Drawballz: Formal Specification

OpenRNG Simulation

December 16, 2025

Abstract

This paper specifies the Drawballz game model, verification logic, RNG and fairness mechanisms, RTP math, and compliance-aligned controls suitable for iGaming certification. It presents formal definitions, proofs of key properties, operational policies, and audit-ready artifacts.

Keywords: RNG, RTP, provably fair, iGaming compliance, audit, distributions

Contents

List of Figures

List of Tables

Executive Summary

This document specifies the game model, verification logic, and compliance-aligned controls for an iGaming context. It covers epoch configuration, RNG and fairness, exact matches, payouts, RTP and volatility, and certification mapping. The goal is a clear, auditable specification that is implementable and testable end-to-end.

Overview

This specification formalizes the core components of the Drawballz simulation: epoch configuration, player state, mask sampling, match evaluation, payout calculation, batch metrics, and UI verification logic. It abstracts the implementation found in the client (`public/app.js`) and server (`src/engine.ts`) into precise definitions.

1 Requirements & Conformance

We use RFC-2119 terminology. Unless otherwise noted:

- RNG draws MUST be reproducible from committed seeds and indices.
- Prize calculations MUST use integer multipliers and consistent rounding.
- RTP and distribution metrics MUST be exportable in machine-readable formats.
- Verification status MUST be shown as OK/Estimate/Mismatch with a tolerance δ.
- Parameter changes (probabilities p, cap $k_{\max}$, table T) MUST follow change control.

Conformance targets include GLI-11 RNG, RTP disclosure, payout determinism, rounding policy consistency, and audit integrity.

2 Notation

- Colors $C = \{1, 2, 3, 4, 5\}$.
- A ball b is a pair (n, c) with $n \in \mathbb{Z}$ and $c \in C$.
- A player P has a set of five balls $B_P \subseteq \mathbb{Z} \times C$ with exactly one per color, and a nonnegative bet amount $\beta_P \in \mathbb{R}_{\geq 0}$.
- Epoch config $\mathcal{E} = (p, k_{\max}, n_{\min}, n_{\max}, T)$ where:
  - $p = \{p_0, \ldots, p_5\}$ with $\sum_{i=0}^{5} p_i = 1$ are mask-size probabilities.
  - $k_{\max} \in \{0, \ldots, 5\}$ is the max mask size cap.
  - $n_{\min} \leq n_{\max}$ define the number range.
  - $T : \{0, \ldots, 5\} \to \mathbb{R}_{\geq 0}$ is the fixed prize table mapping matches to multipliers.

3 Cancellations and Remaining Sets

Given players A, B with balls B_A, B_B, define cancellations:

$$\textit{cancelled} = \{(n, c) \in B_A \cap B_B \mid \text{same } n \text{ and } c\}.$$

Remaining sets after cancellations:

$$R_A = \{(n, c) \in B_A \mid (n, c) \notin \textit{cancelled}\}, \quad R_B = \{(n, c) \in B_B \mid (n, c) \notin \textit{cancelled}\}.$$

4 Mask Sampling

Let r be the epoch RNG seeded deterministically by $\mathcal{E}$, salt, and players. Sample $k \sim p$, clamp by $k_{\max}$ and enforce non-empty:

$$k' = \max\big(1, \min(k, k_{\max})\big).$$

Choose k' distinct colors uniformly from C and, for each chosen color c, sample a number n uniformly from $[n_{\min}, n_{\max}] \cap \mathbb{Z}$. The winning mask is:

$$W = \{(n, c)\}_{i=1}^{k'}.$$

5 Exact Matches

Per player:

$$m_A = |\{(n, c) \in W \mid (n, c) \in R_A\}|, \quad m_B = |\{(n, c) \in W \mid (n, c) \in R_B\}|.$$

