

THIRSTY MODERATOR

Developer handoff

Release v6 | 19 September 2026

A portable, playable demo and the information needed to continue development.

Feature behavior, measured odds, return distributions, deterministic math, production RGS calls, source map and deployment instructions.

96%
theoretical production RTP

25,000x
maximum / base bet

5 modes
1 million outcomes each

Run the prebuilt demo

`node server.mjs`

Open <http://localhost:3000/>

Public demo: no login and play credits only. Real-money wallet authority belongs to the external RGS. Local verification does not constitute Stake staging or operator approval. Current character clips are preserved; visual sign-off is paused.

19 September 2026. Scope: public play-credit demo, editable frontend/game logic, current runtime assets, external RGS integration and reproducible production math export. Character animation changes are paused; this release preserves the current clips and does not claim that their visual quality or lip sync has been approved.

1. Start and deploy

Extract the ZIP, install Node.js 24+, run `node server.mjs` in its root, and open `http://localhost:3000/`. No dependency installation is needed for the prebuilt game. The default host is 0.0.0.0 and port 3000. PORT, HOST and BASE_PATH configure the server. GET /healthz returns a JSON health response.

For a static host, upload only the contents of `web/`. All assets use the game directory, so `/` and `/games/thirsty/` both work. Keep the trailing slash on directory launch URLs. Forward the launch query unchanged. Serve MP4 with `video/mp4` and Range support, JS as `text/javascript`, XML as `application/xml` and fonts with their font MIME. Optional `.br/.gz` companions require Content-Encoding and Vary: Accept-Encoding. Cache `index.html` with revalidation; cache hashed bundles immutably. The included Node server implements these behaviors. Do not add a catch-all HTML rewrite for missing assets or wallet endpoints.

The server publishes only `web/`. It excludes source, documentation, tests, package metadata and local configuration. It supports GET/HEAD, safe subfolder mounts, range/conditional requests, and graceful shutdown. Use your hosting service or reverse proxy for HTTPS and process supervision. Dockerfile and an Nginx static configuration are included; only the Node path has been executed in the local release check.

The default demo starts with 1,000 CR. Settings can refill credits while idle. No login is needed. Each browser keeps its own balance, pending round and preferences. `localStorage` keys are `mod-shift-demo-v1`, `mod-shift-progress` and `mod-shift-preferences`. Demo credit balances are editable browser state, not a financial ledger. Gameplay randomness uses Web Crypto; the packaged demo ID helper also supports HTTP test LANs where `crypto.randomUUID` is unavailable.

2. Modes, costs and headline odds

Every cost and maximum below is relative to the selected base bet unless explicitly labelled purchase cost. For example, a 1 CR bet costs 120 CR for Night Shift and 400 CR for Admin Shift. The round cap is 25,000 CR in either case, not 25,000 times the buy cost.

Mode	Cost / base bet	RTP / cost	Any payout	Profit	25,000x win
Standard (base)	1x	96%	41.4584%	14.848099%	1 / 129,554.8
One mod (push1)	5x	96%	50.907803%	5.739235%	1 / 39,903.92
Two mods (push2)	10x	96%	59.206287%	5.135761%	1 / 18,797.17
Night Shift (bonus)	120x	96%	90.000%	20.000%	1 / 2,259.84
Admin Shift (super)	400x	96%	90.000%	20.000%	1 / 294.52

These are long-run weighted book statistics, not session guarantees. Profit means total payout greater than the entire purchase cost. Any payout can still be a net loss. A cap hit includes every supported route, including Embassy and ordinary accumulated wins. High volatility is the product label; the measured dispersion is included in section 8.

Bonus entry

Paid mode	Normal entry	Direct super entry	Later normal -> super
Standard	0.107326%	0.000688%	0.021549%
One mod	2.000%	0.400%	0.389601%
Two mods	4.000%	1.000%	0.778337%
Night Shift	99.202629%	0.797329%	18.759831%
Admin Shift	0.000%	99.999908%	0.000%

All entries in this table are percentages per paid round. Normal and direct-super entry are disjoint; a later upgrade occurs after a normal entry and must not be added to normal entry to count distinct triggered rounds. One mod preserves normal entry 1/50 and direct super 1/250. Two mods preserves 1/25 and 1/100 respectively. A bought round can very rarely reach the cap in its entry board before the feature opens; that is recorded as no entry. Night Shift can open directly as Admin Shift if its entry board also supplies a bomb.

