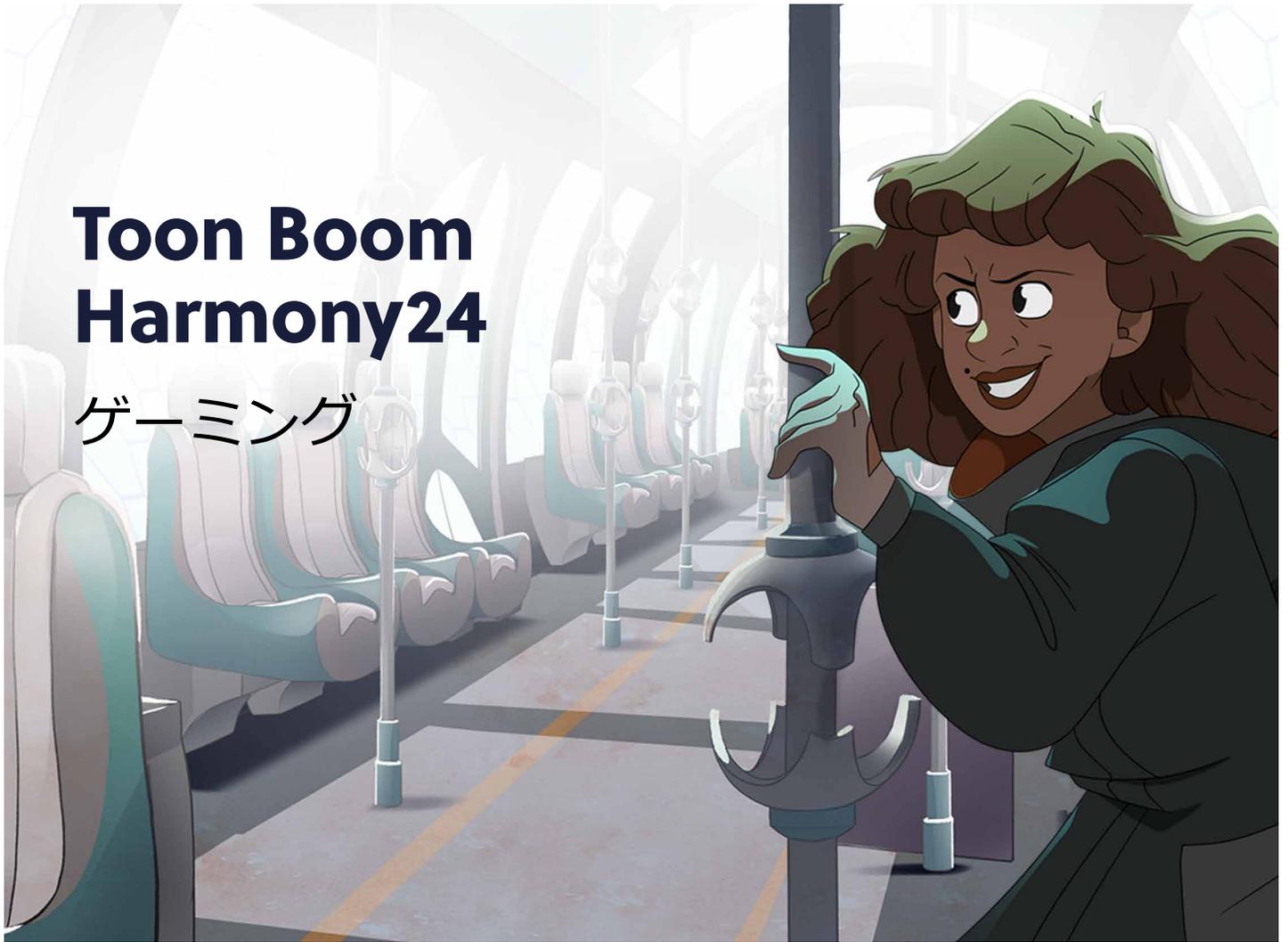


# Harmony 24

## Toon Boom Harmony24

ゲーミング



toon  
boom

---

**TOON BOOM ANIMATION INC.**

55 Mont-Royal Avenue West, Suite 1000  
Montreal, Quebec, Canada  
H2T 2S6

---

**+1 514 278 8666**

[contact@toonboom.com](mailto:contact@toonboom.com)  
[toonboom.com](http://toonboom.com)

## 法的通知

Toon Boom Animation Inc.  
55 avenue du Mont-Royal Ouest  
Montréal Québec, Canada  
H2T 2S6

電話 : +1 514 278 8666

Fax : +1 514 278 2666

[toonboom.com](http://toonboom.com)

## 免責事項

本文書の内容は Toon Boom Animation Inc.の財産であり、著作権で保護されています。全体または一部を複製することは固く禁じられています。

本文書の内容は、該当する使用許諾契約の下での特定の限定的保証と例外および責任制限の対象であり、Adobe®Flash®ファイル形式 (SWF) の特別条件によって補足されます。詳細はライセンス契約および特別利用規約をご覧ください。

本文書の一部のアイコンは、Font Awesome の Font Awesome Free 5.6.1 で提供されました。これらのアイコンは、CC BY 4.0 ライセンスの下で提供されています。Font Awesome の詳細については、<https://fontawesome.com> にアクセスしてください。Font Awesome Free のライセンスについては、<https://fontawesome.com/license/free> をご覧ください。

この文書のアイコンの一部は、Glyphicons の Glyphicons Halflings フォントで提供されました。Glyphicons の詳細については、<https://www.glyphicons.com/> を参照してください

## 商標

Toon Boom® は登録商標です。Harmony™ および Toon Boom ロゴはToon Boom Animation Inc.の商標です。その他のすべての商標はそれぞれの所有者に帰属します。

## 発行日

2024-05-31

著作権 © 2024 Toon Boom Animation Inc. (Corus Entertainment Inc. の子会社) All rights reserved.

# 目次

目次	3
ゲームプレイについて	5
HarmonyゲーミングSDK 2022リリースノート	6
<b>第1章： Harmonyゲーム用データのエクスポート</b>	<b>1</b>
<b>第2章： ゲームアセット作成について</b>	<b>4</b>
ゲームリギングガイドライン	5
ゲーム変形ガイドライン	8
ゲームボーン変形について	9
ゲームカッターのガイドライン	10
メタデータメモの作成	11
ゲームアニメーションのヒント	14
複数のシーンのアニメーション化	15
ワークフロー 1： セパレートカット	16
ワークフロー 2： カットマーカを使用した分離	17
正投影カメラについて	18
<b>第3章： Unity へのエクスポートについて</b>	<b>19</b>
アンカーの設定	20
スプライトシートのエクスポート	21
スプライト解像度	23
Bake_Groupsの使用	25
Easel JSへのエクスポート	27
パレットバリエーション	29
<b>第4章： Harmony Unity SDKについて</b>	<b>32</b>
Harmonyプレビューア	36
アセット操作	38
サンプルUnityプロジェクトについて	45
Unity インターフェースについて	46
HarmonyファイルをUnity にインポート	48

Unity での TBG ファイルワークフロー .....	50
Unity での XML フォルダーワークフロー .....	57
空のゲームオブジェクトへのHarmonyレンダラーの追加 .....	61
Unity での衝突の設定 .....	63
ゲーム用語集 .....	64

## ゲームプレイについて

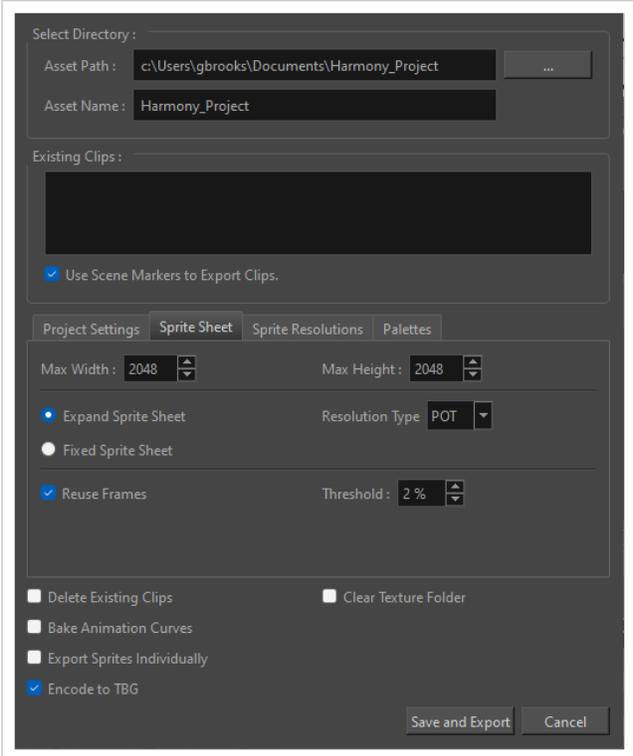
Harmonyを使用すると、Harmonyで作成されたアニメーションを外部のゲームエンジンにエクスポートすることができます。これにより、ゲームエンジンで使用できる単純な動きのキャラクターをデザインできます。

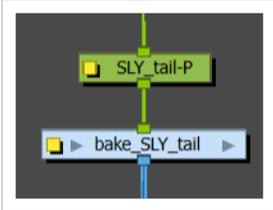
Harmonyデータを組み込むことができるゲームエンジンの例として、Unityが挙げられます。Toon BoomにはHarmony SDKがあり、Unityアセットストアからダウンロードして、Unityにインポートしたキャラクターをコントロールできます。

このガイドでは、Harmonyからアニメーションをエクスポートする方法を示します。さらに、UnityとのHarmony統合のためのHarmonyゲームSDKも詳細に説明されます。

# HarmonyゲーミングSDK 2022リリースノート

## Harmonyからのエクスポート

機能	内容
<p>新しいエクスポーターオプション</p>	<p>ゲームエディター内のサポートを改善するために、[スプライトシートにエクスポート]インターフェイスには、エクスポートされたファイルにより多くの情報を詰め込むための一部の追加機能が与えられました。</p>  <p><b>新しいオプション</b></p> <ul style="list-style-type: none"> <li>• パレット選択タブ - エクスポートされたファイルに含めたくない特定のパレットを除外すると、キャラクターに異なる色が表示されるパレットのすべての可能なバリエーションが含まれます</li> <li>• TBGへエンコード - XML ファイルを TBG カスタムファイル形式に圧縮し、Unity カスタムインポーターを使用して Unity にインポートできるようにします。ファイルがインポートされると、Unity のアセットと同様に扱われ、テクスチャ、スプライト、アニメーションのサブアセットが含まれます。これにより、キャラクターデータの作成と移動、およ</li> </ul>

機能	内容
	<p>びソース管理への変更の確認の人間工学が向上します。</p> <ul style="list-style-type: none"> <li>「保存してエクスポート」のパフォーマンスを改善           <ul style="list-style-type: none"> <li>- エクスポート処理は、XMLフォルダーと.tbhgファイルの両方で、Harmony 21と比較して通常で1/3の時間がかかります</li> </ul> </li> </ul>
「bake_」グループ	<p>現在、ゲーム SDK ではレンダリングできないノードが多数あります。これらのノードの値をエクスポートに取り込む手段として、描画とそれらに影響を与えるノードを「bake_」接頭辞を持つ新しいグループにグループ化する新しい機能があります（例：bake_hips）。</p>  <p>「bake_」接頭辞が付いたグループは、エフェクト、動き、描画の置換に関するアニメーションの一意のコマについて分析されます。スプライトシートにエクスポートすると、このグループの一意の各コマがスプライトとしてエクスポートされます。タイムラインの多くのコマがそのグループ内で一切変更されていない場合、それはすべてスプライトシートで同じエクスポートされたスプライトと見なされます</p>

## Unity での TBG ファイルのインポート

機能	内容
自動インポートプロセス	<p>Unity は TBG ファイルへの変更を自動的に検出し、カスタムインポーターを使用して再インポートします。その後、FBX ファイルに期待するのと同じように、ファイルをプレハブとしてカットにドラッグできます。</p> <p>テクスチャ、スプライト、アニメーション、Tbgストアアセットが生成され、メインの TBG プレハブ アセットの下にサブアセットとして提供されます。これらのアセットは、ソース管理で個別に管理する必要はなく、Unity のライブラリーキャッシュにのみ存在します。アニメーターコントロー</p>

機能	内容
	ラートとスプライトシートアセットは、サブアセットとして動作できないため、キャラクターの横に生成されます。
インポーター設定	<p>インポートプロセスは、[インスペクター]ビューの設定から変更できます。</p> <p>アニメーション設定</p> <ul style="list-style-type: none"> <li>・ 離散化ステップ - スプライトがボーンによって変形されている場合に分割される列/行の数。</li> <li>・ フレームレート - アニメーションが実行されるコマ/秒の数。</li> <li>・ ステップ付き - オーサリングコマ間のフレーム補間を無効にします。</li> <li>・ アニメーターコントローラーの作成 - オーサリングされたコマ間のフレーム補間を無効にします。</li> <li>・ アニメーターコントローラー - 特定の条件下でアニメーションを相互に遷移するためのワークスペースを提供します。ゲームプレイスクリプトで参照することで、ゲームプレイ中にアニメーションを発動させることができます。</li> <li>・ クローンクリップのカーブを維持する：.tbg ファイル内のクリップサブアセットから複製された（およびアニメーターコントローラーで参照される）カーブは、更新された .tbg ファイルの新しいデータでオーバーライドされます。</li> </ul> <p>素材設定</p> <ul style="list-style-type: none"> <li>・ シェーダー - プレハブ内のすべてのスプライトをレンダリングするために使用されます</li> <li>・ SRGB テクスチャ：テクスチャのガンマ色空間が線形色空間レンダリングにマッピングします</li> <li>・ フィルターモード - テクスチャの隣接するピクセルの補間方法</li> <li>・ ミップマップ有効 - 低解像度のテクスチャを生成して、さらに遠くにレンダリングします</li> <li>・ 作成スプライトアトラス - 以下でスプライトアトラスが参照されていない場合、新しいスプライトアトラスアセットがプロジェクトの.tbg ファイルの横に作成されます</li> <li>・ スプライトアトラス - .tbgファイルのすべてのスプライトはスプライトアトラスに追加することが可能で、完了すると複数の新しいスプライトレンダラーを1つのドロージャケットに組み合わせ、レンダリングパフォーマンスを向上させることができます</li> </ul>

## Unity での TBG ファイルのレンダリング

機能	内容
プレハブ階層	Unity で生成されたプレハブの構造は、Harmonyのタイムラインで設定された階層構造と一致する必要があります。

機能	内容
	<p>出力ノードにフィードされる合成のHarmonyの[ノード]ビュー構造は、エクスポート時に除外されます。</p> <p>新しいゲームオブジェクトは、これらの子ペグトランスフォームのいずれかにアタッチできます。これにより、手続き型の機器のアタッチが可能になります。</p> <p>また、任意のゲームオブジェクトを実行時に有効化または無効化できるため、キャラクターへのオプションの視覚的な追加は、必要になるまで非表示にすることができます。階層は、実行時に完全に分解、再構築、または複製することもできます。これにより、キャラクターを爆発させることや、特定の描画を発射体として撃つオプションなどが可能になります。</p>
TBG レンダラー設定	<p>解像度、パレット、素材、色、スキンは、インスペクターとゲームオブジェクトのカスタムスクリプトから編集できます。スプライトレンダラーの素材と色は描画ごとに個別に設定することもできるため、カスタムシェーダーを使用し、キャラクターのアクセサリの色を変更することや、一部の描画に特殊nエフェクトを提供することができます。</p>
2Dアニメーションパッケージの統合	<p>Harmony内のゲームボーンによって変形された描画は、Unityの「2Dアニメーションパッケージ」を使用してUnityで[スプライトスキン]ボーンに変換されます。本パッケージは、スプライトレンダラーのためのマルチスレッドの高性能メッシュ変形と、カットビューでボーンを移動させるためのUIハンドルをサポートします。ボーンチェーンは、アニメーションの更新ステップの後に、照準、傾き、揺れ、手を振るなどのエフェクトをブレンドするために、Unityの「IKマネージャー 2D」を使用して手続き的にアニメーション化できます。</p>
アニメーション	<p>すべてのアニメーションデータはペグごとにHarmonyからエクスポートされるため、すべての中間回転とスケールはUnityのアニメーションウィンドウに表示されます。関連するアニメーションクリップアセットがTBGサブアセットからプロジェクト内の新しいアセットに複製されると、アニメーションを変更できるようになるため、Unity内でイベントを追加したり、追加のカーブを提供したりできます。</p> <p>アニメーションクリップはアニメーターコントローラーを</p>

機能	内容
	使用して選択され、アニメーション間を即時に、またはペグトランスフォームのスムーズなブレンドで切り替える手段を提供します。
シェーダーグラフの統合	互換性のあるシェーダーグラフのシェーダーは、複製および変更できるSDKの一部として提供されており、キャラクターにカスタムゲーム固有の視覚効果を簡単に導入できます。

## Harmonyレンダラーの改善 (XML フォルダー)

機能	内容
Harmonyプロジェクトプレビュー	Unity アセットフォルダーでキャラクターを選択すると、そのキャラクターのプレビューがプレビューウィンドウに表示されます。プレビューウィンドウにはキャラクターのアニメーションのプレビューが再生され、キャラクターをカットビューにドラッグしなくてもスキンを変更するオプションがあります。

## 修正

HarmonyゲームSDKのリリースで、以下の問題が修正されました。

## Harmonyレンダラー

- 以前は、アンカーはアンカートランスフォームの更新中にメモリをリークしていました。この問題は、Harmonyレンダラー C++ プラグインの更新によって解決されます。
- Harmonyレンダラーのレンダリングパフォーマンスが向上しました。HarmonyレンダラーC++ プラグインから新しいメッシュを受け取り、新しいUnityメッシュ生成機能を使用して、UnityのC++バックエンドに直接渡すことができるようになりました。これにより、Harmonyレンダラーキャラクターのアニメーションの更新にかかる時間が大幅に短縮され、一度に画面上により多くのキャラクターを表示できるようになります。



## 第1章： Harmonyゲーム用データのエクスポート

Harmony からゲームエンジンにデータをエクスポートするためには、2つの主要なパイプラインがあります。

- [Harmonyゲーム用データのエクスポート \(1ページ\)](#)
- [Harmonyゲーム用データのエクスポート \(1ページ\)](#)



### Harmonyでのアニメーションの作成

ゲームのキャラクターリグとアニメーションを作成するときは、始める前に検討することがいくつかあります。アーティストとプログラマーは、お互いのニーズが満たされていることを確認するために協力する必要があります。

- ゲームはどのプラットフォーム向けに作成されますか？ Windows、Mac、モバイル、iOS、PS、XBOXなど
- どのゲームエンジンを使用しますか？
- ゲームの外観を作成するためには、どのようなアニメーションスタイルが使用されますか？ 手描き、切り抜き、テクスチャ付きなど

これらは、始める前に検討すべき質問のほんの一部です。これらはすべて、キャラクターのデザイン、構築、アニメーション化の方法に影響を与えます。

たとえば、スマートフォン向けのモバイルゲームを計画している場合は、Wi-Fiに接続しなくてもダウンロードできるよう、ゲームを50MB未満に抑えることをお勧めします。その場合に最も重要な考慮事項は、非常にタイトなスプライトシートを使用して効率的なキャラクターを作成し、多くのアニメーションを再利用してファイルサイズを小さく保つことです。作業工程の間は、次のことを行う必要があります。

- Harmonyでキャラクターをリグ・アニメーション化。
- Harmonyデータを抽出。
- ゲームエンジンにHarmony データをインポート。

カスタムエンジンで作業している場合は、エキスポートされたデータを処理して、カスタムゲームエンジンで使用することもできることを覚えておきましょう。 Harmonyエンジンに合わせてデータを適合させる際にサポートが必要な場合は、 [store.toonboom.com/contact/support](https://store.toonboom.com/contact/support) にお問い合わせください。

PSやXboxなどのコンソール用のゲームを作成している場合は、より大きなテクスチャを自由に作成できます。 コマごと、カットアウトされた文字付き、またはその両方をアニメーション化することができます。

コマごとのシーンでエキスポートするだけの場合は、 のツールをすべて制限なく使用できます。 Harmonyその後、エキスポートされた画像シーンをスプライトシートに処理できます。

## ゲーム生データのエキスポート（スプライトシートへエキスポート）



ゲーム生データのエキスポートは、ファイルサイズを可能な限り小さくしたい場合に適しています。 によって、 Harmonyカットからデータを直接変換し、ゲームエンジンに組み込むことができます。 Toon Boomスケルトン情報、描画情報、キーフレームアニメーションデータのほか、変形（ボーンとアーティキュレーションのみ）、カッター、透明度ノード、タイミング列を抽出できます。

