

UNITY 2022 LTS 版 → EBOOK



Unity での アニメーション制作の 完全ガイド

目次

はじめに
序文:Dave Hunt7
意欲的なプロダクション向けのアニメーション制作8
アニメーターへの支援9
Unity のアニメーションシステム10
Unity にアニメーションをインポートする12
モーションライブラリ14
Autodesk Maya からアニメーションをエクスポートする 16
Blender からアニメーションをエクスポートする18
Unity にアニメーションをインポートする
メッシュ20
ジオメトリ20
マテリアル
スケルトン21
アバター
アニメーション22
FBX インポートルートオブジェクト
ジェネリックアニメーションタイプ 24
アニメーターコントローラー26
パラメーター
レイヤー
ブレンドツリー34
キャラクターコントローラー

-Unity でアニメーションを作成する
Animation ウィンドウの便利なショートカット46
Windows 向けの Animation ウィンドウのズームコントロール47
プレビューモードとレコードモード47
キーフレームに特定の値を設定する47
選択項目で絞り込む47
アニメーションイベント48
上級者向けのアニメーション機能 50
読み取り専用アニメーションのイベント50
ルートモーション
ブレンドシェイプ53
ヒューマノイドアニメーションタイプ55
Animation Rigging61
IK リグの設定63
複数リグ66
フルボディコントロールリグ67
足の IK68
アニメーションでの IK のマージ
Animation Rigging を使用するメリット
アニメーションカットシーンへの Timeline の活用
アニメーションカットシーン74
シーンに Timeline シーケンスを追加する
アニメーションクリップを追加する
トラックオーバーライドを追加する
キーフレームの打ち方82
トラックタイプ
アクティベーショントラック83
オーディオトラック

コントロールトラック	84
プレイアブルトラック	85
シグナルトラック	85
Cinemachine トラック	87
トラックグループ	88
シーケンス	88
上級者向けシミュレーション	89
リジッドボディ物理演算	89
クロス物理演算	95
ラグドール物理演算	99
毛皮と髪1	02
Alembic	04
パーティクル1	06
AI キャラクター1	107
アニメーションのエクスポート	111
FBX モデルのエクスポート	111
Unity Recorder	113
2D Animation1	116
PSD ワークフロー	116
Sprite Editor でのリギング	117
スプライトをボーン、ジオメトリ、ウェイトに接続する	117
 ウェイト	118
スプライトリゾルバーと 2D IK	119
まとめ	20

はじめに



Unity 制作のアニメーション短編映画『Enemies』のスクリーンショット

アニメーションはゲームデザインに欠かせない要素であり、現在では多くのゲームのアニメーションコン テンツは、映画並みのビジュアルクオリティを誇っています。

アニメーターは、キャラクター設定と視覚的なストーリーテリングによって、ゲームのストーリーを進めます。 彼らは、キャラクターのリギング、ポージング、アニメーションシーケンスのキーフレーム設定を担っています。 ゲーム開発におけるアニメーターの役割が変わるに従って、Unity のアニメーションツールやワークフローも 進化しています。 このガイドは、ベテラン、新人を問わず、Unity のアニメーションシステムで利用可能な専用のアニメーション ツール、メソッド、およびテクニックの使用方法を学びたいと思っているアニメーターを対象としています。

Unity にアニメーションをインポートする方法、ジェネリックとヒューマノイドのアニメーションタイプ、 エディターでのアニメーションおよびリギング方法、Timeline を使ったアニメーションカットシーンの 作成方法、上級者向けのアニメーションオプションなどについて学ぶことができます。

フォトリアリスティックなシーンから、手描き風、セルルック、カートゥーン調まで、Unity ならどんなアニメーションでも実現できます。



ランタイムリギングで環境に反応するアニメーションキャラクター

寄稿者

Pete Jepson 氏は、プロのチューターとして、過去10年間、映画制作、アニメーション、写真、グラフィック デザイン、ゲームデザインなどのクリエイティブメディア科目を教えてきました。Unity で 10 年もの経験を 持つ彼は、Udemy.com にて複数のデザインコースを提供しています。

Unity の貢献者

Dave Hunt はコペンハーゲンの Unity でアニメーションとリギングを担当するテクニカルアーティストです。 再現可能なパイプラインとシームレスな相互運用性によって、アーティストやアニメーターをスキルアップする ことに情熱を注いでいます。シアトルで 18 年間ゲーム開発に携わった後、2017 年に Unity に入社しました。 ゲーム開発会社 Bungie でリギングテックアートチームのリーダーとして『Halo』シリーズや『Destiny』 シリーズのゲームに貢献する傍、ワシントン大学のアニメーションキャップストーンプログラムで非常勤講師も 務めました。



