

Tony Yiding Tian

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

EDUCATION

University of Pennsylvania - School of Engineering and Applied Sciences

B.S.E. in Computer Engineering, M.S.E. in Computer Graphics and Game Technology

Philadelphia, PA

May 2027

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

EXPERIENCE

Linux Kernel Policies Research Assistant - PURM Scholar

May 2025 – Present

Learning Directed Operating System (LDOS), Prof. Sebastian Angel

Philadelphia, PA

- Selected as Penn Undergraduate Research Mentorship (PURM) 2025 Scholar, conducting advanced research in distributed systems and kernel optimization under Prof. Sebastian Angel
- 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 tcp kernel probing infrastructure used by 15+ researchers
- Presented research findings on TCP optimization to distributed systems lab, leading to 25% improvement in network throughput for target applications

Product Development Researcher & UX Designer

Jan 2025 – May 2025

YesTech Corporation - Best Friends Network Social Platform

Remote

- Led user-centric product development team of 6 for startup's flagship social media application targeting Gen-Z demographic with 10,000+ beta users
- Architected and designed "Friendship Portal" core feature enabling real-time mood sharing between connected users, driving 40% increase in daily active engagement
- Conducted comprehensive user research including A/B testing across 500+ users, survey design and analysis, and usability studies resulting in 3 major feature iterations
- Collaborated directly with mobile development team to implement data-driven UI/UX improvements, reducing user onboarding time by 35% through iterative design optimization
- Established quality assurance protocols for mobile app testing across iOS and Android platforms, identifying and documenting 50+ critical bugs pre-release

Guest Service Attendant

May 2024 – Aug 2025

Residential and Hospitality Services, University of Pennsylvania

Philadelphia, PA

- Managed large-scale residential operations supporting 1000+ high school students during summer programs, implementing streamlined check-in processes that reduced average wait time by 45%
- Coordinated resource allocation and inventory management for essential residential supplies (temporary credentials, equipment carts, linens), optimizing distribution efficiency by 30%
- Provided technical customer support and troubleshooting for residents including minors, maintaining 95% customer satisfaction rating through effective communication and problem-solving
- Collaborated with facilities management team to resolve technical issues across multiple building systems, reducing average incident response time from 30 to 15 minutes

Beta Test Engineer & Quality Assurance 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)
- Documented and reported 100+ gameplay bugs and performance issues through structured testing protocols, contributing to 15% reduction in post-launch critical bugs
- Provided detailed feedback on character design, combat mechanics, and user experience, influencing design decisions for characters with combined revenue exceeding \$10M

PROJECTS

MatForge - Advanced Material Rendering System Vulkan 1.3, Slang, C++	Nov 2025 – Dec 2025
<ul style="list-style-type: none">Architected production-quality GPU path tracer implementing four SIGGRAPH papers (2023-2024) in unified Vulkan ray tracing pipeline, contributing 2,400+ lines of C++ and Slang shader code for sampling and geometry systemsImplemented Quad-Optimized Low-Discrepancy Sequences (QOLDS) from SIGGRAPH 2024, engineering base-3 Sobol' generator with Owen scrambling across 47 dimensions, achieving +2.57 dB PSNR and 44.7% MSE reduction vs PCG sampling at 512 SPP with ~1% performance overheadDeveloped RMIP (Rectangular MinMax Image Pyramid) intersection shader from SIGGRAPH Asia 2023, enabling tessellation-free displacement mapping through hierarchical texture-space ray traversal with custom Vulkan intersection shadersEngineered automated convergence analysis framework generating statistical benchmarks across sampling configurations, validating rendering quality improvements through PSNR/SSIM metrics and CSV-based visualization pipelineIntegrated techniques into complete Monte Carlo path tracing pipeline supporting glTF 2.0 scenes, HDR environment maps, and KHR_materials_displacement extension for physically-based rendering	
Vulkan Grass Renderer Vulkan, GLSL, C++	Oct 2025 – Nov 2025
<ul style="list-style-type: none">Implemented real-time grass rendering system in Vulkan based on SIGGRAPH I3D 2017 paper, simulating up to 1 million grass blades with physics-based animation at interactive frame rates using compute and tessellation pipelinesDeveloped GPU compute shader for physics simulation applying gravity, Hooke's law recovery forces, and wind with spatial turbulence to quadratic Bezier curve grass blades, processing 1M+ blade state updates per frameEngineered three-tiered GPU culling system (orientation, view-frustum, distance-based probabilistic) achieving 4.31× performance improvement at 1M blades through efficient atomic operations and indirect draw callsBuilt hardware tessellation pipeline with dynamic LOD based on camera distance, generating smooth blade geometry with exponential falloff from 10 to 2 subdivisions for optimal geometry densityCreated comprehensive ImGui control panel with real-time FPS metrics (average, 1% lows), frame time graphs, and automated performance testing system generating CSV benchmarks across 40 configurations	
WebGPU Renderer WebGPU, TypeScript, WGSL	Sep 2025 – Oct 2025
<ul style="list-style-type: none">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 pipelinesEngineered 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 cullingDeveloped G-buffer architecture with 3 render targets (position, normal, albedo) enabling two-pass deferred rendering that decouples geometry complexity from lighting calculationsAchieved 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 patternsBuilt automated performance testing system collecting statistical frame time data across 30 configurations (3 renderers × 10 light counts), generating CSV analysis for rigorous benchmarkingLive demo deployed at webgpu.tonyxtian.com, rendering Sponza scene with real-time performance metrics and interactive controls	
CUDA Path Tracer CUDA, C++	Sep 2025 - Oct 2025
<ul style="list-style-type: none">Monte Carlo path tracer capable of rendering complex 3D scenes with custom 3D models and environment mapsImplemented 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 mapsIntegrated third-party libraries of tinyGLTF to support glTF 2.0 mesh loading and Nvidia OptiX for denoisingUtilized 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/tonytgrt/CUDA-Path-Tracer. A previous standalone performance focused stream compaction project with detailed analysis in Nsight: github.com/tonytgrt/Project2-Stream-Compaction.	
Monte Carlo Path Tracer & Real-Time PBR Renderer C++, GLSL	Jan 2025 – May 2025
<ul style="list-style-type: none">Implemented Monte Carlo path tracer supporting Cornell Box, glass refraction, microfacet materials, and environment mappingDeveloped physically-based rendering (PBR) shader pipeline with albedo/metallic/roughness maps achieving real-time ray tracing on modern GPUs with 60+ FPS real-time performance	