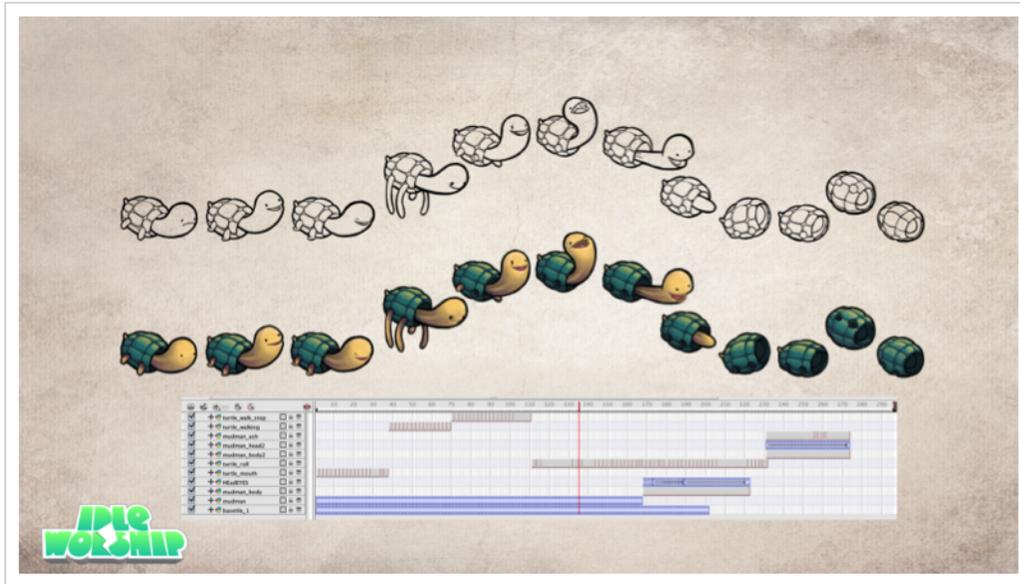
- **メリット：**これは最も軽いエキスポートであり、ファイルサイズを小さく保ち、モバイルアプリケーションに最適です。
- **デメリット：** **で使用できるツールによって多少制限されています。** Harmonyモーフィングやカーブデフォーマー、エンベロープデフォーマーなどのツールを使用できますが、ゲームエンジンで適切に解釈されるように、描画にベイクアウトする必要があります。 カッターエフェクト（マスキング）は使用できますが、カスケードすることはできないため、階層チェーンに複数のエフェクトを含めることはできません。 ゲームボーンデフォーマーは、描画にベイクしなくてもリグで使用できます。

ただし、これらの制限があっても、で非常に優れたカットアウトキャラクターアニメーションを作成し、互換性のあるすべてのデータを抽出することができます。 Harmonyさまざまな描画レイヤーを移動、回転、拡大縮小、傾斜させることで、高度な外観のアニメーションを作成できます。

データを抽出すると、カットファイルで使用されているボディパーツの描画だけを集めたスプライトシートができあがります。 Harmonyまた、アイドル、実行、ジャンプなどの複数のアニメーションをサポートしながら、同じスケルトンと描画を再利用することもできます。

は、このソリューションを Unity ゲーム開発レンダリングエンジンと完全に統合しています。Toon BoomUnity でゲームを作成する場合は、ゲームエンジンでデータを再処理することなく、シームレスなパイプラインを使用できます。

## コマごとのエクスポート（Easel JSへのエクスポート）



スプライトシートに再コンパイルできる画像シーンを使用して、Harmonyからエクスポートできます。

**メリット：**アニメーション化する時に、Harmonyの中のどのツールでも使用できます！

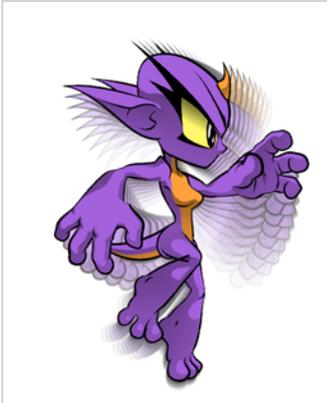
**デメリット：**これらのスプライトシートは非常に重くなり、ゲームで処理するには大きなテクスチャになる可能性があります。これは、コンソールゲームなどの多くのゲームでは問題ないかもしれませんが、iOSやAndroidなどのモバイルデバイス用のゲーム開発を行う際には、テクスチャ空間を非常に軽くする必要があります。また、Wi-Fiに接続しなくてもダウンロードしやすいように、ファイルサイズを50MB未満に保つ必要があります。

コマ単位のアニメーションをEaselJSゲームエンジンにエクスポートするためのスクリプトが用意されています。HarmonyEaselJSはJavaScriptウェブと互換性のあるゲームエンジンです。json データ構造を使用して、各コマをスプライトシートのエンTRIESにマッピングします。データ構造は理解しやすく、必要に応じて別のゲームエンジンに適合または変換できます。

EaselJSにエクスポートするには、スクリプトツールバー（ツールバー\_EaselJSにエクスポート）にスクリプトを追加する必要があります。このスクリプトは、に含まれています。Harmonyスクリプトを[スクリプト作成]ツールバーに追加する方法とインターフェイスパラメーターの詳細については、スクリプト作成ガイドを見るを参照してください。

## 第2章：ゲームアセット作成について

このセクションでは、リギング、変形、カッター、そして多くの役立つアニメーションのヒントに対するガイドラインがあります。



## ゲームリギングガイドライン

以下は、キャラクターをリギングする際に留意すべき一般的なガイドラインの一覧です。ゲームのキャラクターリグを計画するときは、キャラクターのスタイルを念頭に置いて、カスタムのカラーパレットを作成します。ただし、ゲームデータを抽出する場合は、いくつかの考慮すべき制限があります。

- カットを正方形の解像度（例：1024 x 1024）に設定します。Harmonyこれは、[カット設定]ダイアログボックスで行うことができます。リファレンスガイドを参照。
- 線画レイヤーとカラーアートレイヤーにのみ、アートを描画します。オーバーレイとアンダーレイに含まれる情報は、Unity にエクスポートされません。ただし、エクスポートされるものでなければ、参照レイヤーとして使用するのは安全です。
- レイヤーの順序を変更する必要がある場合は、Z 空間のレイヤーをナッジします。ただし、キャラクターリグ内では、有意な Z オフセットはサポートされていません。
- ゲームエンジンのすべてのレイヤーをで別々のカットにします。Harmony深さが異なる 2 人のキャラクターがある場合は、別々のカット ファイルに入れます。
- カットのルートレベルのグループ内のキャラクターリグは、Unity で 1 つの平面にレンダリングされますが、別々のスプライトシートとアニメーションデータセットを使用します。Harmony複数のキャラクターが互いに対話するカットでは、この点に注意してください。
- 3D 空間は使用しないでください。3D空間でセットアップしたい場合は、ゲームエンジンに到達したときにできます。
- 回転ツールを使用してペグレイヤーにピボットポイントを設定し、レイヤー全体にピボットを設定します。描画ピボットよりも、ペグピボットをお勧めします。また、描画レイヤーにアンカーを配置する必要がある場合に、後でゲームエンジンで情報を取得できるため、アニメーション化しない場合も、描画レイヤーにピボットポイントを設定する必要があります。
- モーフィングは使用しないでください。これは、ゲームエンジンではまだサポートされていません。
- モーフィングとデフォメーションの両方を使用して、個々の描画にベイクアウトすることができます。これらの描画は、新しい描画としてスプライトシートに表示されます。テクスチャスペースが増えるため、これを頻繁に行うことに注意してください。
- エクスポートする前に、キャラクターを配置する場所に注意してください。エクスポートしたゲームオブジェクトのマスターピボットは、Harmonyカット (0,0) の中心になります。
- 階層の最後にディスプレイがあることを確認してください。
- リギングするときは、エクスポーターが一部のノードに対応していないことに注意してください。結果として、ノード構造はゲームエンジンによってうまく解釈されません。したがって、複雑なノード構造の使用は避けることをお勧めします。このためには、次の手順を実行します。
  - ペグにアタッチされた描画からキャラクターを構築します。
  - その後、ペグを他のペグに接続して階層を形成することができます。

- ノードビューの上部にある単一のマスターペグに、すべてのペグを接続する必要があります
- 描画は通常、合成に接続して、画面に表示される描画の順序を決定する必要があります
- アニメーション化の間に描画の順序を変更する必要がある場合は、z深度をバンプすることで、描画が最初に表示するものを変更できます
- 描画とペグだけではアニメーションをうまく機能させることができない場合、エクスポートできる他のノードがいくつかあります。

#### カッター

- - 通常、1つの描画のみを他の1つの描画に対してカットします（例：顔を頭の境界まで切る）。あるカッターの出力を別のカッターの入力に「デ이지チェーン」することはできません。

#### ゲームボーンの変形

- - 通常、1つの描画のみを変形させます。ボーンスケルトンは分岐することができます。キネマティック出力は、ボーンの端に描画を添付して使用できます（例：足の端に足）  
「bake\_groups」内では、より複雑なノード構造を引き続き使用できます。



#### メモ

[「バイクグループの使用」](#)

を参照。

これらのヒントを念頭に置いておくと、すべての優れたツールを活用しながら、Harmonyでタイトで効率的な2Dゲームキャラクターを作成しすることができます。

ここに、いくつかやるべき事を記載します。

- タイムラインビューで単純な親子関係階層を作成する。
- ペグレイヤーを使用してキーフレームアニメーションデータを含め、[位置を分離]に設定する。
- 描画レイヤーを使用して描画し、必要に応じて新しい描画を作成します。
- [回転]ツールを使用して、ペグレイヤーにピボットポイントを設定します。
- 後からゲームエンジンで特定のレイヤーのピボットポイントを取得する必要がある場合に、必要なレイヤーを簡単に認識できるよう、レイヤーに適切な名前を付けます。最上位のグループ A に子グループ（グループ B）があり、描画レイヤーがグループ B の子である場合、描画レイヤーはA\_B\_DrawingLayerとして書き出されます。
- Unity でペグをトランスフォームとして表示する場所にアンカーを設定しますを参照）。

鉛筆ツールとブラシツール、テクスチャ線、ソリッドエリア、グラデーションなど、お好きな描画ツールをどれでも使用できます。個々の描画はレンダリングされ、後でスプライトシートに組み立てられます。



**メモ**

UnityゲームエンジンはUnicode文字に対応していないため、ゲーム用のカットでの使用は避けることをお勧めします。

## ゲーム変形ガイドライン

は、変形をゲームエンジンのXML形式にエクスポートできます。Harmonyゲームボーンで構成された階層のみを正常にエクスポートできます。加えて、ゲームボーンを補完するためにキネマティック出力が実装されましたを参照)。

の変形には、幅広い機能セットが付属しています。Harmonyただし、これらの機能の一部はゲームエンジンSDKと互換性がないため、ゲームには使用してはなりません。したがって、ゲームボーンデフォーマーは、次の標準のボーンデフォーマー機能に対応していません。

- 影響範囲
- カーブとエンベロープの変形
- 変形階層に複数のポーズがある
- 変形グループの下の描画ペグのアニメーション化

ゲームエンジンSDKの変形は、Harmonyのボーン変形と全く同じようには働きません。

ほとんどのゲームエンジンに準拠し、高速な計算を維持するために、SDKはアーティキュレーションでボーンをブレンドする線形ベース スキニング アルゴリズムを実装しています。使用するアーティキュレーションの曲率によって、いくつかの違いに気付くかもしれません。ですが、通常の変形と比較すると、ゲームボーンはUnityにインポートしたときにHarmonyに表示されるものに最もよく似ています。



### メモ

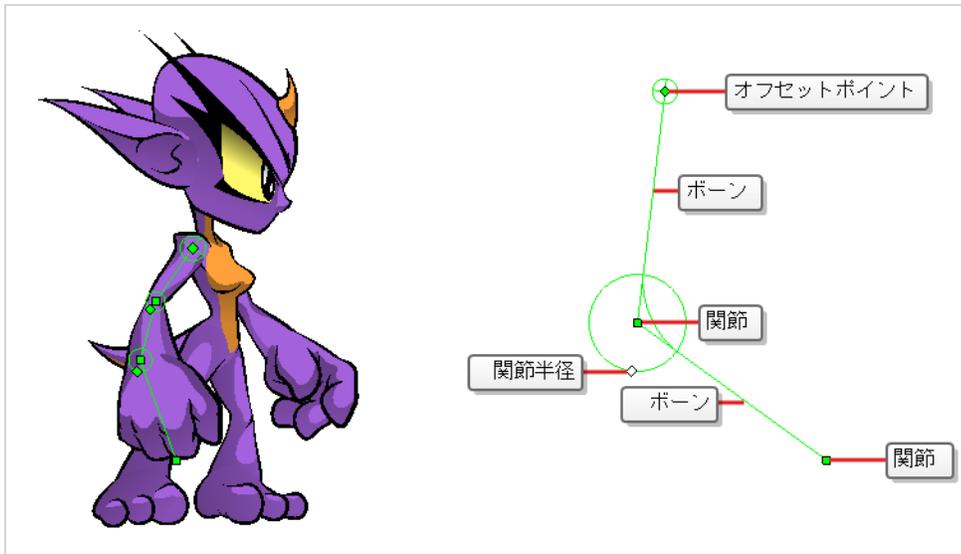
\*現在、ゲームエンジンSDKのCocos2d-x実装では使用できません。

## ゲームボーン変形について

T-RIG-007-003

ゲームボーン変形はボーン変形によく似ています。これにより、各パーツは硬いけれど関節が柔軟な、骨のような構造を作成することを可能にします。これは、キャラクターの腕や脚などの四肢、または胴や指などの関節を動かすことができる他の部分をアニメートする場合に最も役立ちます。例えば、ゲームボーン変形を使用すると、上腕と前腕を異なるレイヤーに描かなくても、1つの描画からなる腕を関節運動させることができるので、上腕と前腕を独立して動かすことができます。は描画を変形して、関節のある見た目にします。Harmonyゲームボーン変形の様々なパーツをその関節周りに回転して伸ばしたり縮めたりすることができます、これにより、パーツの取り外し、ピボットポイント、またはアウトラインのクリッピングを心配することなく、様々なレイヤーで関節をアニメーション化すると同じ機能が得られます。

ゲームボーンの変形は、Unityなどのゲームエンジン用に最適化されたボーンの変形とは異なります。したがって、通常はゲーム開発にのみ使用され、アニメーション制作には使用されません。ボーンとゲームボーンの変形の違いは、ゲームボーンの変形にはバイアスプロパティバイアスと影響領域プロパティがないことです。関節の折り目も少し丸みを帯びて見えます。



---

## ゲームカッターのガイドライン

でのカッターまたはマスクの操作は、カスタム形状の描画を切り取るために使用されます。 HarmonyゲームエンジンSDKは、カッターと逆さカッターの両方を実装しますが、次の制限があります。

- スプライトの場合、レンダリング時に適用できるカッター描画は1つだけです。これは、複数のマット描画のコンポジットにも当てはまります。ゲームエンジンSDKは、レンダリング中に最初のマット描画のみを使用し、他のマット描画を破棄します。
- 変形した描画はカットできませんが、カットされた描画は変形できます。



### メモ

現在、ゲームエンジンSDKのCocos2d-x実装では使用できません。

## メタデータメモの作成

でゲームのアセットを作成する際に、プログラマーのためにカットや、キャラクター、プロップの特定の部分についてメモをとることをお勧めします。 HarmonyアセットとともにUnity にエクスポートされるこれらの埋め込みメモは、メタデータと呼ばれます。

### [メタデータエディタ]ビューにアクセスする方法

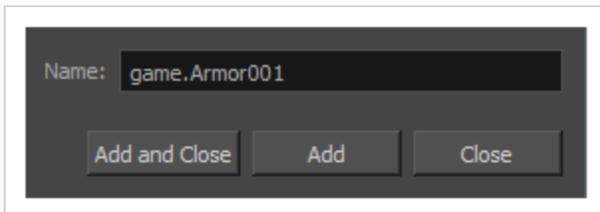
- ビューの右上隅で[ビューを追加] + ボタンをクリックして、**メタデータエディタ**を選択します。
- トップメニューにて、**Windows > スクリプトエディタ**を選択します。

### カットメタデータの作成方法

1. [メタデータエディタ]ビューの[カットメタデータ]セクションで、プラス + ボタンをクリックし、新しいメタデータエントリを作成します。

[メタデータを追加]ダイアログボックスが表示されます。

2. [メタデータを追加]ダイアログボックスで、新しいカットエントリの名前を入力します。



#### メモ

メタデータエディタ内で作成されたアンカーまたはプロップ情報に関しては、命名規則では常に接頭辞「game.」を使用する必要があります。「game.」という用語は、Unity によってメタデータ情報として自動的に認識されます。例：  
*game.Armor001*。

3. エントリを1件だけ追加する場合は、追加して閉じるをクリックします。複数のエントリを追加する場合は、追加をクリックします。継続して複数のエントリを追加する場合は、各エントリの後に追加をクリックします。終わったら、**閉じる**をクリックします。
4. [メタデータエディタ]ビューで、最初のエントリの値フィールドをダブルクリックして編集可能にします。
5. このエントリの値情報を入力します。

6. 継続してすべてのエントリの値情報を追加します。

この情報は、Harmonyアセットとともにエクスポートされます。Unityに入ると、[階層]ビューでアセットが選択された後に、カットメタデータが[インスペクター]ビュー→メタデータで表示されます。

## ノードメタデータの作成方法

1. タイムラインビューで、メタデータを添付するレイヤーをクリックします。

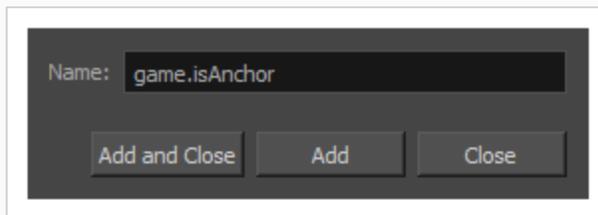
レイヤーの名前が[ノードのメタデータ]セクションの上部近くに表示されます。



2. [ノードメタデータ]セクションで、プラス+ ボタンをクリックして、新しいメタデータエントリを作成します。

[メタデータを追加]ダイアログボックスが表示されます。

3. [メタデータを追加]ダイアログボックスで、新しいノードエントリの名前を入力します。



### メモ

メタデータエディタ内で作成されたアンカーまたはプロップ情報に関しては、命名規則では常に接頭辞「game.」を使用する必要があります。「game.」という用語は、Unityによってメタデータ情報として自動的に認識されます。例：  
*game.Armor001*。

4. エントリを1件だけ追加する場合は、追加して閉じるをクリックします。複数のエントリを追加する場合は、追加をクリックします。継続して複数のエントリを追加する場合は、各エントリの後に追加をクリックします。終わったら、**閉じる**をクリックします。

5. [メタデータエディタ]ビューで、最初のエントリの値フィールドをダブルクリックして編集可能にします。
6. このエントリの値情報を入力します。
7. 継続してすべてのエントリの値情報を追加します。

この情報はHarmonyアセットとともにエクスポートされ、具体的には、この情報は選択したレイヤーに関連付けられます。Unityに入ると、[階層]ビューでアセットが選択された後に、カットメタデータが[インスペクター]ビュー→メタデータで表示されます。

---

## ゲームアニメーションのヒント

ゲームのアニメーションを作成する場合、作成するゲームの種類によっては、アニメーションを制限する必要がある場合があります。たとえば、モバイルゲームを作成していて、ファイルサイズを小さく保ち、すべてのデバイスで高速に再生したい場合は、描画スワップをできるだけ少なくした単純なキーフレームアニメーションに制限します。コンソールゲームを作成している場合は、より多くの描画を自由に作成でき、複雑さが増します。エクスポート先のプラットフォームの制限と、ゲームエンジンの対応可否を、開発者と一緒に調べることができます。

効率的で軽量なアニメーションのヒントをいくつか紹介します。

- 主に回転、拡大縮小、傾斜などの変換を使用します。
- 必要に応じて、追加の描画スワップを作成します。
- カーブデフォーマーとエンベロープデフォーマーまたはモーフィングを使用する場合は、書き出すために描画をベイク処理する必要があります。これを行うときは、描画の数を少なくしたい場合がありますので、注意してください。シーン全体をベイクアウトするのではなく、選択した描画だけをベイクアウトします。ゲームボーンデフォーマーのベイクは不要です。
- [描画]ビューでの描画が大きいほど、スプライトシートのテクスチャサイズで占めるピクセル数が多くなります。リグを設定するときは、トランスフォームツールでキーフレームを使用して個々のレイヤーを拡大縮小しないようにしてください。スケールアップまたはスケールダウンする場合は、[選択]ツールを使用します。これにより、スプライトシート上で同じ相対サイズが維持されます。スプライトシートをエクスポートするときに、スクリプトでスプライトシートの解像度を設定して、小型デバイス用に描画を縮小することもできます。
- カットに公開されている描画のみがスプライトシートにエクスポートされます。たとえば、[ライブラリ]ビューに10個の描画があり、そのうちの2個しかカットに表示されていない場合、その2つだけが書き出されます。これにより、スプライトシートが可能な限りタイトに保たれます。

## 複数のシーンのアニメーション化

ほとんどの場合、キャラクターには複数のアニメーションがあります。これにはアイドルシーン、ランシーン、アクションシーンなどがあります。これらのアニメーションをすべて1つのスプライトシートにエクスポートできるように、特定の構造で作業をする必要があります。

次の2つの異なるワークフローを使用できます。

- ワークフロー 1：セパレートカット
- ワークフロー 2：カットマーカを使用した分離

---

## ワークフロー 1：セパレートカット

まず、ようなキャラクター名でカットファイルを作成します。 **Space Duck**これは、ゲームリグを作成またはインポートできるファイルです。 トップメニューで、ファイル > 新規バージョンとして保存の順に選択し、この新しいバージョンにアニメーションの名前をつけます。 例えば、 **Idle**などです。

同じキャラクターを使用して新しいアニメーションを作成する必要があるたびに、[新しいバージョンとして保存]を実行します。 最後に、次のようなものがあるかもしれません：