3. Board, connections and payouts

The field starts at 6 columns by 9 rows. Neighbours are horizontal or vertical; diagonals do not connect. Five or more matching ordinary symbols, including substituting wilds, form a paying group. A pure-wild group does not pay by itself. Ordinary winners clear; wilds stay in their cells and gain +1 once per clearing step in which they participate. Other tiles fall through the spaces behind held wilds, keeping their IDs; fresh tiles enter from above. Coins, wilds, bombs and other eligible items can enter on cascades.

Ordinary payable, before wild multipliers

Cluster size	High: chair / keypad	Middle: headset / drink	Low: mouse / chat
5	1.00x	0.50x	0.10x
6	1.20x	0.60x	0.20x
7	1.50x	0.80x	0.30x
8	1.70x	1.00x	0.50x
9	2.50x	1.50x	0.80x
10	5.00x	2.50x	1.00x
11-12	12.50x	5.00x	1.50x
13-14	50.00x	7.50x	2.50x
15-16	100.00x	10.00x	3.00x
17+	500.00x	25.00x	5.00x

Internal ordinary identifiers are hammer (chair), keyboard, headset, drink, mouse and chat. Art labels and filenames can differ from math identifiers. Paytable source: `source/lib/game/types.ts` and `payForSize` in `engine.ts`.

A coin group requires at least five orthogonally connected coins/wilds and at least one coin. Its base award is the sum of coin face values, multiplied by every participating wild value. Ordinary groups use the table in the same way. Example: coins totalling 12 with linked x2 and x3 wilds pay 72x base bet, subject to the remaining round cap. The presentation applies wild multipliers one at a time; the credited result is already fixed.

Coin faces start at 1, 2, 5, 10, 25, 50, 100, 250, 500 or 1,000. Boosted ordinary coins cap at 10,000 each. Their bronze/silver/gold/diamond treatment and numbers are rendered separately, so falling tiles retain their values. Money, coin value, wild value and stored Evil reserves are separate data.

4. Bonus and feature rules

Three or more moderators after base resolution award a bonus. Initial spins are twice the wild count, capped at six wilds: 3/4/5/6 wilds award 6/8/10/12 spins. The carried board retains coins, wilds and its expansion level. Other positions become empty.

During a bonus, only coins, wilds and eligible feature items can land. Ordinary prop symbols are excluded from both the outcome and cosmetic passing strip. Held coins and wilds persist. Each new wild, including a new Evil Moderator, awards two spins. A bonus ends when no spins remain, the board is full, or a connected Embassy coin ends it. At the end, qualifying coin groups are paid separately in ascending payout order until the cap is reached. These held groups do not clear and repeatedly repay during the feature.

Bomb / Admin upgrade (upgrade)

Adds its displayed +1, +2, +3 or +5 to each wild, then becomes a coin. Three wilds plus a bomb before feature entry open Admin Shift. The first bomb during Night Shift upgrades it to Admin Shift; later bombs still add values. Admin Shift uses its super coin flow and enables Embassy eligibility. Entry bomb generation is weighted more heavily to +3/+5 in direct super buys, with the exported distribution shown below. Bombs add to wild values; they do not multiply each wild by their displayed amount.

Coin boost (boost)

Multiplies ordinary coin values by 2, 5 or 10, with each coin capped at 10,000. The boost then becomes a coin. Embassy has no transferable numeric value and is excluded.

Magnet / collector (collector)

Selects a random subset of eligible tiles and brings them together near the magnet. Target counts are 4-9; when more than four eligible tiles exist, it selects at most one fewer than the eligible count. It separately limits selected wilds. Other feature symbols are protected. Bonus targeting can favor previously disconnected groups. It does not simply take every wild or every coin. In base resolution, existing connections are paid before the waiting magnet acts. The magnet becomes a coin, and the resulting eligible connection can pay.

Respin / overtime (extra)

Awards its displayed 1-3 extra spins in a bonus, then becomes a coin. It is independent of coin flow, so a sparse spin can still retrigger. Additional wilds award two spins each. Expansion on the last bonus spin is a third rescue mechanism. The occurrence table counts the respin-symbol action separately from those other spin awards.

