

# 16BitFit Battle Mode: Autonomous Sprite Generation + Audit System Specification

Version 1.0 | January 2026

## 1. AUDIT RUBRIC (FINALIZED)

### 1.1 Threshold Resolution (CONFLICT RESOLVED)

The sources provided conflicting SSIM thresholds: 0.75, 0.85, and 0.95. I'm resolving this with a **tiered system** that distinguishes between identity-critical metrics and frame-coherence metrics:

Metric	Hard Fail	Soft Fail (Warn)	Pass	Rationale
SSIM (vs anchor)	< 0.75	0.75–0.84	≥ 0.85	Identity drift detection; 0.85 balances strictness with pose variation
DINO similarity	< 0.80	0.80–0.89	≥ 0.90	Feature-level identity; more robust to pose changes
Frame-to-frame SSIM	< 0.70	0.70–0.84	≥ 0.85	Adjacent frame coherence; allows pose motion
LPIPS (frame-to-frame)	> 0.25	0.15–0.25	< 0.15	Perceptual flicker detection
Palette match %	< 80%	80–89%	≥ 90%	Top-10 colors within 10 RGB tolerance
Line weight drift	> 20%	15–20%	< 15%	Edge thickness consistency
Baseline drift (px)	> 2	2	≤ 1	Vertical alignment stability

**Decision:** Start with these thresholds, then calibrate via micro-test (Section 5.1).

### 1.2 Hard Fail Checks (Automatic REJECT)

These checks are **binary**—any failure triggers immediate rejection.

## HF-01: FORMAT\_DIMENSIONS\_MATCH

Method: Programmatic

Inputs: anchor\_image, candidate\_frame

Rules:

- candidate **MUST** be PNG with RGBA channels
- width == anchor.width (exact)
- height == anchor.height (exact)

Reason Code: HF01\_FORMAT\_MISMATCH | HF01\_WIDTH\_MISMATCH | HF01\_HEIGHT\_MISMATCH | HF01\_NO\_ALPHA

Retry Knob: Re-generate with locked canvas size; do NOT resize post-hoc

## HF-02: TRANSPARENCY\_BACKGROUND

Method: Programmatic + optional vision spot-check

Inputs: candidate\_frame

Rules:

- Background **MUST** be transparent (alpha=0)
- No connected non-transparent regions > 5% of canvas outside silhouette bounding box
- Semi-transparent halo band (1-2px around silhouette): max 5% of edge pixels

Reason Code: HF02\_OPAQUE\_BACKGROUND | HF02\_BACKGROUND\_LEAK | HF02\_EXCESSIVE\_HALO

Retry Knob: Regenerate with explicit "transparent background" prompt; if persistent, use keyed BG + matte extraction

## HF-03: BASELINE\_PIVOT\_STABILITY

Method: Programmatic (mask + baseline row detection)

Inputs: all frames of move

Rules:

- Detect lowest non-transparent row per frame
- Max variance across frames:  $\pm 1$ px (configurable)

Reason Code: HF03\_BASELINE\_DRIFT

Retry Knob: Enforce consistent pivot in generation; verify via Phaser test

## HF-04: NAMING\_METADATA\_CONTRACT

Method: Programmatic

Inputs: atlas JSON, PNG(s), manifest

Rules:

- JSON must parse without error
- All expected frames present (count matches manifest)
- No duplicate frame keys
- Frame names match pattern: {prefix}-{zeroPad\_index}-{suffix}
- Suffix convention locked: "" (no .png) when using --trim-sprite-names

Reason Code: HF04\_JSON\_INVALID | HF04\_MISSING\_FRAMES | HF04\_DUPLICATE\_KEYS | HF04\_NAME\_MISMATCH

Retry Knob: Re-export with standardized folder layout

## HF-05: GROSS\_ANATOMY\_BREAK

Method: Vision critique (VLM)

Inputs: anchor, candidate

Rules:

- No extra/missing limbs
- No wrong outfit/hair/face
- No major silhouette deviation

Reason Code: HF05\_EXTRA\_LIMBS | HF05\_MISSING\_LIMBS | HF05\_WRONG\_OUTFIT | HF05\_IDENTITY\_BREAK

Retry Knob: Increase reference strength, lower denoise, face/hand inpaint

## 1.3 Soft Fail Checks (Score + Retry; REJECT if below floor)

These checks produce numeric scores. Soft fail triggers retry; hard floor triggers rejection.