**カット：**スペースダック

**バージョン：**

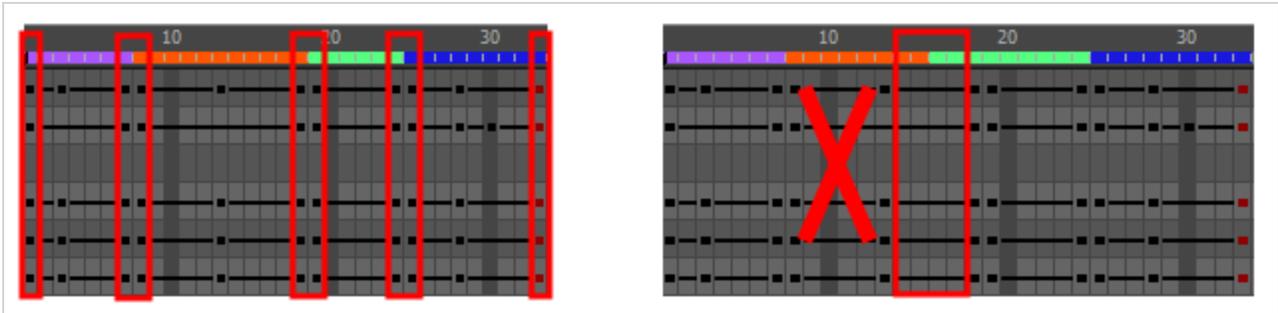
- 待機中
- 実行
- ジャンプ
- 撃つ

エクスポートスクリプトを実行すると、現在のカットからエクスポートフォルダに描画がエクスポートされます。 また、そのフォルダにすでにエクスポートされている他のカットバージョンがあるかどうかとも通知されます。 その場合は、スプライトシートを再コンパイルして、すべてのアニメーションのすべての描画を含めます。 これにより、描画を最大限に再利用できます。

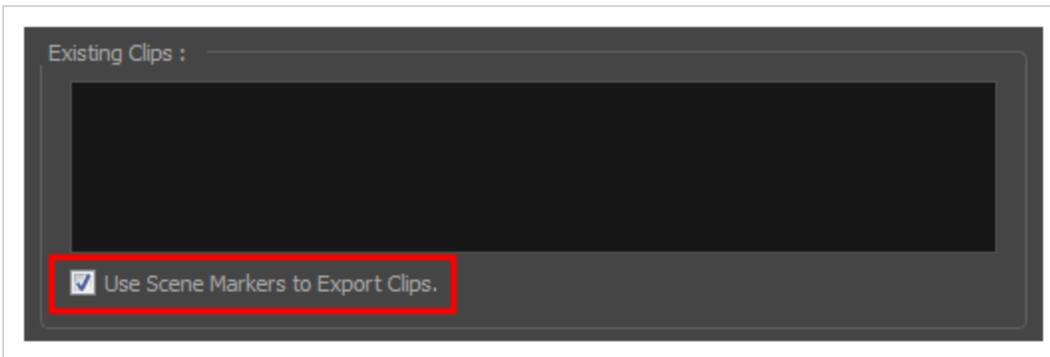
## ワークフロー 2：カットマーカを使用した分離

また、待機中、走る、ジャンプ、シュートなど、すべてのキャラクターアニメーションを1つのカットで1つずつ作成することもできます。次に、カットマーカを使用して、個々のアニメーションをマークおよび分離します。

個々のアニメーションコマ範囲をマークする場合は、キーフレームで開始および終了することを確認してください。補間された動きの途中で開始または終了するコマの範囲のカットマーカを作成しないでください。



スプライトシートをエクスポートするときは、[スプライトシートにエクスポート]ダイアログボックスでカットマーカを使用してクリップをエクスポートするオプションに必ずチェックを入れてください。これはデフォルトで有効になっています。



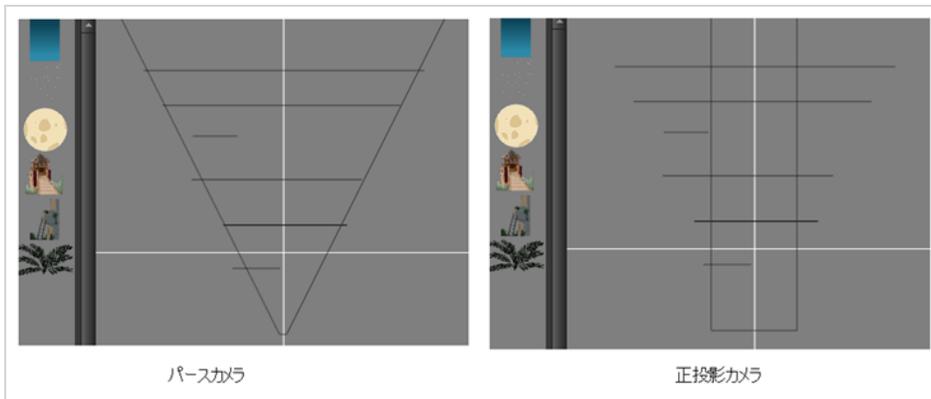
アニメーションクリップは.xml各アニメーションシーンを別々のカットから同じファイルの場所にエクスポートした場合と同じ方法で、ステージに分割されて記載されます。

## 正投影カメラについて

Harmonyでは、次の2種類のカメラを使用できます。

- パース
- 正投影

正投影カメラは、ゲームパイプラインに固有です。カメラの種類をパースから正投影に変更します。消失点のないカメラになります。これは、カメラビューにパースがないことを意味します。オブジェクトは、Z軸上で移動しても、サイズやスケールは変更されません。

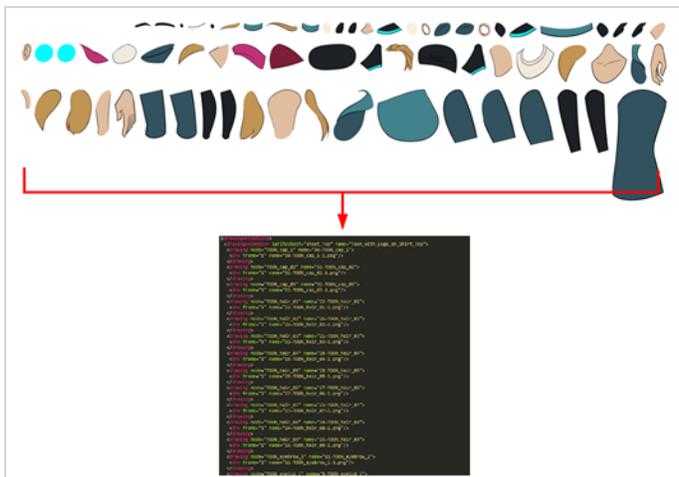


正投影カメラは、[カット設定]ダイアログボックスで設定できます。正投影カメラでカットを作成するには、デフォルトで、新しいカスタムカット解像度を作成する必要があります。



## 第3章： Unity へのエクスポートについて

カスタムゲームエンジンをお持ちの場合は、エクスポートされた データを入力し、通常どおりのプロセスに進みます。Harmonyまたは、規則に合わせてエクスポートスクリプトを変更することができます。



カットをエクスポートする前に：

- ディスプレイは必ず「すべて表示」ではなく、**表示**として設定してください。
- 必ずカットを保存してください。Harmonyはtvgファイルに基づいてエクスポートを行い、追加した未保存の更新内容はエクスポートされません。
- Harmonyカットを正方形の解像度に設定します（例：1024 x 1024）。カットを選択>カット設定を選択します。

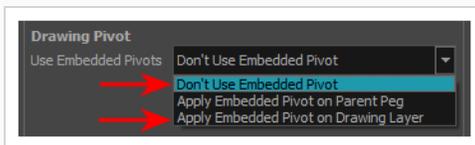
## アンカーの設定

Unity にエクスポートする前に、アンカーとして設定する内容を検討する必要があります。

アンカーは、キャラクターのアニメーション中にゲームエンジンで参照するキャラクター上の点です。

### アンカーの設定方法

1. 新しい描画レイヤーを作成し、作成するアンカーに従って名前を変更します。
2. 描画レイヤーの[レイヤープロパティ]で、ピボットオプションが埋め込みピボットを使用しないまたは**描画レイヤーに埋め込みピボットを適用**に設定されていることを確認します。[親ペグに埋め込みピボットを適用]はUnityにうまくトランジションせず、ピボット情報が失われます。



3. これが完了したら、任意のアニメーションツールを使用して、[高度なアニメーション]ツールバーの[回転]ツールなど、レイヤーのピボットポイントを再配置できます。ピボット座標は、描画レイヤーの[レイヤープロパティ]から確認できます。ピボットが再配置されていない場合、デフォルトの場所は Harmony カット (0,0) の中心です。
4. セットアップが完了したら描画レイヤーを選択し、[ゲーム]ツールバーの[アンカーを切り替え]ボタンをクリックします。🔗 複数選択を行い、複数の描画レイヤーに同時にアンカーを設定することもできます。

描画レイヤーは、[タイムライン]ビューと[ノード]ビューで赤で強調表示されます。これで、アンカーを使用してプロップを取り付けることができます。

## スプライトシートのエクスポート

「スプライトシートにエクスポート」ウィンドウは複数の解像度にエクスポートされ、定義したスプライト解像度の数に応じて、複数の.xmlファイルと1つまたは複数のスプライトシートを生成します。

これにより、同じキャラクターの異なるアニメーションが同じ名前に保存されます。たとえば、待機中、走る、ジャンプのアニメーションがある場合、これらはすべて同じアセット名を共有する必要があります。アニメーションの全体的なコレクションとして考えてください。内部には、その文字セットのすべてのアニメーションに再利用できる描画を持つさまざまな保存されたカットバージョンがあります。各カットバージョンは、リスト内のアイテムとして表示されます。

アニメーションをエクスポートすると、そのカットで使用されている描画のみがエクスポートされます。すべての描画は、最初に個別にエクスポートされ、次にスプライトシートにまとめられます。

複数のアニメーションを同じアセット名に保存した場合（すなわちスペースダック：実行、待機中）、スプライトシートを再度まとめ、そのフォルダー内のすべてのアニメーションのすべての描画を含めた新しいアニメーションファイルを作成しますが、同じスケルトンを再利用します。



### メモ

プログラマーは、によってエクスポートされたXMLデータは、後からより最適化されたバイナリデータ構造に変換できることにご興味があるかと思います。HarmonyXMLをバイナリ形式に変換するには、Xml2Binユーティリティを使用します。このユーティリティは、Toon Boom Harmonyソフトウェアを介して生成されたXMLデータ構造を、圧縮バイナリデータ構造に変換します。このユーティリティは、ゲームSDKの/HarmonyGameSDK/Plugins/で利用できます。

- Plugins/Mac : Mac OSX用にプリコンパイルされたバイナリ。
- Plugins/Windows¥x86 : Windows用にプリコンパイルされたバイナリ。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin : Xml2Binソース。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.mac.Xml2Bin.xcodeproj : Mac OSXのXコードプロジェクト。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.win32/Xml2Bin.sln : Windows向けVisual Studio 2010ソリューション。

他のゲームエンジンと統合する場合、データ構造を処理するC++コードは独自のコードで再利用および解析できます。

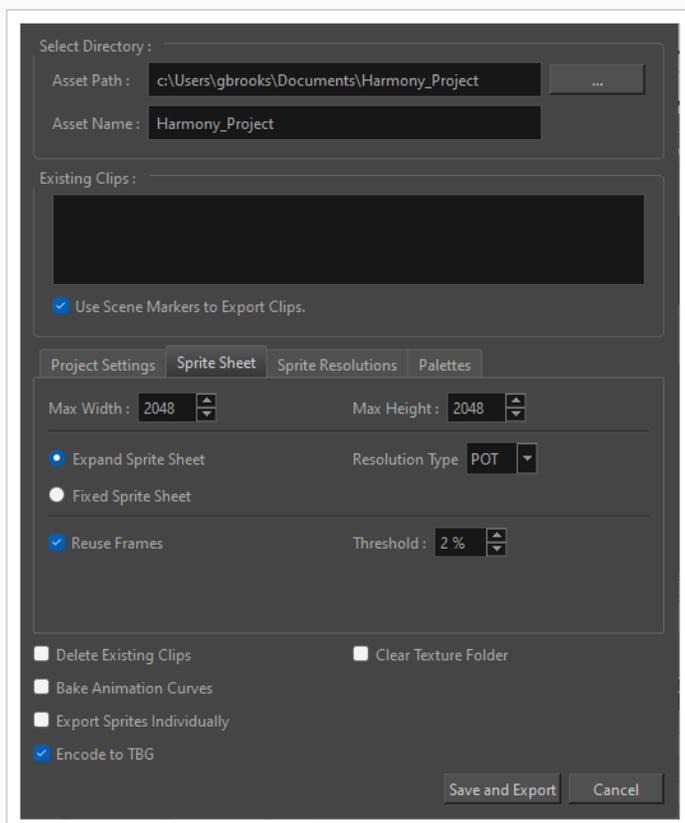
## スプライトシートのエクスポート方法

1. **Windows > ツールバー > ゲーム**を選択して、[ゲーム]ツールバーを追加します。



2. ゲームツールバーの[スプライトシートにエクスポート]  ボタンをクリックして、スクリプトを実行します。

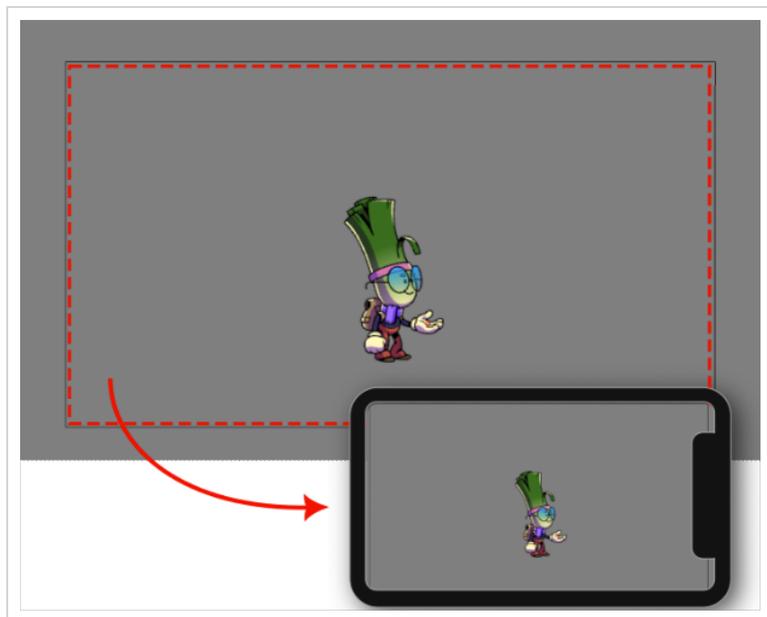
[スプライトシートにエクスポート]ウィンドウが開きます。



3. 自動的に更新する場合は、[保存パス]を Unity プロジェクトの正しいフォルダーに設定します。そうでない場合は、いずれかの場所に保存してから、アニメーションをUnityプロジェクトまたは使用する他のゲームエンジンに転送できます。
4. スプライトシートの環境設定については、
5. **エクスポート**をクリックします。

## スプライト解像度

スプライトのピクセル数は、ゲームの目標解像度に応じて設定できます。ターゲット解像度は、テレビやケータイの画面など、ターゲットとするハードウェアに基づいている場合があります。



[スプライトシートにエクスポート (XML)] ウィンドウの[解像度] タブで、指定した目標解像度ごとに新しいスプライトシートが作成されます。このウィンドウにアクセスするには、**Windows > ツールバー > ゲーム** を選択して、[ゲーム] ツールバーを有効にする必要があります。



スプライトのサイズは、Harmonyで作成されたときのキャラクターのサイズになることに注意してください。

## 複数のスプライト解像度を設定する方法

1. **Windows > ツールバー > ゲーム**を選択して、[ゲーム]ツールバーを有効にします。
2. [スプライトシートにエクスポート]を選択します。  [スプライトシート (XML) にエクスポート]ウィンドウが開きます。
3. [スプライトシートにエクスポート (XML) ]ダイアログで、[スプライト解像度]タブを選択します。このタブでは、現在のすべての目標解像度が一覧表示されます。
4. [追加] アイコンをクリックします。 **+** これにより、新しい目標解像度の新しい行が作成されます。
5. 新しい標的解像度の名前、幅、高さを入力します。

デフォルトの目標解像度は、HD、フルHD、およびQHDです。記載されている画面解像度は、ゲームをプレイできるターゲットデバイスを表しています。

目標解像度とハードウェアに応じて、このリストから解像度を手動で追加または削除できます。ゲームエンジンにインポートすると、レンダリングされたスプライトシートを変更することで、表示する解像度を選択できるようになります。



### メモ

- スプライトの解像度は、描画ストロークのスケールに影響されます。
- 描画やそのペグをスケーリングしても、出力スプライトの解像度には影響しません。
- スプライトの解像度を上げたい場合は、描画の子レイヤーを拡大縮小することができます。

## Bake\_Groupsの使用

Bake\_Groupは、スプライトシート内の単一の統合描画として合わせてエクスポートできる、描画ノードとエフェクトノードのグループです。

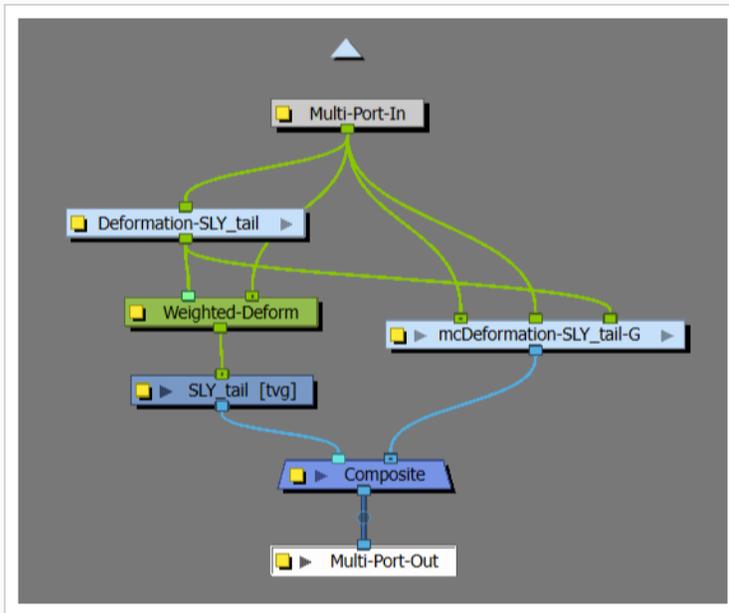
### ベイクされたグループを作成する方法

1. 合わせてベイクするレイヤーをグループ化します。 ノードビューからレイヤーをグループ化することをお勧めします。



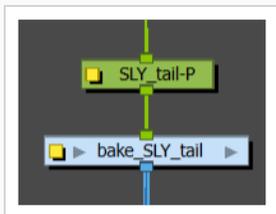
#### メモ

bake\_groupは、1つの入力と1つの出力のみを持つ必要があります。これは通常、エクスポートが正しく機能するために、複合ノードがbake\_group内に含まれていなければならないことを意味します。すべてのペグは最終的にグループ外の単一のペグから供給する必要があるため、描画を誘導するために必要なペグも考慮する必要があります。



2. タイムラインで、グループ名の先頭に「bake\_」の接頭辞を追加します。

「bake\_」の接頭辞を持つグループノードの外観については、以下の例を参照してください。「ヒップ」というラベルの付いた「グループ」は、ベイクを適用するために「bake\_hips」になる必要があります。



グループ名に「bake\_」の接頭辞が含まれている場合、グループは一意的な効果、動き、および描画の置換を含むコマについて分析され、一意の各コマはスプライトシートにスプライトとしてエクスポートされます。

これらの生成されたスプライトの新しい描画アニメーションが作成され、ゲームエンジンに提供されます。ゲームエンジンはこのグループ内のどの構造にもアクセスできず、代わりにグループ全体を、描画置換を伴う単一の描画であるかのように扱います。

## Easel JSへのエクスポート

[Easel JSにエクスポート]ウィンドウでは、アニメーションの画像シーンをフラット化できます。完全にリギングされた人形（パペット）またはアニメーションシーンを含む単一の描画レイヤーがある場合でも、結果は各コマのフラット化された出力であり、スプライトシートにグループ化されます。これによって使用したいツールやエフェクトモジュールにアクセスしやすくなるため、柔軟性と作業の自由度が高まります。ただし、アニメーションの長さ、複雑さ、エクスポートサイズによっては、ファイルが重くなる可能性があります。

### Easel JSにエクスポートする方法

1. **Windows > ツールバー > ゲーム**を選択して、[ゲーム]ツールバーを追加します。



2. ゲームツールバーの[Easel JSにエクスポート]  ボタンをクリックして、スクリプトを実行します。

[Easel JSにエクスポート]ウィンドウが開きます。



3. 自動的に更新する場合は、[保存パス]を Unity プロジェクトの正しいフォルダーに設定します。  
そうでない場合は、いずれかの場所に保存してから、アニメーションをUnityプロジェクトまたは使用する他のゲームエンジンに転送できます。
4. スプライトシート環境設定については、
5. **エクスポート**をクリックします。

## パレットバリエーション

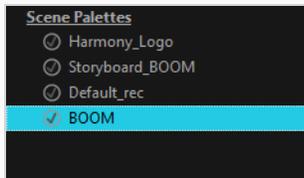
パレットが異なるキャラクターのバージョンを作成することは、キャラクターのカスタマイズをゲームの機能として追加する場合に重要です。Harmonyでは、キャラクターに対して複数のパレットを作成し、ゲーム内でさまざまな外観にすることができます。