Server expansion (expand)

Adds one row of six cells above the board, up to three times per paid round: 6x9 -> 6x10 -> 6x11 -> 6x12. It can arrive in base play, cascades or a bonus. The symbol becomes a coin. IDs, coin values, wild multipliers and Evil reserves survive the row insertion. In base play the new row fills immediately; in bonus play its six new positions start empty. An expansion at zero remaining bonus spins adds one spin to use the space. Expanded rows carry into a triggered bonus and reset on the next paid round. The board scales within its fixed monitor footprint across screen sizes.

Evil Moderator / Lord of the Mods (wild with resetter variant)

Bonus-only, at most one Evil arrival per round. Its arrival cinematic precedes the theft. It banks the uncredited numeric values of ordinary coins and all other wild points above x1, leaving those coins at zero and those wilds at x1. It never deducts wallet money or earlier credited wins. The bank tracks the original symbol IDs, not current cell positions.

To return the bank, Evil must join an orthogonal coin/wild component of at least five tiles containing another wild. Values go back to surviving original IDs. Only then does Evil receive its separate random x2/x3/x5/x10 factor, multiplying its own value. The generator's raw factor weights are 65%/25%/8.5%/1.5%; these are proposal weights, not measured conditional frequencies after the final outcome-book reweighting. Held values expire unpaid if the qualifying return never occurs. Evil is not eligible in base play or the pre-bonus entry state.

Scam (scam)

Bonus-only and eligible when some wild exceeds x1. Its cinematic shows the action, every existing wild is reduced to x1, and the scam tile becomes a coin. Already credited wins and the wallet are unaffected. Scam does not erase Evil's separately held reserve.

God mode (god)

Adds ten to every wild, then becomes a coin. It can appear on eligible fresh base/cascade landings or during a bonus. It requires a wild. The rare-symbol injector limits Evil/Scam/God arrivals to two total per round; Evil has its additional one-per-round limit. These constraints are separate from normal bombs, magnets and expansions.

Ambassy Call coin

Eligible only inside Admin Shift, including a normal bonus that has upgraded. At most one can arrive per round. It triggers the Embassy presentation and is marked as a special max-win coin, with numeric value zero. It is immune to coin boosts and Evil value theft. A qualifying five-or-more coin/wild connection containing it pays the remaining amount to reach the 25,000x round cap. An unconnected Embassy coin pays nothing on its own.

Paid mode	Coin arrives	Coin connects / max win
Standard	1 / 419,580.56	1 / 1,620,741.86
One mod	1 / 8,884.06	1 / 99,536.39
Two mods	1 / 3,513.26	1 / 43,470.3
Night Shift	1 / 581.44	1 / 25,394.3
Admin Shift	1 / 100	1 / 750

These are per paid round, including any feature reached from that round. In a direct super buy, appearance is exactly 1/100 and connection is exactly 1/750 in the integer-weight export; conditional connection among appearances is 13.333333%. Other max-win routes remain possible, which is why the overall Admin Shift cap frequency is higher than 1/750.

5. Measured feature frequencies

Each cell below is the chance of at least one recorded action anywhere in the complete paid round, including any triggered bonus. It is not the probability per cell or per bonus spin. Consequently base-mode Evil/Scam rates are nonzero only because a base spin can first enter a bonus; those symbols still never land in base/pre-entry state. A guaranteed super entry bomb makes its upgrade-action incidence 100%.

At least one activation	Standard	One mod	Two mods	Night Shift	Admin Shift
Bomb upgrade	1.145753%	1.893772%	2.842344%	19.557161%	100.000%
Coin boost	1.16424%	1.694718%	2.27068%	24.700759%	24.793187%
Magnet	1.581628%	2.589401%	3.492239%	40.895393%	37.314514%
Respin symbol	0.069494%	1.55438%	3.26406%	67.32679%	65.716414%
Expansion	1.276279%	2.310344%	3.428731%	42.87866%	42.734225%
Evil theft	0.003862%	0.095177%	0.194188%	4.105993%	6.067279%
Evil return	0.002595%	0.058943%	0.115139%	2.256359%	2.202368%
Evil expiry	0.001267%	0.036234%	0.079049%	1.849634%	3.864911%
Scam	0.000355%	0.007973%	0.019658%	0.481406%	1.771241%
God mode	0.001381%	0.008633%	0.009292%	0.027337%	0.03693%

Evil return and expiry are subsets of theft; they partition the supported theft outcomes in this release. Multiple actions of one kind count once for this table. Joint feature probabilities cannot be obtained by multiplying these marginal rates. All exact integer numerators are retained in docs/evidence/weighted-validation.json.