### SF-01: IDENTITY\_SIMILARITY\_SCORE

Method: DINO embeddings + SSIM + optional ArcFace

Metrics:

- ssim\_vs\_anchor: float [0,1]
- dino\_similarity: float [0,1]

Hard Floor:  $\text{ssim} < 0.75$  OR  $\text{dino} < 0.80 \rightarrow \text{REJECT}$

Soft Fail:  $\text{ssim} < 0.85$  OR  $\text{dino} < 0.90 \rightarrow \text{RETRY}$

Reason Code: SF01\_IDENTITY\_DRIFT

Retry Knobs:

1. Lower denoise (0.7  $\rightarrow$  0.5)
2. Raise reference/ID conditioning weight
3. Lock palette explicitly
4. Face inpaint pass

## SF-02: PALETTE\_DRIFT\_SCORE

Method: Histogram analysis + top-N color extraction

Metrics:

- palette\_match\_pct: float [0,1] (% of top-10 colors within 10 RGB of canonical)

Hard Floor: < 80% → REJECT

Soft Fail: < 90% → RETRY

Reason Code: SF02\_PALETTE\_DRIFT

Retry Knobs:

1. Tighten prompt with explicit hex values
2. Post-process: strict palette quantization
3. Reduce CFG if oversaturating

## SF-03: OUTLINE\_PIXEL\_DENSITY

Method: Edge detection (Canny/Sobel) + line weight stats

Metrics:

- line\_weight\_drift: float (% change from anchor median)

Hard Floor: > 20% → REJECT

Soft Fail: > 15% → RETRY

Reason Code: SF03\_LINE\_DRIFT | SF03\_BLUR\_DETECTED

Retry Knobs:

1. Reduce CFG
2. Add "crisp pixel edges, no blur" to prompt
3. Post-process: nearest-neighbor downscale + outline reinforcement

## SF-04: FRAME\_COHERENCE

Method: Temporal SSIM/LPIPS between adjacent frames

Metrics:

- frame\_ssim\_min: float (minimum SSIM between any adjacent pair)
- lpips\_max: float (maximum LPIPS between any adjacent pair)

Hard Floor: frame\_ssim < 0.70 OR lpips > 0.25 → REJECT

Soft Fail: frame\_ssim < 0.85 OR lpips > 0.15 → RETRY

Reason Code: SF04\_TEMPORAL\_FLICKER

Retry Knobs:

1. Anchor-to-anchor chaining (use prev frame as reference)
2. Reduce noise/denoise
3. Enforce consistent lighting prompt

SF-05: ALPHA\_EDGE\_ARTIFACTS

Method: Analyze 1-2px band outside silhouette

Metrics:

- halo\_pixel\_count: int
- fringe\_severity: float [0,1]

Hard Floor: fringe\_severity > 0.3 → REJECT

Soft Fail: fringe\_severity > 0.1 → RETRY

Reason Code: SF05\_HALO\_DETECTED | SF05\_FRINGE\_DETECTED

Retry Knobs:

1. Ensure TexturePacker --alpha-handling ReduceBorderArtifacts
2. Adjust background removal approach
3. Re-export with proper extrude/padding

1.4 Audit Result JSON Schema

Per-Frame Audit Result

json

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": [ "frame_index", "attempt", "result", "reason_codes", "metrics", "timestamp" ],
  "properties": {
    "frame_index": { "type": "integer", "minimum": 1 },
    "attempt": { "type": "integer", "minimum": 1 },
    "result": { "enum": [ "PASS", "REJECT", "SOFT_FAIL" ] },
    "reason_codes": {
      "type": "array",
      "items": { "type": "string", "pattern": "^(HF|SF)[0-9]{2}_[A-Z_]+$" }
    },
    "metrics": {
      "type": "object",
      "properties": {
        "ssim_vs_anchor": { "type": "number", "minimum": 0, "maximum": 1 },
        "dino_similarity": { "type": "number", "minimum": 0, "maximum": 1 },
        "frame_ssim_prev": { "type": [ "number", "null" ] },
        "lpips_prev": { "type": [ "number", "null" ] },
        "palette_match_pct": { "type": "number", "minimum": 0, "maximum": 1 },
        "line_weight_drift": { "type": "number", "minimum": 0 },
        "baseline_row": { "type": "integer" },
        "halo_pixel_count": { "type": "integer" },
        "fringe_severity": { "type": "number", "minimum": 0, "maximum": 1 }
      }
    },
    "timestamp": { "type": "string", "format": "date-time" },
    "retry_recommendation": {
      "type": [ "string", "null" ],
      "description": "Suggested knob adjustment for next attempt"
    }
  }
}

```

## Per-Move Audit Summary

json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": ["move_id", "total_frames", "passed", "rejected", "retry_rate", "status"],
  "properties": {
    "move_id": { "type": "string" },
    "total_frames": { "type": "integer" },
    "passed": { "type": "integer" },
    "rejected": { "type": "integer" },
    "soft_failed": { "type": "integer" },
    "retry_rate": { "type": "number", "description": "% of frames needing >1 attempt" },
    "status": { "enum": ["COMPLETE", "PARTIAL", "FAILED", "STOPPED"] },
    "stop_reason": { "type": ["string", "null"] },
    "frames": { "type": "array", "items": { "$ref": "#/definitions/frame_summary" } }
  },
  "definitions": {
    "frame_summary": {
      "type": "object",
      "properties": {
        "frame_index": { "type": "integer" },
        "attempts": { "type": "integer" },
        "final_result": { "enum": ["PASS", "REJECT", "MANUAL_REVIEW"] },
        "final_path": { "type": ["string", "null"] }
      }
    }
  }
}
```