Harmonyでは、キャラクターとそのパレットを作成したら、各パレットのSpriteシートにキャラクターをエクスポートできます。



### ゲームキャラクターのパレットバリエーションの作成方法

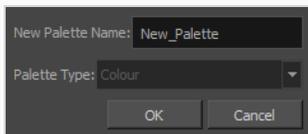
1. [カラー]ビューでパレットを選択します。



2. 次のいずれかの操作を行います。[カラー]メニューから**パレット>クローン**を選択するか、右クリックして**クローン**を選択します。

- パレットを右クリックし、**クローン**を選択します。
- [カラー]メニュー  を開き、**パレット>クローン**を選択します。

[パレットをクローン]ダイアログボックスが開きます

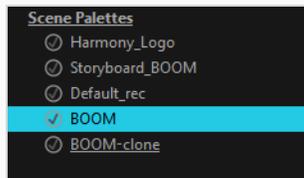


3. 新規パレット名フィールドに新しいパレット名を入力します。デフォルトでは、パレットの名前は元のパレット名の後に「-clone」が続きます。このフィールドの上のパレットリストには、選

択した場所にすでに存在するパレットが一覧されます。必ず別のパレットでまだ使用されていない名前を入力してください。

4. **OK**をクリックします。

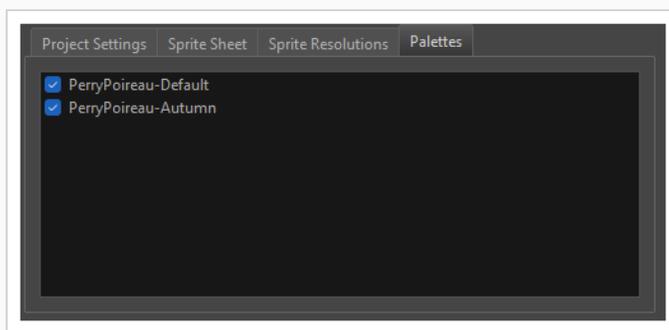
クローンされたパレットがパレットリストに表示されます。



すべてのカットパレットは、[スプライトシートにエクスポート]ウィンドウの[パレット]タブに表示されます。

## パレットバリエーションごとにスプライトシートをエクスポートする方法

1. [ゲーム]ツールバーを有効にするには、次のいずれかの操作を行います。
  - トップメニューで、Windows > ツールバー > ゲームを選択します。
  - 既存のツールバーを右クリックして、**ゲーム**を選択します。
2. [ゲーム]ツールバーで、[スプライトシートにエクスポート]  を選択します。
3. [パレット]タブを開きます。プロジェクトのすべてのパレットと、そのクローンが表示されます。



### メモ

パレットとそのクローンのみが表示されます。パレットの重複は表示されません。

4. エクスポートするパレットの名前を選択します。チェックのついてそれぞれのパレットは、キャラクターとともに独自のスプライトシートとしてエクスポートされます。
5. スプライトシートをエクスポートします。

キャラクターがUnityにインポートされると、レンダリングされたスプライトシートを変更することで、表示されているパレットを選択できるようになります。

## 第4章：Harmony Unity SDKについて

Toon Boomは、Harmonyからエクスポートされたデータをインポートするために必要なすべてのスクリプトを含むUnityアセットストアに、Unityパッケージを提供します。新しいUnityプロジェクトに解凍すると、Toon Boom Harmonyゲーミング SDK フォルダー内で次の構造を使用できます。

- ドキュメント（Unity 統合に関する開発者向けの情報）
- プラグイン（SDK ライブラリ）
- サンプル（デモカット、エクスポータースクリプト、SDK ソース、プレビューア）
- ランタイム（ゲームプレイ中に実行されるHarmonyスクリプト）
- エディター（エディターでのみ実行され、ビルドに表示されないHarmonyスクリプト）
- シェーダー（Harmony固有の機能を適切にレンダリングするシェーダーを提供）
- 素材（Harmony シェーダーを利用したビルド済み素材）

これらのパッケージは、Unity のパッケージマネージャーインターフェイスから手動で管理できます。開発者は、新しいバージョンのUnityにHarmonyスクリプトとの特定の非互換性がある場合に、パッケージをアップグレードおよびダウングレードすることができます。

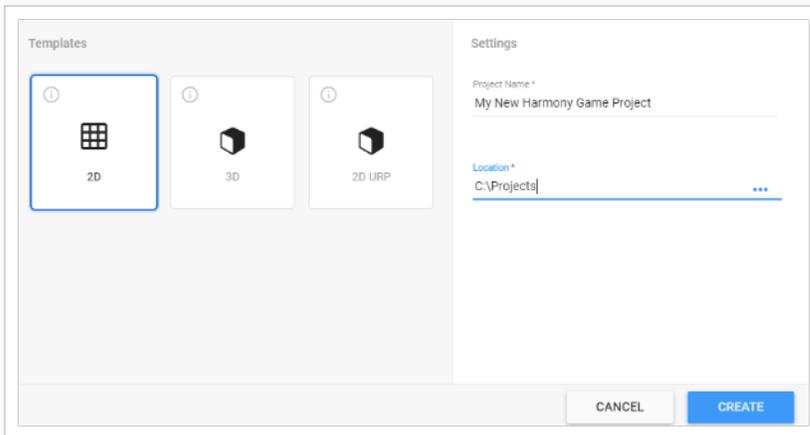
### Unity SDK プロジェクトの作成方法

1. 空のUnityゲームプロジェクトを作成します。空の2D Unityゲームプロジェクトは、2D レンダリングやPhysicsで作業するとき最も便利なので、作成することをお勧めします。



#### メモ

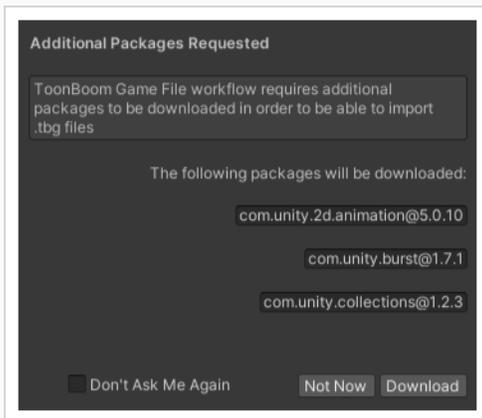
Unity 2020.3.36以降を使用していることを確認してください。以前のバージョンのUnityには、TBGファイルのインポートとレンダリングをサポートするために必要なすべての機能が揃っていません。



2. Unity アセットストアで入手できる Harmony ゲーム SDK をインストールします。SDK を手動でダウンロードした場合は、Unity プロジェクトが開いているときに .unitypackage ファイルをダブルクリックできます。

[要求された追加パッケージ]ポップアップが表示され、実行時に最適化された 2D アニメーションを容易にする、互換性のある Unity パッケージがあることを確認します。

3. **ダウンロード**をクリックします。



Unity が必要なパッケージのダウンロードとすべてのスクリプトのコンパイルを完了すると、プロジェクトは Harmony キャラクターのインポートを開始する準備が整います。

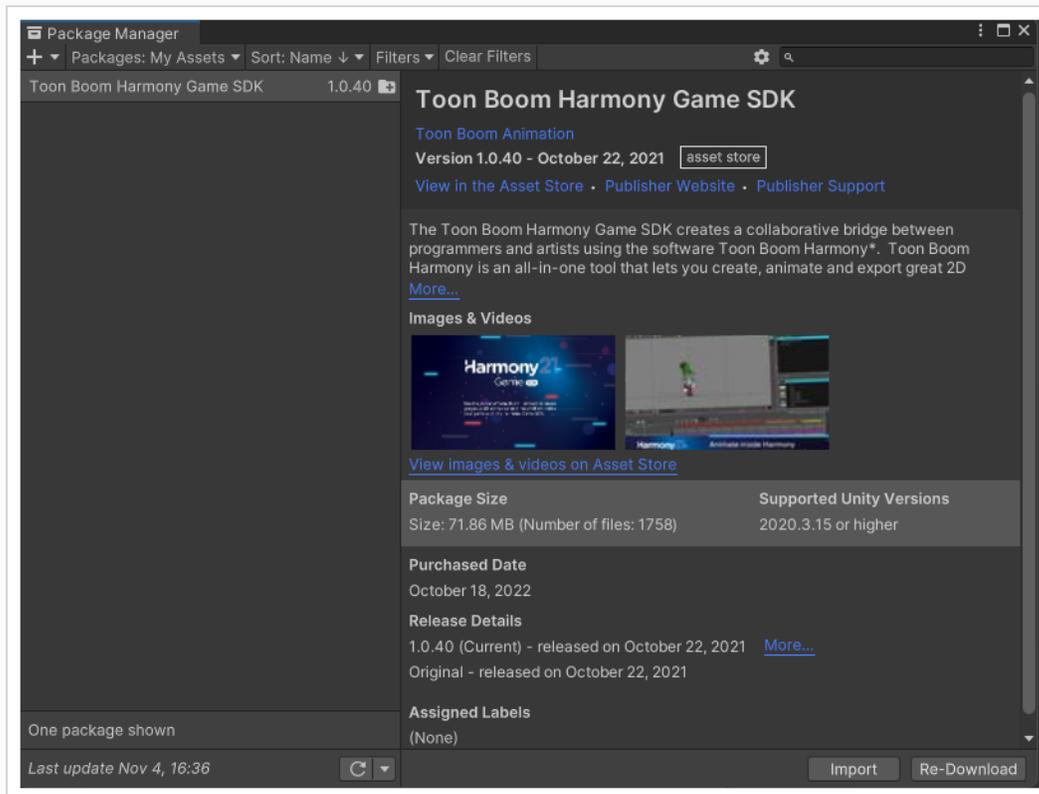


#### メモ

Toon Boom Harmony ゲーム SDK をインストールしたら、Harmony で作成されたすべての新しいプロジェクトに Harmony SDK パッケージをインポートする必要があります。

## パッケージをプロジェクトにインポートする方法

1. [アセットストア]を開くには、ウィンドウ > アセットストア を選択します。[アセットストア]タブが開きます。
2. アセットストアで、パッケージマネージャーを開くを選択します。[パッケージマネージャー]ウィンドウが開きます。



3. パッケージマネージャーの左上にある[パッケージ]ドロップダウンには4つのオプションがあります。
  - パッケージ：Unityレジストリ
  - パッケージ：プロジェクト内
  - パッケージ：マイアセット
  - パッケージ：ビルトイン
4. **パッケージの選択：マイアセット**を選択します。

ダウンロードしたアセットのリストが下のリストに表示されます。[パッケージマネージャー]タブの右に、選択したアセットに関する情報が表示されます。

5. Harmonyパッケージを選択したら、パッケージマネージャーの下部にある[インポート]を選択します。アセットのファイルを表示する[Unity パッケージをインポート]ウィンドウが表示されます。



6. **インポート**を選択します。

## Harmonyプレビューア

このセクションでは、Harmony プレビューアを使用して、Unity をインストールせずにアニメーションを表示する方法について説明します。

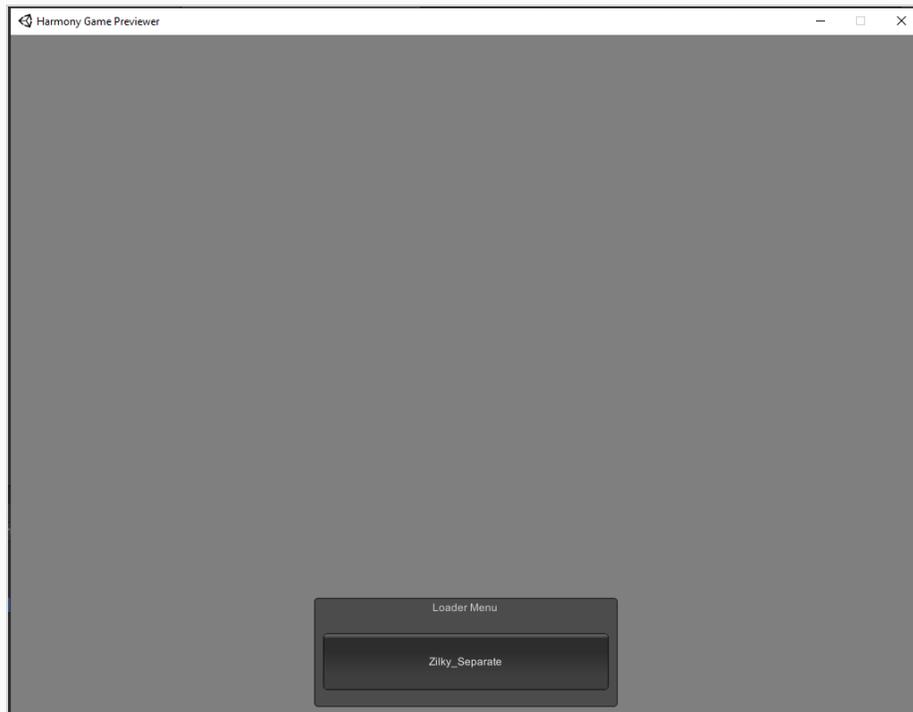
WindowsまたはmacOSのためのビルドは、解凍するだけのHarmonyゲームプレビューアフォルダー内にあります。

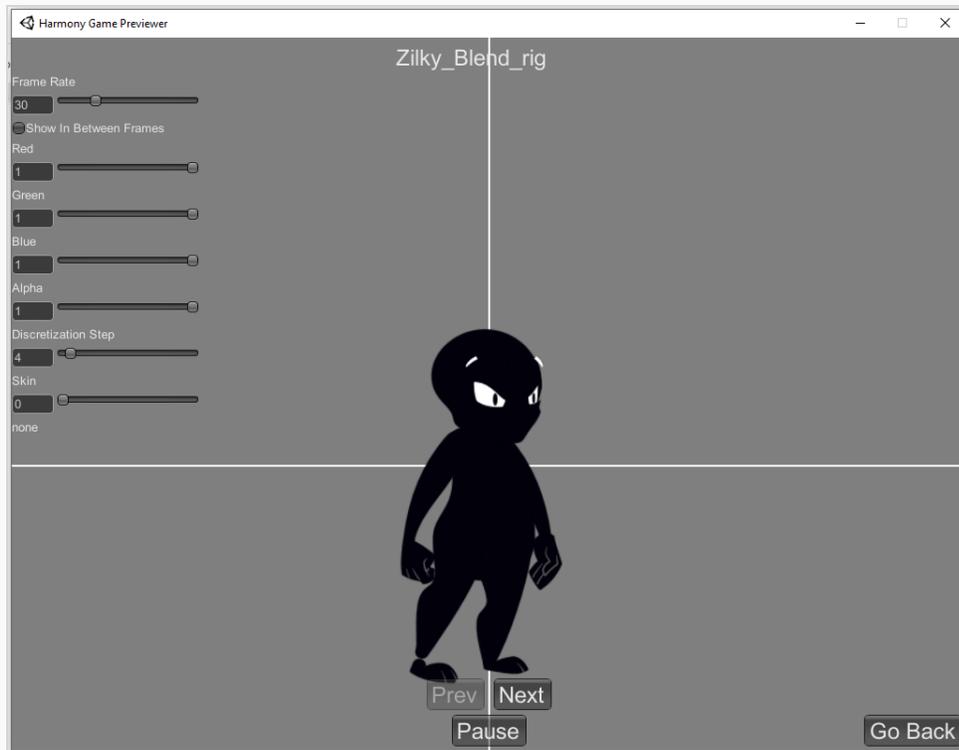


### メモ

独自のフォルダーに解凍することをお勧めします。Unityプロジェクト内のビルドを解凍しないでください。

## プレビューアの使い方





1. *Harmony*ゲームプレビューア.exeまたは*Harmony*ゲームプレビューア.app を実行します
2. ローダーメニューからロードするプロジェクトを選択します。これらのプロジェクトは、*Harmony*ゲームプレビューア.exeまたは*Harmony*ゲームプレビューア.app と同じディレクトリと*Harmony*リソースフォルダーからロードされます
3. [前へ/次へ]ボタンを使用してアニメーションをナビゲートします。
4. 左上のコントロールを使用して、フレームレート、色合い、離散化ステップ、スキン設定を調整します。
5. [戻る]ボタンをクリックして前のメニューに戻り、別のアニメーションをロードします。

## アセット操作

このセクションでは、Unity のコンポーネント、HarmonyメッシュとHarmonyレンダラーについて説明します。

### TBG ワークフロー

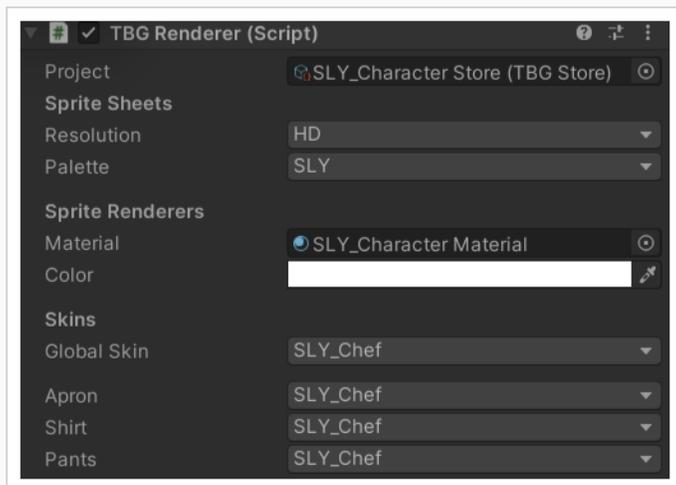
TBG ワークフローを通じてスプライトを Unity にインポートする場合、TBG ファイルのアセット操作オプションがインスペクタービューに表示されます。表示されるアセット操作オプションは、XML ワークフローに表示されるアセット操作オプションとは異なります。

### TBG アセットレンダラー

TBGファイルワークフローのHarmonyレンダラーは、TBGファイルのHarmonyアセットのアニメーションとレンダリングをコントロールし、実行時に変更したい重要な変数を公開するために使用されます。

### TBG ワークフローレンダリングアセットの操作方法

1. カット内のキャラクターを選択します。TBGレンダラー（スクリプト）セクションが右側の[インスペクター]タブに表示されます。



2. キャラクターのアセットをコントロールするには、次の設定をします。
  - プロジェクト：TBGStore アセットへの参照。このアセットはインポーターによって生成され、生成されたアニメーションクリップの対象外であるTBGプロジェクトデータが含まれています。
  - 解決：このキャラクターの表示に使用される解像度。これは、関連付けられたスプライトシート名と一致します。
  - パレット：このキャラクターを表示するために使用されるパレット。これは、関連付けられたスプライトシート名と一致します。

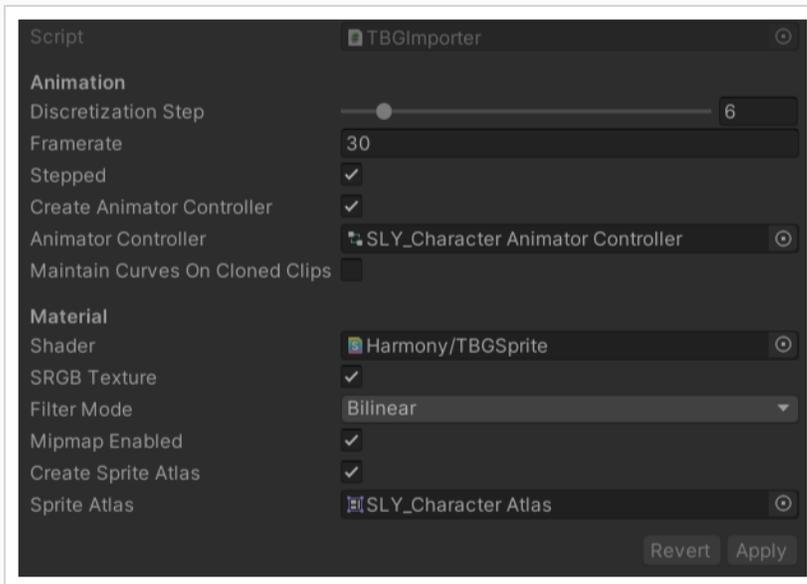
- 素材：このキャラクターをシェーディングするために使用される素材。ハードウェアスキニングは、SDKで提供されるデフォルトのアンリット素材です。
- 色：選択した色をアセットに乗算するオプション。
- スキン：Harmonyからエクスポートしたスキンをリアルタイムで変更できます。各グループには、スキンを変更するための独自のドロップダウンと、すべてのグループのスキンを一度に変更するためのグローバルドロップダウンがあります。

## TBG インポーター

TBGファイルがUnityにインポートされると、HarmonyアセットはUnityプレハブ、アニメーションクリップ、その他のアセットに変換されます。TBGインポーターでは、アセットとアニメーションの作成方法を指定できます。

### HarmonyTBGアセットをUnityにインポートするための設定をコントロールする方法

1. [プロジェクト]ビューで.tbgキャラクターファイルを選択します。インスペクタータブが開きます。
2. [インスペクター]ビューの上部で、[モデル]に対して選択をクリックします。[TBGインポーター]ビューが開き、TBGインポーターオプションが表示されます。



3. アニメーションのインポート方法をコントロールするには、次のように設定します。

## • アニメーション設定

- **離散化の手順**：スプライトがボーンによって変形されている場合に分割される列/行の数を表すスライダー
- **フレームレート**：UnityのアニメーションシステムがHarmonyコマを再生するために使用するアニメーションフレームレート。これは、Harmonyのフレームレートと一致する必要があります。
- **ステップモード**：チェックが付いていると、アニメーションは整数コマ間を通過するため、スムーズに補間されません。チェックが外れていると、アニメーションは整数コマ間で再生され、スムーズなアニメーションになります。
- **アニメーターコントローラーの作成**：チェックが付いていると、このインポーターは生成されたすべてのアニメーションクリップを含む新しいアニメーターコントローラーを自動的に作成し、[プロジェクト]ビューのプレハブの横にアセットを配置します。
- **アニメーターコントローラー**：.tbgファイルに変更が加えられると、参照されているアニメーターコントローラーが新しいアニメーションクリップで更新されます。
- **クローンクリップのカーブを維持する**：チェックが付いていると、.tbgファイル内のクリップサブアセットから複製された（およびアニメーターコントローラーで参照される）カーブは、更新された.tbgファイルの新しいデータでオーバーライドされます。