Expansion distribution

Paid mode	0 expansions	1 expansion	2 expansions	3 expansions	Last-spin rescue
Standard	98.723721%	1.261515%	0.011619%	0.003145%	0.004988%
One mod	97.689656%	2.05239%	0.202279%	0.055675%	0.125969%
Two mods	96.571269%	2.902477%	0.406828%	0.119425%	0.253455%
Night Shift	57.12134%	30.70282%	9.251284%	2.924555%	5.339189%
Admin Shift	57.265775%	31.134745%	9.091541%	2.507939%	4.928313%

The four expansion-count columns are disjoint and sum to 100% before display rounding. Last-spin rescue means an expansion awarded its one extra spin; it overlaps the expansion-count columns.

Initial revealed wild counts

First board wilds	Standard	One mod	Two mods	Night Shift	Admin Shift
0	76.122718%	0.000%	0.000%	0.000%	0.000%
1	21.773679%	95.610803%	0.000%	0.000%	0.000%
2	1.997589%	1.993769%	95.218896%	0.000%	0.000%
3	0.104198%	2.360062%	4.709873%	63.402856%	40.156481%
4	0.001527%	0.031005%	0.061883%	26.841564%	41.079766%
5	0.000272%	0.003954%	0.008414%	8.250553%	15.289529%
6	0.000017%	0.000408%	0.000934%	1.505026%	3.474224%

This table measures the first board only. Cascades can add wilds before feature entry, and bonuses can add still more. It is not the distribution of total wilds by the end of a feature.

Initial bomb values

First board bomb value	Standard	One mod	Two mods	Night Shift	Admin Shift
+1	0.942939%	1.271456%	1.763287%	0.64317%	16.800614%
+2	0.086936%	0.115368%	0.151662%	0.051106%	25.135885%
+3	0.018615%	0.022403%	0.030794%	0.009028%	38.590899%
+5	0.006617%	0.008563%	0.009173%	0.001924%	20.191005%

Each value row measures at least one bomb of that value on the first board. More than one value can coexist, so these rows are not a disjoint distribution and need not sum to 100%.

6. Return distribution and balancing

Total return / purchase cost	Standard	One mod	Two mods	Night Shift	Admin Shift
Zero	58.5416%	49.092197%	40.793713%	10.000%	10.000%
Above zero, below 10%	0.000%	9.0435%	21.108274%	20.000%	20.000%
10% to below 20%	5.79443%	11.178785%	18.326575%	20.000%	20.000%
20% to below 100%	14.640086%	24.597312%	14.490223%	29.939451%	29.915056%
Exactly 100%	6.175785%	0.34897%	0.145454%	0.060549%	0.084944%
Above 100%	14.848099%	5.739235%	5.135761%	20.000%	20.000%

Night/Admin buys have 10% zero return, 20% positive but below 10% of cost, 20% from 10% to below 20%, 30% from 20% through break-even, and 20% profit. Exact break-even is separated in this table. All paid outcomes include entry-board wins as well as any bonus payout. Base/extra-mod modes use their own weighted distributions.

RTP is measured against total purchase cost: $\text{sum}(\text{weight} * \text{payoutHundredths}) / \text{sum}(\text{weight}) / 100 / \text{modeCost}$. Every production mode has total integer weight 3,000,000,000,000,000 and satisfies $\text{weightedPayout} = \text{totalWeight} * 96 * \text{modeCost}$ exactly. Both numerator and denominator must be computed with integers/BigInt; converting the products to ordinary floating-point values loses precision.

The demo selects a weighted group and then a mechanical seed from `source/lib/game/generated/bonus-book.json`. Its target is 96%; its finite-entropy/floating-point selection is not a certified RGS RNG. The exact 96% figures here come from the integer production lookup files that were fully validated before packaging. Tiny integer apportionment corrections are specific to those lookup files. The generator's `draw-config.ts` rates are raw proposal rates and must not be shown as the final player-facing odds.

Do not change payout tables, costs, seed selection, event ordering, eligibility, coin flow or feature logic while keeping the old RTP claim. Recalibrate and re-export after math changes. Frontend timings, sound and artwork can change without changing book payouts. Sampling many spins alone does not prove exact RTP.

