

RigWeaver：はじめてのリギングガイド (v10.0 全面刷新版)

～ イラストとモーションを効率的に繋ぐための実践ガイド ～

最終更新：2026/04/21

[!IMPORTANT] 本ガイドを読む前に：サポート対象アセットの確認 RigWeaverがサポートするのは、**LayerWeaver**を用いて作成されたシンプルな構造のPSDファイルのみです。

他社製品（Cartoon Animator 5等）の**G3キャラクター等の高度なテンプレートには対応していません**。それらの製品は独自の優れた設計を持っており、本ツールでの代用はできません。未対応のアセットを使用するとエラーやリギングの破綻に繋がりますのでご注意ください。

RigWeaverは、クリエイターが描いたイラスト（PSD）に、3Dのモーションデータを論理的に繋ぎ、アニメーション制作の「最初の一步」をスムーズに踏み出すための補助ツールです。

あなたが描いた絵が、3Dの動きを得て動き出す。そこには、本来なら膨大な手作業によるボーン調整や試行錯誤が必要ですが、RigWeaverがその「面倒な計算」を肩代わりします。これは単なる自動化ツールではなく、あなたが思い描くアニメーションを実現するまでの道筋で、技術の壁による挫折を減らすための「制作助手」です。

このガイドでは、キャラクターを動かすための「橋渡しの手順」を、ステップ・バイ・ステップで案内します。

第1章：準備：作画データの「しるし」と必要データ

リギングを成功させるための鍵は、PSD（イラストデータ）の中に、どのボーンに従うべきかという「しるし」を正しく刻んでおくことにあります。

1.1 レイヤー名の命名規則：@（骨の名前）と>（階層）の活用

RigWeaverは、あなたのPSD内のレイヤー名をスキャンし、各パーツがどのボーンに従うべきかを判別します。

- **@ タグ（バインドの指定）**：レイヤー名に **@（骨の名前）** を入れることで、そのパーツがどの骨の動きに追従するかを指定します。
 - **例：@Head**: このタグが付いたレイヤーは、Mohoの頭ボーンに自動的に紐付けられ、動きが同期します。
 - **例：Art@Hand**: 「Art」という画像が「Hand」ボーンに従うようになり、手首の動きに追従します。
- **> タグ（親子の契り）**：特定のボーン配下で、さらに細かい親子関係を作りたい場合に使用します。
 - **例：Sword>Hand**: 剣（Sword）を手（Hand）の子供として配置します。これにより、手が動けば剣も動くという論理的な親子関係が、インポートした瞬間に構築されます。

1.2 バインド設定の効率化：SmartSwitch を見据えた仕込み

後で **SmartSwitch** という「パッキング機能」を使うために、あらかじめレイヤーを整理しておきましょう。

- **グループ化の推奨**: 表情違い（目、口）や、複数のパーツから成るアクセサリ（武器、服の装飾など）は、一つのフォルダにまとめておくのがコツです。
- **DNAの継承**: 各レイヤーに @ タグでボーンの名前を付けておけば、後でそれらをグループ化した際も、「そのグループ全体がどの骨に従うべきか」という情報（DNA）が自動的に上位階層へ引き継がれます。

1.3 必要データの最終確認：必須データセットの配置と整合性

リギングを開始する前に、以下のデータが同じフォルダに揃っているか確認してください。

- **ファイルの「ペアリング」**: RigWeaverがあなたのキャラクターを正しく認識するためには、以下の2つのファイル名が一致している必要があります。
 1. **Character_A.psd**（イラスト本体）
 2. **Character_A_rig.csv**（リギングマップ：どのボーンにどのパーツを貼るかの指示書）

[!NOTE] **_3d.csv（モーションデータ）** について

演技データである **_3d.csv** は、PSDと名前が一致している必要はありません。むしろ、一人のキャラクターに対して「歩き」「走り」「ダンス」といった、異なる名前のモーションファイルを次々と読み込んで「演技を着せ替える」のがWeaverの標準的な使いかたです。