## • 素材設定

- **シェーダー**：プレハブが作成されると、すべてのスプライトレンダラーはこのシェーダーを使用して生成された素材を参照します。
- **SRGB テクスチャ**：オンにすると、生成されたテクスチャのガンマ色空間が線形色空間レンダリングにマッピングされます。
- **フィルタモード**：テクスチャの隣接するピクセルの補間方法。
- **ミップマップ有効**：低解像度のテクスチャを生成して、さらに遠くにレンダリングします。
- **スプライトアトラスを作成する**：オンにすると、このインポーターは生成されたすべてのスプライトを含む新しいスプライトアトラスを自動的に作成し、プロジェクトビューのプレハブの横にアセットを配置します。
- **スプライトアトラス**：.tbgファイルに変更が加えられると、参照されるスプライトアトラスが新しいスプライトで更新されます。

## XML ワークフロー

XMLワークフローを介してスプライトをUnityにインポートする場合、XMLファイルのアセット操作オプションがインスペクタービューに表示されます。表示されるアセット操作オプションは、TBGワークフロー

に表示されるアセット操作オプションとは異なります。

## Harmonyメッシュ

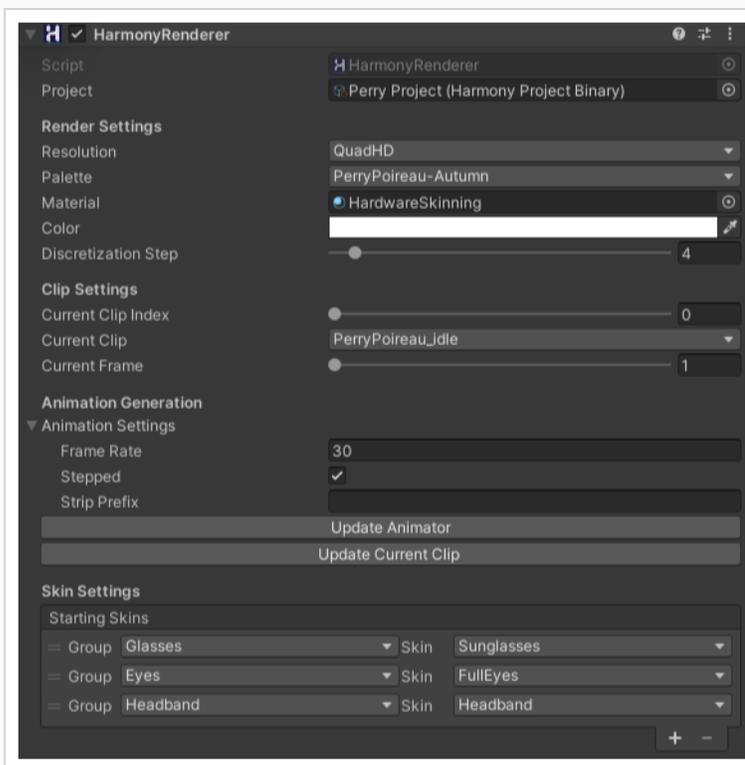
Harmonyメッシュコンポーネントは、アニメーションで公開された描画に基づいてアセットメッシュを作成するために、Harmonyレンダラーによって使用されます。

## XML アセットレンダラー

XMLワークフローのHarmonyレンダラーは、XMLファイルのHarmonyアセットのアニメーションとレンダリングをコントロールし、実行時に変更したい重要な変数を公開するために使用されます。

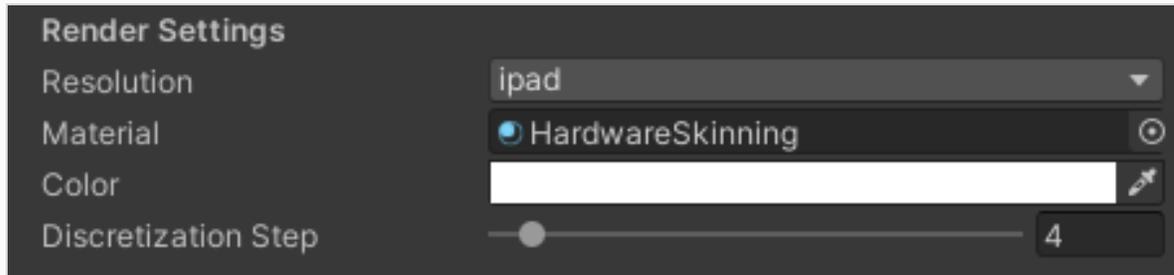
### XML ワークフローレンダリングアセットの操作方法

1. カット内のキャラクターを選択
2. [インスペクター]ビューで、ビューの下部にある[コンポーネントを追加]を選択します。Harmonyレンダラーセクションが表示されます。



3. キャラクターのアセットをコントロールするには、次の設定をします。
  - プロジェクト： Harmonyプロジェクトバイナリアセットへの参照。 このアセットはインポーターによって生成され、効率的なバイナリ形式でHarmonyプロジェクトデータが含まれています。

- **レンダリング設定**：レンダリング設定は、レンダリングに使用する解像度、シェーダー、または変形の品質を変更することにより、アセットを変更するために使用されます。

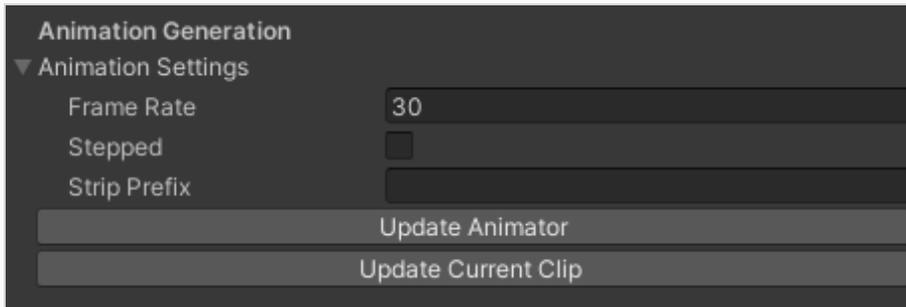


- **解決**：このキャラクターの表示に使用される解像度。これは、関連付けられたスプライトシート名と一致します。
- **パレット**：このキャラクターを表示するために使用されるパレット。これは、関連付けられたスプライトシート名と一致します。
- **素材**：このキャラクターをシェーディングするために使用される素材。ハードウェアスキニングは、SDKで提供されるデフォルトのアンリット素材です。
- **色**：選択した色をアセットに乗算するオプション。
- **離散化の手順**：すべての変形描画の配置を定義するスライダ。
- **クリップ設定**：クリップ設定は、アニメーションウィンドウを開かずにアニメーションを表示するために使用されます。これは、エディターモードでゲームをプレイするときに、アニメーターが[クリップ設定]内の設定を上書きする場合にのみプロジェクトに影響します。



- **現在のクリップインデックス**：表示する数値クリップ インデックス。アニメーションウィンドウでアニメーション化できます。
- **現在のクリップ**：表示するアニメーションを選択するためのドロップダウン。現在のクリップインデックスと一致します。
- **現在のコマ**：現在表示されているコマ番号。
- **アニメーション設定**：これらの設定は、アセット設定のインポートと同じです。これは、Harmony内でアセットの変更をインポートまたは再インポートするときに便利です。変更が完了

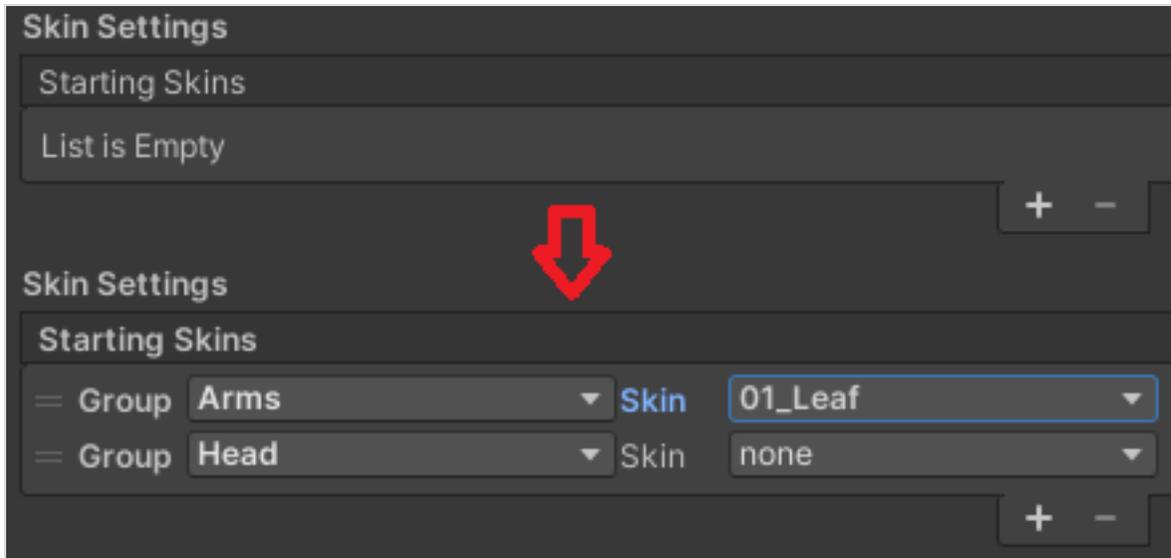
したら、[アニメーターを更新]または[現在のクリップを更新]をクリックして、変更内容を保存します。



- フレームレート：Unity のアニメーションシステムがHarmonyコマを再生するために使用するアニメーションフレームレート。これは、Harmonyのフレームレートと一致する必要があります。
- ステップモード：チェックが付いていると、アニメーションは整数コマ間を通過するため、スムーズに補間されません。チェックが外れていると、アニメーションは整数コマ間で再生され、スムーズなアニメーションになります。
- ストリッププレフィックス：これは、アニメーション名のプレフィックスを削除するために使用されます。例えば、「Wishling\_Idle」を「Idle」にする場合は、接頭辞を「Wishling\_」に設定します。これは、ゲームのアニメーション名がすべてのキャラクターにおいて同じである場合に便利です。
- アニメーターの更新：このボタンをクリックすると、このキャラクターのアニメーションセット全体がHarmonyプロジェクトの内容と一致するように更新されます。これにより、入力した新しい設定で既存のアニメーションが更新され、不足しているアニメーションが追加されます。
- **現在のクリップを更新**：これにより、現在選択されているクリップのみが更新されます。

## スキン設定

スキン設定を使用すると、Harmonyからエクスポートしたスキンをリアルタイムで変更することができます。スキンを変更するには、最初に+ボタンを押してグループを追加する必要があります。Harmonyエクスポートと同じ数のグループを追加する必要があります。グループがない場合は、グループを1つ追加して[すべて]のままにします。変更するグループを定義してから、どのスキンをどのグループに割り当てるかを定義できます。その後、任意のグループを選択して-ボタンを押すと、グループを削除できます。



### XML ワークフローでスキンを変更する方法

1. + ボタンをクリックしてグループを追加します。
2. グループを変更し、スキンをグループに割り当てます。
3. グループを削除するには、いずれかのグループを選択して- ボタンをクリックします。

## サンプル Unity プロジェクトについて

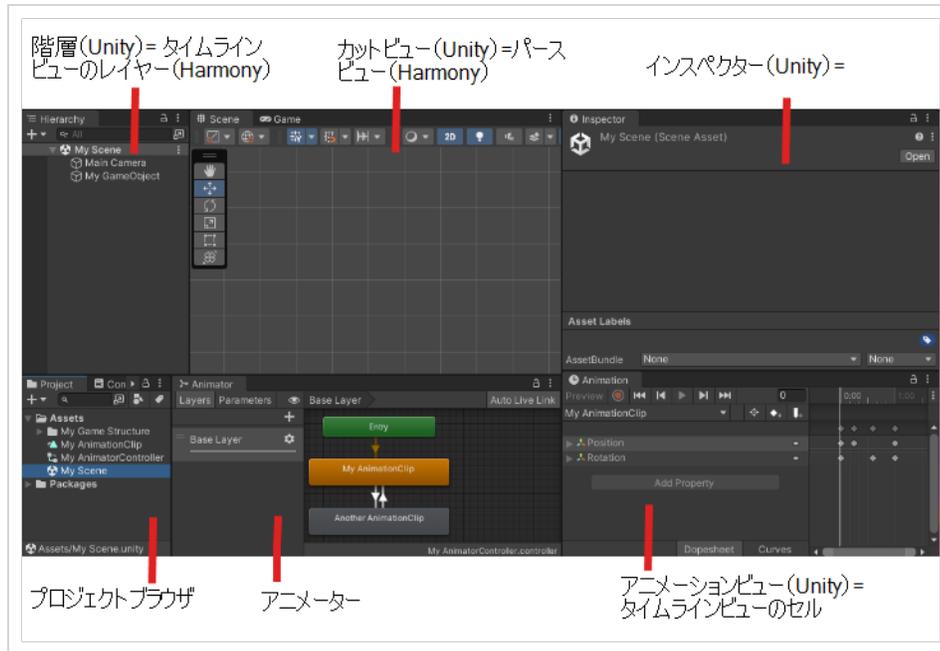
には、からエクスポートされたデータをインポートするために必要なすべてのスクリプトを含む、サンプル Unity プロジェクトが含まれています。Toon Boom Harmonyこのプロジェクト内には、次のフォルダーを含むアセットフォルダーがあります。

- プラグイン
- カット（デモカットとプレビューア）
- スクリプト（すべての Harmony スクリプト）
- ストリーミングアセット（すべての Harmony カットファイルをここにエクスポートする）

スクリプトを構成することによって、アニメーションを StreamingAssets フォルダーに自動的にエクスポートできます。Harmonyこのようにして、Unity は最新のアセットがフォルダーに表示されると、すぐに動的にロードします。エクスポートされた Harmony データをこの場所に手動で配置することもできます。

## Unity インターフェースについて

HarmonyのUnity インターフェースの主なコンポーネントとそれに相当するコンポーネントは、次の通りです。



Unity	Harmony	内容
カットビュー	パースビュー	ここではカットを設定し、環境、プレイヤー、カメラ、敵、その他すべてのゲームオブジェクトを選択して配置します。
ゲームビュー	カメラビュー	ゲーム内のカメラからレンダリングされたビュー。これは、最終版の公開されたゲームを代表します。
インスペクター	レイヤープロパティ	接続されている全てのコンポーネントとそのプロパティを含む、選択したゲームオブジェクトの詳細情報を表示します。
階層	タイムラインビューのレイヤー	カット内の要素の階層を表示し、さまざまなゲームオブジェクトの親子関係を設定することができます。
アニメーション	タイムラインビューのセル	アニメーション化される変換ごとにアニメーションのキーフレームを表示します。

アニメーター		状態と遷移のグラフを表示し、キャラクターがゲームイベントやプレイヤーのアクションにどのように反応するかを設定できます
プロジェクトブラウザ	---	プロジェクトの資産にアクセスし、管理することができます。

Unity が相互作用するいくつかのデータ型と、Harmony での同等のデータ型を次に示します。

Unity	Harmony	内容
スプライト	描画	スプライトレンダラーに表示できる単一の画像。
スプライトレンダラー	レイヤー	一度に 1 つのスプライトを表示します。アニメーション中にスプライトを交換して、キャラクターの体の部位の動きを解釈できます。
キャラクター	ペグ	キャラクターの体の階層を形成するために使用されます。拡大縮小、回転、または変換して、キャラクターの体の部位の動きを解釈することができます。スキューは、スケールと回転の階層から派生する必要があります。
ゲームオブジェクト		常に変換とペアになっています。ゲームオブジェクトは、ゲームプレイを表示およびシミュレーションできるコンポーネントをホストします。SDK は、ゲームオブジェクトで実行されるカスタムビヘイビアを提供し、Harmony 固有のビジュアルを表示します。
アセット		ビジュアル、Physics、ゲームプレイ、効果音に使用される Unity プロジェクトのコアデータ。通常ではこれらはファイルですが、より大きなファイル内のサブアセットとしてホストできます。テクスチャ、アニメーションクリップ、スプライト、アニメーターコントローラー、オーディオクリップ、プレハブはすべてアセットです。

## HarmonyファイルをUnityにインポート

Harmonyでのアートワークとサイクリング可能なアニメーションキャラクターの動きを作成したら、ゲーム統合のためにそれらをUnityにインポートします。

### Harmony ファイルを Unity に直接インポートする方法

1. Unity 2D プロジェクトを作成します。
2. SDK Harmony アセットを取り込むには、プロジェクトをプロジェクトにインポートする必要があります、次のいずれかの操作を行います。
  - Unityアセットストア ([assetstore.unity3d.com/en/#!/content/31211](https://assetstore.unity3d.com/en/#!/content/31211)) にアクセスし、Harmonyゲーム SDK ファイルをコンピュータに保存します。次に、以下からプロジェクトにインポートします。 **トップメニュー>アセット>インポートパッケージ>カスタムパッケージ**。
  - Unity アセットストアでHarmonyゲームSDKを検索し、パッケージをダウンロードしてインポートします。アセットストアには以下からアクセスできます。 **トップメニュー>ウィンドウ>アセットストア**



#### メモ

新しいUnityプロジェクトにつき、Harmonyインポート処理作業を行うために必要なファイルやフォルダーが含まれているため、ステップ2を繰り返す必要があります。

HarmonyからUnityへのデータのエクスポートとインポートには、主に2つのワークフローがあります。

#### • TBGファイル/ TBGレンダラー

これは、次の場合に推奨されるワークフローです。

- Unityのみを使用しており、Unityのやり方に最もネイティブな体験を望んでいます。
- より少ないステップでより速くインポートしたい。
- トランスフォームとスプライトレンダラーの有意義な階層を作成し、調整して遊びたい。これにより、スクリプトの経験が軽い、または全く経験がない方は、キャラクター固有の手続き的効果を追加できます。
- スプライトアトラスとアニメーターコントローラーを組み合わせた Unity のスプライトメッシュを、レイヤー化とトランジションブレンディングと組み合わせて活用することをお勧めします。

これにより、既存のパッケージや単純なスクリプトをインバースキネマティクス (IK) 、ジグル Physics、ラグドールなどに使用する可能性が広がります。

#### • XML フォルダー / Harmonyレンダラー

これは、次の場合に推奨されるワークフローです。

- Unity を使用してプロトタイプを作成し、他のゲームエンジンへのモビリティを求めている。他のエンジンに移植可能な C++ ライブラリーを呼び出すため、XML ワークフローは転送できません。
- カスタムアセットストリーミングソリューションで遊ぶためのスペースが多く必要で、カスタムの高性能手続き的効果を作成するための出発点が必要です。

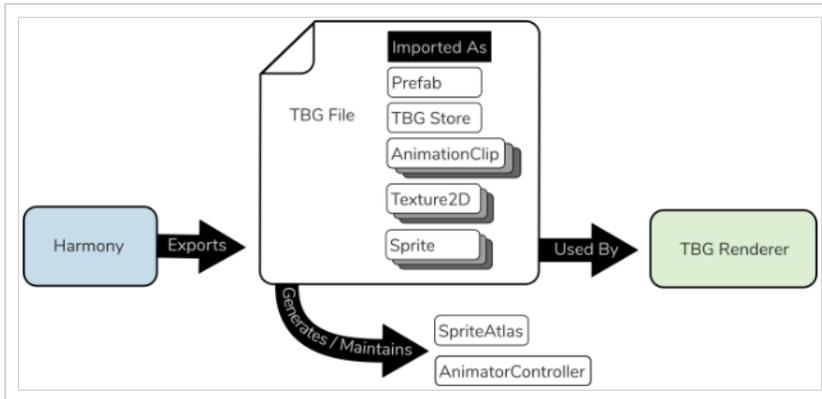


#### メモ

- 必要に応じて、後からワークフローを切り替えることができます。
- どちらのシステムもデフォルトでアニメーターコントローラーを使用してキャラクターのアニメーションを選択して再生するため、キャラクターのアニメーターとのみ対話するスクリプトは同様に機能します。ただし、スキンや解像度を変更したり、手続き的効果を提供したりするスクリプトは、ワークフローごとに書き直すか、カスタムで作成する必要があります。
- 一般的に、HarmonyレンダラーがデフォルトでTBGレンダラーで利用可能な機能のサブセットを持っているため、HarmonyレンダラーからTBGレンダラーへの移行はよりうまく運びます。別の設定または別のワークフローで再エクスポートする必要がある場合は、いつでも元のHarmony .xstageプロジェクトへのアクセスを維持することをお勧めします。

## Unity での TBG ファイルワークフロー

TBG (Toon Boom Gaming) ファイルワークフローでは、Unity はファイルから新しいテクスチャ、スプライト、プロジェクトデータ、プレハブを取得するために、Harmony からエクスポートされた TBG ファイルを参照できる新しいアセットとして即時に解釈します。結果のプレハブをUnityカットにドラッグするだけですぐに表示し、アニメーション化することができます。TBGファイルがHarmonyの新規エクスポートから書きされると、TBGインポーターは既存のプレハブ、スプライト、テクスチャを更新する再インポートプロセスを自動的に実行します。これにより、アーティストはキャラクターをすばやく反復処理し、ゲーム内ですぐに結果を確認できます。



