Anton Zarubin

Senior Software Engineer

TypeScript · Phaser 2/3 · Three.js · HTML5 Playable Ads · Tooling · Performance

- GitHub: https://github.com/zarant77 · LinkedIn: https://www.linkedin.com/in/zarant77
- Upwork: https://www.upwork.com/freelancers/~01d67a6a38050609ab

Senior Playable Ads Engineer with production experience building performance-critical HTML5 playables in TypeScript under strict bundle size and runtime constraints. Focused on reusable systems, ad-tech integration abstractions and developer tooling that improves delivery speed, reliability and analytics correctness across teams.

Core Technologies

TypeScript · JavaScript (ES6+) · HTML5 Canvas · Phaser 2 · Phaser 3 · Three.js · MRAID · Ad network integrations · Bundle size optimization · Asset compression pipelines · Node.js CLI tooling · Remote debugging instrumentation

Production Constraints & Focus

- Size-constrained deliverables (ZIP/bundle optimization, asset budgets)
- Performance-first approach (runtime stability on mid/low-end devices)
- Ad-tech integrations (tracking events, store redirection, network-specific requirements)
- Rapid iteration cycles (dev-mode workflows, build automation, QA feedback loops)

Impact

- Initiated and led internal platform tooling adopted by a 20+ person cross-functional team (developers + game designers)
- Standardized playable integration flows to reduce human error in analytics implementation
- Accelerated production via reusable systems and automated build/packaging tooling
- Improved QA turnaround by enabling remote log capture from real devices

Experience

Senior Playable Ads Engineer

2017–2022

ironSource (acquired by Unity) · Playable Ads · Internal Tooling · Ad Tech

- TypeScript
- Phaser 2/3
- Three.js
- HTML5 Canvas
- MRAID
- Node.js
- CLI tooling
- Build/asset optimization

Built performance-critical HTML5 playables and designed internal platform tools that standardized development workflows, improved integration reliability and increased delivery speed.

Nucleo — Ad Network Abstraction Layer

Internal layer between playables and ad networks providing developer-friendly API over MRAID and network-specific interfaces.

- Unified access to ad network mechanisms without exposing MRAID complexity to playable developers
- Standardized tracking and store redirection flows through a consistent API
- Reduced integration mistakes and improved onboarding experience

GameMaster — Config-Driven Playable Flow Engine

Centralized controller orchestrating playable flow (screens, transitions, events) based on designer-authored configs.

- Loaded JSON configs created by game designers to drive flow and enable A/B testing
- Automated win/lose screens, level transitions (including multi-level playables) and analytics events
- Minimized human factor in analytics implementation, improved testability and delivery speed

Buddy Christ — CLI Build & Optimization Tool

Internal CLI tool for building playables, optimizing assets and running dev-mode workflows across multiple ad networks.

- Automated packaging for different ad networks with consistent build steps
- Implemented asset minification/compression pipeline to meet strict size constraints
- Reduced manual build errors and accelerated iteration cycles via developer-friendly CLI UX

Cockroach — Remote Device Log Proxy

Debug instrumentation embedded into dev builds to stream console logs from real devices to an internal web server.

- Proxied console logs (errors/warnings/info) from device runs where web debugging was insufficient
- Enabled fast diagnosis of device-specific issues (e.g., vendor-custom Android quirks)
- Eliminated blind debugging and improved QA-to-fix turnaround time