- Engineered custom BRDF models for materials including chrome, plastic, and complex surface properties
- Demo: <https://github.com/tonytgrt/TonyTianRenderDemo>

T&T Slots - Biometric-Adaptive Gaming System | ATMega328PB, Bare Metal C Mar 2025 – May 2025

- Engineered embedded gaming system integrating biometric feedback with real-time probability adjustment, featuring custom ATMega328PB firmware managing 3 concurrent hardware subsystems
- Developed custom I2C communication protocol for MAX30102 SPO2 sensor achieving 100Hz heart rate sampling with 95% accuracy, implementing digital filtering algorithms to eliminate motion artifacts
- Architected real-time graphics rendering engine for LCD display supporting 16-bit color depth, custom sprite animation system, and pixelized gameplay with memory-optimized frame buffering
- Implemented adaptive probability algorithm correlating heart rate variability to winning odds, dynamically adjusting game difficulty based on physiological stress indicators with 200ms response latency
- Built multi-modal user interface featuring PWM-driven audio synthesis for dynamic sound effects, tactile button controls with debouncing, and visual feedback systems
- Optimized embedded system performance achieving 1% CPU idle time through interrupt-driven architecture and efficient memory management across 2KB SRAM
- Links: [Project Website](#) — [GitHub Repository](#)

PennOS - UNIX-like Operating System | C, Make, GDB 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
- Developed comprehensive system call interface including p_spawn(), p_wait(), p_sleep(), signal delivery (SIGTERM, SIGSTOP, SIGCONT), and file I/O wrappers with proper resource cleanup and zombie process reaping
- Implemented job control and signal handling system supporting background process management and graceful process termination across parent-child hierarchies

Mini Minecraft - Voxel-based 3D Game | C++, OpenGL, Qt, GLSL 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
- Built efficient chunk-based world management system with frustum culling and LOD optimization, reducing draw calls by 80% through face culling of adjacent blocks
- 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: <https://youtu.be/jRb4EHV5KQI>

TremorChecker - Parkinson's Disease Screening Device | Python, Raspberry Pi Jan 2021 – Jun 2023

- Architected non-invasive Parkinson's disease screening system using electrostatic field detection, achieving 85% correlation with clinical tremor assessments through advanced signal processing algorithms
- Implemented real-time tremor frequency analysis using Microchip MGC3030 3D gesture sensor and Raspberry Pi 4B, processing 130Hz spatial data with custom digital signal processing pipeline
- Developed frequency domain analysis to detect tremor signatures between 4-6Hz characteristic of Parkinson's
- Engineered automated diagnostic software generating comprehensive medical reports with tremor amplitude measurements, frequency analysis, and clinical risk assessment scores
- Conducted clinical validation study with 50+ participants in Shenzhen, achieving 82% diagnostic accuracy compared to neurological assessments, with false positive rate below 15%
- Applied Chinese utility patent and presented research at 2023 International Science and Engineering Fair (ISEF), competing among top 1% of global participants
- Built scalable IoT architecture supporting cloud data synchronization and remote monitoring capabilities for longitudinal patient tracking

TECHNICAL SKILLS

Programming Languages: C/C++, CUDA, GLSL, WGSL, Python, Java, eBPF, Assembly, JavaScript, LATEX

Graphics & Rendering: WebGPU, OpenGL, DirectX, Real-time Ray Tracing, Path Tracing, PBR, Procedural Generation

Systems & Tools: Linux Kernel Development, Docker, Qt, Git, CMake, Embedded Systems, TCP/IP

Frameworks & APIs: Qt, Raspberry Pi, ATMega, VMware, CloudLab, Jupyter Notebook