結果として得られる TBG レンダラープレハブには、Unity に適したリグでキャラクターを表示するためのトランスフォームとスプライトレンダラーの完全な階層が含まれています。さらに、アニメーションの進行中にスプライトが交換され、カッターが関連するスプライトレンダラーに適切に適用されるようにする TBG レンダースクリプトが含まれています。



### メモ

TBGファイルで使用されているXML形式は新しいバージョンであり、より複雑なキャラクターのXMLフォルダーワークフローとは互換性がない場合があります。元のHarmonyファイルを保持し、後から異なる設定で再エクスポートすることをお勧めします。

### TBG ファイルを Unity にインポートする方法

1. Harmonyで、[ゲーム]ツールバーの[スプライトシートにエクスポート]  ボタンを探します。
2. [スプライトシートにエクスポート]ウィンドウで、**TBGとしてエンコード**にチェックを入れます。
3. アセットパスには、キャラクターを保存するファイルパスを入力します。
4. 保存してエクスポートを選択します。エクスポートプロセスを完了し、新しいTBGファイルを生成するロードバーが表示されます。

5. Unity プロジェクトを開きます。Unity は自動的に TBG をプレハブとしてロードします。



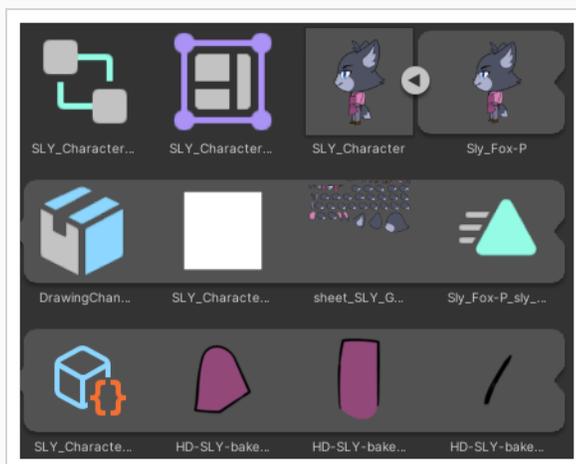
### メモ

キャラクターがプレハブとして表示されない場合は、次のことを確認してください。

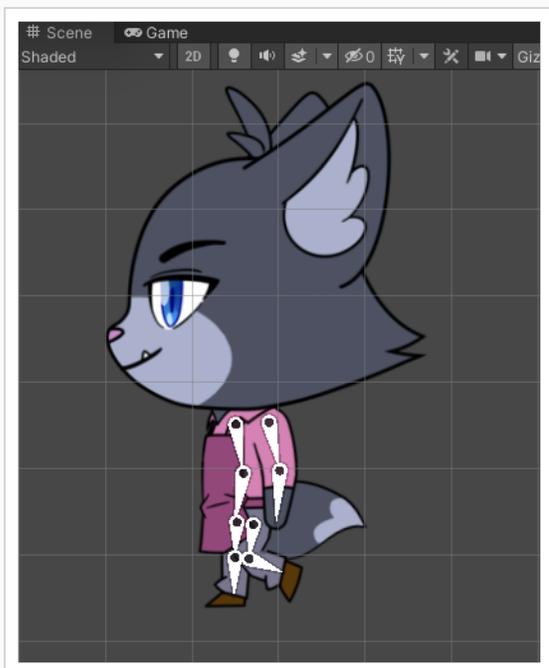
- Unity プロジェクト用に HarmonySDK がインストールされていること。新しいプロジェクトを作成するたびにインストールする必要があります。
- キャラクターの露出を超えて伸びるカットマーカがありません。

Unity プロジェクトには、次のものがが必要です。

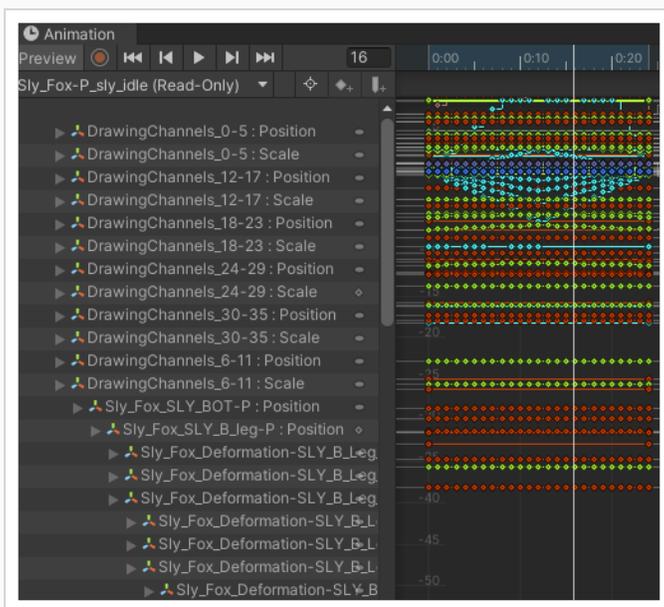
- テクスチャ、スプライト、TBG ストアデータ、アニメーションクリップのサブアセット。
- ファイルのスプライトアトラスと、TBGプレハブの隣にあるアニメーターコントローラーの2つの新しいアセット。



TBGプレハブを階層にドラッグできるようになり、そこで表示され、対話可能になります。



階層でプレハブを選択すると、[アニメーションビュー]からキャラクターのアニメーションを表示できます。

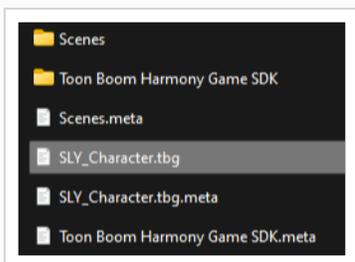


小さな矢印をクリックして[階層]ビューでプレハブを開くと、Harmonyのタイムラインに表示されているものと非常によく似たキャラクターの階層を探索できます。

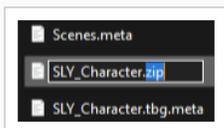


## TBGファイルの内容を手動で検査する方法

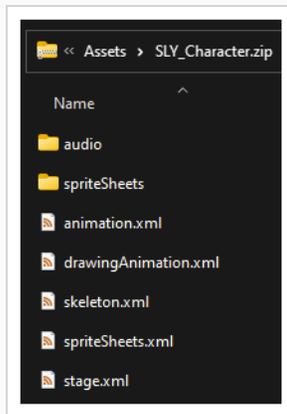
1. コンピュータのファイルエクスプローラーでTBGファイルを検索



2. Windows または Mac を使用している場合は、ファイル拡張子が .tbg で認識されないため、ファイル拡張子の名前を変更します。



名前を変更すると、WindowsとMacでは.zipファイルの内容を調べることができるため、SpriteSheetsが適切なサイズであること、またはデータファイルに期待どおりの内容が含まれていることを確認できます。



### メモ

Unity に問題が発生しないようにするには、Unity での作業を再開する前に、ファイル拡張子を .tbg に戻すことができます。通常、この検査プロセスは、Unity がインストールされていないコンピューターを使用している場合にのみ実行する必要があります。

## TBG レンダラプレハブへのインバースキネマティクス (IK) の追加

Unity 独自の2Dインバースキネマティクス (IK) 機能を利用して、キャラクターの手足に手続きのターゲットを追加できます。このプロセスには、TBG ワークフローのサンプルカットのSLY\_Characterを使用できます。

### TBG レンダラプレハブにインバースキネマティクス (IK) を追加する方法

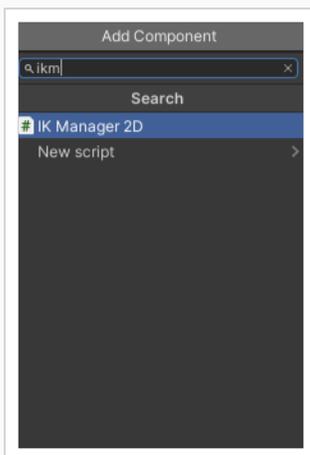
1. [プロジェクト]ビューから[カット]ビューにSLY\_Characterを見つけてドラッグし、Fキーを押してビュー内のキャラクターをコマに収めます。



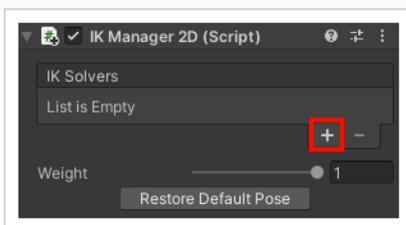
### メモ

階層でSLY\_Characterが選択されていることを確認します。

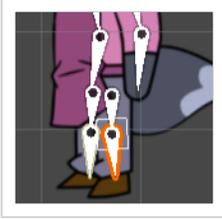
2. [インスペクター]ビューで、IK マネージャー 2D コンポーネントを追加します。



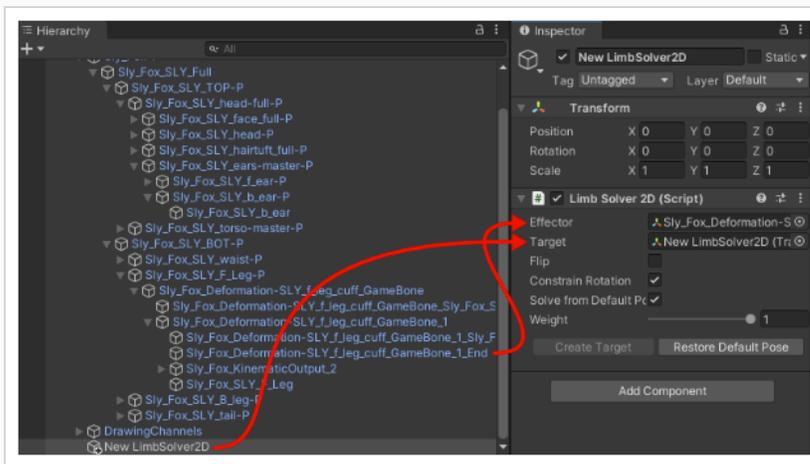
3. IK マネージャー 2D コンポーネントで、[+]ボタンをクリックして新しいIK ソルバーを追加し、**リム**を選択します。



4. スライの脚の右下のボーンをクリックして、階層内の位置を表示します。名前に「...\_GameBone\_1\_end」を含む、兄弟ゲームオブジェクトを使用します。



5. 「New LimbSolver2D」ゲームオブジェクトを選択した状態で、「New LimbSolver2D」をLimbSolver2Dスクリプトのターゲットスロットにドラッグし、「\_leg\_cuff\_GameBone\_1\_end」ゲームオブジェクトをエフェクタースロットにドラッグします。



これが完了したら、新しいLimbSolver2D変換の周りを移動して、ターゲットに一致するようにリムが曲がっているのを確認できます。



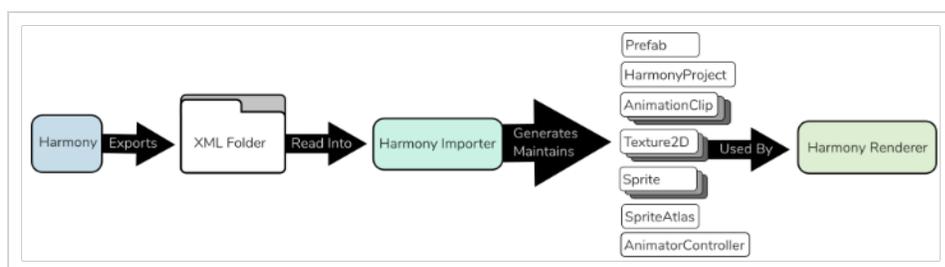
膝が曲がる方向を切り替えるには、リムソルバー2Dコンポーネントの「反転」チェックボックスをクリックします。

## Unity での XML フォルダーワークフロー

XMLフォルダーワークフローでは、開発者が独自のHarmonyインポーターアセットを作成して、HarmonyからエクスポートされたXMLフォルダーを参照することができます。このインポーターを使用して、Harmonyプロジェクトアセットと新しいHarmonyレンダラープレハブを生成できます。

再エクスポートされたXMLファイルごとに、開発者はインポートパスとエクスポートパスを手動で設定し、Harmonyインポーターアセットからインポートプロセスを実行する必要があります。

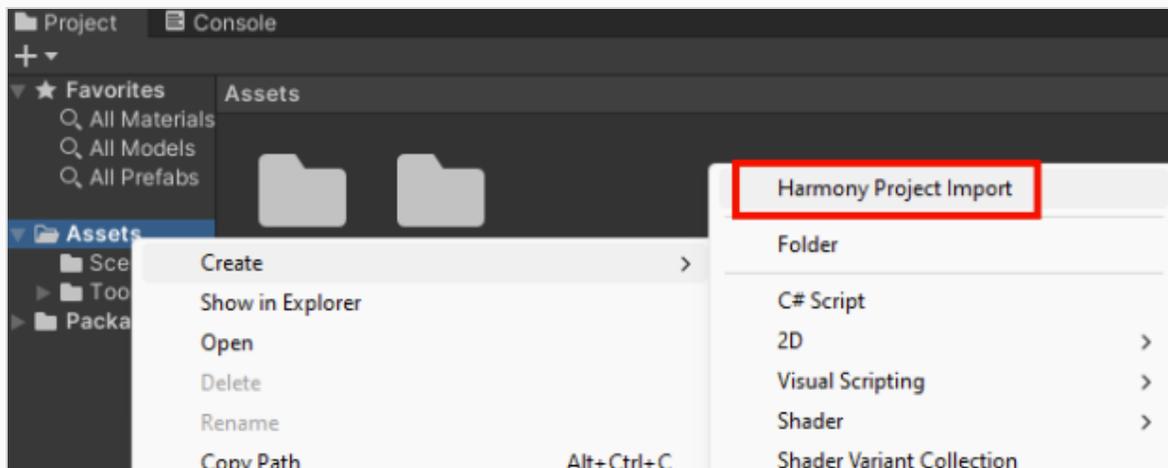
このワークフローは、XMLデータが配置できる場所に関して柔軟性を提供します。例えば、Unityプロジェクトの内部または外部に存在できます。ただし、このワークフローの欠点は、より多くの手順が含まれるため、より多くの間違いが発生する可能性があることです。たとえば、ファイルを移動させるときは、TBGインポーターのファイルパスを手動で更新する必要があります。



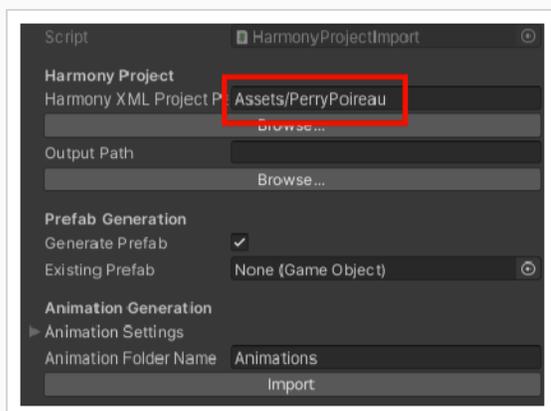
結果として得られるHarmonyレンダラープレハブは、C++プラグイン内でメッシュを生成することにより、高性能のキャラクターレンダリングを可能にします。これは、複雑なカスタムレンダリング機能が不要な、キャラクターの集まりに適しています。

### XML フォルダーを Unity にインポートする方法

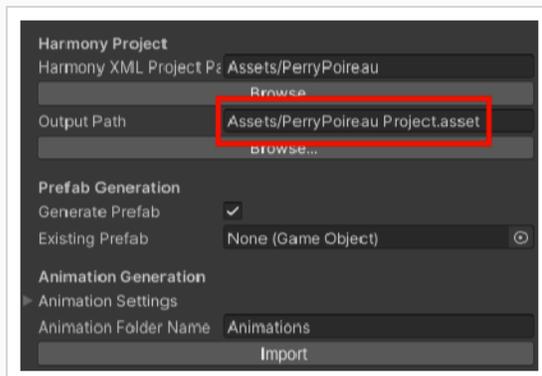
1. Harmonyの中で、[ゲーム]ツールバーの[スプライトシートにエクスポート]  ボタンを探します。
2. 保存してエクスポートを選択します。[スプライトシートにエクスポート]ダイアログで[TBGとしてエクスポート]オプションがオフになっていることを確認します。
3. Unityプロジェクトのアセットフォルダー内のフォルダーにXMLを保存します。
4. Unityで次のいずれかを実行して、新しいHarmonyインポーターアセットを作成します。
  - トップメニューの[アセット]ドロップダウンで、**作成 > Harmonyプロジェクトをインポート**を選択します。
  - [プロジェクト]ビューでアセットフォルダーを右クリックし、**作成 > Harmonyプロジェクトをインポート**を選択します。



5. 将来的な混乱を避けるために、Harmonyインポーターに「インポーター」の接尾辞を付けて名付けます。
6. エクスポートされたフォルダの相対パスをコピーし、新しいHarmonyインポーターにペーストします。

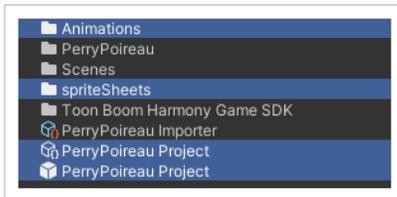


7. Harmonyインポーターアセットの相対パスをコピーし、Harmonyインポーターにペーストし、パスの「インポーター」を「プロジェクト」に置き換えます。

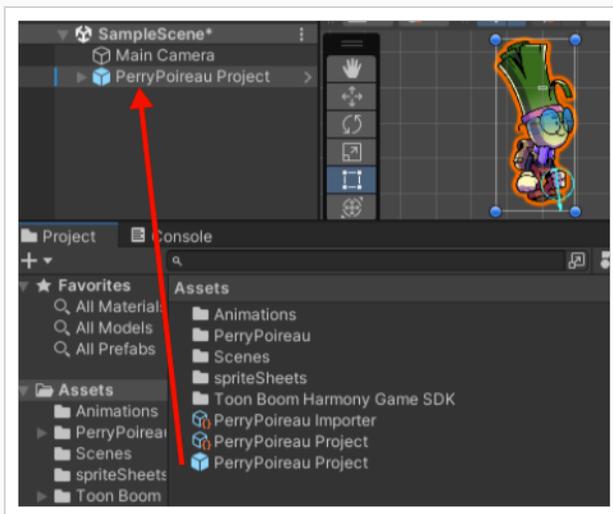


## 8. インポートをクリックします。

新しいHarmonyプロジェクトアセットと、Harmonyプロジェクトを参照するHarmonyレンダラーを備えた新しいプレハブが作成されます。



プレハブアセットを[プロジェクト]ビューから[階層]ビューと[カット]ビューにドラッグして、ゲーム内のキャラクターを表示できます

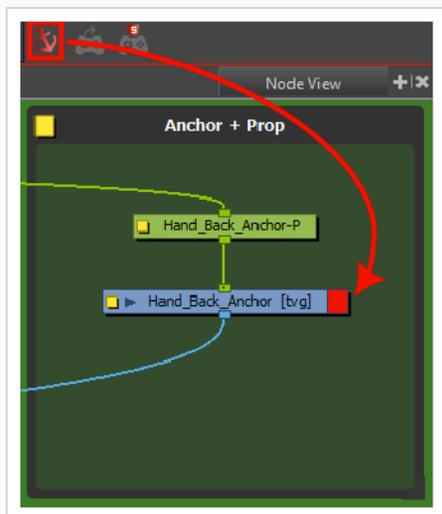


## アンカーのインポート

Harmony では、アンカーは Unity のゲームプレイスクリプトに対してどのペグや描画が関心のある位置であるかを示す方法を提供します。任意の描画またはペグにアンカーを割り当てることができます。

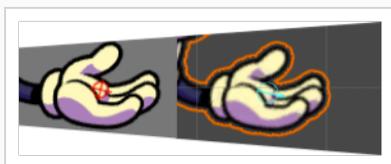
### アンカーを Unity にインポートする方法

1. タイムラインビューでレイヤーを選択するか、[ノード]ビューでノードを選択し、[ゲーム]ツールバーの[アンカーを切り替え]ボタンを選択します。



2. キャラクターを Unity にインポートします。

次にXMLのインポートプロセスが実行される際に、新しいアンカーはUnityのキャラクターのインポートされたプレハブの下に、新しい[Harmonyアンカー]として表示されます。



この新しいゲームオブジェクトは、キャラクターで再生されている現在のアニメーションの位置、回転、およびスケールに従います。これにより、ゲームプレイオブジェクトを子としてアンカーに動的に取り付けて、持ち運び、装備、使用することができます。



#### メモ

アンカー位置を移動しても、Harmonyレンダラーのビジュアルには影響しません。アンカーだけでなく、Unityでキャラクターの構造全体を表示して影響を与えるには、TBGワークフローの使用を検討することをお勧めします。

## オーディオソースの追加

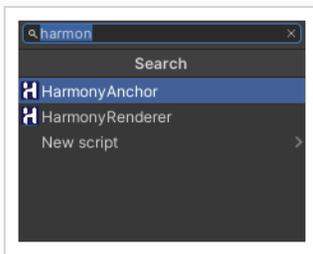
Harmonyでオーディオ付きのカットファイルを作成すると、オーディオはプロジェクトフォルダーにエクスポートされます。Harmonyインポーターは自動的にオーディオソースを作成し、生成されたプレハブに追加します。Harmonyレンダラーは、Harmony内で作成されたように、アニメーション再生中のポイントでオーディオクリップファイルの再生をトリガーできます。