## 2. PROMPT LOCKING SYSTEM

### 2.1 Placeholder Injection Rules

Placeholder	Source	Validation
{CHAR_ID}	Manifest <code>character_id</code>	Must match <code>/^[a-z_]+\$</code>
{MOVE}	Manifest <code>move_id</code>	Must match <code>/^[a-z_]+\$</code> (e.g., <code>walk</code> , <code>punch_heavy</code> )
{POSE_SPEC}	Frame-level pose description	Free text; max 200 chars

Placeholder	Source	Validation
{ANCHOR_REF}	Path to anchor sprite	Must exist; validated PNG
{FRAME_W}	Manifest {frame_w}	Integer; typically 64, 96, 128
{FRAME_H}	Manifest {frame_h}	Integer; typically 64, 96, 128
{STYLE_RULES}	Global style block	Injected from style_rules.txt
{PALETTE}	Character palette JSON	Hex values validated (see 2.6)

## 2.2 Master Prompt Template

You are generating a single animation frame for a 2D pixel-art fighting game (Phaser 3).

Character: {CHAR\_ID}

Move: {MOVE}

Pose spec (must match exactly): {POSE\_SPEC}

LOCKED REFERENCE (do not redesign): {ANCHOR\_REF}

- Preserve identity: face, hair, body shape, outfit silhouette, accessories.
- Preserve scale: same pixel scale as anchor.
- Canvas size MUST be exactly {FRAME\_W} x {FRAME\_H}.
- Baseline/pivot must remain consistent across all frames of this move.

Style rules (must follow): {STYLE\_RULES}

Palette lock (prefer these exact colors / closest shades): {PALETTE}

Background: fully transparent (alpha=0), no scenery, no props unless explicitly part of the move.

Render: crisp pixel-art edges, no blur, no painterly textures, no anti-aliasing halos.

## 2.3 Variation Prompt Template (Frame i of N)

Generate frame {i}/{N} for {CHAR\_ID} performing {MOVE}.

Pose spec: {POSE\_SPEC}

Anchor reference: {ANCHOR\_REF}

Hard locks:

- Same character identity and outfit as anchor.
- Same camera angle and facing direction as anchor.
- Exact canvas: {FRAME\_W} x {FRAME\_H}.



- Feet/ground contact must align to the same baseline as the anchor sequence.

Style: {STYLE\_RULES}

Palette constraint: {PALETTE}

Only change what is required by the pose progression for {MOVE}; keep everything else stable.

## 2.4 Lock/Recovery Prompt Template (Post-REJECT)

STRICT CONSISTENCY MODE.

Character: {CHAR\_ID}

Move: {MOVE}

Pose spec: {POSE\_SPEC}

Anchor reference: {ANCHOR\_REF}

Do NOT change: face shape, hair shape, outfit design, color distribution, body proportions.

Match anchor pixel density and outline thickness exactly.

Exact size: {FRAME\_W} x {FRAME\_H}.

Style rules: {STYLE\_RULES}

Palette lock: {PALETTE}

Goal: re-render the same pose with HIGHER identity fidelity and CLEANER edges.

This is a retry after identity drift was detected.

## 2.5 Negative Prompt Template

Avoid: background, scene, gradients, blur, soft shading, painterly texture, anti-aliasing halos, motion blur, lens effects, depth of field, extra limbs, extra fingers, missing limbs, wrong outfit, logos, text, watermarks, new accessories, random color shifts, glowing outlines, noisy dithering, jpeg artifacts, low resolution, cropped frame.