序文:Dave Hunt

アニメーションは、クリエイティブ表現の最も純粋な形のひとつと言えるでしょう。よく出来たアニメーションは、 まるで生きているかのような錯覚を起こします。Frank Thomas 氏は、自身の名著『The Illusion of Life: Disney Animation(生命を吹き込む魔法)』の中で、ディズニーの黎明期に、彼自身と Ollie Johnston 氏、 そして同僚たちが、今日でもプロのアニメーターが実践しているアニメーションの基本原理をいかに開拓し、 体系化したかについて書いています。

私も子供の頃、おもちゃのロボットや粘土や木で作った人形でストップモーションのアニメーションを作り、 その楽しさを知りました。自分のアイデアに命が吹き込まれるのを見たときの爽快感は、私をさらに意欲的な プロジェクトへと駆り立て、最終的にはワシントン大学のコンピューターアニメーションキャップストーン プログラムへと導いてくれました。自分が選んだ道が、カリフォルニア州バーバンクにある Frank Thomas 氏の 自宅での彼との出会いにつながり、そこで彼から直接自分の学生アニメーションのプロジェクトを評価して もらうことになるとは、思いもよりませんでした。何がアニメーションの犬を本物の犬のように見せるのかを 調べ、それを表現する 91 歳の彼の瞳が若々しく輝いていたのを、私は決して忘れることはないでしょう。 彼は生き物を常に観察し、その生き物の動きの美しい繊細さに注意を払うことで、それを自身のアニメー ションに取り入れていたのです。

真の巨匠と出会ってから、私はアニメーションの楽しさを他の人とも分かち合いたいと思うようになりました。 幸運にも私は、ワシントン大学のキャップストーンプログラムで 15 年以上アニメーションを教える機会に 恵まれながら、2000 年からは、本業でアニメーターやゲーム開発のテクニカルアーティストとしても働いて います。そして私が思うのは、AAA のゲームスタジオで働いていても、インディーゲームジャムで少人数の 仲間と作業をしていても、アニメーションから得られる創造の喜びは同じだと言うことです。

私はこの eBook の著者と協力し、初心者からプロの方までが、Unity のアニメーションの扉を開くお手伝いが できることをとても嬉しく思っています。Unity のアニメーションでは、インタラクティブなアニメーションが クリエイティブな表現にさまざまな可能性を与えてくれます。 私は今でも、親友の Jay と初めて Unity 3 用のカエルのキャラクター、Lerpz を使った三人称視点の プラットフォーマーのチュートリアルをダウンロードしたときのことをよく覚えています。私たちは数行の コードが記載された本を持っていました。それは、歩行のアニメーションのクリップを再生しながらプレ イヤーの入力を受けとり、キャラクターを前進させるために必要なコードです。そのおかげで、想像力が 許す限り、独自の物語や世界、インタラクティブな体験を生み出すことができたのです。私たちはふたり とも、当時すでにベテランのアニメーターであり、ゲーム開発者でもありましたが、それでもその簡単な 解説文は、この新しいコンテキストでのスタートを切るのに大いに役立ちました。Unity のアニメーション ツールを自分のものにするために何が必要なのか、ぜひこの eBook のページから見つけてください。 あなたが世に送り出すクリエイティブな作品を目にするのを、とても楽しみにしています。

意欲的なプロダクション向けのアニメーション制作

多くのゲームには、多種多様なキャラクター、ストーリー展開、多面的な環境などが盛り込まれています。 そして、それらすべては、同様に多種多様な開発者やアーティストのチームによって作られていることが 多々あります。このような意欲的なプロジェクトでは、プロジェクトに適したツールをコンテンツ制作者に 提供し、何千ものアニメーションを管理できるようにする、強固なアニメーションパイプラインが役立ちます。

キャラクターのコンセプトからモデリング、リギング、アニメーションへと移行するプロセスは、ゲーム開発に おけるキャラクター制作の一連の流れとしてよく知られています。しかしそれでは完璧ではなく、一般的に 作品作りはそれほど単純なプロセスではありません!

例えば、コンセプト段階で、キャラクターの脚が長すぎて自然な運動サイクルが作れなかったり、腕や胴体の 構造が、リアルな武器の持ち方をするには向いていなかったりする場合があります。理想としては、キャラ クター作成プロセスにおいて、キャラクターデザインの非破壊なイテレーションを可能にするべきです。 Dave Hunt と Forrest Söderlind 氏は、GDC 2015 で仕事の組み立て方について紹介しています。彼らは、 大規模なプロダクションに適応したあるシステムを開発しました。このシステムには、ゲームプレイ、ストーリー テリング、そしてワールドビルディングを効率的に組み合わせることを可能にする、Maya のモジュール化 されたプロシージャル制御のリグ構築フレームワークが使用されていました。彼らのプロセスにおける 主なステップは、以下の画像で確認できます。このワークフローは Unity でも実現できます。



