

Coleman Engbrecht

Systems Engineer

cole.engbrecht@gmail.com | <https://coleengbrecht.github.io>
<https://www.linkedin.com/in/coleman-engbrecht-73b818306>

Skills:

C++, JavaScript, Dart, Vulkan/OpenGL/DirectX, GLSL/HLSL, Console APIs, Linux, Unreal, Unity, Git, Perforce, Flutter, React Native

Experience:

Engine Programmer, Hypersect

2025 - Present

- Ported a networked multiplayer game project to Amazon's Luna platform
- Developed a low-latency, mobile-first web app controller to be used on the Luna platform
- Helped port an in-house game engine to console platforms
- Wrote a tile based light culling algorithm to replace a per-object, CPU bound light culling algorithm. Which enabled an uncapped light limit and about a 10% perf improvement for most scenes
- Optimized a rendering engine to reduce the amount of constant buffer updates by 3x, improving perf by about 10-15%
- Debugged and fixed thread-scheduling related latency issues on multiple platforms, made use of ETW and console debugging tools
- Wrote an asset streaming system used to stream in large audio files, making sure the audio timeline doesn't de-sync with the game timeline
- Implemented a data-driven gamepad effect system that supports multiple effect types including: sound based vibration, trigger vibration and resistance
- Wrote a physically based animation system for characters and weapons, supporting multiple constraint types
- Implemented a robust save version upgrade system that does: rule-based graph rewrites on the save data tree structure to ensure forward compatibility when schema data changes. Development builds enable additional verification checks to ensure the rules are written correctly
- Improved the quality, correctness and performance of a novel 3D sprite sorting algorithm that utilizes topological sorting and heuristic based strongly connected component breaking. I removed non-deterministic behavior in the heuristic, implemented a spatial acceleration structure to speed up graph edge building, and split the algorithm into separate passes for static and dynamic objects in order to improve temporal stability under camera rotation
- Built a networked vehicle system using Unreal's Chaos physics
- Added features and improved the feel of a networked character controller in Unreal including: soft character vs character collision, input buffering for jumps and sprinting, camera post-process effects for stamina, better gamepad controls for sprinting, heuristic driven first-person item interaction

Software Developer, gskinner

2019 - 2025

- Integrated deep support for the Zoom SDK utilizing CallKit and other native iOS APIs into a finance app built in Flutter. Including support for answering VoIP call notifications in app and displaying video through Flutter's native views
- Built multiple cross platform OS integration plugins for Flutter, including custom desktop window decorations for Mac, Windows, Linux

- Re-wrote Pixi JS's input manager to provide better support for pen devices on Android, implemented an advanced gesture detection system using the custom input manager
- Improved loading performance for certain assets in a web view up to 10x by utilizing web APIs that avoid CPU to GPU round trips
- Implemented hundreds of custom UI components and views in both Flutter and React Native
- Wrote a CD pipeline that packaged and released software for the Windows Store and Ubuntu Snap Store
- Utilized compute shaders in Unity to efficiently solve data visualization problems, debugged and solved shader compiler issues leading to undefined behavior on some platforms
- Wrote a level editor for a rhythm game in Unity
- Wrote a software 3D renderer and OBJ model loader in Flutter that could hit 90+ fps on low-end mobile devices
- Implemented a SDF ray marching renderer and several post-process effects using GLSL shaders in Flutter
- Communicated with clients to fine tune technical requirements and implementation details for deliverables

Game Engine Development, Oak Engine

2015 - Present

- Developed a cross platform editor and runtime that has been used to release several small games
- Implemented a connection oriented network protocol that supports multiplexing of multiple channels with different reliability and ordering behavior
- Implemented a reflection and serialization framework that enables scene serialization, automatic symmetric diffing for undo/redo, network message serialization, real-time game state replication with record and replay functionality, property editing with the ability to define widgets per data type and annotation
- Implemented a simple and fast entity component system with efficient cache behavior and support for entity links and transform hierarchies
- Implemented a low latency fiber job system utilizing atomic operations and futexes that enables async asset loading and parallelized rendering work
- Implemented a game export feature that packages all code and assets into a single executable file
- Integrated address and undefined behavior sanitizers into low level context switching code and custom memory allocators
- Implemented hot reloading for assets, code and shaders
- Implemented a GUI framework with support for constraint based auto-layout
- Implemented 2D and 3D rigid body physics engines with: island solvers, SAT and GJK collision detection, robust manifold generation, dual quaternion mechanics, sub-stepping solvers, soft constraints, and warm starting
- Implemented dual quaternion based skeletal animation system with: animation blending, blend spaces, full body IK, joint attachments
- Implemented a navigation engine with nav-mesh generation from level geometry and A* path finding
- Wrote a 3D, physically based, GPU driven renderer that supports fully dynamic direct and indirect lighting, post processing and tone-mapping
- Implemented a software voxel ray tracer that scales down to low end hardware, tools that robustly generate voxel volumes from triangle meshes
- Implemented an integrated path tracer to provide an in-engine reference for real-time algorithms
- Implemented a compute based particle system that can simulate and render millions of particles in real-time
- Multi-threaded audio mixer with basic effect support

Aircraft De-icing Technician, AeroMag

2017 - 2019

- Communicated effectively over the radio in a safety critical environment

Education:

Computer Engineering (CNT), North Alberta Institute of Technology

2018-2019