7. Presentation and controls

Normal, Turbo and Instant affect the board presentation, not the predetermined result or character playback rate. Turbo applies the configured board speed factor; Instant advances most board phases to a short settled hold. Feature cinematics and the bonus-intro acknowledgement are protected from speed changes. Clicking the board acts as Stop/speed-up. Space starts a spin; held Space requests repeated turbo spins and is prevented from activating other controls. Jurisdiction flags can disable those controls in live sessions.

The passing strip runs at a shared 280 ms per symbol step in normal presentation, including bonus and two-wild anticipation. It cycles 3-4 visible passing items near each final reveal and uses its own cosmetic hash. A roughly one-in-50 cosmetic opportunity can show a high coin/feature before a low result; this does not alter math or award probabilities. Base/pre-entry boards have actual symbols rather than empty outcome cells. Anticipation highlights unresolved positions after two wilds. Numeric values stay attached to moving tiles, and reveal ticks, coin dings, connection transfers and retrigger cues follow the board clock.

Win-count acceleration stages are one below 100x, two from 100x to below 250x, three from 250x to below 1,000x and one from 1,000x upward. Each press advances immediately, including during title transitions. The cap is 25,000x. The displayed count is presentation; settlement comes from the completed round record.

Character clips are packed-alpha video played at their own rate, with separate current idle/action assets and voice variants. Base and bonus music are separate tracks. Web audio unlocks after interaction; sound controls and page visibility pause/resume behavior remain active. Compact/mobile layouts hide the side character to keep the field and controls clear. No studio inspector route is part of the public release. Settings still includes supported demo feature previews, which are explicitly no-charge presentations and cannot alter the wallet.

8. Diversity evidence and deterministic replays

Mode	Outcomes	Distinct structures verified	Saved examples	SD / cost
Standard	1,000,000	8,768	600	80.329
One mod	1,000,000	25,921	600	30.334
Two mods	1,000,000	40,545	600	22.207
Night Shift	1,000,000	88,151	600	5.624
Admin Shift	1,000,000	371,684	600	4.111

The structural audit ignores seed numbers, cell IDs, positions, monetary amounts and ordinary-symbol skins. It retains event ordering, combinations, cluster sizes, wild participation, retriggers, upgrades and late saves. These are verified lower bounds, not all possible structures. The archive contains 600 supported distinct examples per mode (3,000 total) under docs/evidence/catalogs/, each with a positive-weight event ID, mechanical seed and payout. Independent rounds may repeat; a million distinct seeds is not a promise of a million visibly unique experiences.

Named examples in diversity-audit.json cover zero/partial/profit returns, every feature, one/two/three expansions, disconnected wilds, late upgrades, dry retriggers, last-spin rescues, chained saves, Embassy miss/hit and caps. A base-only max-win route is supported. SD / cost is standard deviation of return measured in purchase-cost units; it describes dispersion rather than a guarantee that a player will avoid long losses.

Generate a supported event without wagering: `node tools/sample-event.mjs --mode=super --event=106398 --out=sample-event.json`. That example includes three expansions, Evil return and Scam and pays 75.40x base bet. This writes a Stake event envelope for developer inspection; it is not a public wallet API or an alternative outcome-selection mechanism.

9. Frontend and RGS routes

The deployable app lives at its configured root URL. There are no private credentials or login pages in the ZIP. A launch with neither `rgs_url` nor `sessionID` uses the local demo. A launch containing an incomplete or invalid RGS context is rejected and never falls back to a local draw. Live play accepts an HTTPS RGS address; plain HTTP is allowed only when both page and RGS are loopback for development.

Live launch example: `https://your-host.example/games/thirsty/?rgs_url=https%3A%2F%2Fyour-rgs.example&sessionID=ISSUED_SESSION&lang=en¤cy=USD`

Replay launch example: `https://your-host.example/games/thirsty/?replay=true&rgs_url=https%3A%2F%2Fyour-rgs.example&game=YOUR_GAME&version=YOUR_VERSION&mode=super&event=106398&amount=1000000`

The values `YOUR_GAME/YOUR_VERSION` must come from the operator's uploaded game. A replay performs the replay GET and never places/settles a wallet bet. Amount defaults to 1,000,000 integer units. lang defaults to en; the existing visible game copy is English rather than a complete translation catalogue.