Character development



GDC 2015 トークで紹介されたキャラクター開発の主な段階、画像のキャラクターは Unity デモ『Adam』のもの

アニメーターへの支援

ゲーム開発では、多くの場合、テクニカルアーティストが、アーティスト、開発者、デザイナーの効率的な コラボレーションを支えています。

スクリプティング、リギング、テクニカルアートは、自動化されたプロセスによって、大規模なプロダクションで 何千ものファイルの管理を可能にする分野です。

ここでは、アニメーションファイルを整理するための一般的なヒントをいくつか紹介します。ただし、すべての プロジェクトはそれぞれ異なるため、常に自身のプロダクションに適したパイプラインを作ることを目標に してください。

- 命名規則:キャラクターは、多くのオブジェクト、ジオメトリ、ボーン、アクセサリから構成されています。
 チームメンバー全員が階層を操作する方法を理解できるよう、名称を標準化するのがおすすめです。
 シンプルさと読みやすさの丁度良いバランスを見つけましょう。標準化されたわかりやすい命名規則は、
 チームのアニメーター向けにカスタムツールが作成された場合にも役に立ちます。
- ー シーン構成:あなたが Unity 上で直接作業しようとしているアニメーターなら、サンドボックスシーンの 作成や、一般的なゲームプレイアニメーション向けのプレハブモードでの作業を検討してみてください。
- アセットバージョンの追跡と自動化:Unity の AssetPostprocessor クラスは、アセットインポート時に コードを実行したり、Presets で自動インポート設定を適用したりできます。これは、アセットがチームの 基準に準拠しているかどうかを効率的に検証するのに役立つため、実際のコンテンツ制作により集中 することができます。



- Unity でのモックアップ作成と FBX Exporter の使用:デザイナーは、大まかな意図とタイミングを 効率的に表現することを可能にする Timeline などのシステムを使用して、Unity でアニメーションや シネマティックシーケンスのモックアップを作成できます。プロトタイプされたアニメーションを更に ブラッシュアップするために、FBX Exporter を介してアニメーターの好みの DCC ソフトウェアに エクスポートすることができます。
- ビジュアライザーとエディターのカスタムツール: Unity の エディターツールは優れた柔軟性を 提供しており、ビジュアルコントロールリグや、アーティストが Unity のアニメーションツールの多くの API を使いやすくできるその他のツールなど、必要に応じてカスタムインターフェースを作成することが 可能です。
- Animation Rigging パッケージを活用した Unity での IK: ランタイムリギングを有効化すると、 キャラクターをゲームワールドと接触させることができます。表面を手でなぞる、ドアの取っ手を握って 回すといった微妙な動きをリアルに見せるには、ボーンチェーンに修正を加える必要があり、このような ケースは無数にあります。Unity の Animation Rigging パッケージを使用すれば、このような詳細な 動きを作ることができ、キャラクター制作プロセスに大いに役立ちます。

Unity のアニメーションシステム

Unity 初心者の方がエディターの見方や使用方法を学ぶ上で最適なリソースのひとつは、Unity Learn チュートリアルです。特に、3 つのパスウェイコース (Unity Essentials・Junior Programmer・Creative Core) がおすすめです。

Unity マニュアルもアニメーションにおける包括的なセクションを含む重要なリソースです。以下の短い 序文はマニュアルからの引用です。

Unity のアニメーションシステムは、以下の機能を提供します。

- オブジェクト、キャラクター、プロパティを含む Unity のすべての要素に対するアニメーションの簡単な ワークフローと設定
- ー インポートされたアニメーションクリップおよび Unity 上で作成されたアニメーションへのサポート
- ヒューマノイドアニメーションリターゲティング(あるキャラクターモデルのアニメーションを別の モデルに適用する機能)
- アニメーションクリップを整えるためのシンプルなワークフロー
- ー アニメーションクリップ、遷移、およびそれらのインタラクションを確認できる便利なプレビュー
- ー ビジュアルプログラミングツールによるアニメーション間の複雑なインタラクションの管理
- ー 異なるロジックでボディの異なる部位にアニメーションをつける機能
- ー レイヤーおよびマスク機能

このガイドのスクリーンショットは、スクリプタブルレンダーパイプライン(SRP)フレームワークの上に構築 された Unity のマルチプラットフォームレンダリングソリューションであるユニバーサルレンダーパイプライン (URP)を使用して作成されています。URP は Unity で利用可能な3 つのレンダーパイプラインのうちの 1 つで、他の2 つはHD レンダーパイプライン(HDRP)とビルトインレンダーパイプラインです。

アーティスト、テクニカルアーティスト、デザイナー向けの URP、HDRP、および Unity のツールセットに関する 詳細は、以下の eBook をご覧ください。



ダウンロード

ダウンロード

ダウンロード

Unity に アニメーションを インポートする

多くのスタジオでは、Autodesk Maya や Blender などのソフトウェアを使用してアニメーションを作成し、 それを Unity にインポートしています。