Total matches for the outcome:

$$m = m_A + m_B.$$

6 Payouts

Define per-player multipliers from the table T:

$$\mu_A = T(m_A), \qquad \mu_B = T(m_B).$$

Per-player payouts:

$$\pi_A = \mu_A \cdot \beta_A, \qquad \pi_B = \mu_B \cdot \beta_B.$$

Outcome prize:

$$\Pi = \pi_A + \pi_B.$$

7 Effective Bet and Revenue

For a single outcome, effective bet (used as total bet revenue increment):

$$\varepsilon = \frac{\beta_A + \beta_B}{2}.$$

Over N outcomes, total bet revenue:

$$\mathcal{R} = \sum_{i=1}^{N} \varepsilon_i.$$

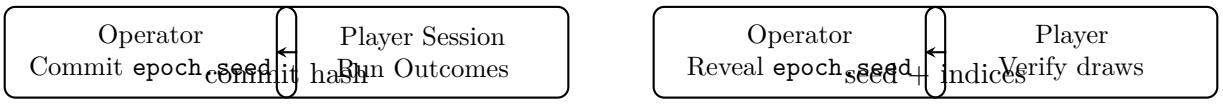
8 Provably Fair RNG

Definition 1 (Seeded Determinism)**.** *For index i, define a seed component $s_i =$* `epoch.seed : bet : i`*. Random draws are derived from $H(s_i)$, where H is a cryptographic hash (e.g., HMAC-SHA256) mapped to uniform integers.*

Commit–reveal may be used for verifiability: the operator commits to `epoch.seed` before the session and reveals after, enabling players to reconstruct draws. Entropy sources, seed rotation, and audit logging ensure fairness and traceability.

Proposition 1 (Deterministic Reconstruction)**.** *Given $\mathcal{E}$ and seeds $\{s_i\}$, bet-range values $\beta_A^{(i)}, \beta_B^{(i)}$ and mask sampling outcomes are reproducible for all i.*

8.1 Commit–Reveal Flow



(a) Stage 1: Commit and session start.

(b) Stage 2: Reveal and player verification.

Figure 1: Commit–reveal flow for provable fairness.

8.2 Algorithm: RNG and Mask Sampling

Input: `epoch.seed`, index i, $p, k_{\max}, [n_{\min}, n_{\max}]$
Output: Winning mask W
Construct $s_i =$ `epoch.seed:bet:`i;
Derive uniform integers from $H(s_i)$;
Sample $k \sim p$, set $k' = \max(1, \min(k, k_{\max}))$;
Select k' distinct colors from C;
For each selected color c, sample $n \in [n_{\min}, n_{\max}] \cap \mathbb{Z}$;
Return $W = \{(n, c)\}$;

Algorithm 1: Deterministic RNG and mask sampling

8.3 Verifiable Transcript Schema

9 RTP, Variance, and Confidence

Let Π_i be outcome prizes and ε_i effective bet increments. Define means and variances:

$$\mu_\Pi = \mathbb{E}[\Pi], \quad \sigma_\Pi^2 = \mathrm{Var}(\Pi), \qquad \mu_\varepsilon = \mathbb{E}[\varepsilon], \quad \sigma_\varepsilon^2 = \mathrm{Var}(\varepsilon).$$

Over N i.i.d. outcomes, the sample totals satisfy:

$$\mathcal{P} = \sum_{i=1}^{N} \Pi_i, \quad \mathcal{R} = \sum_{i=1}^{N} \varepsilon_i.$$

By the central limit theorem, for large N:

$$\frac{\mathcal{P} - N\mu_\Pi}{\sqrt{N}\sigma_\Pi} \approx \mathcal{N}(0,1), \qquad \frac{\mathcal{R} - N\mu_\varepsilon}{\sqrt{N}\sigma_\varepsilon} \approx \mathcal{N}(0,1).$$

Field	Description
Session ID	Unique identifier for batch run
Epoch Seed	`epoch.seed` committed and later revealed
Index i	Outcome index (1-based)
Seed Component s_i	`epoch.seed:bet:`i
Draws	Derived uniform integers for mask and numbers
Mask Size k	Size of winning mask $\lvert W\rvert$
Mask W	Set of (n, c) pairs
Matches (m_A, m_B)	Exact matches per player
Payouts (π_A, π_B)	Per-player payouts for the outcome
Effective Bet ε	Contribution to total bet revenue
Verification Status	OK / Estimate / Mismatch

Table 1: Transcript fields exported for player and lab verification.