1.4 戦略的リギング：なぜ「3D演出」が先なのか

Weaverシリーズが推奨する最大の効率化は、**「リグを作る前に、3Dでカメラワークを確定させる」**という点にあります。

- **無駄な作業の排除**: 従来の2D制作のように、最初に「360度対応の万能リグ」を作る必要はありません。
- **必要な角度だけを特定**: 3Dで先に演出を固めることで、そのシーンで「本当に必要なキャラの角度と、その枚数」が最初から分かります。使わないアングルのリグを延々と作る「後戻り」と「時間泥棒」を、この段階で完全に排除できるのです。
- **最小限で最大限の効果**: 必要な分だけを描き、必要な分だけをリギングする。この「引き算の制作」こそが、独学者が個人で作品を完成させるための最強の武器になります。

第2章：キャラクターが動き出すまでの「3段階」の活用法

RigWeaverは、すべての設定を完璧に終わらせるまで「待ち」をする必要はありません。あなたの作業進捗に合わせて、段階的にキャラクターを動かし始めることができます。これを使いこなすことで、後からの「描き直し」という悲劇を未然に防ぐことができます。

2.1 段階1：ボーン・プレビズ（動きのタイミングと演出の先行確定）

イラストがまだ描けていない、あるいはデザインが固まっていない初期段階でも、3Dモーションデータ（**_3d.csv**）さえあれば、Mohoの上で「ボーン（骨）だけ」を動かすことができます。

- **何ができる？**: 演技のタイミング、カメラとの距離感、舞台上でのキャラクターの配置など、演出の根幹をこの段階で確定させることができます。

- **メリット**: まだ「絵」がない状態でも、実際のモーションに基づいた「動くビデオコンテ」が作れるため、演出上のミスを上流工程で排除できます。

2.2 段階2：レイヤー・プロキシ（キャラクターの厚みと重なるの検証）

ラフなイラストができたなら、それをボーンに「仮貼り」して即座に動かしてみましょう。

- **何ができる？**: キャラクターが動いたときに「腕が胴体に隠れるか」「前後関係（奥行き）に違和感がないか」を、実際の画像で検証できます。
- **ポイント**: 面倒なリギング設定をすべてスキップして、まずは「絵として成立するか」を物理的にチェックできるフェーズです。ここでアセットの「描き込み不足」や「アライメントのズレ」を早期に見えます。

2.3 段階3：フルリギングと清書シンクロ（プロダクション完成フロー）

すべての準備が整い、いよいよ最終的なアニメーションを出力する最高密度の完成フェーズです。

- **何ができる？**: 3Dの正確な動きに、あなたが描く2Dの緻密なベクターライン（清書）が完璧に同期します。
- **Weaverの主な利点**: 「動いている下書きの上をなぞれば、描き込んだ線が即座に動きと同期する」。このスムーズな清書体験こそが、Weaverシリーズが提供する大きなメリットです。

第3章：最短5ステップ：迷わないための操作手順（前編：舞台構築）

いよいよ実践です。RigWeaverのプログラムは、Mohoのメニューバー **Scripts** -> **RigWeaver** に並んでいます。ボタンを「**左から順番に**」押していくだけでリギングは完了しますが、その前に「一番大切な準備」があります。

3.0 Step 0：PSDのインポートと「倍率の確定」（最重要の準備）

RigWeaverを起動する前に、まずはあなたの描いたキャラクター（PSD）をMohoの舞台へと呼び出し、**「舞台の大きさに合わせて倍率を調整」**する必要があります。

- **操作**: Mohoのメニュー **File** -> **Import** -> **Moho Layer...** から、LayerWeaverで出力したリギング用PSDを選択、インポートします。
- **最重要ポイント**: 読み込み直後のPSDは、Mohoのワークスペースに対して巨大すぎたり小さすぎたりすることがあります。**リギングを開始する前に、イラストがワークスペース（画面中央の青い枠）の中にちょうど良く収まるように、レイヤーの拡大縮小ツールで倍率を確定させてください。**
- **なぜ重要か？**: RigWeaverの各工程は、この「確定された座標」を基準に計算を行います。倍率調整を忘れたままボタンを順番に押すと、レイヤーの位置が激しくズレたり、思わぬ場所に骨が生成される原因となります。

3.1 Step 1：Setup & Viewer（プロジェクト設定と3Dガイドの投影）

舞台に置かれたイラスト（肉体）の上に、「3Dの演技（魂）」を重ね合わせる儀式です。

- **操作**: **1A. Load 3D Data** を実行します。
- **何が起きる？**: モーションデータ（**_3d.csv**）が読み込まれ、キャンバス（ワークスペース）の高さに収まるサイズで「**Viewer（赤いスケルトン）**」が現れます。

