Anton Zarubin

Senior Software Engineer

Slot Engines · RTP Modeling · Simulation · Deterministic Systems · Multiplayer Architecture

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

Senior Software Engineer focused on deterministic gambling systems, slot engine architecture and simulation-driven balancing. Experience in building reproducible, config-driven game logic systems with real-time multi-client support, RTP/volatility modeling and large-scale simulation tooling. Strong focus on predictability, testability and performance.

Core Technologies

TypeScript | Node.js | Phaser 3 | Deterministic systems | Game logic architecture | Simulation systems | RTP / volatility modeling | Config-driven design | CLI tooling | Performance optimization

Gambling Systems Engineering

- Design of deterministic slot engines with reproducible outcomes and no hidden state
- Config-driven reels, symbols and payout systems
- RTP and volatility modeling through large-scale simulation
- Real-time multi-client architecture (CLI + Web client)
- Separation of game logic from rendering and UI layers
- Strategy testing systems (bet progression, scenarios)
- Deterministic spin resolution pipeline (bet → spin → outcome → payout)

Slot System Features

- Reel-based slot architecture with configurable symbol distribution
- Modular symbol pipeline and payout evaluation system
- Support for multiple RTP/volatility profiles via configuration
- Reproducible simulation runs using deterministic RNG
- Integration-ready core independent from rendering layer
- Scalable simulation execution (10k–100k+ spins per run)

Impact

- Built a deterministic slot engine structured for reproducible simulation and balancing
- Enabled rapid iteration of reels, payouts and symbol behavior through config-driven design
- Implemented large-scale simulation workflows for RTP and volatility validation
- Designed system architecture supporting multiple clients consuming the same core logic
- Eliminated hidden state to guarantee reproducibility and testability

Experience

Gambling Systems Engineer

Personal / Showcase Project

Slot Engine · Simulation · Multiplayer Architecture · Game Logic

TypeScript | Node.js | Phaser 3 | Deterministic core | Simulation CLI | Web client | Config-driven architecture

Designed and implemented a deterministic multiplayer slot engine with full control over configuration, probability and simulation for balancing, testing and system experimentation.

CatSpin Arena — Deterministic Multiplayer Slot Engine

Multiplayer slot engine built on a deterministic core with reproducible outcomes and no hidden state.

- Implemented deterministic spin logic ensuring reproducible outcomes across all clients
- Designed real-time multi-client architecture (CLI + Web)
- Separated core game logic from UI and rendering layers
- Built config-driven reels, symbols and payout evaluation pipeline

Slot Implementation (Phaser)

Built multiple slot prototypes using Phaser 3 with reel systems and spin flow logic integrated with deterministic backend.

- Implemented reel generation and symbol distribution based on config-driven data
- Built spin lifecycle (bet → spin → resolve → payout)
- Integrated frontend rendering with deterministic engine
- Designed modular reel and symbol systems for rapid iteration
- Handled animation timing, easing and visual feedback

Simulation & RTP Modeling

Simulation tooling for balancing, validation and statistical analysis.

- Developed CLI-based simulation system (10k–100k+ spins per run)
- Enabled RTP and volatility analysis through deterministic execution
- Built strategy testing scenarios for bet progression systems
- Validated statistical behavior across different configurations

Engineering Approach

Architecture-first implementation focused on predictability and performance.

- Used config-driven design for rapid balancing iteration
- Eliminated hidden state to ensure reproducibility
- Optimized execution flow for batch simulation workloads
- Designed system as a reusable core independent from UI