KEVIN 'TYLER' COX

Unreal Engine 5 C++ Gameplay Systems Developer

(843) 718-4024 kevincox103@gmail.com Charleston, SC

github.com/Ty-lerCox linkedin.com/in/tyler-cox-1715065a/ ty-lercox.github.io/portfolio/

Summary

Developer packaging SDK-style gameplay modules with clean APIs, UE5 unit tests, and developer docs. Complementary distributed-services experience (Java/C#) with Kubernetes, Kafka, and Grafana/Prometheus/Loki/Tempo; secure integrations via Keycloak OIDC/SAML and TLS/mTLS. Working knowledge of MySQL (schema design & SQL optimization). Remote from Charleston, SC; open to on-call rotation and light travel.

Education

Computer Science

Trident Technical College

Technical Skills

Unreal Engine 5, C++17/20 (5+ years), SDK-Style Libraries & API Design, UE5 Unit Testing (Automation), Client-Server Integrations (HTTP/REST), UMG/HUD, Dialogue & Quest Systems, AI Movement & Spawning, Redux-style State (actions/effects/state), Data-Driven Design, Profiling & Optimization, Tooling & Debugging, MySQL (Schema design & SQL optimization), Kubernetes (deployments), Kafka, Grafana/Prometheus/Loki/Tempo, Keycloak (OIDC/SAML), TLS/mTLS, C#/.NET, Java, Python, Git

Soft Skills

Communication, Collaboration & Cross-Team Coordination, Systems Thinking, Problem Solving, Self-Direction, Iteration & Rapid Prototyping

Additional Skills

API Documentation, Testability & Maintainability, Runbooks & Incident Response (on-call ready), Cinematic UI/UX, CI/CD (GitHub Actions, GitLab

Work Experience

Independent / Self-Employed Sep, 2021 - Present

UE5/C++ Game Systems Developer

Design and implementation of core gameplay systems and tools in Unreal Engine 5 with emphasis on maintainable, data-driven C++.

- Built branching Dialogue System with multi-option choices and conditional availability based on quest state (current/completed/required).
- Engineered Quest System supporting multi-objective tasks, required counts, dependencies, and side effects (trigger cinematics, spawn NPCs, state changes).
- Established Redux-style architecture in C++ (actions, effects, state) to isolate subsystems and improve testability and maintainability.
- Created AI movement and a needs model; authored advanced spawner logic for unpredictable NPC distribution with region rules, cooldowns, and variance.
- Implemented story-management and transition volumes to control floor/zone visibility and scene flow.
- Developed cinematic HUD widgets and web-inspired UI components to accelerate iteration and improve UX.
- Delivered inventory and shop subsystems integrated with global game state; built a codex/collection system to track collectibles and progression.
- Integrated an in-game guide chatbot (LLM) with command interface; used context caching to reduce token usage and latency.
- Packaged systems as reusable SDK-style C++ modules with public headers, versioned APIs, integration guides, and demo levels to illustrate usage.
- Wrote UE5 unit tests for gameplay subsystems (e.g., quest progression, inventory interactions) using the automation testing framework.
- Practiced data-driven configuration, profiling, and optimization across systems.

Expediters International (Expediters) Aug, 2014 - Present

Angular Application Developer (Additional Professional Experience)

Primary employer; application development, DevOps, and observability (transferable engineering practices).

- Led state management patterns (NgRx, Angular Signals) and modular architecture across multiple internal applications.
- Stood up observability with Grafana stack (Prometheus metrics, Loki logs, Tempo traces) and alerting; emphasized instrumentation and telemetry—skills applicable to game profiling and tooling.
- Automated CI/CD with GitHub Actions/GitLab Runners and Ansible; supported Kubernetes deployments; improved iteration speed and reliability.

Runners), Ansible, Secure Integrations (PKI/CA), Kanban & Delivery Flow

Languages

English

- Implemented SSO (Keycloak OIDC/SAML), Kerberos integrations, and PKI/CA for TLS/mTLS; enforced secure defaults.
- Modernized data flows from bespoke Kafka producers to database-level CDC → Kafka with medallion layers (bronze/silver/gold).
- Used Python for rapid prototypes; production services in Java Spring Boot and C#/.NET; practiced feature flags, DORA metrics, and Kanban for flow.
- Documented integration steps, runbooks, and platform configurations; supported incident response; willing to participate in on-call rotation.

Projects

Dialogue System (UE5/C++) Apr, 2025 - May, 2025

https://github.com/bedivere-lea

Branching, conditional dialogue with quest-aware availability and consequence tracking.

- Multi-option choices; gating by current/completed/required quests; integrates with Redux-style game state; API usage documented with sample/demo level.

Quest System (UE5/C++) Apr, 2025 - May, 2025

https://github.com/bedivere-lea

Objectives with counts, dependencies, and side effects wired to gameplay events.

- Triggers cinematics; spawns NPCs; updates global state via actions/effects; integration points documented for reuse.

AI Needs & Advanced Spawning May, 2025 - Jun, 2025

https://github.com/bedivere-lea

AI behavior model with needs (Sims-like) and region-based spawner rules.

- Unpredictable spawn distribution; cooldowns and variance; tunable via data assets; profiling-informed tuning to maintain frame-time targets.

Story/Level Flow Controls May, 2025 - Jun, 2025

https://github.com/bedivere-lea

Volume-based story transitions with floor/zone visibility controls.

- Performance-aware visibility; clean narrative gating; improved player readability; documented API hooks for scene transitions.

Inventory & Shop Subsystems Jun, 2025 - Jul, 2025

https://github.com/bedivere-lea

Item catalogs, purchasing, and persistence integrated with global game state.

- Predictable side effects; data-driven tuning; testable modules; sample/demo level illustrates API usage and events.

Codex/Collection System Jun, 2025 - Jul, 2025

https://github.com/bedivere-lea

Collectible tracking and progression feedback loops.

- Player progress surfaces; unlock conditions; consistent data schema; developer notes and API reference for extension.

Guide Chatbot Integration (LLM) Jul, 2025 - Aug, 2025

https://github.com/bedivere-lea

In-game user-guide chat agent using language models.

- HTTP/REST integration; command interface; context caching to lower inference cost and latency; developer notes for configuration.

Redux-Style Game State Library Jul, 2025 - Aug, 2025

https://github.com/bedivere-lea

Shared actions/effects/state managers used across gameplay subsystems.

- Separation of concerns; maintainable C++ modules; easier testing and debugging; packaged as SDK-style library with API docs and UE5 unit tests for key reducers/effects.