Here is the comprehensive specification for the **SpriteGen QA & Evaluation System**.

This design treats **Baseline Stability** and **Identity** as "Hygiene Factors" (failures result in immediate rejection or harsh penalties) while treating **Structure** (Pose) more lightly to allow for necessary animation fluidity.

1. Hard Gates ("The Kill Switch")

Filter out unusable frames before wasting compute on detailed scoring. If any condition is met, the result is an immediate REJECT.

Gate Metric	Threshold	Rationale
Baseline Error	> 3 px	In 2D fighting games, pivot shifts >3px create visible "jitter" or "ice skating" that breaks the ground connection.
Identity Drift	vision_identity_drift_score ≥ 4	On a 1-5 scale, a score of 4 or 5 implies the character is unrecognizable. No amount of pixel polish can fix a wrong face/costume.
Palette Breach	palette_delta > 0.15	High delta (>15%) implies hallucinated colors (e.g., wrong team colors) or lighting artifacts that break the game's indexed palette system.
Canvas Safety	bbox_touching_edge == True	(Implicit Check) The sprite is cropped/cut-off and cannot be used in a sprite sheet.

---

2. Score Formula (Total ∈ [0, 100])

The scoring function uses a Weighted Sum with Exponential Decay for stability.

Why Exponential? A linear penalty is too lenient. A 1px jitter is significantly worse than 0px, and 2px is nearly unusable. Exponential decay enforces "S-Rank" only for pixel-perfect stability.

$$ \text{Score} = 100 \times (w_{St} S_{St} + w_{Id} S_{Id} + w_{Sy} S_{Sy} + w_{Te} S_{Te}) $$

A. Weights (Priorities)

- **w_{St} (Stability) = 0.30** (Highest operational priority: Game Feel)
- **w_{Id} (Identity) = 0.35** (Highest visual priority: Character Lock)
- **w_{Sy} (Style) = 0.20** (Line work and shading consistency)
- **w_{Te} (Technical) = 0.15** (Palette cleanliness and structural integrity)

B. Component Normalization

1. **Stability (S_{St}):** Exponential decay based on pixel error.
    - $$ S_{St} = e^{-(\text{baseline_error_px})} $$
    - *Impact:* 0px = **1.0** (Perfect) | 1px = **0.37** (Harsh Penalty) | 2px = **0.13** (Fail).
2. **Identity (S_{Id}):** Linear mapping of Vision Score (1-5).
    - $$ S_{Id} = 1.0 - \frac{\text{vision_identity_drift_score} - 1}{4.0} $$
    - *Note: Since score ≥ 4 is gated, the effective input range is 1-3.*
3. **Style (S_{Sy}):** Composite of Vision Style and Edge Map.
    - $$ S_{Sy} = 0.6 \left(1.0 - \frac{\text{vision_style} - 1}{4.0} \right) + 0.4 (\text{edge_map_similarity}) $$
4. **Technical (S_{Te}):** Palette accuracy and Structure.
    - $$ S_{Te} = 0.6 (1.0 - \text{palette_delta}) + 0.4 (\text{ssim}) $$
    - *Note: SSIM is weighted low to allow for valid animation changes (squash/stretch).*

---

3. Thresholds & Gating Logic

Rank	Score Range	Status	Action Strategy
Diamond	**90 - 100**	**PASS**	**Auto-Commit.** Pixel-perfect stability (0px) and high fidelity.
Gold	**75 - 89**	**PASS**	**Acceptable.** Likely has 1px jitter or minor color noise.
Silver	**60 - 74**	**SOFT FAIL**	**Conditional Retry.** If this is the best of 3 attempts, flag for human review (Batch "B"). Otherwise retry.
Bronze	**0 - 59**	**HARD FAIL**	**Discard.** Trigger parameter tuning (see Retry Mapping).

---

4. Tie-Break Rules

When the Agent has a batch of candidates with scores within ±2 points, use this waterfall to pick the winner:

1. **The "Grounded" Rule (Lowest baseline_error_px):**
    - **Logic:** A sprite with 0px jitter is exponentially better than 1px jitter for game mechanics. Always prioritize the most stable feet.
2. **The "On-Model" Rule (Lowest vision_identity_drift_score):**
    - **Logic:** If stability is equal, pick the face that looks most like the Anchor.
3. **The "Clean Lines" Rule (Highest edge_map_similarity):**
    - **Logic:** Pick the one with the cleanest pixel-art outlines (easier to downscale/clean).
4. **Last Resort:** Lowest palette_delta.

---

5. Retry Guidance Mapping

*If a frame is **REJECTED**, map the failure reason to the specific "Knob" in your Python generation agent.*

Failure Reason	Primary Metric	Agent Knob Tweak (Action)
"Floating/Jitter"	baseline_error	**Action:** Increase **ControlNet (Pose)** weight (+0.1). Do not re-prompt; the issue is spatial, not semantic.
"Who is this?"	vision_identity	**Action:** Increase **IP-Adapter / Reference** weight. Reduce **CFG Scale** (high CFG burns facial features).
"Wrong Art Style"	vision_style	**Action:** Increase **Style LoRA** weight. Add "3d render, vector art" to Negative Prompts.
"Messy/Dirty"	edge_map_sim	**Action:** Adjust **Denoising Strength**. If blurry: *Decrease* strength. If noisy/hallucinated lines: *Increase* strength.

"Wrong Colors"	palette_delta	**Action:** Force **Palette Pre-pass**. Histogram-match the latent noise to the anchor sprite before sampling.

---

6. Tuning Over Time (Human-in-the-Loop)

Avoid manual "magic number" guessing. Use a lightweight Logistic Regression update loop.

1. **Data Collection:** Save a CSV log: [metrics_vector, final_score, human_label].
    - human_label: 1 (Pass), 0 (Reject).
2. **Weekly Weight Update:**
    - Run a regression analysis to find which metrics correlate most with human_label=1.
    - **Example:** If humans consistently **Reject** frames that the system gave **80** (due to high edge_map_similarity), but the frames had bad palettes, the regression will show palette_delta needs a higher weight.
3. **Threshold Drift:**
    - If **False Positive Rate > 15%** (System passes bad frames), raise the PASS threshold from 75 to 78.
    - If **False Negative Rate > 15%** (System rejects usable frames), lower the Hard Gate for ID from 4 to 3 (allow slightly more drift).