Tomas Stegemann / sunsite.pages.dev / stegemanntomas@outlook.com
B.Sc. Computer Science, Minor in Interdisciplinary Design
College of Science and Engineering, University of Minnesota Twin Cities

Skills

C, C++, C#, Python, OpenGL, Vulkan, GLSL, Unity, Git, CMake, Make, GDB/LLDB, Valgrind/Leaks, graphics programming, systems programming, game development, typography

Experience

Gameplay and Engine Programmer, Entropic Engineering (October 2024 - Present)

Environment: C, Python, OpenGL
Developed bespoke engine and gameplay systems that run in real-time on an embedded system with tight resource constraints and novel hardware. Designed engine API to support both PC and embedded backends. Leveraged device hardware to create novel mechanics not seen in other games. Developed a variety of content-authoring tools incorporated into a central editor application. Additionally, served as primary gameplay designer.

Game Developer, College of Design (August 2023 - December 2023)

Environment: Unity, C#, OpenGL, C++, GLSL
Served primarily as a tools programmer, shader programmer, and programmer/designer for all systems and gameplay. Developed an educational game for teaching course material about trend forecasting to design students, as well as content-authoring tools for the game. Communicated with clients to shape and refine the project according to their needs.

Undergraduate Research Assistant, Interactive Visualization Lab (November 2022 - June 2023)

Environment: Unity, C#
Served primarily as a shader programmer and simulation/interaction programmer. Developed an interactive voxel-based canoe carving simulation to help indigenous communities share their expertise. Gave interactive demos of the simulation and employed user feedback in improving it. Collaborated closely with other lab members.

Projects

Graphics and Physics Programming with Hardware APIs

C/C++ primarily with OpenGL, Vulkan, GLSL. Examples include a CCD Inverse-Kinematic hydra demo, pinball, a mass-spring cloth simulation, a PBD rope simulation, and real-time rendering engines supporting features like precomputed lightmapping, physically-based and artistic shading, asset and scene loading, live scene editing, and more.

Graphics Programming “From-Scratch”

A rasterizer and a raytracer written in C/C++. The rasterizer renders OBJ meshes and the raytracer renders OBJ meshes as well as common implicit primitives. Each employs my personal libraries for image I/O, asset importing, and linear algebra.

Game and Simulation Development

Team and solo projects in Unity, Unreal Engine, and Godot. Highlights: A 3D painting demo employing compute shaders, two hand-drawn and procedurally animated games with shooting mechanics, and an FPS where I explored navmeshing spatial partitioning and triangulation techniques

Leadership: Gopher Graphics Club Workshop Officer, Video Game Development Club Programming Officer