大規模なプロのプロジェクトでは、モーションキャプチャを使用することが多く、役者を雇ってコンテンツの リアルな動きを実現するモーションを演じてもらいます。役者は 3D スケルトンに合わせて要所要所に マーカーが付いたモーションキャプチャスーツを着用します。高速赤外線カメラが役者の動きをとらえ、 そのデータは Autodesk の MotionBuilder などのソフトウェアに渡されます。顔の動きも同様にキャプチャ されます。役者の顔全体にマーカーを付け、リアルな顔の動きを記録します。

その後、キャプチャしたアニメーションを Unity にインポートします。

ストックホルムにいる Unity のデモチームは、モーションキャプチャを使い、短編アニメーション『Adam』 『The Heretic』『Enemies』を作成しました。



ゲームやその他のリアルタイム 3D コンテンツにおけるリアルな人間の顔や体の動きは、多くの場合モーションキャプチャに依存している。



このブログ記事では、アニメーターの Krasimir Nechevsky 氏が、Unity で作成された『Adam』における プレビズ、モーションキャプチャ、アニメーションのプロセスについて語っています。

Unity 制作の短編映画『<u>Adam</u>』の制作の舞台裏

また、Apple iPhone と併せて Unity の ARKit を使用すると、顔の動きや表情をキャプチャして、キャラクターの 顔のリグや口の動きをコントロールすることもできます。この方法のデモについては、Unity プロジェクト『The Realtime Rascals』をご覧ください。



キャラクターの顔をコントロールするライブフェイスキャプチャ

モーションライブラリ



Adobe Mixamo モーションライブラリ

モーションキャプチャスーツやモーションキャプチャソフトウェアがない場合は、モーションキャプチャライブラリ から既製のアニメーションクリップを使用することができます。これは、時間を節約し、少ない予算でアニメー ションを作るには効果的ですが、実際の役者の演技と同じレベルの忠実度を得ることは難しいでしょう。

Adobe Mixamo や Reallusion ActorCore は何千もの短いモーションクリップを提供するモーションライ ブラリと呼ばれるウェブサイトで、これらのクリップは Unity を含むお使いの 3D ソフトウェアにダウンロード することができます。

多くの無料キャラクターから選択するのに加えて、自作のキャラクターをアップロードすることもできます。 各キャラクターの骨格構造が異なる場合、ヒューマノイドアニメーションを使用していないと、あるキャラ クターで再生されるアニメーションが別のキャラクターでは再生されない可能性があります。ヒューマノイド アニメーションタイプを使用する場合、まずはヒューマノイドの設定に必要な T ポーズをダウンロードして ください(これについては、このガイドの後述セクションで説明します)。

	Pose	
~	T-pose	~
	Keyframe Reduction	
~	none	~
		DOWNLOAD
	~	Pose T-pose Keyframe Reduction None

DOWNLOAD SETTINGS

Mixamo からモデルをダウンロードする際、Unity 形式の FBX と T ポーズ設定を選択する必要がある。

Unity Asset Store は既製アニメーションを探すのに便利な場所です。アニメーションカテゴリを検索し、 Package Manager から Unity にパッケージをダウンロードします。アニメーションは自動で設定され、 すぐに使用できます。



あらかじめ用意されたアニメーションに加え、プロジェクトの質を高めるための既製ツールもダウンロード可能。Sakura Rabbit 氏(上の画像のキャラクターと アニメーション作成者)は、Final IK、MagicaCloth、Face Capture などの Unity Asset Store パッケージを使用している。



ヒューマノイドアニメーションの 使い方

この動画では、ダウンロードしたアニメーションを Unity に インポートし、ルートモーションを使用してヒューマノイド アニメーションタイプとして設定する方法を紹介しています。 紹介されているワークフローは以下の通りです。

- IK を含む、Animation Rigging を使用した
 アニメーションの修正
- ー レイヤーを使ったアニメーションの合成
- アニメーションフローをコントロールする属性の設定
- ー キャラクターコントローラーの設定
- ー AI キャラクターの設定

Autodesk Maya からアニメーションをエクスポートする

Maya の IK ハンドルは、Maya から Unity にエクスポートできません。Maya のリグによってコントロール されたモーションは、すべてエクスポート前にアニメーションクリップにベイクする必要があります。Maya で 「send to Unity」を選択すると、これを自動で行うことができます。

ただし、この自動エクスポートオプションではさまざまな仮定を作成しますが、すべての状況に当てはまる とは限りません。大手スタジオでは、テクニカルアーティストがスクリプトを使って、自分たちのチームに 最適な方法でエクスポートするためのカスタムツールを作成することがよくあります。



