

# LayerWeaver：はじめての作画ガイド (v12.9 実戦思想反映版)

～ イラストを動かすための、効率的なデータ作成ガイド ～

**[!IMPORTANT] 本ガイドを読む前に：サポート対象アセットの確認** 本ツール（LayerWeaver）は、Weaverシリーズ独自の仕様に基づいた**比較的単純なレイヤー構造を持つPSD**を作成するための補助ツールです。

**G3テンプレート規格**は、Cartoon Animator 5 (CTA5) の開発元である **Reallusion社** によって設計された極めて洗練された独自規格です。LayerWeaverは、この既存規格をゼロから自動生成したり、その代用となるものではありません。

本ツールの役割は、あなたの描いた絵をこれら「既存の素晴らしい規格」へと正しく適合させるための「**翻訳機（ブリッジ）**」です。出力されたPSDを他社ツールで利用する場合は、各ツールの仕様に合わせた最終的な手動調整が必要になることをあらかじめご了承ください。

LayerWeaverは、あなたが描いたイラストを効率的に「動くパーツ」へと切り分け、リギングの下準備を自動化するための補助ツールです。このシステムには「**テンプレートにあるサンプルの画像を消して、自分の絵を主役（Art）として置く**」という、シンプルかつ絶対的なルールがあります。

## 0. GUIメニュー・リファレンス（操作画面の対応表）

操作中、ボタンの名称や色が変化することがあります。画面のシグナルをよく見て進めましょう。

### メイン操作ボタン（通常時）

- **[1] 1. 分析 & プレビュー生成:** 解析の第一歩（レイヤー生成）です。
- **[2] 2. 最終PSDを構築 & 表示:** 切り分けられた破片を、新しいPSDとして結合・再構築します。
- **[3] 3. 表示中のPSDを保存...:** 構築されたリギング用アセットを書き出します。
- **[4] 4. MotionWeaver 起動 (3d/\_rig.csv 生成):** 演技データとの紐付け工程へ移行します。

## 1. 準備工程：イラストの持ち込み（入替）と「階層ルール」

LayerWeaverを正しく動かすには、Krita上での「**自分の絵の置き方**」が何よりも重要です。

### 1.1 基本ルール：サンプルのArtを消して、自分の絵を置く

テンプレートの **Human** グループなどの中にあるサンプル画像を、自分の絵と入れ替えてください。

**[!IMPORTANT] なぜ「入れ替え」が必要なのか？（システムの仕様）** システム（プログラム）は解析開始時に、ルート（一番上の階層）をスキャンし、「**表示・非表示に関係なく、最も面積が大きいノード**」を1つだけ強制的に「主役（Art）」として選び出します。

ルート階層での非表示設定は無視されます。不要なサンプルキャラは非表示にするのではなく、物理的に削除してから自分の絵を置くのが、最も確実でクリーンな方法となります。

## 1.2 マーカー（裁断カッター）の調整

**Markers** 内の「黒い丸」は、ボーンの回転軸（ピボット）であると同時に、\*\*パーツを切り分ける「裁断機（カッター）」\*\*です。マーカーのサイズ（正円の直径）は、イラストのパーツよりも「左右に少しはみ出すくらい」大きく調整してください。これが綺麗な裁断の秘訣です。

### 1.2.1 首（Neck）の長さと作画に関する重要ルール（首が曲がらない現象の防止）

キャラクターの首を正常に回転（曲げる）させるためには、イラスト上の「**首の長さ（のりしろ）**」が極めて重要になります。首が短すぎると、PSDを書き出してCTA5等のアニメーションツールにインポートした際、首ボーン（Neck）の位置に対して首の画像ピクセルが十分に重ならず、自動ウェイトが適用されなくなります（ボーンを動かしても首の画像が曲がらず、頭が平行移動するだけで切り離されてしまう等の原因になります）。

