic underwater/lava distortion effects using UV coordinate manipulation and real-time crosshair overlay rendering
• Developed dual physics simulation system: gravity-based collision detection with terrain for ground movement, and buoyancy calculations for water/lava interaction, plus creative fly-mode with 6-DOF movement
• Implemented real-time block manipulation (mining/placing) with ray-casting intersection testing and immediate mesh updates, supporting 16 different block types with unique textures and properties
• Project demo showcasing all features: youtu.be/jRb4EHV5KQI

EXPERIENCE

Linux Kernel Policies Research Assistant - PURM Scholar May 2025 – Aug 2025
Learning Directed Operating System (LDOS), Prof. Sebastian Angel Philadelphia, PA
• Developed eBPF-based kernel monitoring infrastructure collecting real-time TCP networking metrics retrieved from 5 crucial kernel tcp functions of `tcp_v4_rcv`, `v4_connect`, `state_process`, `congestion_control`, and `cubic`
• Engineered high-performance data analysis pipeline processing 10,000+ TCP state transitions per second, identifying critical performance bottlenecks in kernel networking policies and congestion control algorithms
• Architected and contributed 2000+ lines of C and Python code to open-source KernMLOps repo, implementing kernel probing infrastructure used by 15+ researchers. GitHub Repo: github.com/tonygt/KernMLOps

TECHNICAL SKILLS

Graphics/Rendering: Nsight Profiling, Path Tracing, Deferred Rendering, Rasterization, Animation systems, PBR
Programming: CUDA, C/C++, WGSL, GLSL, Parallel algorithms, Memory management, Rendering pipeline
Tools/APIs: Nvidia Nsight, WebGPU, Vulkan, Visual Studio, Qt, OpenGL, Git, CMake, MakeFile, Clang, GDB