External RGS route	Method	Request fields	Client purpose
<code>/wallet/authenticate</code>	POST	sessionID, language	Balance, bet ladder, jurisdiction, active round recovery
<code>/wallet/play</code>	POST	sessionID, amount, mode	Obtain one authoritative round
<code>/bet/event</code>	POST	sessionID, event = frame:N	Optional coalesced presentation checkpoint
<code>/wallet/end-round</code>	POST	sessionID	Settle the active round after presentation
<code>/wallet/balance</code>	POST	sessionID	Refresh balance only while idle
<code>/bet/replay/{game}/{version}/{mode}/{event}</code>	GET	Encoded path values	Read-only event replay

These are client calls to the RGS supplied in the launch URL. They are not local endpoints served by `server.mjs`. JSON POST requests use no cookies, a 15-second timeout, no cache and no redirect following. The RGS must allow the frontend origin through CORS if it is on a different origin. The contract is described in `docs/rgs-openapi.json` and implemented in `source/lib/game/rgs.ts`. An invalid-session error requests reconnection; an uncertain play/settlement result requires re-authentication and active-round recovery rather than rerolling or retrying a debit blindly.

Authenticate provides integer `minBet`, `maxBet`, `stepBet`, `defaultBetLevel` and `betLevels` plus jurisdiction flags. Implemented flags include `disabledBuyFeature`, `disabledTurbo`, `disabledSuperTurbo`, `disabledAutoplay`, `disabledSlamstop`, `disabledSpacebar`, `disabledFullscreen`, `displayNetPosition`, `displayRTP`, `displaySessionTimer`, `socialCasino` and `minimumRoundDuration`. Unknown/cancelled network state blocks further play until recovered. Server authority must enforce the actual balance, betting permission and settlement once; frontend checks alone do not do that.

Units and event boundary

- Wallet amount, bet, cost and payout use integers at 1,000,000 units per currency unit. 1 CR = 1,000,000.
- Internal round multiplier, frame totals, event-book payoutMultiplier and finalWin.amount use hundredths of the base bet. $25,000x = 2,500,000$.
- The live WireRound payoutMultiplier uses base-bet multiples. For a 75.40x payout, the wire value is 75.4, whereas the event-book value is 7540.
- WireRound costMultiplier must agree with the selected mode. Its payout must agree with the event final total and bet; mismatches are rejected.
- Round frames carry immutable symbol IDs plus before/after boards. Expansion changes board length by six; never assume a fixed 54-cell board after round start.
- Stake event serialization uses roundStart, frame events (expansion is carried in frame.expansion) and finalWin. roundFromStake validates and reconstructs the presentation. The live frontend never generates a local replacement outcome.

10. Source map and safe editing

Path	Responsibility
source/stake/main.tsx	Standalone React entry
source/components/game/ModShift.tsx	Session/UI state, playback and controls
source/components/game/three/Board3D.tsx	Board, held identities and expansion view
source/components/game/retro/	Menus, source-style controls and win panels
source/lib/game/engine.ts	Deterministic mechanical round construction
source/lib/game/types.ts	Modes, cost, payable, board and event types
source/lib/game/bonus-book.ts	Weighted seed selection for demo
source/lib/game/generated/bonus-book.json	Frozen v6 calibrated compact book
source/lib/game/rare.ts	Evil reserve, Scam and God rules
source/lib/game/stake-book.ts	Event serialization / strict decoding
source/lib/game/rgs.ts	Authoritative RGS client and replay
source/lib/game/motion.ts / reel-pass.ts	Timelines, anticipation and passing strips
source/lib/game/audio.ts	Music, sprites and synchronized cues
source/lib/game/character-performances.ts	Active clips, voices and cues
source/components/game/FrameMotionStage.tsx	Packed-alpha character playback
web/art, web/brand, web/models	Current images, atlases, media, fonts and geometry
vite.config.ts	Relative asset URLs, dependency build and safe bundle cleanup
server.mjs	Zero-dependency static serving only

Install pinned dependencies with `npm ci`. Run `npm run typecheck`, `npm test` and `npm run verify:math` before `npm run build`. Build preserves the static asset tree and regenerates only its compiled bundle directory and entry. Do not add a service worker that serves stale books or mixes versions. Treat `web/` as an atomic release: publish the whole new directory and then switch the public version URL.