Sakura Rabbit 氏によるキャラクターの Maya でのリギング設定

より詳細なコントロールが必要な場合は、以下の手順に従い、手動でキャラクターをエクスポートしてください。



- 1. エクスポートしたいキャラクターを選択します。
- アウトライナーから腰をクリックしてスケルトンの ルートノードを選択します。
- 「Select」>「Hierarchy」から連結されたすべての ボーンを選択します。
- 4. Shift を長押ししながら、キャラクターのメッシュ オブジェクトを選択します。
- 5. 「File」>「Export Selection」に移動します。FBX ファイルとしてエクスポートすることを選択します。
- ここで、アニメーションをエクスポートすることも できるようになりました。チェックを入れないと、 モデルのみがエクスポートされます。キャラクターが T ポーズなら、より簡単に Unity でヒューマノイド アニメーションの設定を行うことができます。
- アニメーションをエクスポートしたい場合は、 アニメーションをベイクするオプションに必ず チェックを入れてください。開始フレームと終了 フレームを設定できます。これで、3D メッシュと 一緒にエクスポートされるキーフレームアニメー ションファイルに、すべてのモーションがベイクされ ます。

Maya でキャラクター全体を選択



Maya でのエクスポート時にアニメーションをベイクする

Blender からアニメーションをエクスポートする

Blender からエクスポートする場合は、以下の手順に従ってください。

1. エクスポートしたくないオブジェクトは、オブジェクトの横に表示される可視性(目)のアイコンを クリックして非表示にします。エクスポートしたいキャラクターのみが表示されるようにします。

	∑ ~ ⊑•
Scene Collection	
 Collection 	v 💿 🖸
📔 🕨 🔭 Armature 🛛 🎝 🏷 🏷 🗸	00
🕨 📽 Camera 🗐	\smile O
🕨 🖷 Sun 🔆	\sim O

エクスポートしない項目の可視性をオフにする

Operator Presets		~ + -
Path Mode	Auto	× @
Batch Mode	Off	~ =9
In alcosta		
✓ Include		
Limit to	Selected Objects	
	Visible Objects	
	Active Collection	
Object Types		
	Camera	
	Lamp	
	Armature	
	Mesh	
	Other	
	Custom Properties	
\checkmark Transform		
Scale		
Apply Scalings	All Local	~
Forward	-Z Forward	~
αU	YUp	~
	Apply Unit	(
	Use Space Transform	
	Apply Transform	A
Compto		
> Geometry		
✓ Armature		
Primary Bone Axis	Y Axis	~
Secondary Bone Axis	X Axis	~
Armature FBXNode T	Null	~
	Only Deform Bones	
	Add Leaf Bones	
V 🗸 Bake Animation		
	Key All Bones	
	VLA Strips	
	All Actions	
	V Force Start/End Keying	1
Sampling Rate		
Simplify	1.00	
- + Expor	FBXCap	el

ポートオプション

- 「Armature」を選択します。コントロールリグは Blender から Unity にエクスポートできないので選択しないで ください。
- 3. 「File」>「Export」>「FBX」に移動します。
- FBX エクスポートオプションで「Visible Only」を選択します。 これにより、コントロールリグや他の項目はファイル内に エクスポートされません。アーマチュアとメッシュのオブ ジェクトタイプのみを選択します(複数のオプションを 選択するには Shift を長押しして選択します)。
- 「Armature」で「Only Deform Bones」にチェックを 入れ、「Add Leaf Bones」のチェックを外します。変形 ボーン(deform bones)はメッシュを変形させるスケル トンボーンです。「Bake Animation」にチェックが入って いない場合は、メッシュのみがエクスポートされます。 「Bake Animation」にチェックを入れると、アニメーション クリップのキーフレームにベイクダウンされるアニメー ション(コントロールリグにより設定されたあらゆる アクションを含む)がエクスポートされます。ノンリニア エディター内のアニメーションストリップは、個別の埋め込み アニメーションクリップとしてエクスポートされます。



Unity にアニメーションをインポートする



Unity にインポートされたアニメーション付きのキャラクター

FBX はアニメーションが埋め込まれた 3D モデルに適した形式です。ご利用の PC のファイルブラウザから Unity エディターの Project ウィンドウにドラッグすることで、FBX ファイルを Unity にインポートできます。



FBX ファイルの構造 (Unity の Starter Assets - ThirdPerson CharacterController パッケージより)

FBX ファイルには、メッシュ、ジオメトリ、マテリアル、スケルトン、アバター、アニメーションなど、多くの コンポーネントやプロパティが埋め込まれています。



メッシュ

メッシュは、オブジェクトの形状を定義する 3D モデルデータです。Unity では、Mesh アセットがモデルのすべての頂点、法線、 ブレンドシェイプ、UV データを保持して います。

FBX ファイルのメッシュデータ

ジオメトリ

ジオメトリは、メッシュによって駆動される 物理モデルです。これには、シーンでの 見え方を定義するスキンメッシュレンダラー コンポーネントが含まれており、スケルトン にスキンされているため、スケルトンの ボーンの動きに合わせて動きます。スキン メッシュレンダラーコンポーネントのジオメ トリには、マテリアルを割り当てることが できます。