- **工学的価値:** これが「魂のガイド」となります。スケルトンの大きさは常にワークスペース基準で自動調整されるため、**Step 0でPSDをワークスペースに合わせてあれば、ここで現れるスケルトンとイラストのサイズが概ね一致しているはずです。**（微調整が必要な場合は、ここでイラストの倍率をスケルトンにぴたっと合わせてください）

3.2 Step 2：Build Skeleton（骨格構築とイラスト・ポーズの完全同期）

次に、PSDの中に刻まれた「しるし（マーカー）」を探し出し、肉体と同期した本番用の骨組みを錬成します。

- **操作:** **2A. Build Hierarchy** を実行します。
- **何が起きる?:** リギングマップ（**_rig.csv**）に基づき、イラストの関節位置を正確に捉えたボーン階層が自動で組み立てられます。
- **魔法の較正 (Step 2B: Align Base Pose):** イラストのポーズ（初期姿勢）が、3Dデータのデフォルト姿勢と少しズレていても安心してください。このステップを実行すると、RigWeaverがイラストのポーズを基準として判断し、すべての骨の初期角度をイラストに合うように自動修正（同期）します。

第4章：最短5ステップ：迷わないための操作手順（後編：魂の注入）

舞台と骨組みが整いました。ここからは、キャラクターに3Dの躍動を流し込み、最終的な「動く体」として完成させるエキサイティングな工程です。

4.1 Step 3：Import Motion（動きの注入とビデオコンテの完成）

Viewerに宿っていた3Dの演技、あるいは**あらかじめ用意しておいた任意の _3d.csv ファイル**を、あなたのキャラクターの骨組み（Skeleton）へと移し替えます。

- **操作:** **3. Retarget Motion** を実行します。
- **オプション設定 (Retarget Options):** 実行すると設定ダイアログが表示されます。目的に合わせて以下を選択してください。
 - **Start motion at current frame:** チェックを入れると、現在のタイムラインの位置（例：24フレーム目）から演技を開始します。別の動きの後ろに繋げたい場合に有効です。
 - **Clear existing motion before applying:** チェックを入れると、すでにタイムラインに入っている古い動きをすべて完全に消去してから、新しい演技を流し込みます。
- **演技の選択:** 設定完了後、ファイル選択ダイアログが開きます。キャラクターにさせたい動き（CSVファイル）を選んでください。
- **何が起きる?:** 選択した演技データが指定した条件でボーンへと転写され、イラストのポーズを維持したままキャラクターが動き出します。
- **この時点の状態:** いわゆる「ビデオコンテ」の完成です。PSDからインポートされた画像はまだボーンに付きませんが、骨（スケルトン）が正しく動いているか、演技のタイミングが演出プランと合っているかを、この段階で**ガイドラインとして納得のいくまで確認**することができます。もし動きがイメージと違うなら、別のCSVを選んで何度でも流し込み直し、納得のいく「演技の土台」を固めてください。

[!IMPORTANT] 最新機能：タイムラインマーカーからの「マルチアクションベイク」 現在のバージョンでは、タイムライン上の任意の場所に複数のマーカー（インポート元CSV絶対パスと開始フレームを記録した **path|start**）を並べることで、後続の **4B. Bake & Bind** 実行時にすべてのアクションを自動スキャンし、一括でボーンにベイクおよびバインドを適用できるようになりました。これによ

り、手動でのインポートや切り替えの手間が劇的に削減され、複雑な演技の繋ぎ合わせがタイムライン上で視覚的かつ安全に行えます。

[!TIP] プロのコツ：ポーズ間自動補間 (Pose-to-Pose) ワークフロー

タイムライン上に持続時間0の「静止画マーカー」を置くと、PNGの書き出しと同時に「1フレームのCSV (`_f[frame]_s`)」がエクスポートされます。この切り出されたキーポーズをMohoのタイムラインに流し込むことで、**「キーポーズ間をMoho独自のベクター・ボーン自動補間で繋ぐ (Pose-to-Pose)」**極めて効率的なアニメーション制作が可能です。長大なモーションをすべてベイクする高負荷を避けつつ、決めポーズだけを自動配置して手動で細部を調整する、手戻りのない2Dアニメーション制作が実現します。