- **確実に曲げたい場合のコツ**: 首のパーツを「**頭の後ろまでしっかりと描き足す（頭部と深く重なる長さにする）**」くらい長く作画してください。首のパーツが十分に長ければ、ボーンの可動ウェイトが正常に適用され、滑らかに首が曲がるようになります。
- **あえて曲げずに移動（スライド）させる選択肢**: デザインやキャラクターの性質によっては、「首自体はグニャグニャと曲がらず、頭の動きに合わせて平行移動するだけでいい」という場合もあります。CTA5は、首の描き足し長さ（重なり具合）によって、ウェイトを乗せて曲げるか、あるいは単にスライド移動させるかを自動で判別している可能性があります。作画意図に合わせて首の長さを調整してください。

[!TIP] **推奨ブラシ：ピクセルアート系（ボケなし）** マーカーを描いたり調整したりする際は、Kritaのブラシプリセットから「**Digital**」→「**Pixel Art**」カテゴリを選び、「**Pixel Art Fill**」などのブラシを使ってみてください。これらのブラシは最初から「ボケなし・不透明度100%」です。これにより、システム内部の `alpha_threshold`（不透明度閾値）の判定を最大限に活かすことができ、わずか **1ピクセルの隙間** であっても確実にパーツを分割・抽出できるようになります。

## 1.3 【応用テクニック】棒人間によるプレビズ用アセットの作り方

LayerWeaverは、正式なイラストを描く前の「モーションのプレビュー（アニメティック）」用の骨組み素材を作るのにも最適です。しかし、Krita上で透明なレイヤーに「マーカーの点だけ」を置いて、システムはそれを正しく認識できません。

[!IMPORTANT] **仕様上の正解：マーカー間をブラシで繋いで「島（Island）」を作る** LayerWeaverは、マーカーの点だけでなく「マーカーとマーカーの間にあるピクセルの塊（島）」を検知して骨の長さを計算します。したがって、プレビズ用の棒人間を作る際は、**マーカーを置いた後、そのマーカー同士をブラシでサッと繋いで（あるいは丸などで囲んで）不透明なピクセルの「島（Island）」を作っておくことが技術的な必須条件（正解）**となります。

この手法を使って「ジャンプ」や「攻撃」などのキーポーズを棒人間として描き、PSDで保存してMotionWeaverに通せば、あなただけのオリジナルな初期ポーズ（`_3d.csv`）が完成します。これらをMohoで繋ぐことで、完全オリジナルの手付けアニメーション（Pose-to-Pose）の土台を爆速で組み上げることができます。

## 1.4 【応用テクニック】半透明なパーツ（スカーフ等）をキャラクターに登録する方法

スカーフやベールなどの半透明な装飾品をキャラクターの一部として綺麗に取り込みたい場合は、「**マーカーで塗りつぶして関節（Hole）として検出し、ベースパーツに統合する**」というテクニックが有効です。

### [!TIP] 手順と仕組み

1. **マーカーで塗りつぶす**: **Markers** レイヤーで、登録したい半透明パーツの領域を塗りつぶします。
2. **分析を実行する**: [1. 分析 & プレビュー生成] を実行します。マーカーで塗られた半透明領域は、アルファ閾値の影響を受けずに関節 (**j** 番号) として確実に検出されます。
3. **テーブルで統合する**: UI画面の対応表 (テーブル) で、検出された **j** 番号 (例: **j3**) を、胴体 (**Torso** などのベースパーツ) の「**Joints**」列にカンマ区切りで追記します (例: **j1, j3**)。
4. **ビルドする**: [2. 最終PSDを構築 & 表示] を実行すると、半透明パーツのピクセルがベースパーツに綺麗に合成・統合された状態でPSDが出力されます。

## 2. 工程 1：1. 分析 & プレビュー生成 (ボタン [1])

準備が整ったら、ボタン「**1. 分析 & プレビュー生成**」をクリックします。

### 2.1 🎨 職人の仕上げ (パーツレイヤーと関節マージンを適切に整える)

クリック後、**islands** (パーツ) と **hole** (関節) レイヤーが自動生成されると、ボタンが **黄色く点灯** し、あなたの修正を待つポーズ状態になります。

**注意**: この時点ではまだ「色」も「番号」も付いていません。自動生成されたこれら2つの重要レイヤーを、Kritaのブラシや消しゴムを使い、両方のバランスを最適化するように「**適切に整える**」作業を行ってください。

#### [!TIP] 修正のテクニック:

- **爆速の小技**: 簡単に済ませたいなら、パーツレイヤー (islands) を細い消しゴムでピッと切り離すだけで構いません。
- **王道の修正**: 調整が難しい場合は、一度生成されたレイヤーを削除し、マーカー (Markers) を調整してから再度ボタン [1] を押してください。