Inspector		a :
🚰 🔽 Armature_Mesh		Static 🔻
Tag Untagged	 Layer Default 	
= I Tronoform		0 -t :
		• • •
Position	x 0 Y 0 Z	0
Rotation	X 0 Y 0 Z	1
Scale		
🔻 🗞 🗹 Skinned Mesh Rende	rer	07‡ :
	合 Edit Bounds	
Bounds		
Center	X -5.960464¢ Y -0.0921425 Z	0.0262486
Extent	X 0.5281845 Y 0.8809299 Z	0.2205018
Quality	Auto	
Update When Offscreen		
Mesh	HArmature_Mesh	
Root Bone	Hips (Transform)	\odot
Materials		3
Element 0	M_Armature_Body	
Element 1	M_Armature_Arms	0
Element 2	M_Armature_Legs	0
		+ -
▼ Lighting		
Cast Shadows	On	•
light Probes	Blend Probes	Ţ
Anchor Override	None (Transform)	
Additional Settings	None (Hunsteini)	Ŭ
Skinned Motion Vectors	~	
Dynamic Occlusion	~	
Rendering Layer Mask	Light Layer default	

FBX ファイルのジオメトリ



マテリアル

マテリアルは、レンダリング時にオブジェ クトの表面がどのように見えるかを定義 するアセットで、Unity シェーダーを使用 してジオメトリに適用されるテクスチャを 組み合わせ、シーンでのモデルの外観を 定義します。

FBX ファイルのマテリアル設定

スケルトン

スケルトンは、一連のボーン(二足歩行 のキャラクターの場合、少なくとも 15個)で、シーン内でのキャラクターの 動きにアニメーションを付けるために メッシュにスキニングされます。スケル トンは、シーンビューとゲームビュー ではデフォルトで非表示になって います。スケルトンには、「Hierarchy」 からアクセスできます。スケルトンを 表示したい場合は、この eBook の 後述のアニメーションリギング セク ションで説明されているボーン レンダラーを使用してください。



FBX ファイルにアニメーションを付けるために使用されるスケルトン

アバター

アバター はキャラクターのボーン レイアウトを定義します。これは、Unity Animator システムを使ってキャラクターや オブジェクトにどのようにアニメーション を付けるかを定義します。一般的なアバ ターは、動物やクリーチャーに適した二足 歩行ではない構造にすることができます。



FBX ファイルのボーン名を定義するヒューマノイドアバターレイアウト



アニメーション

FBX ファイルには、複数のアニメーションクリップ を埋め込むことができます。各アニメーションク リップには、キャラクターのスケルトンにアニメー ションのキーフレームが打たれたモーション データが保存されています。

ジオメトリなし(ただしアニメーションあり)の FBX ファイルをエクスポートすると、三角形の アイコンで示されたスケルトンとアニメーション クリップが含まれます。

FBX ファイルに埋め込まれたアニメーションクリップ



アニメーションのみの FBX ファイル

FBX インポートルートオブジェクト

FBX インポートルートオブジェクトは、FBX ファイルの設定を含むメインの FBX オブジェクトです。



FBX ファイルのインポート設定

Unity マニュアルで紹介されているその他のリソース:

- ー 「Model」タブ
- ー 「Rig」タブ
- 「Animation」タブ
- 「Materials」タブ



Unity では、キャラクターのアニメーションのコントロールに、主にジェネリックとヒューマノイドという 2 つの アニメーションタイプを使用しています。ヒューマノイドは、すべての二足歩行のキャラクターに適用できる 汎用的な構造を提供します。ジェネリックは各キャラクターの骨格構造に固有のものです。アニメーション カーブは、位置、回転、スケール、および正規化されたマッスルスペースです。

Inspector		a :
Armature Import S	Settings	ᅷ᠄
		Open
Model R	ig Animation Materials	
Animation Type	Generic	▼
Avatar Definition	Create From This Model	-
Root node	None	-
Skin Weights	Standard (4 Bones)	-
Strip Bones	 Image: A set of the set of the	
Optimize Game Objects		
	Revert	Apply

FBX ファイルのリグ設定

ジェネリックアニメーションタイプは、アニメーターがフォワードキネマティクス (FK)を使用して Unity 内で 直接アニメーションのキーフレームを打ちたい場合や、異なる骨格構造を持つキャラクター間でアニメーション を共有する必要がない場合に使用できます。 自分の FBX ファイルをインポートすると、デフォルトで**ジェネリック**に設定されます。このアニメーション タイプは、そのキャラクターと骨格レイアウトに固有のものです。ジェネリックアニメーションタイプを使用した 場合、異なる骨格設定を持つキャラクター用の他のアニメーションは、このキャラクターでは機能しない 可能性がありますが、ジェネリックアニメーションは、同じ骨格レイアウトを持つキャラクターでは問題なく 機能します。