The package contains only the standalone game's imported source chain and its applicable tests. Unused template pages, old character rigs/renderers, original body production files, model-generation environments, local videos, `node_modules`, caches, recordings, `.git` state and secrets are excluded. Active symbol/panel Spine atlases, game fonts, models and runtime media are retained because the frontend needs them. Refer to `THIRD-PARTY-NOTICES.txt` for preserved dependency notices and current asset provenance.

11. Production math export and submission handoff

Five million complete event records and corresponding integer-weight rows were validated before this packaging pass. Those event/LUT files total multiple gigabytes and are unnecessary to run the public demo. The ZIP instead includes the exact compact seed book, frozen hashed engine sources, integer apportionment logic, exporter, validator and evidence manifest. No paid backend is embedded.

After `npm ci`, choose a new absolute output directory with at least 10 GB free and run:

```
npm run verify:math
node tools/export-stake-books.mjs --out=/absolute/new/math-output
node tools/validate-stake-books.mjs --dir=/absolute/new/math-output
```

On Windows, quote the full absolute path when it contains spaces. Export/validation are CPU and disk intensive. They use Node 24's built-in Zstandard support. Do not place output inside `web/`. The exporter refuses an already indexed release directory. It outputs `index.json`, `books_MODE.jsonl.zst`, `lookUpTable_MODE_0.csv` and `calibration.json`. The validator adds `validation.json`. Keep modes `base/push1/push2/bonus/super`, costs, event IDs and weights together. Each lookup row is `id,weight,payoutMultiplier` with no header; the payout column is in hundredths. Upload `index.json` and the event/weight files it references through the platform's math workflow; upload `web/` as the frontend version.

`Production-math-manifest.json` records the hashes of the previously validated event/LUT files and math sources. Re-exporting with identical code/book preserves deterministic events and weighting; compressed byte hashes can depend on the compressor/runtime version. Validate the decoded content and exact integer totals, not only a compression hash. The included source verifier checks all frozen hashes and seed counts and performs sampled event round-trips; it does not pretend to regenerate or revalidate all five million records in seconds.

For future math edits, `tools/calibrate-candidate.mjs` is the portable calibration workflow. In a separate checkout, run `node tools/calibrate-candidate.mjs --version=7 --out=/absolute/new/candidate`. It draws one million candidates per mode by default, fits the existing RTP/entry/return-band/Ambassy targets, and writes a candidate `bonus-book.json` plus calibration report. It never replaces the shipped book automatically. Review support, diversity and exact payout targets, promote the reviewed candidate to `source/lib/game/generated/bonus-book.json` in that separate version, then export and validate all rows. The current `verify:math` script and shipped evidence deliberately describe v6; regenerate/update that evidence and its checks for a new version. A changed design may make the old target combination infeasible; calibration must fail rather than silently invent a guaranteed payout.

Remaining external steps are actual Stake upload, staging sessions, `wallet/resume/jurisdiction/replay` checks against the operator's RGS and approval. The route contract and local tests are ready to support that work; they are not evidence of platform approval. Character visual sign-off remains paused separately. There is no guarantee of winning within any number of buys, and no adaptive wallet-based balancing.

12. Verification and references

See RELEASE-MANIFEST.json for this archive's final extraction/build/test results and SHA-256 file inventory. The carried math validation and diversity evidence are under docs/evidence/. The editable tests cover RGS error/recovery behavior, event parsing, cap handling, feature eligibility, board identity/expansion, presentation clocks and game rules. The release server adds root/subpath, MIME, video range, caching, inaccessible-source and missing-route checks.

Primary implementation references: source/lib/game/engine.ts, types.ts, rare.ts, bonus-book.ts, stake-book.ts and rgs.ts. Weighted figures are computed directly from the versioned integer validation report. Proposal probabilities come from draw-config.ts and evil-release.ts and are labelled separately.

Stake platform reference: <https://stake-engine.com/docs>. Endpoint references: <https://stake-engine.com/docs/api/authenticate>, <https://stake-engine.com/docs/api/play>, <https://stake-engine.com/docs/api/end-round>, <https://stake-engine.com/docs/api/event> and <https://stake-engine.com/docs/api/bet-replay>. Recheck the platform's current requirements during staging; this document records what this release implements.