4.2 Step 4A：Convert 2D（3D角から2.5D/2D角への空間翻訳・転写演算）

3Dの複雑な回転を、2D/2.5Dアニメとして一番きれいに見える「魅力的な角度」へと翻訳（流し込み）する、Weaverシリーズの知能が最も発揮される工程です。

- **操作:** 4A. **Build 2D Rig** を実行します。
- **何が起きる?:** 3Dの「鉄の論理」で作られた動きが、2.5D空間の「情緒的な嘘」を許容するコントローラボーンへと変換されます。
- **物理の嘘:** ここで生成される2D用ボーンは、3Dではありえないパースの誇張や、イラスト特有の「見栄えの良い角度」を数学的に正当化し、破綻のないアニメーションを実現します。
- **推奨設定 (2D Modeの切り替え):** 以前のバージョンとは異なり、最新のシステムでは安定した行列投影と二重アンラップ（反転防止）処理が極めて堅牢になったため、「**2D Mode**」**チェックボックスは「チェックなし (OFF=3Dモード)」で実行することが標準の推奨ワークフロー**となっています。これにより、3Dならではの立体的な伸縮やしなやかな身体のねじれが極めて自然かつダイナミックに表現されます。「2D Mode」を有効にするのは、3D的なパース歪みを完全にゼロにしてイラストのフラットなシルエットを100%保護したい特殊なアセット用途に限定されます。

4.3 Step 4B：Hybrid Bind（画像と骨の紐付け：SmartSwitch の威力）

いよいよ、肉体（画像）と骨（ボーン）を物理的に接続します。

- **操作:** 4B. **Bake & Bind** を実行します。
- **何が起きる?:** 画像がボーンに吸いつき、キャラクターの「肉」が動きに追従し始めます。
- **ハイブリッドの賢さ:** 関節は柔らかく（Flexi）、装飾品は硬く（Layer）といったバインド設定が、CSVの定義に従って自動で適用されます。

4.4 Step 5：Finalize（階層整理・ベーキング・アニメーション確定）

リギングという「儀式」の締めくくりです。

- **操作:** 5. **Flatten Rig** を実行します。
- **何が起きる?:** リギング作業のために使っていた一時的な階層（マトリックス）が破壊・整理され、すべての動きがワールド座標へと「焼き付け (Bake)」られます。
- **成果:** これにより、非常に安定した、本番制作に耐えうる「プロダクション・リグ」が完成します。

第5章：効率的な清書フロー：動くイラストの上から「清書」する

リギングが終わったからといって、あなたのクリエイティブが終わるわけではありません。むしろここからが、あなたの絵に真の魂を宿す「仕上げ」の本番です。

5.1 清書同期の実践：グループの中で「設定を継承」する

Weaverシリーズにおいて、最も効率的な手法がこの「清書同期」です。動いているラフ（下書き）を、本番用の美しい線（ベクター）で清書していく工程です。

- **清書用の「器（グループ）」**：もしパーツが画像単体なら、それをグループ（SmartSwitch）化し、名前に「魔法の呪文（@タグ）」を付けます（例：`Arm_Group@Arm`）。
- **設定の継承**：そのグループの中にベクターレイヤーを作成して清書を行うことも、元のPSD画像（ラスター）をそのままボーンで歪ませることも可能です。「動いている下書きをなぞって清書するだけ」で、本番のラインが動きと同期し、2Dならではの誇張表現を自由自在に加えられます。

5.2 グループによる本番化：ラフをガイドに「線を磨く」

画像を差し替えるのではなく、既存のリグを「人形の骨組み」として使い、その内部で線を磨き上げ、完成させていきます。

- **重なりと品質の死守**：ひとつのパーツグループ（例：`Eye@Head`）の中に、ガイドとなる画像と、清書用のベクター線を共存させます。これにより、最終的な線のつながりや重なりを、動きの中で常に確認しながら清書を進めることができます。