アニメーションのタイプは、FBX ファイルの「Rig」タブのドロップダウンメニューからいつでも変更できます。 アバターがまだない場合は、このモデルからアバターを作成することもできます。T または A ポーズのモデル からアバターを作成するのが最適です。

Ν	lodel Rig Animation Materials	
Animation Type	Generic	•
Avatar Definition	Copy From Other Avatar	•
If you have already created an A its Avatar definition. With this option, this model will no	Avatar for another model with a rig identical t ot create any avatar but only import animatic	o this one, you can copy ons.
Source	🖈 ArmatureAvatar	\odot
Skin Weights	Standard (4 Bones)	-
Strip Bones	~	
		Revert Apply

アニメーションのみの FBX ファイルのリグ設定

アニメーションのみの FBX ファイルは、デフォルトではジェネリックアニメーションタイプを使用します。FBX キャラクターファイルからアバターを参照し、アニメーションをそのキャラクターのスケルトンに正しく合わせる 必要があります。

クリップセクションの下にある「+」アイコンをクリックすると、FBX インポート設定の「Animation」タブで アニメーションクリップを定義できます。同じクリップの中に複数のモーションがあることがありますが、 その場合はそれぞれのモーションを独立したクリップに分ける必要があります。何も定義されていない場合、 クリップ名は「Take01」となります。

Clips			Start	End
Idle			1.0	149.0
ldle (1)			1.0	90.0
			+	
Idle (1)				0:
Length 2.967				30 FPS
Î I ^{1:0,0}	2:00	j®:qc , , , , ,	 4:00 , ,	
Start 1			End	90
Loop Time				

アニメーションの複製と長さの調整

インまたはアウトのマーカーを調整するか、フレーム番号を Start または End スロットに入力することで クリップのタイミングを変えることができます。「Apply」をクリックするとキャラクタープレハブにアニメー ションクリップをインポートできます。

また、クリップをループさせることや、ループがシームレスでない場合はサイクルオフセットを設定することもできます。これを数フレーム調整することで、クリップの最初と最後のキーフレームを確実に一致させることができます。

アニメーターコントローラー



Unity の アニメーションステートマシン

アニメーターコントローラーは、キャラクターのすべてのアニメーションの保存と駆動を行います。「Project」 タブを右クリックし、「Create」>「Animator Controller」からアニメーターコントローラーを作成します。 キャラクターがアニメーターコントローラーを保持するには、Animator コンポーネントが必要です。

🔻 ≻ 🗹 Animator		Ø	- 1 -	:
Controller	🖫 Controller			\odot
Avatar	* Armature Avatar			\odot
Apply Root Motion				
Update Mode	Normal			▼
Culling Mode	Cull Update Transforms			•
Clip Count: 0 Curves Pos: 0 Quat: 1 Curves Count: 0 Con	0 Euler: 0 Scale: 0 Muscles: 0 Generi stant: 0 (0.0%) Dense: 0 (0.0%) Stre	c: 0 am:	PPt 0	r: 0



より手っ取り早い方法は、Project ビューの「Hierarchy」にあるキャラクタープレハブにクリップをドラッグする ことです。これにより、キャラクターに「Animator」コンポーネントが追加され、アニメーターコントローラーに アニメーションクリップが追加されます。

コントローラーには、「Window」>「Animations」>「Animator」から Animator ウィンドウを開くか、 Project ビューの「Animator Controller」をダブルクリックすることでアクセスできます。これで、アニメー ションクリップをドラッグしてキャラクターに割り当てられるようになりました。アニメーションが、キャラクター に設定されているアニメーションタイプと一致していることを確認してください。ジェネリックキャラクター にはジェネリックアニメーション、ヒューマノイドキャラクターにはヒューマノイドアニメーションを割り当てる 必要があります。



Animator ウィンドウレイアウト

上記の Animator ウィンドウのスクリーンショットにあるオレンジ色のブロックは、アニメーションステート ノードです。これは、最初に再生されるデフォルトのアニメーションです。ノード上で右クリックし、「Make Transition」を選択し、選択したアニメーションノードに移動することで、特定のノードから次のノードへの 遷移を追加します。

上の例では、Idle から Walk_N に移動して再び戻るといった、一定のループを作り出しています。ジャンプの 他の 2 つのノードは、Idle または Walk_N への直接遷移がないため、再生されません。アニメーションを アクティブにするには、オレンジノードから生じるデフォルトステートに接続された任意のノードからの遷移、 または「Any State」ノードからの遷移が必要です。

遷移線をクリックすると、いつどのように遷移が発生するかの条件を含む、その遷移の設定が表示されます。



2 つのアニメーション間の遷移のブレンド時間

上の画像で、赤枠で強調されている小さな青い部分は、2 つのアニメーションのブレンド時間を示しています。 ブレンドが長ければ長いほど、遷移は滑らかになります。これにより、あるアニメーションから次のアニメー ションに切り替わる際に、目立つノイズを防ぐことができます。