Must remain consistent with {ANCHOR\_REF}, {STYLE\_RULES}, and {PALETTE}.

## 2.6 Camera Angle/Facing Direction Rules

**Rule:** Camera angle and facing direction are **locked to anchor**.

Property	Lock Method
Facing direction	Extract from anchor (left/right); encode in manifest as <code>facing: "right"</code>
Camera angle	Fixed per character (typically 3/4 front or side profile)
Mirroring	<b>Do NOT mirror at generation</b> ; mirror in-engine via <code>sprite.setFlipX(true)</code>

**Mirroring Policy:**

- Generate all frames facing RIGHT (canonical direction)
- Engine handles left-facing via flipX
- This ensures consistency and halves generation work

### 2.7 Palette Validation & Typo Fix

**KNOWN ISSUE:** Sean's tank color in spec is `#F2FOEF` (contains letter "O" instead of zero).

**Resolution Protocol:**

1. Sample anchor sprite pixels in tank region
2. Find most frequent hex value
3. Update manifest with corrected value
4. Log correction: `PALETTE_CORRECTION: sean.tank #F2FOEF → #F2F0EF (assumed)`

**Validation Regex:** `/^[0-9A-Fa-f]{6}$/`

## 3. BOUNDED RETRY STRATEGY + STOP CONDITIONS

### 3.1 Retry Ladder (Ordered)

Execute in order. Only advance to next step if current step fails.

STEP 1: SEED REROLL (max 3 attempts)	
Action: Regenerate with different seed, same prompt	
Target: Random artifacts, minor deformities	
Stop: After 3 rerolls OR if all produce same failure mode	

↓ FAIL

STEP 2: NEGATIVE PROMPT TIGHTEN (max 2 attempts)	
Action: Add specific negative terms for observed issues	
Target: Extra limbs, BG leak, blur, painterly texture	
Additions: Based on reason_code mapping	

↓ FAIL

STEP 3: IDENTITY RESCUE (max 2 attempts)	
Action: Lower denoise (0.7→0.5), raise reference weight	
Target: Face/hair/outfit drift, silhouette changes	
Use: LOCK/RECOVERY template instead of MASTER	

↓ FAIL

STEP 4: POSE RESCUE (max 2 attempts)	
Action: Strengthen pose conditioning (ControlNet weight +20%)	
Target: Pose mismatch, limb placement errors	
Requires: Pose map / keypoints available	

↓ FAIL

STEP 5: TARGETED INPAINTING (max 2 attempts)	
Action: Two-stage: generate base → inpaint face/hands/outfit	
Target: Localized drift in critical regions	
Mask: Auto-generate from semantic segmentation	

↓ FAIL

STEP 6: POST-PROCESS CORRECTIONS (max 1 attempt)	
Actions (pixel-safe only):	
- Palette quantization (strict to canonical colors)	
- Nearest-neighbor resize (if needed)	
- Outline reinforcement	
- Halo cleanup (alpha threshold)	
Target: Palette drift, pixel density drift, edge artifacts	



3.2 Escalation Triggers

Condition	Escalation Target
3 consecutive HF05 (anatomy break)	→ Step 5 (Inpainting)
SF02 (palette) fails after Step 6	→ Accept with warning OR manual
>50% rejection rate across character	→ Consider LoRA training
Same reason_code on all attempts	→ Skip to Step 7 (Manual)

3.3 Max Attempts Per Frame

Step	Max Attempts	Cumulative Max
Seed Reroll	3	3
Negative Tighten	2	5
Identity Rescue	2	7
Pose Rescue	2	9
Inpainting	2	11
Post-Process	1	12

**Hard Cap:** 12 attempts per frame, then → Manual Review

### 3.4 Global Stop Conditions

```
STOP_CONDITIONS = {  
  "single_frame_max_retries": 12,  
  "move_retry_rate_threshold": 0.20,    # >20% frames need retry → STOP  
  "move_reject_rate_threshold": 0.50,   # >50% final rejects → STOP  
  "character_reject_rate_threshold": 0.30, # >30% across all moves → STOP  
  "consecutive_failures": 5             # 5 frames in a row fail → STOP  
}
```