[!TIP] 演技の再利用：Mohoアクションの活用

RigWeaverで流し込んだ素晴らしい演技は、Mohoの標準機能である「アクション（Actions）」として登録し、保存することができます。同じキャラクタープロファイル（骨格構造）のPSDであれば、アクションを読み込むだけで即座に同じ演技を再現可能です。

第6章：2Dの「表現の嘘」を味方につける

3Dのデータは「物理的に正しい」ですが、それが必ずしも「2Dとして魅力的」とは限りません。この章では、アニメーションをさらに洗練させるための工芸的な工夫を学びます。

6.1 アングル補正の極意：3Dの生データに「2Dのケレン味」を加える

3Dモーションを流し込んだ直後の動きは、時として「生々しすぎる」ことがあります。

- **補正の考え方**：Mohoのタイムライン上で、特定のキーフレーム（例えば決めポーズ）のボーンの角度を、手動でわずかに「嘘をつく」ように調整します。
- **2Dのケレン味**：3Dでは正面を向いていても、2D的には「あえて少し斜めに見せる」ことで、キャラクターのシルエットをより美しく、力強く見せることができます。

6.2 仮想ボーンによる形状維持：関節の破綻を「増築」で防ぐ

激しい動きをした際に、首や関節の重なりが破綻してしまうことがあります。これはPSDの構造上の限界ですが、ボーンの「増築」で解決できます。

- **仮想ボーン（Virtual Bone）**：`templates.json`を設定することで、PSD上には描かれていない制御用の「補助ボーン」を生成できます。

- **形状のガード**: 例えば「襟（えり）」専用のボーンを作り、首の回転に合わせて襟が不自然にめくれないうように制御することで、イラストのクオリティを死守します。

第7章：レスキュー：破綻（怪物化）を防ぐチェックリスト

リギング中にキャラクターが急に「怪物」のような形になってしまっても、落ち着いてください。その多くは、単純な設定の不整合が原因です。

7.1 ボーンの裏返り、ねじれの解消法

キャラクターが180度裏返ったり、関節が不自然にねじれる場合があります。

- **原因**: 3Dモーション（`_3d.csv`）の回転軸と、2Dイラストの描かれた向きが一致していない可能性があります。
- **解決策**: **Step 1A** の実行時に設定する **Angle Offset**（角度オフセット）を見直してください。また、Moho上で対象のボーンを選択し、「ボーンのリセット」を行うことで、初期状態の正気を取り戻すことができます。

7.2 座標ズレの修正：原点回帰の儀式

ボーンとイラストが大きく離れてしまった場合は、以下のチェックリストを確認してください。

- **魔法の呪文の再確認**: PSD内のレイヤー名（@や>）が間違っていないですか？
- **親子関係の断絶**: すべてのボーンが **Hip**（腰）を起点とした正しい家系図になっているか確認してください。
- **無限ループの警告**: もしスクリプトが「CRITICAL ERROR: Circular reference」という警告を出して止まったなら、仮想ボーンの親子関係が自分自身に戻っている（ループしている）合図です。
`templates.json` の設定を見直しましょう。

第8章：ステップアップ：量産工学（BatchWeaver）への入り口

おめでとうございます！ここまで辿り着いたあなたは、リギングのワークフローを習得し、3Dと2Dを繋ぐ効率的なアニメーション制作の手法を手に入れました。

8.1 100本のモーションを一括変換する：クリエイターから監督へ

RigWeaverの真の力は、量産フェーズで発揮されます。

- **BatchWeaver**: **Step 4** で書き出した **batch.bat**（一括処理ファイル）を実行することで、1つ1つ手作業で行っていたバイク工程を、数十、数百のファイルに対して全自動で行うことができます。
- **監督の視点へ**: あなたが1枚のキャラクターを描き、一度リギングを完成させれば、あとは何十種類もの演技データを流し込むだけで、大量のアニメーション素材が数分の間に自動生成されます。

おわりに：一歩ずつ、理想の動きへ

RigWeaverは、技術の制約による「制作の挫折」を減らし、表現の可能性を広げるためのツールです。正しい手順（Step 0～5）をマスターした今、あなたは複雑な設定に悩まされることなく、キャラクターの演技を追求するスタートラインに立ちました。

Weaverシリーズは、あなたの創作の熱量を形にするための「確かな足場」として、これからも進化を続けます。さあ、次のキャラクターの制作に取り掛かりましょう！