

Thomas Fong

tfong@engineering.upenn.edu | <https://www.linkedin.com/in/tzfong> | tzfong.com | (650) 660-0307

SUMMARY

Computer science and graphics programming student with experience in GPU APIs (OpenGL, Vulkan) for real-time rendering, physically-based offline rendering, shading languages (GLSL, Slang), operating system design, as well as Java, C, C++, and data structures and algorithms, seeking research and applied opportunities in software engineering.

EDUCATION

University of Pennsylvania, School of Engineering and Applied Sciences

Philadelphia, PA

Candidate for Bachelor of Science in Engineering, Digital Media Design (Graphics Programming)

May 2027

Candidate for Master of Science in Engineering, Computer Graphics and Game Technology

GPA: 3.99/4.00

Minor: Mathematics

Current Coursework

Computer and Network Security, Database and Information Security, GPU Programming, Machine Learning Mathematics

Past Coursework

Advanced Rendering, Computer Algorithms, Computer Animation Mathematics, Computer Graphics, Computer Systems, Discrete Mathematics, Data Structures and Algorithms, Linear Algebra, Multivariable Calculus, Operating Systems, Theory of Computing, UI/UX Design, 3D Modeling

TECHNICAL SKILLS

In Progress: Unity, HTML, CSS, JS

Current: Java, C, C++, Vulkan, Operating Systems Design, Slang Shading Language, CUDA, OCaml, Blender, Adobe Photoshop, Adobe Illustrator, OpenGL, GLSL, Figma, Python, QT, PyTorch

PROJECTS

Vulkan-CUDA Interop Pathtracer Graphics Engine | C++, CUDA, Slang, Vulkan

May 2026–Present

- Architected a hybrid rendering engine from scratch pairing a real-time Vulkan preview viewport with a CUDA Monte Carlo pathtracer, using CUDA-Vulkan interop to physically simulate light-material interaction.
- Eliminated host-side transfer overhead via zero-copy memory sharing, exporting Vulkan device memory as external OS handles mapped directly to CUDA external resources.
- Coordinated CUDA-Vulkan execution using external semaphores for cross-API synchronization.
- Built a bindless rendering system to allow material data to be shared across the Vulkan rasterizer and CUDA pathtracer.
- Modeled physically-based metallic/roughness materials with the Cook-Torrance BRDF and GGX microfacet distribution, producing convincing metal and plastic surfaces.
- Added glTF loading with host-side acceleration-structure and scene-graph construction.
- Implemented gamma correction and tone mapping (Reinhard, ACES, AgX) through Vulkan Compute.

GPU Boids Flocking Particle Simulator | CUDA, C++

May 2026–June 2026

- Implemented a GPU boids flocking simulation (separation, alignment, cohesion) in CUDA and C++.
- Cut per-frame cost with a uniform spatial grid to prune neighbor searches, raising throughput from ~19 to ~420 FPS on an RTX 4070 Mobile.
- Layered a coherent-memory optimization on the grid to improve memory access locality, reaching ~780 FPS (~40× speedup over the naive baseline).

Mini-Unix Kernel | C

March 2026–April 2026

- Built a Unix-like kernel in C with a priority-based round-robin scheduler and a kernel/syscall suite.
- Managed the full process lifecycle, zombie reaping and reparenting orphaned processes to a root process on parent exit.
- Added a FAT file system with memory block allocation and file I/O.
- Wrote a Bash-like shell with builtins (e.g. echo, cat, sleep) to drive the kernel and file system.

Physically-Based Deferred Real-Time Renderer | C++, OpenGL, GLSL

March 2026–April 2026

- Built an OpenGL deferred pipeline featuring PBR materials (Cook-Torrance BRDF), screen-space reflections, and diffuse/glossy convolution for real-time metal and plastic surfaces.
- Approximated subsurface scattering via raymarching.

OpenGL Monte Carlo Pathtracer | C++, OpenGL, GLSL

February 2026–March 2026

- Implemented a Monte Carlo physically-based pathtracer in C++ and GLSL, including global illumination with multiple importance sampling (MIS), sample-warping, and BXDF implementations for diffuse, specular, glass, reflective and transmissive materials.

OpenGL Mini-Minecraft | C++, OpenGL, GLSL

November 2025–December 2025

- Generated procedural biome-based terrain and caves.
- Rendered with shadow mapping, a procedural skybox, and post-processing.
- Authored vertex/fragment shader effects: swaying leaves, underwater distortion, and procedural grass coloration.

Mesh Editor | C++, OpenGL, GLSL

October 2025 – October 2025

- Built a mesh editor that loaded and stored obj files using the half-edge data structure, implementing features like vertex movement, face coloring, edge splitting, and Catmull-Clark subdivision.

HELL V. HARDWARE | Character Artist, Model Artist, Rigger, Animator

September 2025–November 2025

- Designed, modeled, textured, rigged, and animated Lilith, a playable character in Hell V. Hardware, a 3D fighting game in Unity, using Blender, Krita, Procreate, and Unity.

CPU Rasterizer | C++

September 2025

- Built a software 3D rasterizer with perspective projection, Z-buffering, and vertex attribute interpolation.
- Added FS anti-aliasing, wireframe mode, and Lambertian shading.

EXPERIENCE

University of Pennsylvania | Data Structures and Algorithms Teaching Assistant (CIS 1210)

August 2025 – Present

- Co-led weekly recitation session with ~12 students to bolster understanding of Java programming and essential computer science algorithms and data structures topics, such as runtime analysis, hash tables, and graph traversal algorithms.
- Held office hours to assist students with written and coding homework assignments, along with course content questions.

University of Pennsylvania | Discrete Mathematics Teaching Assistant (PACT)

June 2025 – July 2025

- Taught college-level mathematics for computer science, including combinatorics, graph theory, and discrete mathematics, to ~40 middle-high school students during the 1 month PACT (Program in Algorithmic and Combinatorial Thinking) program.
- Co-led weekly recitation sessions and office hours to bolster content understanding.

University of Pennsylvania | Communications Intern

June 2024 – March 2025

- Designed print NSO events calendar and inserts using InDesign, with thousands of copies printed and distributed for use by 2500+ students and parents during orientation.
- Managed Outlook email correspondence and newsletters with students and parents
- Designed logos, graphics, flyers, and brochures to brand, decorate, and promote NSO materials, social media accounts, programs, and events using Adobe suite software (InDesign, Illustrator, Photoshop).
- Edited video reels for use in social media posts.

HONORS & AWARDS

IEEE Eta Kappa Nu, Lambda Chapter | Engagement Chair

April 2025 – Present

- Invited to join as part of the top eighth in academic performance in the CIS department class of 2027.