## 3. 次のステップ：▶ もう一度「分析」を実行 (黄色いボタン)

パーツレイヤーと関節マージンの「整え」が終わったら、黄色いボタン「**▶ もう一度「分析」を実行**」をクリックして解析を確定させます。

### 3.1 🔍 解析結果の確認 (プレビューの番号とテーブル修正)

このクリックにより、Krita内のプログラムがあなたの修正したレイヤーをスキャンし、「**解析結果**」を表示します。

[!CAUTION] 「**自動判定**」は万能ではありません (**Tポーズ推奨**) プログラムがパーツを自動的に関節に紐づけられるのは、原則として「**腕を横に広げたTポーズ**」の場合のみです。それ以外のポーズ (腕を下げている等) では、正しく判定できないのが普通です。

**テーブルは「修正するもの」と認識してください** 画面に表示される対応表 (テーブル) は、あくまでプログラムの「推測」です。必ず自分の目で確認し、間違っている箇所 (不自然な島の番号など) は、**ユーザーが手動で書き換えて統合を完了させる** 必要があります。

[!TIP] テーブルの行の並び順 = PSDの重なり順 (Z順) 画面上のテーブル (および保存される `rigging_map.csv`) の行の並び順は、出力されるPSDのレイヤーの重なり順に直結します。

- **決定要素:** 最終的な重なり順は、Krita上のマーカーのレイヤー順や `islands.csv` の順序ではなく、`rigging_map.csv` (および初期表示されるテーブル) の行の並び順で決定されます。
- **上の行にあるパーツほど:** PSD内では「手前 (前面)」に配置されます。
- **下の行にあるパーツほど:** PSD内では「奥 (背面)」に配置されます。

例えばキャラクターが左を向いている場合、手前に来る「左足」を「右足」よりも上の行に配置することで、正しい前後関係を保つことができます。通常はTポーズを前提としたデフォルトの並び順 (`rigging_map.csv`) のままで問題ありません。

## 4. 工程 2：2. 最終PSDを構築 & 表示 (ボタン [2])

解析結果 (テーブル) を修正し、正しく紐付けられたら、ボタン「**2. 最終PSDを構築 & 表示**」をクリックします。

### 4.1 構築後の最終検品：パーツを「回転」させてチェックする

構築が完了し、新しいPSDドキュメントが開いたら、そのまま保存せずに必ず以下の「実動テスト」を行ってください。

1. **軸を合わせる:** 修正したいパーツレイヤーを選択し、**Ctrl + T** (変形)。
2. **中心を合わせる:** 変形の中心点 (ピボット) を、そのパーツの裏側に見える「**関節マージン (赤い番号)**」に刻まれた点 (ピボット位置) に重ねます。
3. **回して確認:** パーツをドラッグして回転させてみてください。
4. **のりしろの修正:** 回転させた時に隙間などがあれば、その場で描き足したり削ったりして、パーツの重なり (のりしろ) を完璧に整えます。

## 5. 工程 3 & 4：保存 と 起動

- ボタン [3]「**3. 表示中のPSDを保存...**」: 保存します。
- ボタン [4]「**4. MotionWeaver 起動**」: アニメーション制作へ移行します。

## 6. 【応用】タグ制御と物性設計によるビジュアルプログラミング

Weaverシリーズの主な設計思想は、レイヤー名に特定のタグを付けるだけで、外部の設定ファイルを記述することなくリギングの準備を完了できる点にあります。これは、イラストレーターがキャンバス上で行う効率的な設定作業です。

### 6.1 命名ルール (Tagging Logic)：リギングエンジンへの自動指示

レイヤー名に含まれる特定の記号は、後続の構築エンジンに対する実行命令として機能します。

- 「@」タグ：空間吸着とバリエーション展開
  - **基本:** 子パーツ名 @ 吸着先アンカー名 (例: `Sword @ RHand`)
  - **効果:** 親ボーンの回転軸 (ピボット) に子パーツを磁石のように吸着させます。

