

Ze Claw



A 7×7 cluster-pays slot with a claw-machine prize engine

Designed & built by **Jack Drach** · jackdrach.com

1 Overview

Game name	Ze Claw
Theme	Neon arcade claw machine — plush toys, gumball capsules, and a chrome grabber
Grid	7×7 (49 cells)
Mechanic	Cluster pays with tumbling reels
Bonus engine	"Claw drop" — a weighted prize stream injecting wilds, multipliers, sticky spots, and cash into the bonus board
Bet range	\$0.20 – \$10.00 (8 steps)
Max win	30,000× bet (per bonus session)

The hook: Ze Claw reskins the cluster-pays-with-tumbles framework around an arcade claw machine, and replaces the genre's usual "collect-and-upgrade" bonus with a **claw drop** — every cascade in a bonus rolls a chance to drop the grabber, delivering a randomly weighted prize from a tiered pool. The result is a base game that grinds steady cluster wins toward a rare scatter trigger, and two purchasable bonus modes whose entire variance is *earned through the prize stream* rather than pre-seeded multipliers. The premium Jackpot mode is tuned as a true "mega-win-or-bust" feature, with the 30,000× cap genuinely reachable.

2 Key Stats

BASE RTP 91.6% 200k spins	PRIZE BUY RTP 93.4% 100k bonuses	JACKPOT BUY RTP 94.8% 100k bonuses	MAX WIN 30k× per bonus
--	---	---	-------------------------------------

METRIC	VALUE	NOTES
Grid / win rule	7×7, clusters of 5+	Orthogonal connection (no diagonals)
Base game RTP	91.59%	200,000 simulated base spins
Prize Buy RTP	93.38%	100,000 simulated bonuses

METRIC	VALUE	NOTES
Jackpot Buy RTP	94.76%	100,000 simulated bonuses
Base hit frequency	31.36%	% of base spins returning a win
Natural bonus trigger	~1 in 312 spins	3+ scatters → Prize mode
Volatility	High / Extreme	Base & Prize / Jackpot
Jackpot 500×+ rate	~9.9%	Share of buys returning ≥500×
Jackpot cap-hit rate	~0.5%	Share of buys hitting 30,000×

All RTP figures were measured by a Monte-Carlo harness running the **real game logic** — not a separate abstract model. See §5.

3 Core Mechanics

Base spin

A spin fills all 49 cells from a weighted symbol pool. The board is scanned for clusters; any group of **5+ identical symbols connected orthogonally** pays. Winning cells clear, survivors fall under gravity, and new symbols drop from the top — then the board re-scans. This **tumble** loop repeats until no new clusters form.

Symbols & payable

Seven paying symbols across three tiers, plus a Wild and two Scatters. Payouts (×bet) interpolate between five cluster-size breakpoints — **5 / 7 / 10 / 12 / 15+** — rather than fixed brackets.

SYMBOL	TIER	WEIGHT	@5	@7	@10	@12	@15+
Gold Teddy	Premium	14	1.0	2.5	7.5	15.0	150.0
Unicorn	Premium	17	0.75	2.0	6.0	12.5	100.0
Robot	High	18	0.5	1.75	4.0	10.0	60.0
Duck	High	16	0.4	1.25	3.0	7.0	40.0
Pink Token	Low	12	0.3	0.75	2.0	5.5	30.0
Blue Token	Low	11	0.25	0.4	1.25	2.75	25.0
Yellow Token	Low	10	0.2	0.25	1.0	2.5	20.0

Symbol weights are inversely correlated with pay (higher-value symbols are rarer); weights total 98 across the seven payers. Scatters are rolled separately per new cell.

Win-count multiplier curve

Every cell tracks a **win count** — the number of times it has joined a winning cluster during the current bonus. Each cell contributes `min(32, 2^winCount)` to its cluster's multiplier sum, with the first win contributing 0× (the "free" first hit):

WIN COUNT	0	1	2	3	4	5+
Multiplier	0×	0×	2×	4×	8×	32× capped

The cluster payout formula:

```
payout = base_pay(symbol, cluster_size) × max(1, multSum)
```

where `multSum` is the sum of every contributing cell's multiplier (win-count-derived **plus** any sticky-spot value on that cell). The `max(1, ...)` ensures a fresh cluster still pays its base.

Wilds

The Wild is a single-cell, sticky symbol. It substitutes for any paying symbol (not scatters), can count toward **multiple** different-symbol clusters in one tumble, and inherits its cell's multiplier — but **its own win count never grows** (wilds inherit, they do not compound). Wilds placed by claw prizes persist for the rest of the bonus.