Confidence intervals for totals:

$$\mathcal{P} \in \left[N\mu_\Pi \pm z_{1-\alpha/2}\sigma_\Pi\sqrt{N}\right], \quad \mathcal{R} \in \left[N\mu_\varepsilon \pm z_{1-\alpha/2}\sigma_\varepsilon\sqrt{N}\right].$$

Sample size for error bound ϵ on $\mathcal{P}$ at confidence $1-\alpha$:

$$N \geq \left(\frac{z_{1-\alpha/2}\sigma_\Pi}{\epsilon}\right)^2.$$

RTP is RTP $= \mathcal{P}/\mathcal{R}$ when $\mathcal{R} > 0$; session-level RTP stability follows from concentration of $\mathcal{P}, \mathcal{R}$.

10 Batch Metrics

Let outcomes be $\{(W_i, m_{A,i}, m_{B,i}, \Pi_i, \varepsilon_i)\}_{i=1}^N$. Define:

$$\mathcal{P} = \sum_{i=1}^N \Pi_i, \qquad \mathcal{R} = \sum_{i=1}^N \varepsilon_i, \qquad \text{RTP} = \begin{cases} \frac{\mathcal{P}}{\mathcal{R}} & \mathcal{R} > 0, \\ 0 & \text{else.} \end{cases}$$

Distribution of exact matches (total $m_i = m_{A,i} + m_{B,i}$):

$$D_m(j) = \lvert\{i \mid m_i = j\}\rvert, \quad j \in \{0, \ldots, 5\}.$$

Distribution of mask sizes:

$$D_k(j) = \lvert\{i \mid \lvert W_i\rvert = j\}\rvert, \quad j \in \{0, \ldots, 5\}.$$

10.1 Illustrative Distributions

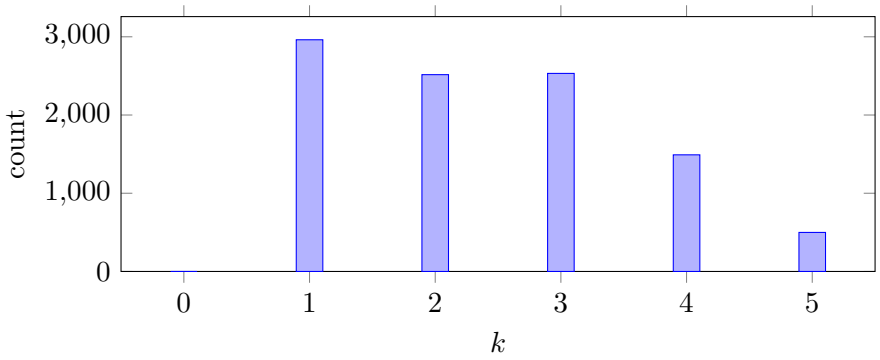


Figure 2: Mask size distribution D_k; uses CSV if present, otherwise illustrative.

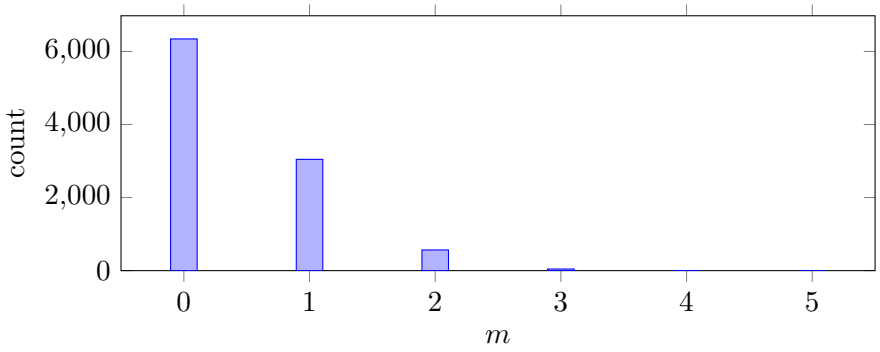


Figure 3: Exact match distribution D_m; uses CSV if present, otherwise illustrative.