## 空のゲームオブジェクトへのHarmonyレンダラーの追加

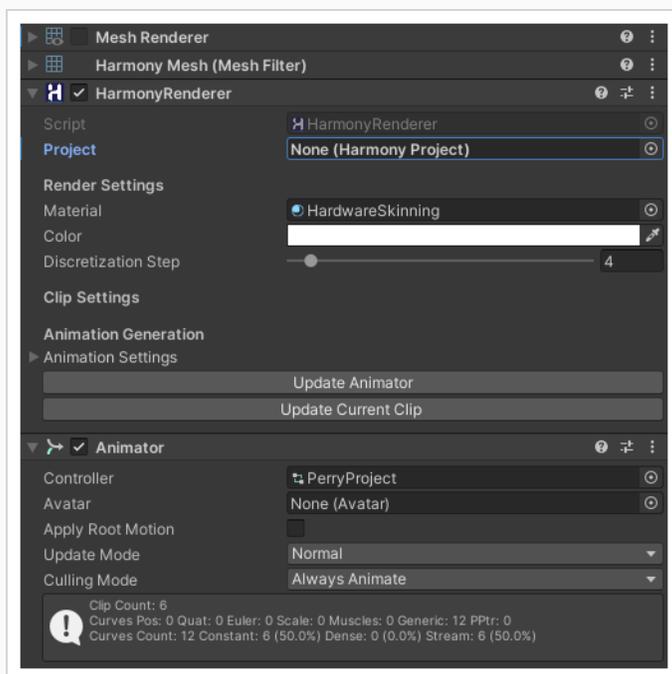
Harmony インポーターからのプレハブがない場合でも、空のゲームオブジェクトに新しい Harmony レンダラーコンポーネントを作成して、同じ最終的な結果を得ることができます。

### 空のゲームオブジェクトの使用方法

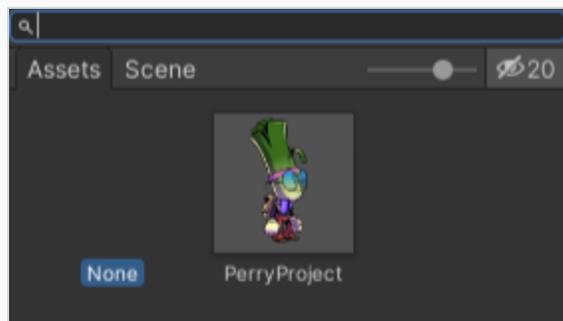
1. **ゲームオブジェクト > 空を作成**を選択します。
2. 階層内で明確になるよう、空のゲームオブジェクトの名前を変更します。PerryPoireau デモファイルを使用しているので、次のいずれかを実行して、ゲームオブジェクトの名前を**SpaceCat**に変更します。
  - 名前をダブルクリックし、階層で名前を変更します。
  - ゲームオブジェクトを選択し、インスペクターで名前を変更します。
3. この時点では、空のオブジェクトです。Harmonyデータへのアクセスは、スクリプトを介して行われます。
  - PerryPoireauゲームオブジェクトを選択します。
  - インスペクターで、**コンポーネントを追加>スクリプト>Harmonyレンダラー**に進みます。



インスペクターに、メッシュレンダラー、Harmonyメッシュ（メッシュフィルター）、Harmonyレンダラーという4つの新たなセクションが表示されます。



ここから、この新しく作成されたHarmonyレンダラーにHarmonyプロジェクトをアタッチする必要があります。空（Harmonyプロジェクト）スロットの右側にある円をクリックすると、XMLフォルダーワークフローを使用してHarmonyプロジェクトを作成する手順を使用して、すでにインポートしたプロジェクトを選択できます。



## Unity での衝突の設定

境界ボックスの情報を抽出すると、何かをキャラクターと衝突させる場合に便利です。2Dのキャラクターを3Dカットに配置する場合、Physicsを使用して2Dのキャラクターを3Dの平面に衝突させることができます。

コライダーは、Unity で使用するためにいくつかの異なるタイプのコライダーを提供するように改善されました。ポリゴンコライダーとボックスコライダーを選択して、より正確な境界ボックスを作成することができます。

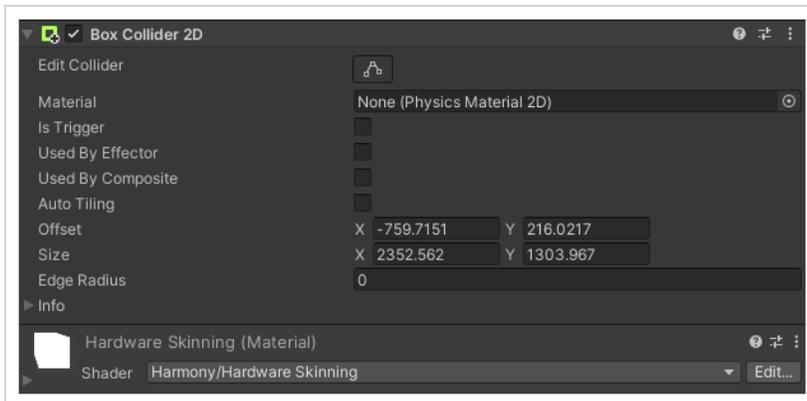
### 衝突の設定方法

1. インポートしたキャラクターを選択します。
2. インспекターで、**コンポーネントを追加 > Physics 2D > ボックスコライダー2D**を選択します。
3. キャラクターのPhysicsを有効にするには、**コンポーネントを追加 > Physics 2D > 剛体2D**を実行します。



### メモ

インポートしたファイルで3D Physicsを使用することもできます。



これで変数を調整して結果を確認できるようになります。

## ゲーム用語集

<b>.tbg ファイル</b>	Toon Boom Gameファイル形式 (.tbg)、Harmonyから作成され、Unityおよび将来的には他のゲームエンジンによって取り込まれます。
<b>2Dアニメーションパッケージ</b>	高度な 2D スプライト処理とアニメーション（変形、リギング、テッセレーション）を容易にする Unity 用のファーストパーティプラグイン（Unity 製）
<b>2原画</b>	カットアウトアニメーションでは、2原画とは、キャラクターを断片に分割して、アーティキュレーションのあるパペットを作成するアクションです。キャラクターを分解するために、アーティストはキャラクターのモデルから手や腕といったパーツをカットして、別々のレイヤーにペーストします。次に関節が固定され、ピボットが設定されます。セルアニメーションにおける2原画は、一般的に2つのキーポーズ間にあるアニメーションポーズです。キーポーズとはアニメーションのメインポーズです。分解は補助的なポーズとして、動きと回転カーブ（通常「ストーリーアーク」と呼ばれます）の描写に役立ちます。
<b>bake_group</b>	bake_groupsは[ノード]ビューのグループであり、「bake_...」に改名されます。加えて、bake_groupは1つの入力ポートと1つの出力ポートのみをサポートすることができます。Harmonyからのエクスポート中、これらのグループの変更について分析が行われ、グループ内のデータの一意のバリエーションと同じ数のスプライトと、必要に応じて各スプライトを表示するアニメーションがエクスポートされます。bake_group内では、グロー、自由変形、カラーアート/ラインアートなど、あらゆるエフェクトを使用できます。
<b>CMYK</b>	シアン、マゼンタ、イエロー、ブラックの頭文字をとったもの。これらのカラーは標準モデルとして「オフセット印字」と呼ばれるプロセスで使用されます。
<b>CUT/カット</b>	映画やTV番組における1つのカット。シーンは複数のCUT/カットで構成されています。カットはシンプルなカットまたはトランジションによって別のカットに切り替わります。
<b>dpi</b>	ドットパーインチ（Dots per inch）は、電子化されたプリンターの解像度を表す一般的な単位です。画面の解像度として用いられることもあり、この場合、より正確には「ピクセルパーインチ（pixels per inch）」と呼ばれます。いずれの場合も、ドットは画像を形作る最小単位です。

<b>FBX</b>	FBXファイル形式 - 3ds Maxによって普及した3Dモデリングおよび3Dゲーム開発業界の標準ファイル形式
<b>film-1.33</b>	標準的な4:3画素アスペクト比を満たすワイド画面フィルム形式に最適な解像度。
<b>film-1.66</b>	16:9画素アスペクト比を満たすワイド画面フィルム形式に最適な解像度。
<b>HDTV</b>	テレビ制作の現在の解像度基準である、High Definition Televisionの頭字語。HDTV制作の解像度は1280×720 (HD対応) か1920×1080 (フルHD) のいずれかであり、それらのフレームレートは一般に23.976~30の範囲です。
<b>HSV</b>	色相 (Hue) 、彩度 (Saturation) 、および明度 (Value) 。色相 (色合い) 、彩度 (陰影) 、および明度 (明暗または輝度) でカラーを定義する手法。
<b>IK マネージャー 2D</b>	Unity 用のファーストパーティプラグイン (Unity 製) で、2D キャラクターのインバースキネマティクス (IK) を容易にします。
<b>NTSC</b>	北米で使用されるレガシースタンダードのアナログテレビ放送システム。
<b>PAL</b>	テレビとコンピューター画面のヨーロッパ規格に最適な解像度。長方形画素を異なる方向に表示します。
<b>RGB</b>	赤 (Red) 、緑 (Green) 、青 (Blue) 。この3つのカラー成分の量を指定することで、カラーを定義する手法です。
<b>sRGB テクスチャ</b>	Unity のインポート設定でチェックされた sRGB は、テクスチャ内のデータが Unity の線形レンダリングモードと一致する線形カラースペースでエンコードされたことを Unity に示します
<b>TBG</b>	Toon Boom Gameファイル形式 (.tbg) 、 Harmonyから作成され、Unityおよび将来的には他のゲームエンジンによって取り込まれます。
<b>TBG レンダラー</b>	TBGインポートプロセス中にプレハブにアタッチされたランタイム動作で、解像度、パレット、スキンの切り替え、カッタースプライトの更新を管理します。
<b>TBG ワークフロー</b>	Harmonyからキャラクターをエクスポートし、TBGファイルを中間ファイルタイプとして使用してゲームエンジンにインポートするプロセス

<b>TBGストア</b>	TBG レンダラーによって実行時に参照されるキャラクターに関する有用なデータを格納する、TBG インポート中に生成されたアセット
<b>Unity カスタムインポーター</b>	プロジェクトごとにC#スクリプトインターフェイスで定義され、ファイルの拡張子をマップして（例：「.tbg」）ファイルデータを Unity アセットに変換するプロセスに置き換える、Unityのインポーター。
<b>Unityアセット</b>	Unity によって生成され、Unity ネイティブであり、ゲーム構築プロセス中にゲームにパックされる Unity 内のデータの塊。これらはUnity のツールによって、および外部ファイルのインポート中に生成されます。
<b>XML</b>	XMLファイル形式 - JSON形式と同様の機能で、データを構造的に保存するための標準ファイル形式
<b>XML ワークフロー</b>	Harmonyからキャラクターをエクスポートし、XMLフォルダーを中間ファイルタイプとして使用してゲームエンジンにインポートするプロセス
<b>アクションライン</b>	アクションポーズの主要中心線。アクションポーズを描くとき、アクションラインを気にすることでポーズが動的になり、そのシルエットがアクションの方向に合うようになります。
<b>アクション方向</b>	アクションがたどるパス。
<b>アスペクト比</b>	カット、コマまたはフィルム形式の幅と高さの寸法比率。現在、最も人気のアスペクト比は16：9です。従来のディスプレイは4:3のアスペクト比を使用してきました。
<b>アニメティック</b>	絵コンテ内のパネルを順番に並べ、各パネルが表すアクションの大ききな尺についてサウンドトラックと同期するように各パネルのタイミングを調整して作ったムービー。アニメティックは、絵コンテを最終的なムービーの非常に大ききなドラフトに変換し、各カットと各アクションに割り当てられた時間を決定し、そのアクションをサウンドトラックと同期させるために使用されます。
<b>アニメ</b>	日本のアニメ。
<b>アニメー</b>	個々のコマを描画または編集することによって作成された動きのシミュレーショ

<b>シオン</b>	ン。
<b>アニメーションクリップ</b>	プレハブ内のさまざまなデータフィールドの一連のカーブを含む Unity アセット (位置、回転、スケールの変更)
<b>アニメーターコントローラー</b>	アニメーターコントローラーは、キャラクターの現在の状態に基づいてゲームプレイ中に再生するアニメーションを決定します
<b>アニメーターコントローラーを作成</b>	TBGファイルの初回インポート時にアニメーターコントローラーを作成するオプション
<b>アルファチャンネル</b>	透明度情報を持った画像チャンネル。画像にはもともと赤、緑、青 (RGB) という3つのチャンネルがあります。4番目のチャンネル (A) がアルファチャンネルです。この4番目のチャンネルにはマットまたは透明度に関する情報が保存されています。アルファチャンネルを持たない画像は常に不透明となります。
<b>アンカー</b>	Harmonyのノードをアンカーとしてマークし、実行時にゲームエンジンでの位置、回転、スケールを公開できます。TBGファイルにはアンカーは不要です。
<b>アンダーレイ</b>	アニメーションでは、アンダーレイはメインアニメーションの背後に配置された舞台装置の特定部分です。
<b>イーズ</b>	アニメーションにおいて、イーズは「速度」とも呼ばれ、動きの加速または減速を指します。この動きは、機能曲線または一連のアニメートされた描画によって作成されます。イーズインとイーズアウトには、スローインとスローアウトという用語も広く使われています。
<b>イーズアウト</b>	アクションが徐々に減速すること。「スローアウト」とも呼ばれます。
<b>イーズイン</b>	アクションが徐々に加速すること。「スローイン」とも呼ばれます。
<b>インスペク</b>	ゲームオブジェクトのコンテンツを表示する Unity のビュー - トランスフォーム、

<b>タービュ</b>	レンダリングとPhysicsのためのコンポーネント、カスタムビヘイビア
<b>インバース キネマティ クス (IK)</b>	階層を持つ3Dキャラクターとカットアウト人形（パペット）をアニメートするために主に使用される機能。インバースキネマティクスは、四肢のひとつ（手など）から人形（パペット）をアニメートして、身体の他の部分を自然に動かします。
<b>インポ ーター設 定</b>	アセットをファイル内のデータから Unity が理解できるデータに変換する方法を変更する、[インスペクター]ウィンドウからアセットで利用可能な設定
<b>ウォークサ イクル</b>	キャラクターの歩行を描く、「その場」での一連の描画。背景をパンすることで、移動しているような錯覚を生み出します。描画を無数に作らなくてもいいように、アニメーターは必ずキャラクター用のウォークサイクルを作成します。
<b>オーバ レイ</b>	椅子や茂みなど、メインのアニメーションの前方に配置されたカット環境の一部。
<b>オニオン ス キン</b>	シーンの前の画面および次の描画を確認できる機能。
<b>カット</b>	1つのカットから次のカットへ、あるコマから次のコマにまったく視覚効果なくトランジションするとき。
<b>カットア ウトア ニメ ーション</b>	「カットアウトアニメーション」と呼ばれるプロセスは、複数のピースからなるキャラクターを1コマずつ動かしてアニメートするアクションです。カットアウトアニメーションはCGまたは紙を使った従来の手法で作成することができます。
<b>カット ビュー</b>	Unity でのゲームの視覚表示。階層ビューを補完し、カットビューはプレイヤーにどのように見えるかを示す階層ビューは、ゲームがエンジンにどのように見えるかを示します。
<b>カメラブレ</b>	カメラブレが発生するカットでは、カメラが複数の方向にわずかにすばやく動きます。これによって衝撃や振動、あるいは道路のこぶといったものを表現します。
<b>カラーカ ード</b>	カラーカードとは単色が塗られたカメラと同じ大きさのカードです。背景画像がない場合に、カラーカードを使って背景を単色で塗りつぶします。
<b>カラーホ</b>	円形で表示されたカラースペクトル。

<b>イール</b>	
<b>キーフレーム</b>	あらゆるアクションの開始および終了ポイントを指定する、アクション内にある重要な位置。キーフレームは、ある軌跡の特定の瞬間（コマ）にあるコンピューター生成された位置です。
<b>キーポーズ</b>	滑らかなトランジションの開始および終了ポイントを指定する、アクション内にある重要な位置。キー、またはキーポーズは、動きを描写するアニメーションシーンの中で、メインとなる描画です。例えば、腕を振っている場合、腕の動きの極点と極点にある腕がキーとなります。この描画をフリッピングすることで、アニメーターはすべての描画がなくても動きの概要を確認できます。
<b>キーボードショートカット</b>	押されたときにアプリケーションで特定のコマンドを実行するために割り当てられている、キーボードキー、またはキーボードキーの組み合わせ。
<b>キャプション</b>	絵コンテでダイアログ、エフェクト、サウンド、またはスラッグングについての情報を書き込むテキストフィールド。
<b>キャラクターデザイン</b>	アニメートされたフィルムの各キャラクターは、ポスターのような形で、さまざまな角度から描かれます。これは「モデルシート」と呼ばれ、アニメーターの参考資料になります。
<b>クリーンアップ</b>	ラフな描画のテストおよび承認が行われた後、画像にあるすべてのノイズ（余分な線やメモなど）が除去され、最終描画が作成されます。最終描画では、インク入れやペイント、撮影が可能になります。クリーンアップ作業とは、ラフ描画上できれいな線をトレースして最終版を完成させること、またはスキャン作業で残った汚れと余分な線を除去することを指します。
<b>クロスディゾルブ</b>	次のカットがフェードインすると同時にフェードアウトすることによって、1つのカットが次のカットにトランジションするとき。
<b>ゲームボーン</b>	ゲームエンジンでのボーンの動作をエミュレートするHarmonyの変形は、Harmonyの通常のボーン変形に似ていますが、ブレンドアルゴリズムが異なります。
<b>コマ</b>	アニメーションでは、「コマ」とはそのカットで描画が表示されるセルの数を指します。図面が長く表示されるようにするには、露出をセル数の多くにわたって拡張する必要があります。

<b>コマ撮り キーフレー ム</b>	コンピューター生成されたフレーム補間がないキーフレーム。
<b>サイクル</b>	連続的な動きをシミュレートするために繰り返しが可能な一連のアニメーションコマ。たとえば、キャラクターは、2つのステップを前に1つずつ進めながらアニメートし、次にこのアニメーションを循環させることによって歩行させることがよくあります。
<b>サウンドス クラブ</b>	再生ヘッドを前後に動かしながらリアルタイムにサウンドを聴くことができます。ロパクを微調整する際に非常に便利です。
<b>サウンドト ラック分解</b>	アニメーションフィルムのサウンドトラックを個々のサウンドに分解して、各サウンドのコマごとの正確な位置を生成します。
<b>サブアセッ ト</b>	既存のアセット内にパックされ、通常はメインアセットに関連付けられているアセット。TBGファイルの場合、キャラクターはテクスチャ、スプライト、アニメーションクリップ、素材、TBGストアなどの多くのサブアセットを持つ、メインのプレハブアセットで構成されます。
<b>サムネール</b>	参照用に使用される非常に小さな画像。
<b>シーン</b>	ストーリーやムービーの中で1つのまとまりを形成する、一連のカットまたはCUT/カット。通常、場所や時間が同じものがまとまっています。
<b>シェーダー</b>	提供されたメッシュとテクスチャのデータ（着色、照明、滑らかさ、曲げ、ブレンドなど）の表示方法をゲームに通知する手順。
<b>シェーダー グラフ</b>	Unity が UI とスクリプトインポーターで提供する Unity ファイル形式は、シェーダーの視覚的な作成と編集を可能にします - コードを書く必要はありません
<b>ジャンプ カット</b>	2つのカット間の唐突なカット。通常、ジャンプカットは視覚的に美しいものではありません。一般的には、1つのカットが終了して、類似した画像で次のカットが始まる際に使用されます。2つのカット間の差異を補うものがない分、少し飛躍しているように映ります。
<b>ショットの</b>	シーンが発生するエリア全体を視聴者が確認できるカット。例えば、ひとりの子供

<b>確立</b>	が家の前の地面で遊んでいたとすると、ショットの確立では、視聴者はアクションの中心点の周囲にある家、地面、通りの一部、建物のカットを確認できます。これによって視聴者はストーリーの場所やカットの位置関係を把握することができます。
<b>シンボル</b>	シンボルはアニメーション、アートワークまたはレイヤーを単一のオブジェクトに統合し、1つのレイヤーでコントロールできるようにします。カットアウトパペットの各ボディーパーツからシンボルを作成することもできます。シンボルにはどんなものでも配置することができます。シンボルを使用して、人形（パペット）をアニメートしたり、点滅などの再利用可能なアニメーションを作成することができます。
<b>スキン</b>	レイヤーのタイムラインにある描画の代替コレクションは、Harmonyの[スキン]ビューを使用して作成および切り替えることができます。例としては、変更可能なキャラクターの代替コスチューム、またはゲームプレイ中にオプションで表示または非表示にされるキャラクターの体の新しい部位が含まれます
<b>ステップ付き</b>	アニメーションをアーティストの意図したフレームレートに維持するか、より高いフレームレートで表示したときに中間コマを推測するか
<b>ストーリーアーク</b>	アクションが直線的に発生することはめったにありません。通常は、語り手がストーリーアークと呼ぶ形で展開します。ストーリーアークの目的はキャラクターまたはシチュエーションを、ある状態またはシナリオから、次の状態または次のシナリオへと展開させることです。
<b>ストレートアヘッドアニメーション</b>	シーン全体を、最初の位置から最後の位置まで順に描く手法。この手法にはプランニングがほとんどありません。キャラクターがどこで結末を迎えるか、そしてどのようにそこへ行き着くかは、視聴者にもアニメーターにも予想できません。自由でクリエイティブな手法である一方、でたらめな結果を招くこともあります。
<b>ストローク</b>	実際のブラシや鉛筆のストロークを参考しない場合、ストロークは、カラーゾーンを定義したり開いたカラーゾーンを閉じるために使用される目に見えない輪郭です。
<b>スプライトアトラス</b>	Unity バージョンのスプライトシート - スプライトのコレクションを取得し、それらを単一の画像にフラット化します。アトラス内の各スプライトは、レンダリング中にアトラス内のデータを指すため、多くの異なるスプライトの性能オーバーヘッドが削減されます。