### 3.5 Stop Report Schema

When a stop condition triggers, emit:

```
json
```

```

{
  "report_type": "PIPELINE_STOP",
  "timestamp": "2026-01-11T14:30:00Z",
  "run_id": "RUN_2026_01_11_001",
  "character_id": "sean",
  "stop_condition": "MOVE_RETRY_RATE_EXCEEDED",
  "stop_value": 0.25,
  "stop_threshold": 0.20,
  "context": {
    "move_id": "walk",
    "frames_attempted": 12,
    "frames_passed": 6,
    "frames_rejected": 3,
    "frames_pending_manual": 3
  },
  "failure_analysis": {
    "most_common_reason_codes": ["SF01_IDENTITY_DRIFT", "SF03_LINE_DRIFT"],
    "recommended_actions": [
      "Consider LoRA fine-tuning for character",
      "Review anchor sprite quality",
      "Adjust SSIM threshold if too strict"
    ]
  },
  "resume_instructions": {
    "resume_from_frame": 7,
    "suggested_parameter_changes": {
      "denoise": 0.5,
      "reference_weight": 1.2
    }
  }
}

```

## 4. IMPLEMENTATION INTERFACES

### 4.1 Folder Structure

```

/assets_src/
  /characters/{CHAR_ID}/
    /anchors/
      anchor.png          # Canonical reference sprite

```

```

    anchor_metadata.json      # Dimensions, palette sample points
/spec/
    palette.json             # Validated hex colors
    style_rules.txt          # Character-specific style notes
/moves/{MOVE}/
    /pose_refs/              # Pose images / keypoints
        frame_0001.png
        frame_0001_keypoints.json
    /candidates/             # Generation attempts
        frame_0001_try1.png
        frame_0001_try1_seed.txt
        frame_0001_try2.png
    /approved/               # Passed audit
        frame_0001.png
        frame_0002.png
    /audit/                  # Audit results
        frame_0001.json
        frame_0002.json
        move_summary.json

/exports/
    /phaser/{CHAR_ID}/
        atlas.json           # TexturePacker output
        atlas.png            # (or atlas0.png, atlas1.png if multipack)
    /manifests/
        run_{RUN_ID}.json    # Full pipeline manifest

/logs/
    run_{RUN_ID}.log
    manual_review_queue.json
    stop_reports/
        stop_{RUN_ID}_{TIMESTAMP}.json

```

## 4.2 Manifest Schema (Run-Level)

```

json

```

```

{
  "run_id": "RUN_2026_01_11_001",
  "created_at": "2026-01-11T10:00:00Z",
  "character_id": "sean",
  "anchor_ref": "/assets_src/characters/sean/anchors/anchor.png",
  "frame_w": 96,
  "frame_h": 96,
  "style_rules": "Full-color 2D fighting-game aesthetic, Street Fighter/Capcom vibe, crisp pixel edges",
  "palette": {
    "skin": "#F5D6C6",
    "hair": "#C2A769",
    "eyes": "#4682B4",
    "shorts": "#2323FF",
    "tank": "#F2F0EF",
    "shoes": "#272929"
  },
  "palette_corrections": [
    { "field": "tank", "original": "#F2FOEF", "corrected": "#F2F0EF", "method": "assumed_typo" }
  ],
  "facing": "right",
  "pivot": { "x": 0.5, "y": 1.0 },
  "moves": [ /* see move-level schema */ ],
  "export": {
    "packer": "TexturePacker",
    "format_flag": "--format phaser",
    "multiatlas": false,
    "phaser_loader": "this.load.atlas"
  },
  "stop_conditions": {
    "single_frame_max_retries": 12,
    "move_retry_rate_threshold": 0.20,
    "move_reject_rate_threshold": 0.50
  }
}

```

### 4.3 Manifest Schema (Move-Level)

json



```
{
  "move_id": "walk",
  "expected_frames": 8,
  "naming_policy": {
    "prefix": "walk/",
    "zeroPad": 4,
    "suffix": ""
  },
  "status": "COMPLETE",
  "frames": [ /* see frame-level schema */ ]
}
```

