CODY COKER

GRAPHICS & RENDERING ENGINEER · LEAD XR ENGINEER

601.508.1549 · j.cody.coker@gmail.com · Durham, NC
linkedin.com/in/cody-coker · github.com/j-coker · portfolio website (codycoker.dev)

Graphics engineer with a decade of shipped Unity and C# work and a specialization in real-time rendering on tile-based mobile GPUs. I write HLSL, build scriptable render features, ship volumetric effects, and spend a lot of time profiling on-device to understand what the GPU is actually doing. Currently lead engineer at Moth+Flame, where I carry the rendering architecture for VR training products in use by thousands of enterprise and military users.

SELECTED GRAPHICS & RENDERING WORK

Raymarched planetary volumetric clouds with temporal accumulation

Stylized cloud and atmosphere wrapped around a sphere, driven by particle-based wind bands stamping into a heightmap. Per-frame raymarch cost amortized across multiple frames via reprojection and history accumulation, paired with extensive use of precomputed LUTs. Runs as a scriptable render feature in URP.

GPU-driven volumetric dust with persistent accumulation

Volumetric dust for an Mk19 grenade launcher training app. The first attempt was a true 3D fluid sim that didn't fit the frame budget. The version that shipped is a 2D heightfield projection driven by rendertexture stamps and a compute shader for dilution and settling, with a temporally amortized raymarch resolving the 3D appearance. Later ported to Quest standalone via a separate projection-mapping approach that leveraged the experience's static user position and overcast conditions.

Multithreaded ECS-based Earth-scale terrain

Whole-planet procedural terrain on Unity DOTS. Mesh chunks generate asynchronously across parallel jobs as the camera moves, so the main thread never stalls. A heightmap index map lets each region pull from purpose-built high-fidelity terrain instead of one stretched texture. A floating-origin system recenters the world on the camera to keep float precision intact at Earth scale.

Instanced indirect SDF text rendering

A huge amount of on-screen text drawn as SDF quads via a single instanced indirect draw call, so the cost is one call regardless of how much is on screen. CPU-side instance data laid out for sequential cache-friendly access, which is also where the per-instance fade behavior lives. Used to label thousands of geographic features across a planet's surface at framerate.

GPU quadtree planetary LOD

Continuous level of detail with the entire quadtree living and updating on the GPU. The CPU never sees a node or triangle. Each frame, ping-pong append/consume buffers refine the tree (split toward camera, collapse far away), and surviving leaves emit triangles into an append buffer drawn with DrawProceduralIndirect. Helper passes use CopyCount and an indirect dispatch, so the GPU drives its own draw with zero readback.

PROFESSIONAL EXPERIENCE

Lead XR Software Engineer · Moth+Flame *Remote · April 2023 – Present*

Lead engineer carrying the graphics and rendering architecture across a multi-product VR training catalog deployed to enterprise and military customers. Team of 3 covering what was previously around 8, so the role spans more than just rendering: graphics, build pipeline, networking, business logic, and developer tooling.

Products ship natively to Quest, Pico, Focus, and desktop VR, with current work targeting Apple Vision Pro and Quest 3.

- Shipped custom shaders, scriptable render features, and volumetric effects across the product line. Photorealistic and stylized art directions, all inside native mobile XR frame budgets.
- Optimized rendering on Quest using the full mobile XR profiling stack: RenderDoc (Meta fork), Unity Profiler and the Frame and Rendering Debuggers, Perfetto, ovrgpuprofiler, and Meta Quest Developer Hub.
- Built our CI/CD pipeline from scratch. Four platforms (Quest, Pico, Focus, desktop VR), per-build content packaging for offline-deployed customers, automated APK signing, distribution to downstream training endpoints.
- Built developer tooling that let non-engineers produce training content at scale: a full in-VR module editor used extensively by the creative studio team, and a separate Electron-based authoring app for aircraft maintenance procedures. The authoring app talks to Unity over a local WebSocket (so technical producers can resolve references against the live scene, while external authors don't need Unity at all) and syncs to a server backend for parallel multi-author collaboration with live updates.
- Led code review, technical mentorship, and engineering documentation across the team.