<b>スプライト シート</b>	複数のスプライトがすべて1つの画像にフラット化されました。一部のシートは固定サイズのグリッドで、他のシートにはさまざまなサイズのスプライトがあります。シートからスプライトデータを適切に抽出できるように、ゲームエンジンは各スプライトがシート内のどこにあるかを理解する必要があります。
<b>スラッグ グ</b>	ダイアログとそれに対応するアクションの開始時間および停止時間を指定します。
<b>スローア ウト</b>	アクションが徐々に減速すること。「イーズアウト」とも呼ばれます。
<b>スローイ ン</b>	アクションが徐々に加速すること。「イーズイン」とも呼ばれます。
<b>セル</b>	セルアニメーションで使用されるセル（「セルロイド」とも呼ばれます）は透明なシートで、カメラに送られる前にアニメーションに仕上げ処理が施されます。セルの前面に絵の輪郭が描かれ、次に裏面が彩色されます。
<b>セルア ニメー ション</b>	すべてのアニメーションシーンを紙に手描きしてから、スキャンやセルへのインク入れを行うアニメーション制作プロセス。
<b>ゾーン</b>	カラーを塗ることができるエリア。
<b>ダイア ログ</b>	ムービーまたはアニメーションでキャラクターが話す台詞。
<b>タイム コード</b>	ムービークリップにプリントされたタイミング情報。現在画面に表示されているカット、時間、分、秒を表します。
<b>タイム シート</b>	縦方向に列、横方向にコマが複数記載されたシートです。カットのタイミングを指示するために使われます。各列はカットレイヤーを表します。各列の描画番号は、表示する必要がある特定コマ全体にわたって示され、反復されます。タイムシートはアニメーター、ディレクターおよびその他のスタッフが使用して、画像のシーンとタイミング、ダイアログ、サウンドエフェクト、サウンドトラック、カメラ動作を追跡します。「ドープシート」とも呼ばれます。
<b>タイム ライ ン</b>	カットの要素、タイミング、キーフレームを横方向に表したものの。

<b>ダブルバウンズウォーク</b>	ダブルバウンズウォークのキーフレームとパッシング点では、直立姿勢で描かれた基準線より身体が低くなります。中割りでは身体がこの線より上になるため、跳ねているように見えます。
<b>テンプレート</b>	あらゆるプロジェクトで再利用可能な、ライブラリに保存されているアセット。描画、一連のキーフレーム、サウンドファイル、パネル、カットアウトキャラクター、エフェクト、軌跡、アニメーションなど、アニメーションで使用されるあらゆるものがテンプレートになります。
<b>ドープ</b>	特定の描画を一定範囲のコマに割り当てます。
<b>ドープシート</b>	画像のシーンとタイミング、ダイアログ、サウンドエフェクト、サウンドトラック、カメラ動作を追跡するために、アニメーター、ディレクターおよびその他のスタッフが使用します。「タイムシート」とも呼ばれます。
<b>ドラッグ</b>	アイテムをクリックして押したままマウスカーソルを移動してアイテムを移動する操作。これは、マウスの左ボタンかペン先端で行うことができます。
<b>トランジション</b>	切り替わる2つのカットの間に置かれたエフェクト。よく使用されるトランジション効果はクロスディゾルブとワイプです。
<b>トレース&amp;ペイント</b>	ラフアニメーションが、クリーンアップとラインまたは鉛筆の最終テストを通過した後、各描画は最終的なアニメーションに向けてトレースおよびペイントされます。デジタル化が進んだ現在では、従来のセルロイドまたはアセテートの手法以外にもさまざまな手法で行われることがあります。
<b>ナッジ（微調整）</b>	選択したエレメントを、キーボードの矢印キーを用いて少しだけ（上下、左右、前後に）動かすこと。選択範囲をほんの少し、正確に移動させる際にはナッジを使用します。
<b>パッシング点</b>	パッシング点は、キャラクターの歩行シーンを描くときに、片方の脚がもう片方の脚を追い越すポイントです。
<b>パネル</b>	絵コンテにおいて、パネルは1つのCUT/カットの1つのコマです。カットは1つまたは複数のパネルからなります。
<b>パレット/</b>	パレットまたはマスターパレットは、キャラクターまたはプロップに割り当てられ

<b>マスターパレット</b>	た色のグループです。プロジェクト全体を通じてパレットを使用することで見た目の一貫性が保たれ、アニメーションの途中でのカラーが変わることを回避します。「マスターパレット」とも呼ばれます。
<b>パレットスタイル</b>	パレットスタイルは既存のパレットの第2バージョンで、色合いと明度にわずかな違いがあります。パレットスタイルを使用してパレットの夜間バージョンを作成することができます。「クローンパレット」とも呼ばれます。
<b>パン</b>	カットの任意の方向にカメラを動かすこと。
<b>ビットマップ</b>	画素のメッシュと、その各画素の個々の色で定義された画像。ビットマップグラフィックは、高度に詳細な、または写実的なアートワークにとってベクターグラフィックより好ましいことで知られています。しかし、ビットマップグラフィックは、画質を損なわずに拡大縮小、回転、または傾斜させることはできません。
<b>ピボット</b>	ペグまたは描画が回転するポイント。
<b>ファストアウト</b>	アクション終了時の劇的な加速。
<b>ファストイン</b>	アクション開始時の劇的な加速。
<b>フィールド</b>	カメラの位置とステージまでの距離を記録し追跡するために、セルアニメーションで使用されている測定ユニット。
<b>フィールドチャート</b>	アニメーションおよびレイアウトアーティストがカメラフィールドのサイズとカメラ動作を定義するために使用するすべてのフィールドユニットを含むガイド。
<b>フィルターモード</b>	隣接するピクセルをブレンドする方法 - 最近接はピクセルの周りのハードエッジ（ピクセルアート）を表示しますが、2本線と3本線は隣接するピクセルのカラーデータをブレンドして画像を滑らかにしようとします
<b>フェードイン/フェードアウト</b>	フェードインまたはフェードアウトは、シーンを開始または終了するために使用されるトランジション効果です。フェードインが発生すると、完全な透明から完全な不透明へと、最初のカットが徐々に現れます。フェードアウトが発生すると、完全な不透明から完全な透明へと、最後のカットが徐々に消えていきます。

<b>フォワード キネマティ クス</b>	フォワードキネマティクス (FK) は主に、階層を持つ3Dキャラクターとカットアウト人形 (パペット) をアニメートするために使用されます。肩などの親パーツの1つからパペットをアニメートして、腕の残りの部分を単一のピースとして連携させて動かすために使われます。
<b>フリッピング グ</b>	セルアニメーションにおけるフリッピングとは、アニメーションシーンの描画を非常にすばやくめくり、アニメーションを動かすアクションです。フリッピングは、オブジェクトのミラートランスフォームrを作成する作業でもあります。
<b>フレーム レート</b>	フレームレートはコマが再生されるスピードです。通常は1秒あたりのコマ数で計算されます。例えば、カットは1秒間に12、24、25、30、60コマ、または任意のコマ数で再生することができます。これは、撮像装置が「コマ」と呼ばれる固有の連続的な画像を生成する、周波数 (速度) の測定値です。この用語は、コンピューターグラフィックス、ビデオカメラ、フィルムカメラおよびモーションキャプチャーシステムでも同じように使われます。フレームレートは多くの場合1秒あたりのコマ数 (fps) で表され、プログレッシブスキャンモニターではヘルツ (Hz) で表されます。
<b>フレーム補 間</b>	2つのキーフレーム間に作成される、コンピューター生成された動き。キーフレーム間にフレーム補間を作成するかしないかを選択することができます。
<b>プレハブ</b>	ゲームオブジェクト、トランスフォーム、コンポーネント、ビヘイビアの階層は、階層ビューで構築してプロジェクトビューに保存することができます。プレハブのインスタンスを階層に配置して、同じオブジェクト (多くのキャラクター、多くの銃弾、多くの木) の複数の複製を持つことができます。
<b>プレビュー ウィンドウ</b>	[インスペクター]タブのセクションでは、通常は通常はモデル、素材、テクスチャで、アセットの外観を表示します
<b>ペーパーレ ス作画</b>	ペーパーレス作画とは、デジタルでアニメートを行う作業のことです。ペーパーレス作画でメインとなる作業は、ソフトウェアで直接1コマごとにアニメーションを描くことです。
<b>ペグ</b>	セルアニメーションで、セルレイヤーが動く際にアクションを正確に登録するためのツール。より高度なパペトリギングを行うデジタルアニメーションでは、ペグレイヤーを使用することができます。ペグレイヤーは、描画を含まない軌跡レイヤーです。これはモーションパスで、これを使用してパス関節を追加することができます。後者の場合は、インバースキネマティクス (IK) ツールを使用することも

できます。

<b>ベクター</b>	ベクターベースの画像は点とベジェ曲線で構成されています。コンピューターは点を読み取ってセグメントをトレースし、これらをつなぎ合わせて画像の形状を再現します。ベクター画像には固定されたサイズや解像度はありません。システムがセグメントを再計算して形状を再構築するため、グラフィックを好きなだけ拡大、変形することができます。計算が終了するとベクター画像は画素に転換され、表示されます。
<b>ベジェ曲線</b>	フランス人数学者ピエール・ベジェによって考案された、曲線を定義する手法。ベジェ曲線は数学的曲線またはパラメトリック曲線です。ベジェ曲線はコントロールポイントによって定義されます。Harmonyでは、ベジェ曲線には開始点、終了点、および開始点と終了点の間の曲線に影響を与える2つの内部コントロールポイントがあります。
<b>ペンタブ/ ペン</b>	マウスと連携して、またはマウスの代わりに使用して、コンピューター画面でマウスポインタ（「カーソル」と呼ばれることもあります）を動かすデバイス。
<b>ポーズトゥ ポーズ アニ メーション</b>	ポーズトゥポーズ アニメーションプロセスは、「キーポーズ」と呼ばれるすべてのメインアクションポーズを作成し、その後キーの間に補助的なポーズを配置する作業です。この補助的なポーズは「2原画」と呼ばれます。最後にアニメーターが中割り描画で隙間を塗りつぶし、滑らかなアニメーションを実現します。
<b>ホールド</b>	アニメーションでキャラクターが1つの位置に留まって動かないコマ。ホールドは任意の2つのキーフレーム間に作成できます。
<b>ボーン</b>	キャラクターを変形させるためのガイドとして使用される、キャラクターの体の剛体セグメント-通常は脚と腕
<b>マウス チャート</b>	ロパクに使用される、8つのアニメーション音素（A、B、C、D、E、F、G、そして沈黙を表すX）に基づくチャート。
<b>マスターパ レット</b>	キャラクターやプロップに割り当てられた色のグループ。パレットを利用することで制作の全過程を通じて見た目の統一性が保たれ、作品全体で確実に同じ色を使用することができます。単に「パレット」とも呼ばれます。
<b>マルチプ レーン</b>	複数階層の描画を通して1つのCUT/カットに奥行きを与えるエフェクト。マルチプレーンのカットでは、カメラから異なる距離にレイヤーを配置することで、カメラ

	<p>が移動した際に奥行きがあるかのような錯覚を与えます。マルチプレーンを利用すると、すべての遠近感と縮尺は自動的に計算されます。</p>
<b>ミップマップ</b>	<p>Unity の[インポート]設定でチェックされたミップマップは、ユーザーがより小さい規模/より遠い距離でレンダリングするために生成された、テクスチャの低解像度バージョンを希望していることを示しています</p>
<b>モーションキーフレーム</b>	<p>コンピューター生成されたフレーム補間があるキーフレーム。</p>
<b>モーフィング</b>	<p>起点の描画と終点の描画の間に、コンピューター生成された描画を作成する機能。モーフィング機能で作成されたアニメーションは他のプロジェクトで再利用できます。</p>
<b>モデル/キャラクターモデル</b>	<p>制作時に各アーティストが従わなくてはならない、キャラクター、プロップ、または場所の最終的なデザイン。色指定はアニメーションのペイントに使用する必要がある公式のカラーデザインです。</p>
<b>ライトテーブル</b>	<p>特定のレイヤーで作業しながら透かして他のレイヤーを見ることができるデバイス。</p>
<b>ライブラリ</b>	<p>あらゆるプロジェクトやカットで再利用できるテンプレートとアセットを収めたストレージエリア。</p>
<b>ラフ</b>	<p>ラフとはアニメーション映画で、参照用に使われるものの、最終イメージの一部にはならない描画を指す一般的な名称です。レイアウトはラフになります。アニメーションやデザインの概要となるスケッチです。ラフは主にスケッチの線と図形からなりますが、デザインの詳細が含まれることもあります。</p>
<b>リギング</b>	<p>カットアウトパペットのさまざまなパーツを取り付けるプロセス。</p>
<b>レイアウト</b>	<p>絵コンテとアニメーションをつなぐステップ。モデルに基づいて絵コンテを配置する作業です。つまり、モデルパックのデザインに従ってキャラクターを描き、アニメーターが作業を開始できるようにすることです。レイアウトアーティストは背景を描き、カットとカメラの動きに合わせたカメラおよびフィールドガイドを作成します。最後に、アニメーターがモデルに基づいてメインのアクションポーズを描きます。</p>

<b>レイアウトアーティスト</b>	背景を描き、カメラとフィールドガイドを作成してカットとカメラの動きを一致させるアーティスト。モデルに基づいてメインのアクションポーズを描きます。
<b>レイアウトとポーズ</b>	アニメーターが作業を開始できるようにモデルに基づいて（正しい縮尺で）絵コンテを配置する作業。
<b>レイアウトプロセス</b>	絵コンテとアニメーションをつなぐステップ。
<b>レイアウト計画</b>	アニメーションを計画して第一段階に着手する際に、カットの主な特徴を描いたもの。参照用に使用されます。
<b>レイヤー</b>	アニメーションにおけるレイヤーは、個々の列、レベル、またはキャラクターです。カットのレイヤーが重ね合わされて最終的な画像となります。
<b>レンダリング</b>	コンピューターによるアニメートの最終ステップ。レンダリング中、コンピューターは画面に表示される各画素を取り込んですべての構成要素を処理し、モーショングラフを追加してから最終イメージを作り出します。合成プロセス後に最終イメージを計算するプロセスです。
<b>ローレゾ</b>	ウェブ用のビデオに理想的な形式。品質よりビデオのサイズとダウンロードの速さが優先されます。ローレゾ画像では細部は表現されません。
<b>ロトスコープ</b>	アニメーターが実写映画の動きを1コマごとにトレースしてアニメーションに使用する技術。実写映像をスケッチして、アニメートされたシーンを作成します。
<b>安全フレーム</b>	カットのコマの中央にあるゾーンで、テレビフレームによってクロップされる心配がありません。TVコマはオリジナルのコマサイズから縁をカットするため、安全フレームを維持することで、フィルムがテレビで上映されても、カットのメインアクションをはっきり視認できます。
<b>音素</b>	言語の音の単位。
<b>画素</b>	モニターやテレビ画面に表示される画像の最小要素。ピクセル (pixel) とは画素 (picture element) を短縮した呼称であり、グラフィック画像内にある単一の点を指します。画像の小さなサンプルであるピクセルは、「ドット」とも呼ばれ、正方

	<p>形をしています。滑らかなフィルタリングによって作られた、非常に小さなセクションです。デジタル画像を拡大していくと、画素が見えてきます。画素は、さまざまな色と明度を持った小さな正方形に見えます。</p>
<b>解像度</b>	<p>通常画素で計算されるカットのサイズ。例えば、NTSCの解像度は720 x 480です。解像度は、HDTV、film-1.33、film-1.66、NTSC、PAL、低など、最終出力と一致する必要があります。</p>
<b>回転テーブル</b>	<p>アニメーションディスク/テーブルと同じく、描画中に作業スペースを回転して、より快適な作業を可能にするデバイス。</p>
<b>絵コンテ</b>	<p>アニメーション内のすべてのカットやCUT/カットを視覚的に確認できる設計図。絵コンテでは、何が起こるか、いつ起こるか、カットのオブジェクトがどのように配置されるかが指定されます。</p>
<b>機能</b>	<p>エレメント、その他の軌跡およびエフェクトパラメータを付け加えることができる、コンピューター生成された動き、軌跡、またはパス。機能曲線にキーフレームとコントロールポイントを追加することで、機能をコントロールすることができます。</p>
<b>軌跡</b>	<p>エレメントがなぞる、コンピューター生成されたパスや軌跡。コントロールポイント、キーフレームおよび速度によって軌跡をコントロールすることができます。</p>
<b>脚本</b>	<p>映画やTV番組の情報がすべて含まれるテキスト。アニメーションでは、場所の説明、ダイアログ、時間など、すべてが脚本に記載されています。プロジェクトの第一歩は脚本からスタートします。</p>
<b>ロパク</b>	<p>キャラクターの口を、ダイアログサウンドトラックのサウンドに同期させるプロセス。ダイアログのサウンドに合うように口の形が1コマごとに調整されるため、まるでキャラクターが話しているかのような錯覚を与えます。鳥がさえずったり、月に向かって狼が吠えるなど、ロパクは台詞だけでなくあらゆるサウンドシーンに利用できます。</p>
<b>合成</b>	<p>合成とは、レンダリングの前にカットのエレメントすべてを合成して、最終的な成果物を作成する作業です。例えば、合成アーティストはカットのすべてのアニメーションシーン、背景、オーバーレイ、アンダーレイをインポートして、正しい位置に配置します。アーティストは次にカメラフレームを設定し、必要であればアニメートします。最後にアニメーターがプロジェクトのすべてのCGエフェクトを作成</p>

	します。
<b>仕上げ</b>	仕上げプロセスは、カラーモデルに従って、空白ゾーンをペイントして最終的なアニメーション描画のラインに彩色するアクションです。
<b>自動インポートプロセス</b>	ファイルの作成時、または変更/上書き/更新時に、ユーザーの介入なしにファイルを自動的にインポート
<b>自動フィード</b>	描画をスキャナーに自動的にフィードする手法。複数の描画がシートフィーダーにスタックされます。スキャナーが起動すると、ユーザーが操作しなくても描画が連続してスキャンされます。
<b>自動口パク作成</b>	エレメントの描画を、サウンド用に生成されたマウスチャートに自動的にマッピングします。これによってボイストラックを口パクする時間を節約できます。
<b>軸</b>	オブジェクトが回転する仮想ライン。2DグラフィックスにはX（水平）とY（垂直）の2つの軸があります。3DグラフィックスにはX（水平）、Y（垂直）およびZ（深度）の3つの軸があります。常に回転し続けるアニメーションでは、オブジェクトがどの軸を中心に回転するか、軸要素によって指定されます。負の数はアニメーションを反時計回りに回転させ、正の数はアニメーションを時計回りに回転させます。
<b>手動口パク検出</b>	口の位置の描画を手動で取り替えて、ボイストラックに一致させること。このプロセスでは、サウンドスクラブ（コマごとに分割された音波を聴くこと）と描画置換の両方が用いられます。
<b>色域</b>	特定のデバイスが描写できる色彩の範囲。
<b>色指定</b>	アニメーションをペイントする際に使用しなければならない公式のカラーデザイン。モデルはキャラクターやプロップ、場所の最終的なデザインであり、各アーティストはこれに従って制作にあたる必要があります。
<b>色収差</b>	光学において、色収差（CA）、色消しまたは色彩変形は歪みのひとつであり、レンズの不具合により、すべての色を同一の光軸点に合わせることはできません。
<b>速度</b>	アニメーションにおいて、速度は「イーズ」とも呼ばれ、動きの加速または減速を

	<p>指します。これは、機能曲線または一連のアニメートされた描画によって生み出されます。イーズインとイーズアウトには、スローインとスローアウトという用語も広く使われています。</p>
<b>中なびき</b>	<p>メインのアクションによって引き起こされる二次的な動き。例えば、マントを着たキャラクターが走っているとします。メインのアクションは、走っている身体です。マントもその動きに従うこととなりますが、同時に動くわけではなく、数コマ遅れて反応し、メインのモーションカーブをなぞります。</p>
<b>中割り</b>	<p>キーポーズの間にある描画。ポーズ間の滑らかなトランジションを生み出すために描かれます。</p>
<b>背景</b>	<p>カメラのフィールドを塗りつぶし、キャラクターやプロップの背後にあるアートワーク。通常、このアートワークはアクションが行われる舞台装置や舞台を表します。</p>
<b>離散化ス テップ</b>	<p>スプライトが変形した場合にどの程度細かくセグメント化されるか、数値が大きいほど、激しく変形したスプライトの形状が滑らかになります</p>
<b>立体3D</b>	<p>実際に異なる2つの画像を出力する画像またはムービーで、それぞれが人間の目の1つの視点からのものです。3Dディスプレイやバーチャルリアリティゴーグルといった特別な装置を通して見たり投影したりすると、これによって、画像内のエレメントが実際に視聴者の目の前にあるかのような錯覚が生じます。</p>