Tongwei D.

716-622-9793 tongweid@gmail.com LinkedIn California, CA
Portfolio: https://dw218192.github.io/portfolio/

EDUCATION

University of Pennsylvania (GPA: 4.0/4.0) **PA, USA**
MSE in Computer Graphics and Game Technology Aug.2022 - Dec.2023(expected)
Core courses**:** *Physically based Rendering, GPU Programming, Procedural Generation, Computer Animation*

University of Southern California (GPA: 3.8/4.0) **CA, USA**
BS in Computer Science Aug.2017 - Jun.2021

SKILLS

- **Programming:** C/C++, Python, C#, CUDA, GLSL, HLSL, Maya MEL
- **Tools:** Visual Studio, Unity, Unreal Engine 4&5, Git, Maya, Houdini
- **Tech:** OpenUSD, NVIDIA Warp, Vulkan, OpenGL, WebAssembly, LLVM

WORK EXPERIENCE

System Software Engineer | NVIDIA ***Jan.2024 – Present***
- Developed various **synthetic data generation (SDG)** tools focused on character AI and crowd simulation in NVIDIA's Isaac Sim for training object detection and multi-object tracking models.
- Helped prototype and optimized the **MEGA Omniverse blueprint**, a comprehensive framework designed to simulate, test, and optimize robotic fleets within digital twins of industrial environments.
- Implemented CI/CD pipelines using **GitLab CI/CD** for new synthetic data generation projects, enabling automated builds, tests, and deployments for the first time, replacing manual and error-prone release processes.
- Implemented an **end-to-end testing framework** for SDG projects, ensuring data integrity, reproducibility, and robustness across multiple datasets and use cases.

Research Assistant, Software Developer Intern | Penn Computational Intelligence Lab ***May.2023 – Sep.2023***
- Incorporated Microsoft's **mimalloc** allocator into **WASI-SDK**'s C standard library.
- Implemented a multi-way segment tree to achieve O(log(n)) address-tag pair look-up for a custom kernel, achieving a **5x** speed boost with only **1.3x** worst-case memory overhead.
- Built a specialized memory allocator based on mimalloc for disaggregated computing applications.

Unity Engine Tool Programmer | NetEase Games ***May.2020 – Aug.2020***
- Used **Unity Editor API** to develop pipeline automation and scene object placement tools using **C#**, greatly improving the working efficiency of level designers.
- Refactored existing animation debugging tools, achieving approx. 50% performance gain and 100% less memory usage.
- Leveraged **compute shaders** to optimize terrain brush tools by transferring some workloads to GPU.
- Held regular meetings with the art team to gauge tool usage and collect performance data.

PROJECT EXPERIENCE

Differentiable Rendering for Material Optimization ***Nov. 2025 – Jan.2026***
- Developed a **OpenUSD**-native differentiable path tracing renderer using **NVIDIA Warp**.
- Implemented path-replay backpropagation for differentiable path tracing, computing gradients with respect to PBR metallic–roughness parameters through Monte Carlo light transport.
- Implemented **latent-space inverse rendering** with **Adam + LR scheduling** to recover metallic–roughness material parameters from image-space losses.

Maya Plugin for Procedural Flower Modeling ***Feb. 2023 – Apr.2023***
- Implemented a Maya Plugin using C++ and **MEL** to procedurally generate tree-like structures and flowers based on extended **L-System** production rules and grammar with positional information.
- Created a user-friendly GUI using the **Qt** framework, enabling convenient configuration and visualization of the parameters.

GPU Sound Propagation Simulator ***Nov. 2022 – Dec.2022***
- Developed an efficient 3D sound propagation system by segmenting space into discrete planes and simulating 2D wave propagation on each slice as an approximation.
- Ported the C++ library to Unity to allow for easy access by game developers.

CUDA Path Tracer and Denoiser ***Sep. 2022 – Nov.2022***
- Implemented a Monte-Carlo Path Tracer using **CUDA C++** with .obj Mesh loading and texture mapping.
- Implemented an edge avoiding denoiser for high-quality, low-SPP denoising using G-Buffer data.
- Achieved a 10x speed-up for triangle intersection tests in complex scenes using GPU-based octree and stream compaction.
- Utilizing **NVIDIA Nsight** kernel profiler, conducted a thorough performance analysis to catch significant factors that impact the running time of the path tracer.

2D Puzzle Game ***Mar. 2021 – May.2021***

- Designed and developed a 2D puzzle game in Unity, featuring recursive puzzles and brain teasers for players to solve.
- Utilized various decoupling patterns (e.g., observer pattern, game logic as data) to ensure code extensibility and maintainability.
- Implemented a simple boolean expression parser that enables level designers to set conditions for triggers and manage the dependencies between scene objects.