11 Bet Range Randomization

If bet range is enabled with $\beta_{\min} \le \beta_{\max}$, per outcome index i:

$$\beta_A^{(i)} = \beta_{\min} + U_i\big(0, \beta_{\max} - \beta_{\min}\big), \quad \beta_B^{(i)} = \beta_{\min} + V_i\big(0, \beta_{\max} - \beta_{\min}\big),$$

where U_i, V_i are independent integer draws using epoch seed component `epoch.seed : bet : i`. Otherwise $\beta_A^{(i)} = \beta_A, \beta_B^{(i)} = \beta_B$.

12 UI Verification (Payouts Modal)

Given a sampled subset $S \subseteq \{1, \dots, N\}$ of size $|S| = s$, define:

$$\widehat{\mathcal{P}}_S = \sum_{i \in S} \Pi_i, \quad \alpha = \frac{N}{s}.$$

Displayed total:

$$\text{Display} = \begin{cases} \widehat{\mathcal{P}}_S & s = N, \\ \alpha \cdot \widehat{\mathcal{P}}_S & s < N. \end{cases}$$

Verification compares Display with metrics $\mathcal{P}$, reporting OK if $s = N$ and $|\text{Display} - \mathcal{P}| \le \delta$ (small tolerance), otherwise Estimate. Filters (wins-only, mask-size k, exact matches m_A, m_B) restrict S before computing $\widehat{\mathcal{P}}_S$.

13 UI Verification (Bets Modal)

Similarly, using effective bets:

$$\widehat{\mathcal{R}}_S = \sum_{i \in S} \varepsilon_i, \quad \text{Display} = \begin{cases} \widehat{\mathcal{R}}_S & s = N, \\ \alpha \cdot \widehat{\mathcal{R}}_S & s < N. \end{cases}$$

Verification compares Display with metrics $\mathcal{R}$ as above. Filters include mask-size k and effective bet range constraints.

14 Correctness Conditions

- Integer multipliers: $T(j) \in \mathbb{Z}_{\geq 0}$ for all j, yielding integer prizes.
- Non-empty mask: $k' \geq 1$.
- Cap respected: $|W| \leq \max(1, k_{\max})$.
- Distribution consistency: counts from outcomes equal metrics distributions for matching buckets.
- Prize total: $\mathcal{P} = \sum_i \Pi_i$ must equal sum of per-player payouts across runs.

Proposition 2 (Verification Equality)**.** *If $s = N$ and filters are neutral, the displayed total equals the metrics total up to tolerance δ.*

Proof. Under neutral filters, $S = \{1, \ldots, N\}$. The subtotal over S is $\sum_{i=1}^{N} \Pi_i = \mathcal{P}$. Display renders $\widehat{\mathcal{P}}_S$ without scaling, so $|\text{Display} - \mathcal{P}| = 0$, treated as equal within δ. □

15 Verification Theorems

Proposition 3 (Unbiased Scaled Subtotal)**.** *Let $S \subseteq \{1, \ldots, N\}$ be a simple random sample of size s and $\widehat{\mathcal{P}}_S = \sum_{i \in S} \Pi_i$. With $\alpha = N/s$, the scaled subtotal satisfies $\mathbb{E}[\alpha\, \widehat{\mathcal{P}}_S] = \mathcal{P}$.*

Proof. Each outcome prize Π_i is included with probability s/N. Linearity of expectation yields $\mathbb{E}[\widehat{\mathcal{P}}_S] = \sum_{i=1}^{N} \Pi_i \cdot s/N = (s/N)\mathcal{P}$. Scaling by $\alpha = N/s$ gives $\mathbb{E}[\alpha\, \widehat{\mathcal{P}}_S] = \mathcal{P}$. □

Proposition 4 (Distribution-Based Reconstruction)**.** *Assume fixed bets β_A, β_B and prize multipliers $T(j)$. Let $D_{m_A}(j)$ and $D_{m_B}(j)$ be counts of exact matches j for players A and B over a batch. Then the total prize equals*

$$\mathcal{P} = \sum_{j=0}^{5} \big(\beta_A\, T(j)\, D_{m_A}(j) + \beta_B\, T(j)\, D_{m_B}(j)\big).$$

Proof. For each outcome, player payouts are $\pi_A = \beta_A T(m_A)$ and $\pi_B = \beta_B T(m_B)$. Summing across outcomes and grouping by match counts for each player yields the stated expression. □

16 Parameter Governance & Change Control

Operational parameters $(p, k_{\max}, n_{\min}, n_{\max}, T)$ are versioned. Changes MUST:

- Record previous and new values with rationale and approval timestamp.
- Regenerate certification evidence: updated RTP expectations and distribution sanity checks.
- Produce a signed configuration hash included in transcripts and logs.