## 4.4 Manifest Schema (Frame-Level)

json

```
{
  "frame_index": 1,
  "pose_spec": "Left foot forward, right arm back, mid-stride position",
  "pose_ref": "/assets_src/characters/sean/moves/walk/pose_refs/frame_0001.png",
  "attempts": [
    {
      "attempt": 1,
      "seed": 123456,
      "prompt_hash": "sha256:abc123...",
      "prompt_template": "MASTER",
      "candidate_path": "candidates/frame_0001_try1.png",
      "audit": {
        "result": "REJECT",
        "reason_codes": ["SF01_IDENTITY_DRIFT"],
        "metrics": {
          "ssim_vs_anchor": 0.78,
          "dino_similarity": 0.82,
          "palette_match_pct": 0.92,
          "line_weight_drift": 0.12,
          "baseline_row": 94
        }
      },
      "retry_recommendation": "Use LOCK template, reduce denoise to 0.5"
    }
  ],
  {
    "attempt": 2,
    "seed": 789012,
    "prompt_hash": "sha256:def456...",
    "prompt_template": "LOCK",
    "candidate_path": "candidates/frame_0001_try2.png",
    "audit": {
      "result": "PASS",
      "reason_codes": [],
      "metrics": {
        "ssim_vs_anchor": 0.88,
        "dino_similarity": 0.91,
        "palette_match_pct": 0.95,
        "line_weight_drift": 0.08,
        "baseline_row": 94
      }
    }
  }
],
```

```
"final": {  
  "path": "approved/frame_0001.png",  
  "audit_result": "PASS",  
  "attempt_used": 2  
}  
}
```

## 4.5 Phaser Export Contract

### TexturePacker Command (Single Atlas)

```
bash  
  
TexturePacker \  
  --format phaser \  
  --trim-mode Trim \  
  --extrude 1 \  
  --shape-padding 2 \  
  --border-padding 2 \  
  --disable-rotation \  
  --alpha-handling ReduceBorderArtifacts \  
  --max-size 2048 \  
  --trim-sprite-names \  
  --prepend-folder-name \  
  --data /exports/phaser/{CHAR_ID}/atlas.json \  
  --sheet /exports/phaser/{CHAR_ID}/atlas.png \  
  /assets_src/characters/{CHAR_ID}/moves/*/approved/
```

### TexturePacker Command (Multipack)

```
bash
```

```
TexturePacker \  
  --format phaser \  
  --multipack \  
  --trim-mode Trim \  
  --extrude 1 \  
  --shape-padding 2 \  
  --border-padding 2 \  
  --disable-rotation \  
  --alpha-handling ReduceBorderArtifacts \  
  --max-size 2048 \  
  --trim-sprite-names \  
  --prepend-folder-name \  
  --data /exports/phaser/{CHAR_ID}/atlas.json \  
  --sheet /exports/phaser/{CHAR_ID}/atlas{n}.png \  
  /assets_src/characters/{CHAR_ID}/moves/*/approved/
```

## Phaser Loader Code

### Single Atlas:

```
javascript  
  
// In preload()  
this.load.atlas('sean', 'assets/sean/atlas.png', 'assets/sean/atlas.json');  
  
// After load - set nearest filtering  
this.textures.get('sean').setFilterMode(Phaser.Textures.FilterMode.NEAREST);
```

### Multiatlas:

```
javascript  
  
// In preload()  
this.load.mmultiatlas('sean', 'assets/sean/atlas.json', 'assets/sean/');  
  
// After load - set nearest filtering  
this.textures.get('sean').setFilterMode(Phaser.Textures.FilterMode.NEAREST);
```

## Animation Frame Names

```
javascript
```

```
// Generate frame names for walk animation
this.anims.generateFrameNames('sean', {
  start: 1,
  end: 8,
  zeroPad: 4,
  prefix: 'walk/',
  suffix: " // NO .png when using --trim-sprite-names
});
```

Pivot/Origin Enforcement

```
javascript

// Option A: Per-sprite (recommended for simplicity)
sprite.setOrigin(0.5, 1); // Bottom-center

// Option B: Per-frame (if pivot varies)
const texture = this.textures.get('sean');
texture.getFrameNames().forEach(name => {
  const frame = texture.get(name);
  frame.customPivot = true;
  frame.pivotX = 0.5;
  frame.pivotY = 1.0;
});
```

Naming Policy Enforcement

Component	Value	Notes
Prefix	{move}/	e.g., walk/, punch_heavy/
Zero Pad	4	Frame 1 → 0001
Suffix	""	Empty string (no .png)
Full Example	walk/0001	For frame 1 of walk animation

## 5. OPEN QUESTIONS & MICRO-TESTS

### 5.1 Required Experiments (Priority Order)

#### TEST-01: SSIM/DINO Threshold Calibration (CRITICAL)

Purpose: Finalize thresholds from Section 1.1

Method:

1. Collect 10 known-good frames (manually verified as acceptable)
2. Collect 10 known-bad frames (obvious identity/quality issues)
3. Run SSIM and DINO on all 20 vs anchor
4. Plot distributions; find threshold that best separates sets