Products I worked on at Moth + Flame:

- **Procedures.** VR training for aircraft repair, maintenance, and operation. Deployed to 20+ military organizations across 70+ modules covering 10+ airframes. Completely overhauled legacy architecture and built the full rendering solution and procedure-authoring tooling.
- **Conversations.** VR training catalog for difficult interpersonal conversations. Around 10,000 enterprise users across 150-ish modules. Built the in-VR module editor that became essential to the studio's content pipeline as well as numerous other client-side features and some backend work.
- **Mk19 Training.** VR training for the Mk19 automatic grenade launcher. Texture-projection rendering exploits the static user position to ship photorealistic visuals at low cost. Volumetric dust system with persistent accumulation (see Selected Work).
- **F-16 BFM Training.** VR training for F-16 pilots (later expanded to F-15 and F-35 teams), visualizing tactical engagement geometry from first-person and god's-eye views. Procedural flight-path geometry built indirectly from real telemetry.
- **U.S. Space Force Orbital Space Defense Training.** Built solo, very well-received. VR scenario placing the user as a Space Force operator handling a control-room encounter with a hostile foreign space asset.

Senior XR Software Engineer I–II · Moth+Flame

Remote · May 2021 – April 2023

- Owned and shipped major features across Conversations and Procedures. Custom shaders for both photorealistic and stylized art directions on mobile VR hardware.
- Cleaned up tightly coupled legacy rendering and mechanics to make them extensible and per-product configurable.
- Worked directly with PMs, designers, and the creative studio team to take training modules from concept to shipped product.

Software Engineer · PLAYSTUDIOS

Remote · July 2020 – May 2021

- Owned critical user-facing features in myKonami Slots, a mobile Unity and C# free-to-play title with hundreds of thousands of daily active users.
- Wrote custom shaders for high-fidelity visuals on low-end mobile hardware, including a separable Gaussian blur used across the app and still in production years later.

Creative Technology Director · Greater Jackson Alliance

Hybrid Remote · Nov 2016 – July 2020

- Self-initiated technical product work in support of regional economic development. Operated with broad autonomy to identify problems and build solutions.
- Built JXNMetro360 Augmented, a Unity and Vuforia AR mobile app that projects an interactive map of the Jackson, MS metro onto a physical printed map, used to walk industry prospects through featured sites.

- Captured aerial drone photogrammetry of industrial sites and processed it into digital-twin 3D models used for remote site visualization and analysis. This was well before “digital twin” was a marketing term, and the work was self-initiated.
- GIS analysis and data visualization for industry-attraction campaigns.

PERSONAL PROJECTS

Aminal Space ***2024 – Present · Unity, C#, HLSL***

Solo-developed contemplative colony builder set across a stylized solar system. The cloud and atmosphere, Earth-scale terrain, and GPU quadtree LOD in the Selected Work section are all from this project.

Project EXO ***2018 – Present · Unity, C#, HLSL · projectexo.webflow.io***

Science-informed, near-future space strategy simulator with deeply simulated emergent systems. The SDF text rendering in the Selected Work section originated here, alongside JFA-based screenspace borders and log-depth reprojection. More on the portfolio site.

Songs of the Eons ***2017 – 2020 · Unity, C#, C++ · patreon.com/SongsOfTheEons***

Highly simulation-driven nearest-neighbor Goldberg polyhedron-based planetary-scale fantasy sandbox where the player guides civilizational evolution across eons on a planet generated from real geomorphological processes.

TECHNICAL SKILLS

Graphics & Rendering · HLSL shader authoring, Unity SRP (URP/HDRP), Scriptable Render Features, custom render passes, ShaderGraph, Vulkan/Metal rendering concepts, Forward+/deferred rendering, PBR and stylized lighting, volumetric raymarching, GPU compute, indirect rendering, GPU-driven culling, custom post-processing

XR Platforms · Meta Quest 2/3/Pro, Apple Vision Pro (visionOS), Pico, HTC Focus, OpenXR. Single-pass stereo, fixed foveated rendering, TBDR-aware strategy.

Profiling & Optimization · RenderDoc (Meta fork), Unity Profiler / Frame Debugger / Rendering Debugger, Perfetto, ovrgpuprofiler, Meta Quest Developer Hub. Frame-budget management, draw-call reduction, overdraw analysis, bandwidth/ALU/latency bottleneck diagnosis.

Languages & Engines · C# (10+ years, Unity), HLSL, GLSL, C++, JavaScript/TypeScript, Python. Unity 5 through Unity 6, Unity DOTS/ECS. Rust.

Tooling & Pipeline · Custom editor extensions and inspectors, in-engine and external content-authoring tools, multi-platform CI/CD for Unity, asset bundling and per-build content packaging.

EDUCATION

B.S. Geography, University of Southern Mississippi, 2012–2016
Continuing self-directed coursework · Pluralsight, Udemy, FreeCodeCamp (C#, C++, Unity, software design patterns, frontend).

References available upon request.