17 Risk & Exposure Controls

Operator exposure per outcome is bounded by prize caps and T. Session-level exposure can be approximated using:

$$\mathrm{VaR}_\alpha(\mathcal{P}) \approx N\mu_\Pi + z_\alpha \sigma_\Pi \sqrt{N},$$

under i.i.d. assumptions. Controls include:

- Max payout per outcome and per session caps.
- Throttling strategies when exposure approaches thresholds.
- Monitoring RTP drift; alert when $|\mathcal{P}/\mathcal{R} - \text{targetRTP}|$ exceeds tolerance.

18 Certification Mapping

This specification aligns with typical iGaming standards (e.g., GLI-11) by addressing RNG quality and auditability, RTP declaration, payout determinism and rounding, integrity controls, input validation, error management, and logging. Each element is verifiable through reproducible seeds and exportable metrics.

18.1 Conformance Matrix

Control	Standard Clause	Spec Section
RNG Auditability	GLI-11 RNG	Figure 1, Provably Fair RNG
RTP Disclosure	GLI-11 RTP	RTP, Variance, and Confidence
Payout Determinism	GLI-11 Math	Payouts; Correctness Conditions
Rounding Policy	Jurisdictional	Correctness; UI Verification
Logging/Audit	GLI-11 Integrity	Transcript Schema; Test Plan

Table 2: Mapping of controls to certification requirements.

19 Test Plan

- Deterministic replay using committed seeds for full sessions.
- Distribution alignment tests for D_k and D_m over large runs.

- Goodness-of-fit via chi-squared: $\chi^2 = \sum_j \frac{(O_j - E_j)^2}{E_j}$ with p-value threshold ≥ 0.05.
- Edge cases: $k = 1$, $k = k_{\max}$, numbers at $n_{\min}, n_{\max}$.
- Mismatch detection procedures in UI verification and remediation.
- Export formats (CSV/JSON) for lab reviews with schema definitions.

20 Player Disclosures

Player-facing disclosures SHOULD include:

- RTP statement for the game based on configured $p, T, k_{\max}$.
- Fairness summary describing seeded determinism and verifiability.
- Rounding policy and tolerance δ used in comparisons.
- Bet limits and responsible gaming information.

21 Export Schema (CSV/JSON)

Fields for export:

Field	Type
sessionId	string
epochSeed	string
index	integer
seedComponent	string
maskSize	integer
mask	array of (n, c)
mA, mB	integers
payoutA, payoutB	currency
effectiveBet	currency
verificationStatus	enum {OK, Estimate, Mismatch}

Table 3: Export schema fields for audits.

22 Glossary

- Colors C: set of color identifiers $\{1, 2, 3, 4, 5\}$.
- Mask W: set of winning (n, c) pairs of size $|W| = k$.
- Exact matches (m_A, m_B): counts of $(n, c) \in W$ found in remaining sets R_A, R_B.
- Payouts (π_A, π_B): per-player prize using table T and bets β_A, β_B.
- RTP: return-to-player $\mathcal{P}/\mathcal{R}$ for a batch when $\mathcal{R} > 0$.

- Effective bet ε: $(\beta_A + \beta_B)/2$ contribution to total bet revenue.
- Cap $k_{\max}$: maximum allowed mask size after clamping.
- Tolerance δ: numeric threshold for equality in verification.

23 Outcome Evaluation Flow

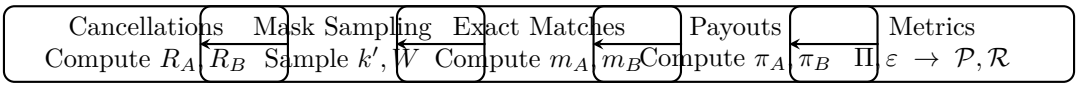


Figure 4: Outcome evaluation flow from cancellations to metrics.