- **応用（スイッチグループ）**：グループレイヤーの名前に **@Head** 等を付与すると、その中の「瞬き」や「口の形」といった全バリエーションが一括で吸着・固定（Layer Binding）されます。
- **「>」タグ：動的な骨格（ボーンチェーン）の構築**
  - **基本**: **親ボーン名 > 子グループ名 > 子ボーン名**
  - **効果**: Krita上の命名だけで、出力後のPSD内に複雑なボーンの親子関係を構築します。
  - **(0) パラメータ**: **Scarf\_Tip(0)** のように記述すると、関節の回転軸としての機能のみを残し、メッシュへの影響範囲をゼロに設定できます。これにより、関節を曲げた際の不自然な歪みを抑止できます。

## 6.2 動き方の設定（Material Design）：しなやかな動きと硬い動き

一枚の絵に「どのように動くか」という物性を与えます。

- **チューブ表現 (Organic / Flexi-Bind)**:
  - **特徴**: 一つの描画パーツが、複数のボーンの動きに合わせて「しなやか」に曲がります。
  - **推奨**: 腕、脚、尻尾、長い髪、マント。
  - **設定**: 「>」タグで繋がれたボーンチェーンに沿って、ゴムのように滑らかな曲線を描きます。
- **ロボット表現 (Rigid / Region-Bind)**:
  - **特徴**: パーツ自体の変形を許さず、ボーンの回転に100%同期して「カチッ」と動きます。
  - **推奨**: 剣、盾、兜、メカの装甲。
  - **設定**: 「@」タグで単一のボーンに吸着させたパーツに自動適用され、正確な関節可動を再現します。

## 6.3 特殊自動化：リップシンクと表情制御の境界線

- **リップシンク自動マッピング**: **@Mouth** タグが付与されたグループ内に口の形を描くと、検出順に基づき音素が自動割り当てされます。
  - **5音素（日本語標準）**：「あ・い・う・え・お」の順で5つ並べるだけで、基本会話に対応。
  - **15音素（フルスペック）**：高度なリップシンクにも、特定の順序で描くことで対応可能です。
  - **ツール別仕様**:
    - **CTA5**: 出力された PSD の構造がシンプルであるため、読み込み後に基本的なリップシンク機能が動作する場合がありますが、公式な G3 規格としての動作を保証するものではありません。
    - **Moho**: 同様の運用が可能です。Moho 側のスイッチレイヤー設定やプロジェクトに応じた微調整が必要です。
- **「Weaverがやらないこと」を知る**: Weaverは、目パチのスマートボーン化や複雑なメッシュ変形までには行いません。これらは専用ツールのエディタを直接使ったほうが、結果的に高品質・低工数で仕上がるためです。Weaverは、そのための「整理された下地（構造）」を提供することに特化しています。

---

## 7. トラブルシューティング：SOS プロトコル

Weaverシリーズは、不自由な完成品ではありません。

- **エンジニアを頼る**: 設定ファイルの構文ミスや、エラーの原因究明で、貴重な時間を浪費するのは「失敗」です。
- **思想**: 絵描きは作画に専念し、技術的な壁は早めに相談して解決する。この「一人で悩まず効率的に進める」ことこそが、Weaverの想定するスムーズなワークフローです。

## 8. 仕上げ：マルチプラットフォームへの展開（各ツールでの活用）

LayerWeaver で出力された PSD は、ボーン構造と物性データの「下書き」が埋め込まれた汎用リグ PSD です。

- **Cartoon Animator 5 (CTA5) 等での利用:** 書き出された PSD は、シンプルなボーン付き素材として他社ツールへ持ち込むことが可能です。ただし、それら専用の高度なテンプレート（G3等）として設計されているわけではありません。読み込み後は、各ツールの作法に従って、ウェイトの再調整や表情の設定などを行うための「土台」として活用してください。
- **Moho での利用:** PSD を読み込んだ後、`Weaver_SmartSwitch` スクリプトを実行してください。LayerWeaver で設計した命名規則に基づき、ボーン制御とメッシュ結合の基本的な構造が構築されます。

---

*LayerWeaver：はじめての作画ガイド - 効率的なアセット制作のための第一歩。*