Tsingtao Zhang

tsingzhang1@gmail.com, Rochester, NY (willing to relocate), 585-615-0628
LinkedIn, Portfolio, Github

Skills

Game Engine and Graphics API: Unreal 5, Unity, OpenGL, Direct 3D 11
Programming Language: C/C++, C#, HLSL, GLSL, Python
Software: RenderDoc, PyQt, Blender, Photoshop, Substance Painter, Git, Perforce, Figma
Area of Focus: Asset Pipeline, Tool Development, Asset Creation, Procedural Content Generation, VR, Multiplayer

Work Experience

Role: Contract Technical Artist July - Oct 2025
The Forge Interactive Inc. Using Blender, Substance Painter, HLSL, Python
- Contributed in assets and particle effects in The Forge custom engine demo.
- Optimized San Miguel scene assets for mobile platform demo by refining meshes and textures through The Forge custom PBR pipeline. Using Blender and Substance Painter, corrected topology issues and baked high-poly details.
- As the first technical artist in the team, iteratively provided feedback to programmers, established and documented the asset production workflow, tailored for The Forge asset pipeline.
- Developed file managing tools using Python, helped streamline the asset production workflow.
- Developed particle and lighting effects including rain, fireflies, lanterns, and candles using The Forge's proprietary particle editor, while resolving shading issues in legacy codebase, and optimizing parameters for indirect lighting and lightning strikes using HLSL.
- Collaborated with graphics programmers to provide technical requirements for particle system, lighting system, and UI. Developed an iterative feedback loop on rapid development with multiple people across time zones.

Projects

Role: Technical Artist and Gameplay Programmer Sep 2024 - June 2025
Duolatera: A VR Multiplayer Puzzle Game, using Unreal 5, C/C++, Niagara, Python, Blender, Perforce, HLSL
- Implemented PCG content generation and spline auto-snapping tool, reducing level layout time by 90%.
- With 3D asset creating skills, established asset production pipeline and an art bible, led an external art team of 5.
- Using Python and Unreal Editor Utilities Widget, created an automated asset import tools for Unreal Engine, reducing 90% of related manual work.
- Created a texture conversion tool using PyQt and OpenCV, converting albedo textures into RGB Channel Masks.
- Using PyQt, developed an asset renaming tool, autonomously managing all assets avoiding human error.
- Created procedural and stylized material and VFX using Material Graph, HLSL, and Niagara system.
- Using Unreal IK system, built IK retargeted/predicted avatar animation based on player's movement.
- Implemented the online multiplayer gaming feature, allowing 2 players to cooperate remotely through Steam.

Role: Technical Artist July 2025 - now
A Runtime Procedural Tower Defense Game Prototype, using Unity, Compute Shader, HLSL, RenderDoc, Unreal 5
- Developed a procedural wall builder system in Unity. Dynamic brick mesh instances are calculated and generated along the spline, which can be changed by the player anytime during gameplay.
- Optimized rendering performance by implementing GPU-accelerated compute shaders to calculate transforms for 5,000 cube instances per frame, achieving a 185% performance improvement (70 to 200 FPS).
- Conducted cross-engine performance analysis by developing a comparable procedural fence builder in Unreal Engine, identifying limitations in runtime PCG updates through codebase investigation.

Role: Mocap Engineer, Multiplayer Programmer
XR Karaoke: A virtual performance, using Unreal, C++, VR/XR, Motion Capture Feb - April 2025
- Built an XR environment supporting audience watch performance in VR headset via internet.
- Extracted mocap data from Unreal Animation Blueprint, replicated from server to more than 10 clients.
- Led and facilitated 10 people from art to programming backgrounds, set up a virtual performance system.

Role: Graphics Programmer March - May 2024
Ocean Simulator: A real time ray-tracing water shader, using OpenGL, GLSL, C/C++
- Created a real-time interactive ocean renderer using GLSL in OpenGL with ray-tracing, performing above 30 FPS.
- Using linear algebra, ray reflection and refraction, added in a real-time caustics effect 10 times faster than backward ray tracing method, additive blending the underwater illumination.
- Created a clicking-promoted water circle waves on the surface interactively, on top of the default wave patterns.

Education

Rochester Institute of Technology, Rochester, NY. Aug 2023 - Aug 2025
M.S., Game Design and Development
China Agricultural University, Beijing, China. Sep 2018 – June 2022
B.Eng., Agricultural Structure Environment Engineering