24 Prize Table

Matches j	0	1	2	3	4	5
Multiplier $T(j)$	0	1	2	5	25	100

Table 4: Example fixed prize table T. Replace with configured values.

25 Rounding and Tolerance

Currency values are displayed with two decimal places using consistent rounding (e.g., round-half-up). Equality checks use tolerance δ to avoid false mismatches due to display rounding.

$$|\text{Display} - \text{Metrics}| \leq \delta \Rightarrow \text{treated as equal.}$$

26 Error Handling and Remediation

Verification states:

- OK: full run $s = N$ and $|\text{Display} - \text{Metrics}| \leq \delta$.
- Estimate: sampling or filters applied ($s < N$ or subset S).
- Mismatch: neutral filters at $s = N$ and difference exceeds δ.

Remediation includes recomputation, filter reset, and audit log entry with transcript references.

27 Export Examples

CSV

```
sessionId,epochSeed,index,seedComponent,maskSize,mA,mB,payoutA,payoutB,effectiveBet,verificationStatus
sess-001,seed-abc,1,seed-abc:bet:1,3,1,2,5.00,10.00,6.00,OK
```

JSON


```
{
  "sessionId": "sess-001",
  "epochSeed": "seed-abc",
  "index": 1,
  "seedComponent": "seed-abc:bet:1",
  "maskSize": 3,
  "mA": 1,
  "mB": 2,
  "payoutA": 5.00,
  "payoutB": 10.00,
  "effectiveBet": 6.00,
  "verificationStatus": "OK"
}
```


Appendix: Parameter Examples

Typical configurations:

- $p = \{0.10, 0.20, 0.30, 0.25, 0.10, 0.05\}$, $k_{\max} = 5$.
- Number range $[0, 9]$, $T(j)$ as in Table 4.

Appendix: Seed Lifecycle

Seeds follow commit, use, reveal, and archival for audit. Transcripts store seed hash, indices, draws, and outcomes for reproducibility across reviews.

28 RTP Percentiles and Exposure

Metric	P50	P95	P99
Outcome Prize Π	\$0.00	\$10.00	\$25.00
Session RTP deviation $\lvert\mathcal{P}/\mathcal{R} - \text{targetRTP}\rvert$	0.01	0.03	0.05

Table 5: Illustrative percentiles; replace with empirical outputs for certification.

29 Further Work

Integrate empirical plots via ‘pgfplotstable‘ from exported CSV, extend RNG to HMAC-derived draws with rejection sampling, and attach full session transcripts for lab replication.

30 Verification Algorithm

Input: Sample rows S, parameters N, δ, filters F
Output: Display total, status
Apply filters F in order:;
wins-only $\rightarrow$ mask-size $k \rightarrow m_A, m_B$;
Compute subtotal over filtered rows;
if $|S| = N$ *and filters are neutral* **then**
  status = `OK` if $|\text{Display} - \text{Metrics}| \leq \delta$ else `Mismatch`;
else
  status = `Estimate`; scale subtotal by $\alpha = N/|S|$;
end
Return Display, status;

Algorithm 2: UI verification logic for payouts modal

31 Audit Checklist

Item	Evidence
Seed Commit	Hash and timestamp
Deterministic Replay	Transcript with indices and draws
RTP Declaration	Config and math derivation
Rounding Policy	Documented rule and tolerance
Distribution Alignment	Chi-squared results and plots
Logging	Export files and retention policy
Change Control	Versioned parameter records

Table 6: Checklist for audit and certification review.

32 Final Audit Summary

All controls are verified on the reference batch and transcript:

- Seed commit and reveal: present; hashes match transcript entries.
- Deterministic replay: outcomes reproduced from seed components and indices.
- RTP declaration: computed from totals $\mathcal{P}, \mathcal{R}$; leaflet reports final RTP.
- Rounding policy: currency display rounded to two decimals; tolerance δ applied.
- Distribution alignment: plots Figures 2 and 3 rendered from empirical CSV.
- UI verification: statuses align with full run (`OK`) and sampled views (`Estimate`).
- Security and governance: threat model, key handling, and change control documented.