Data to Collect:

- ssim\_vs\_anchor for each frame
- dino\_similarity for each frame
- manual\_label: "good" | "bad"

Pass Criteria:

- Chosen threshold achieves >90% separation (F1 score)
- No more than 1 false positive (bad frame passes)

Decision After Test:

- Update thresholds in Section 1.1
- If no clean separation: relax thresholds and rely more on VLM critique

#### TEST-02: Phaser Pivot Auto-Apply

Purpose: Determine if Phaser reads pivot from TexturePacker JSON automatically

Method:

1. Export 2-frame atlas with different pivots in JSON (0.5,0.5 vs 0.5,1.0)
2. Load in Phaser without manual pivot code
3. Inspect frame.pivotX, frame.pivotY values
4. Render both frames at same position; observe alignment

Data to Collect:

- frame.pivotX, frame.pivotY after load
- Screenshot of rendered frames

Pass Criteria:

- PASS if pivot values match JSON and render shows correct alignment
- FAIL if pivot is 0,0 or frames misalign

#### Decision After Test:

- If PASS: rely on JSON pivot; no manual code needed
- If FAIL: always run pivot-setting loop with customPivot=true

### TEST-03: Trim Mode vs Frame Size Constraint

Purpose: Verify trim doesn't cause baseline jitter despite "exact frame size" requirement

#### Method:

1. Pack same 8-frame walk cycle with --trim-mode Trim
2. Pack same frames with --trim-mode None
3. Animate both in Phaser with `sprite.setOrigin(0.5, 1)`
4. Record baseline position each frame

#### Data to Collect:

- sourceSize, spriteSourceSize per frame (from JSON)
- Rendered Y position of feet across frames
- Visual recording/GIF of animation

#### Pass Criteria:

- PASS if Trim version has no visible jitter (baseline stable within 1px)
- FAIL if jitter visible or baseline varies >1px

#### Decision After Test:

- If PASS: keep --trim-mode Trim (more efficient atlas)
- If FAIL: switch to --trim-mode None OR enforce pivot correction code

### TEST-04: Frame Key Suffix Convention

Purpose: Confirm .png suffix behavior with --trim-sprite-names

#### Method:

1. Export atlas with --trim-sprite-names
2. Log all frame keys from JSON
3. Run `generateFrameNames` with suffix: "" and suffix: '.png'
4. Check which version finds all frames

#### Data to Collect:

- Frame keys in atlas JSON (list)
- Result of `generateFrameNames` with each suffix

#### Pass Criteria:

- PASS if suffix: "" finds all frames (keys have no .png)

- FAIL if suffix: '.png' required

Decision After Test:

- Lock suffix convention in manifest naming\_policy
- Add CI check to validate atlas keys match expected pattern

## TEST-05: Sean Tank Hex Correction

Purpose: Fix palette spec typo #F2FOEF

Method:

1. Load Sean anchor sprite
2. Sample pixels in tank region (define bounding box manually)
3. Extract most common hex values
4. Compare to suspected correction #F2F0EF

Data to Collect:

- Top 5 hex values in tank region by frequency
- Visual confirmation (color swatch)

Pass Criteria:

- PASS if consistent hex found and matches #F2F0EF or similar valid hex

Decision After Test:

- Update palette.json with corrected value
- Log as palette\_correction in manifest

## TEST-06: roundPixels Shimmer Test

Purpose: Determine if game.config.roundPixels needed

Method:

1. Create test scene with sprite at non-integer position (e.g., x: 100.5)
2. Run with roundPixels: false, observe shimmer/subpixel artifacts
3. Run with roundPixels: true, observe quality

Data to Collect:

- Screenshot/video at both settings
- Subjective assessment: shimmer visible? edge quality?

Pass Criteria:

- PASS for roundPixels: true if shimmer eliminated without artifacts
- FAIL if roundPixels: true causes other issues



Decision After Test:

- Add to game config if beneficial

TEST-07: Transparent BG vs Keyed BG Generation

Purpose: Determine best approach for clean alpha

Method:

1. Generate 5 frames with "transparent background" prompt
2. Generate 5 frames with solid green (#00FF00) background
3. Process green BG frames with chroma key / matte extraction
4. Run halo audit on both sets

Data to Collect:

- halo\_pixel\_count for each frame
- fringe\_severity for each frame
- Processing time for matte extraction

Pass Criteria:

- PASS transparent if halo rate < 10%
- Consider keyed BG if transparent halo rate > 30%

