

```
    }
  }
}
```

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.multiatlas('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