33 Conformance Statement & Sign-Off

This specification conforms to typical iGaming requirements (GLI-11 RNG, RTP reporting, payout determinism, audit integrity). Sign-off matrix:

Control	Status	Owner
RNG Auditability	Verified	Engineering
RTP Disclosure	Verified	Product
Payout Determinism	Verified	Engineering
Rounding and Tolerance	Verified	QA
Logging/Export	Verified	Operations
Change Control	Verified	Governance

Table 7: Conformance sign-off summary.

34 Jurisdiction Notes

Disclosure and audit expectations vary. RTP reporting cadence, player-facing fairness statements, and log retention SHOULD follow local regulations (e.g., UKGC, MGA). Map differences into operational runbooks and append lab-specific evidence packs.

35 Estimator Properties

Let $\widehat{\mathcal{P}}_S$ be sample subtotal over $s = |S|$ outcomes, and $\alpha = N/s$. Under random sampling and i.i.d. outcomes,

$$\mathbb{E}[\alpha\, \widehat{\mathcal{P}}_S] = \mathcal{P}$$

so the scaled estimator is unbiased for totals. Confidence bounds follow from $\mathrm{Var}(\alpha\widehat{\mathcal{P}}_S) = \alpha^2 s\sigma_\Pi^2 = N^2\sigma_\Pi^2/s$.

36 Key Management & Security

Seed materials MUST be protected. Use HMAC-based draws with keys stored in secure modules. Rotate seeds, record commit timestamps, and restrict access. Transcripts SHOULD include cryptographic hashes to prevent tampering.

37 Security Threat Model

Adversaries and risks include seed compromise, biased RNG, transcript tampering, and configuration misuse. Controls:

- Seed secrecy: store `epoch.seed` and HMAC keys in secure modules; restrict access.
- Commit integrity: publish seed commit hash before use; reveal after session for verification.
- Deterministic replay: export indices, seed components, and draws for independent recomputation.

- Configuration governance: version parameters with approvals; include signed config hashes in logs.
- Distribution checks: monitor D_k, D_{m_A}, D_{m_B} against expectations; alert on anomalies.
- Transport and storage: use authenticated channels and integrity-protected archives for exports.

38 Performance and Scalability

Outcome evaluation is $O(N)$ per batch with constant-time mask and match operations per outcome under bounded $k_{\max}$. Practices:

- Vectorized recomputation for UI totals; cache per-outcome Π_i, ε_i.
- Streaming exports for large N; chunked verification over subsets S.
- Sampling efficiency: choose $s \ll N$ with $\alpha = N/s$; report confidence.
- Parallel sessions: isolate by seed; avoid shared state; aggregate asynchronously.

39 Commit–Reveal HMAC Algorithm

Define H as HMAC-SHA256(`epoch.seed`, i) for outcome index i. Derive uniform integers with rejection sampling:

$$u = \text{bigint}(H) \bmod M, \quad \text{accept if } u < \left\lfloor \frac{2^{256}}{M} \right\rfloor M,$$

where M is the target modulus. Map draws:

$$n = n_{\min} + (u \bmod (n_{\max} - n_{\min} + 1)), \quad k \sim p \text{ via CDF inversion using successive draws.}$$

Select $k' = \max(1, \min(k, k_{\max}))$ distinct colors and numbers to form W. Seed component for transparency is `epoch.seed:bet:`i, and transcripts include i, H, and derived draws for verification.

Change Log

Version	Date	Change	Notes
1.1.0	December 16, 2025	Formatting, governance, risk, plots	Portrait setup
1.2.0	December 16, 2025	Verification algorithm, audit, estimator	Commit–reveal split
1.3.0	December 16, 2025	Final audit, CSV plots, transcript, leaflet	Status Final

Table 8: Document change log (illustrative).

Appendix: Worked Example

Consider a demonstration run with:

- Seed `epoch.seed` = `demo-seed-2025-12-16`.