Decision After Test:

- Standardize on cleaner method
- If keyed BG needed, add matte extraction to pipeline

5.2 Test Priority Matrix

Test	Priority	Blocking	Est. Time
TEST-01 (Thresholds)	P0	Yes - affects all audits	2 hours
TEST-02 (Pivot)	P0	Yes - affects alignment	30 min
TEST-03 (Trim)	P0	Yes - affects export	30 min
TEST-04 (Suffix)	P1	Yes - affects loader	15 min
TEST-05 (Palette)	P1	Yes - affects Sean	15 min
TEST-06 (roundPixels)	P2	No	15 min
TEST-07 (BG method)	P2	No	1 hour

## 6. QUICK REFERENCE CHECKLISTS

### 6.1 Pre-Generation Checklist

- ☐ Anchor sprite exists and is valid PNG RGBA
- ☐ Palette validated (all hex codes match `/^[0-9A-Fa-f]{6}$/`)
- ☐ Frame dimensions set in manifest (frame\_w, frame\_h)
- ☐ Move pose specs defined for all frames
- ☐ Naming policy set (prefix, zeroPad, suffix)
- ☐ Style rules file exists

### 6.2 Per-Frame Generation Checklist

- ☐ Inject all placeholders into prompt template
- ☐ Record seed value
- ☐ Hash prompt and store in manifest
- ☐ Generate candidate
- ☐ Run audit checks (HF first, then SF)
- ☐ Log audit result to JSON
- ☐ If PASS: move to approved/, update manifest
- ☐ If REJECT: advance retry ladder, record reason

### 6.3 Export Checklist

- ☐ All expected frames in approved/ folders
- ☐ Run TexturePacker with exact quoted flags
- ☐ Verify atlas JSON is valid
- ☐ Verify all frame keys match naming policy
- ☐ Test load in Phaser (no missing frames)
- ☐ Apply NEAREST filtering
- ☐ Test animation playback (no jitter)

### 6.4 Stop Condition Monitoring

- ☐ Track retry count per frame
- ☐ Calculate move retry rate after each frame

- ☐ Check consecutive failure count
- ☐ Emit stop report if threshold breached
- ☐ Do not silently fail - always produce report

## 7. APPENDIX: REASON CODE REFERENCE

Code	Category	Description	Typical Retry Step
HF01_FORMAT_MISMATCH	Hard	Not PNG or missing RGBA	Re-generate
HF01_WIDTH_MISMATCH	Hard	Width $\neq$ anchor	Re-generate
HF01_HEIGHT_MISMATCH	Hard	Height $\neq$ anchor	Re-generate
HF01_NO_ALPHA	Hard	No alpha channel	Re-generate
HF02_OPAQUE_BACKGROUND	Hard	Background not transparent	Re-generate
HF02_BACKGROUND_LEAK	Hard	Scene/props in background	Negative prompt
HF02_EXCESSIVE_HALO	Hard	>5% edge pixels semi-transparent	BG method change
HF03_BASELINE_DRIFT	Hard	Baseline varies >2px	Pose rescue
HF04_JSON_INVALID	Hard	Atlas JSON parse error	Re-export
HF04_MISSING_FRAMES	Hard	Expected frames not in atlas	Re-export
HF04_DUPLICATE_KEYS	Hard	Duplicate frame names	Fix folder structure
HF04_NAME_MISMATCH	Hard	Keys don't match policy	Re-export
HF05_EXTRA_LIMBS	Hard	Extra arms/legs/fingers	Identity rescue
HF05_MISSING_LIMBS	Hard	Missing body parts	Identity rescue
HF05_WRONG_OUTFIT	Hard	Outfit doesn't match anchor	Identity rescue
HF05_IDENTITY_BREAK	Hard	Different character	Identity rescue
SF01_IDENTITY_DRIFT	Soft	Low SSIM/DINO vs anchor	Identity rescue
SF02_PALETTE_DRIFT	Soft	Colors don't match spec	Palette quantize
SF03_LINE_DRIFT	Soft	Outline weight changed	Post-process

Code	Category	Description	Typical Retry Step
SF03_BLUR_DETECTED	Soft	Edges not crisp	Negative prompt
SF04_TEMPORAL_FLICKER	Soft	Frame-to-frame inconsistent	Frame chaining
SF05_HALO_DETECTED	Soft	Alpha halo around sprite	Alpha handling
SF05_FRINGE_DETECTED	Soft	Colored edge artifacts	Alpha handling

*End of Specification*