Scatters & triggers

Scatters do not form clusters or pay directly. **3+ scatters** trigger the Prize bonus, with the spin grant scaling by count:

SCATTERS	3	4	5	6	7+
Spins awarded	10	12	15	20	30

A separate, very rare Jackpot-scatter (~1 in 9,300 base spins) can trigger Jackpot naturally; in practice Jackpot is reached almost exclusively via Bonus Buy.

Caps: 30,000× max win per bonus · 100 tumbles max per spin · 50 spins max per bonus · cell multiplier capped at 32×.

4 Bonus Features

Both bonuses run the same engine: a clean board, a spin grant, and a **claw drop** rolled after every cascade. When the claw fires it draws one prize from a weighted pool; a rare **golden claw** draws from a hard sub-pool of the best prizes. Modes differ only in claw rate, golden rate, prize pool, and buy cost — variance is earned entirely through the prize stream, with no pre-seeded multipliers.

PARAMETER	PRIZE MODE	JACKPOT MODE
Entry	3+ scatters, or buy 95×	Buy 500× (≈\$100 at \$0.20)
Validated RTP	93.38%	94.76%
Spins granted	10 (up to 30 by scatter count)	10
Claw rate / tumble	2.0%	9.0%
Golden claw rate	1% of claws	10% of claws
Prize pool	8-prize "keep-it-going" pool	10-prize pool weighted to board-changers
Variance	High	Extreme

The Jackpot claw prize pool

TIER	PRIZES	ROLE
DECENT 80%	extra_spins (+3/5/10), single/double wild, small_cash (25/50/75×), seed_bomb (5 cells → 8× ticket), small_sticky_spot (10/20×)	Bread-and-butter that keeps boards alive
MEDIUM 17%	rolled_wild_count (2–3 wilds), multiplier_rain (3–5 cells × 3/6/10/15×)	Board-shifting boosts
INSANE 3%	big_rolled_sticky_spot (12/25/50×, top-skewed), board_frenzy (wilds + multiple sticky spots; 20% mega)	Dream prizes that carry a session to the cap

The **golden claw** in Jackpot guarantees a draw from the four hardest prizes (both MEDIUM + both INSANE), so at a 10% golden rate a player sees roughly one guaranteed power-prize per bonus.

RTP contribution — base game

The base game's 91.59% RTP splits cleanly between direct play and the bonus it feeds:

SOURCE	RTP CONTRIBUTION	SHARE OF BASE RTP
Direct base-spin cluster wins	60.29%	65.8%
Naturally-triggered Prize bonuses	31.30%	34.2%
Total	91.59%	100%

Each natural trigger is worth ~97.5× bet on average, firing ~1 in 312 spins — confirming a base game that grinds toward a meaningful, relatively rare scatter event.

5 Math & RTP Design

Philosophy & validation

Ze Claw's math was validated against a **Monte-Carlo harness that executes the real game logic** — the same cluster detection, tumble, win-count, and claw-prize code paths the live game runs — rather than a simplified parallel model. This was deliberate: an abstract RTP model can agree with itself while diverging from what the game actually pays. Figures here come from large samples through that harness (200k base spins, 100k per bonus mode).

A second Python simulator was maintained as a cross-check. Keeping three implementations (live game, JS harness, Python sim) in sync was the project's central engineering discipline, and reconciling them surfaced real bugs — including a wild-multiplier-compounding divergence and a fused gravity-and-refill bug that inflated cascade rates. The harness, verified value-for-value against the live game, was the source of truth for all tuning.

The Jackpot tuning problem

Jackpot began tuning at ~**320% RTP** against a 96.5% target — over 3× too hot. The instructive part was *what actually drove it*. The intuitive culprits — the big sticky-spot and board-frenzy prize values — were tested first and **barely moved RTP** (halving the INSANE prizes shifted RTP ~4 points). The real driver was the **win-count multiplier compounding engine**: across a bonus, cells climb the doubling curve, and unbounded compounding let too many sessions snowball into the cap.

The clean, high-leverage lever was the **cell multiplier cap**. Lowering it from 512× to **32×** throttled the deepest compounding chains without distorting *how often* big wins occurred — a cap sweep (96 → 64 → 48 → 32) traced an almost-linear RTP

response and landed Jackpot at ~105%, just above target, with the distribution shape intact.

