

Tony Yiding Tian

Philadelphia, PA | (267)249-1202 | tonytg@seas.upenn.edu | github.com/tonytrgt

EDUCATION

University of Pennsylvania - School of Engineering and Applied Sciences

Philadelphia, PA

Computer Engineering, B.S.E.

May 2027

- GPA: 3.74 — Relevant Courses: GPU Programming, Operating Systems Design & Implementation, Data Structure & Algorithm, Advanced Rendering, Interactive Computer Graphics, Computer Animation, Embedded Systems

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× - 160× framerate in complex scenes).
- Project Repo and Demo: github.com/tonytrgt/CUDA-Path-Tracer. A previous standalone performance focused stream compaction project with detailed analysis in Nsight: github.com/tonytrgt/Project2-Stream-Compaction.

PennOS - UNIX-like Operating System | *C, Shell, Kernel*

Mar 2025 – May 2025

- Architected and implemented a complete user-level operating system in C with team of 4, featuring 8000+ lines of systems code with full process lifecycle management
- Designed Process Control Block (PCB) data structure managing 50+ concurrent processes with metadata including PID allocation, priority levels, parent-child relationships, signal handling, and user/kernel stack management
- Implemented preemptive multi-level priority scheduler supporting 3 priority levels with Round Robin time-slicing (10ms quantum), preventing starvation through priority aging and achieving 95% CPU utilization
- Built POSIX-compliant interactive shell supporting 15+ built-in commands (ps, kill, jobs, fg/bg), I/O redirection, pipeline chaining, and batch script execution with robust error handling

Mini Minecraft - Voxel-based 3D Game | *C++, OpenGL, Qt*

Oct 2024 – Dec 2024

- Collaborated in team of 3 to develop fully-featured voxel game engine in C++ using OpenGL in Qt
- 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
- 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
- Project demo showcasing all features: [Game-Demo](#)

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 repository](#), implementing kernel probing infrastructure used by 15+ researchers

Beta Test Engineer & QA Analyst

Dec 2022 – Oct 2023

miHoYo - Genshin Impact (AAA Mobile Gaming)

Remote

- Selected as exclusive beta tester for Genshin Impact, a \$4B+ revenue mobile game with 60M+ monthly active users, participating in pre-release testing cycles every 6 weeks
- Conducted comprehensive quality assurance testing for 10+ character releases, each generating \$1M+ in revenue, ensuring gameplay balance and identifying critical performance issues before public launch
- Performed systematic testing of open-world gameplay mechanics including combat systems, puzzle design, and performance optimization across multiple mobile platforms (iOS, Windows)

TECHNICAL SKILLS

Programming: CUDA, C/C++, GLSL, Parallel algorithms, Memory management, Rendering pipeline

Graphics/Rendering: Nsight Profiling, Path Tracing, Deferred Rendering, Rasterization, Animation systems, PBR

Tools/APIs: Nvidia Nsight, Visual Studio, Qt, OpenGL, WebGPU, Git, CMake, MakeFile, Clang, GDB, GCC