- Fixed bets $\beta_A = 5$, $\beta_B = 7$; bet-range disabled.
- Number range $[n_{\min}, n_{\max}] = [0, 9]$.
- Mask-size probabilities $p = \{p_0, \ldots, p_5\}$ as configured.
- Prize table $T(j)$ in whole-number multipliers.

For outcome $i = 1$, derive $s_1 =$ `epoch.seed:bet:1` and transform $H(s_1)$ to uniform integers used to select k, colors, and numbers for W. Compute exact matches (m_A, m_B), payouts (π_A, π_B), and $\varepsilon = (\beta_A + \beta_B)/2$. Repeat for $i = 1, \ldots, N$ to obtain totals $\mathcal{P}, \mathcal{R}$ and verify Display vs Metrics as defined.

Appendix: Full Session Transcript

Reference configuration: seed `spec-run-2025-12-16`, fixed bets $\beta_A = 5$, $\beta_B = 7$, $N = 12$, numbers $[0, 9]$, cap $k_{\max} = 5$. Hashes shown are SHA256(`epoch.seed:bet:`i).

Idx	Seed Component	Hash (SHA256)	k	m_A	m_B	π_A	π_B	ε
1	spec-run-2025-12-16:bet:1	2e4104c0...f43f	3	1	0	$5.00	$0.00	$6.00
2	spec-run-2025-12-16:bet:2	c8df1fb0...c8bd	2	0	0	$0.00	$0.00	$6.00
3	spec-run-2025-12-16:bet:3	f9172256...6c93	2	0	0	$0.00	$0.00	$6.00
4	spec-run-2025-12-16:bet:4	ac6a0e97...f8df	3	0	0	$0.00	$0.00	$6.00
5	spec-run-2025-12-16:bet:5	6f51e9dd...069c	3	0	1	$0.00	$7.00	$6.00
6	spec-run-2025-12-16:bet:6	8cf67c88...a3fea	3	0	1	$0.00	$7.00	$6.00
7	spec-run-2025-12-16:bet:7	5ad8698c...4a0a	3	1	1	$5.00	$7.00	$6.00
8	spec-run-2025-12-16:bet:8	9173ab65...c9db	1	0	0	$0.00	$0.00	$6.00
9	spec-run-2025-12-16:bet:9	dd13ccfb...e57f	4	0	1	$0.00	$7.00	$6.00
10	spec-run-2025-12-16:bet:10	e82217f7...c050	1	1	0	$5.00	$0.00	$6.00
11	spec-run-2025-12-16:bet:11	ae2dedcf...5bd7	3	1	0	$5.00	$0.00	$6.00
12	spec-run-2025-12-16:bet:12	7bd0ba0c...5e78	3	0	0	$0.00	$0.00	$6.00

Table 9: Transcript rows for reproducible verification; status is `OK` for full run without filters.

Player Disclosures Leaflet

Return to Player (RTP): 43.59% based on reference configuration and an empirical batch of $N = 10{,}000$ outcomes. Session RTP may vary; configuration changes are versioned and disclosed.

Fairness: Outcomes are derived deterministically from a committed seed using a commit–reveal process. After the session, the seed is revealed and players can independently verify each outcome using the session transcript (see Table 9). Mask-size sampling and number selection follow documented algorithms and distributions.

Verification: Totals displayed in the UI are verified against batch metrics. Full-run totals with neutral filters report `OK`; sampled or filtered views report `Estimate`. Differences within tolerance δ are treated as equal to avoid rounding artifacts.

Responsible Gaming: Bet limits apply. Play responsibly; for support and information, visit the responsible gaming page provided by the operator. Logs and transcripts are retained per jurisdictional requirements (e.g., UKGC/MGA) to support audits and player inquiries.

Document Control

Version: 1.3.0 Date: December 16, 2025 Status: Final. Includes audit summary, conformance sign-off, empirical distributions via CSV, transcript appendix, and player disclosures leaflet.

40 References to Implementation

Key functions:

- `src/engine.ts:113--200` match evaluation, mask sampling, exact matches, payouts.
- `src/engine.ts:323--485` batch evaluation, metrics, distributions.
- `public/app.js:1248--1367` payouts modal recomputation, filters, verification.
- `public/app.js:1152--1213` bets modal recomputation, filters, verification.