The tail-vs-RTP discovery

The original target paired 96.5% RTP with a "2–4%" tail of 500×+ wins. Tuning proved these **mathematically incompatible at a 500× buy**: the bottom ~80% of bonuses structurally contribute ~80–88× regardless of tuning, so for total RTP to reach ~96.5% the tail must supply the remainder — forcing a tail of ~10%, not 2–4%. The achievable, healthy target was therefore RTP ≈ 96.5% with a ~10% tail, and Jackpot finalised at **94.76% RTP / 9.9% tail / 0.5% cap-hit** — a textbook extreme-variance curve (≈55% of buys under 50×, a smooth middle, a fat reachable top end).

Prize mode

Because the cell cap is global, lowering it to 32× also pulled Prize down — from a stale ~111% to ~82%. Prize was then tuned up to **93.38%** by lifting the flat cash prize to 245× and adjusting pool weights — a reminder that the cap is a shared lever and every mode had to be re-measured after it moved.

Key tuning levers (by leverage)

#	LEVER	EFFECT
1	Cell multiplier cap	Dominant RTP control; throttles compounding depth without flattening the tail
2	Claw rate	Scales the entire prize stream; held in reserve to protect the signature mechanic's feel
3	Prize-value rolls	Shape the <i>magnitude</i> of the tail far more than its RTP
4	Mid/common prize values	Recover RTP without cutting the dream tail

6 Design Intent

Ze Claw is built to feel like a real arcade claw machine: most grabs come up empty, but the machine is *visibly* full of prizes you're playing toward.

Base game is a steady grind — a ~31% hit frequency keeps small cluster wins landing often enough to stay engaging, while the rare (~1-in-312) scatter trigger is the event everyone is chasing. Over a third of base RTP lives in that trigger, making the bonus feel like the payoff rather than a footnote.

Prize mode is the accessible bonus: high variance, a forgiving claw stream that keeps boards alive, and frequent enough wilds and sticky spots to build satisfying mid-size wins.

Jackpot mode is the centerpiece and emotional core — a deliberate "mega win or bust" feature. Most buys lose; a meaningful ~10% clear the buy cost handsomely; and a real ~0.5% slam the 30,000× ceiling. The variance is *earned* through the claw stream rather than handed out by a pre-seeded board, which makes a big Jackpot hit feel built rather than given. The entire tuning effort was oriented around protecting that fat, reachable tail — never flattening it to hit a number — because the dream is the product.

What makes the design distinctive: the claw drop reframes the cluster-pays bonus as a *prize-delivery* event rather than a multiplier-accumulation grind, and the math was honest about its own constraints — discovering and accepting that the RTP/tail relationship at a 500× buy *forces* a ~10% tail, rather than chasing an impossible target.

7 Tech

Stack	Vanilla HTML / CSS / JavaScript (no framework, no build step)
Rendering	DOM-based grid and UI; CSS-driven animation (transform / opacity / filter only)
Audio	Custom-synthesised SFX (cluster, tumble, claw alarm, bonus triggers) + looping background track
Math validation	Node Monte-Carlo harness running live game logic + Python cross-check simulator
Outcome model	Single source of truth: live game, harness, and sim kept value-for-value in sync; harness verified against the game as the authority for all RTP figures
Info screen	Data-driven payable/rules overlay reading every value live from game constants — cannot drift from real pay values
Deployment	Static site, hosted on the jackdrach.com portfolio

Verify Before Publishing

Values to confirm against the latest build, since not all were locked from a single source:

- **Base hit frequency (31.36%)** and the **base RTP split** — from the 200k base-harness run; confirm against the most recent base-game harness (the base loop was the last piece built).
- **Natural trigger rate "1 in 312"** vs the code's scatter-chance comment ("~1 in 308"). Both in range; use whichever the latest sim reports.
- **Jackpot tail 9.9% / cap-hit 0.5%** — from the final 100k confirm. Re-run if any Jackpot prize value changed since.
- **Avg returns (Prize 88.7× / Jackpot 473.8×)** if publishing avg-return figures alongside RTP.
- **Volatility labels** are descriptive, not a computed volatility index. A studio expecting a formal index would need a dedicated measurement.
- **Symbol names** — the info screen displays "Alien / Red Ball / Blue Ball / Orange Ball" via a display-name override; this doc uses the code IDs (Robot / Pink-Blue-Yellow Token). Decide which to present publicly.
- **Source comments:** several inline comments in `game.js` are stale (e.g. a "250/500/1000×" comment whose real rolls are 12/25/50×). This doc uses the real constant values — worth cleaning the comments for credibility if a recruiter reads the source.