2 つの青い矢印をドラッグすることで、2 つのアニメーションのブレンド時間を伸縮できます。

「Has Exit Time」オプションを選択すると、最初のアニメーションが完全に再生されてから次のアニメーションに遷移します。条件が設定されていない場合は、これを選択してください。

パラメーター

アニメーターコントローラーは、プレイヤーが特定のボタンを押すなどしてアニメーションを動かすことが できるような、インタラクティブなものである必要があります。インタラクティブにするには、コントローラーに パラメーターを追加する必要があります。パラメーターとは、アニメーション遷移で条件として設定できる変数 入力のことです。

「+」アイコンをクリックして「Walking」というパラメーターにブーリアンを追加します。「Jump」という 「Trigger」パラメーターを設定することもできます。



Animator ウィンドウでパラメーターを作成

次に、Idle から Walk_N への遷移で、「Walking」の条件を「true」に設定します。「Has Exit Time」オプションの 選択を解除します。すると、Idle アニメーションを完全に再生する必要なく、すぐに遷移が行われます。



Walking パラメーターからブール条件が設定され、遷移が生じるタイミングをコントロールする。

Walk_N から Idle へ戻る遷移で、「Walking」の条件を「false」に設定します。

キャラクターのパラメーターをコントロールするには、C# スクリプトが必要です。



スクリプトが Animator の 2 つのノードの遷移をコントロールする仕組み

スクリプトは、キャラクターをコントロールするために、「Start()」内で「animator = GetComponent<Animator>();」を使用して、キャラクターの Animator コンポーネントを取得します。 Update メソッドの間は、プレイヤーが W または上矢印キーを押すのを待っている状態です(これは垂直 方向の入力です)。キーが押されると、Animator の歩きのブーリアンが「**true**」に設定されます。これにより、 Idle から Walk_N への遷移が行われます。

プレイヤーが上矢印キーを押している間、Walk_N のアニメーションが再生され続けるようになります。 プレイヤーが上矢印キーを押すのをやめると、Animator の歩きのブーリアンが「**false**」に設定されます。 Walk_N から Idle への遷移が行われ、プレイヤーが再び上矢印キーを押すまで Idle アニメーションが 再生され続けます。

また Update メソッドは、プレイヤーがジャンプキーを押すのを待ちます(デフォルトでは、これはスペースキーに 設定されています)。キーが押されると、Animator はジャンプトリガーを実行し、スペースバーが押される度に ジャンプアニメーションを1回再生します。

ジャンプのアニメーションには、休止と歩行のいずれのステートからもジャンプをする可能性があるため、 両方からの2つの遷移が用意されており、両方の遷移の条件が「Jump」に設定されています。



Walk_N と Idle の両方からジャンプが起こるタイミングをコントロールするために、トリガーパラメーター条件が設定される。

これを実行するもうひとつの方法は「**Any State**」ノードからの遷移を持つことです。これは、Animator 全体の どのクリップからでも呼び出せることに注意してください。アニメーションループをアクティブに保つため、 完了すると Idle に戻る遷移があります。



いずれかのアニメーションノードからの遷移のために Any State ノードを使用

アニメーションノードをクリックすることで、再生速度を調整できます。また、「Tag」フィールドで特定のアニメーションタグを割り当てることができ、コーディング時に便利です。下の例では、Walk_N アニメーションには モーションタグがありますが、Idle アニメーションにはありません。これにより、コードはキャラクターが動いて いるときを識別できるようになります。



設定タグを含むアニメーションステート設定

AnimatorStateInfo 構造体を使用して、スクリプトからアニメーションを監視します。これは、Animator の 現在のアクティビティに関する情報を毎フレームスクリプトに渡します。下の例では、Update メソッドで、 Animator から Base Layer (レイヤー0)の情報を取得しています。

モーションタグの付いたアニメーションクリップが再生されると、「transform.Translate」コマンドを使って キャラクターを前進させます。



コードで Animator を監視

AnimatorStateInfo 構造体を使用すると、アニメーションの再生や遷移のタイミングを確認でき、 Animator コンポーネントによりコードのタイミングを計る強力な方法を提供してくれます。

レイヤー

キャラクターのアニメーションの分割には、レイヤーを使うこともできます。キャラクターのボディパーツに それぞれ異なるアニメーションを使うことができ、複数のアニメーションを組み合わせて独自の新しいアニメー ションを作ることが可能です。

基本的な歩きのアニメーションは、Animator 内の Base Layer と呼ばれるレイヤー 0 にあります。



アバターマスクを使って、どのボディパーツに 基本的な歩きのアニメーションを使い、また 別のパーツに別のアニメーションを使うかを 定義できます。レイヤーシステムと併用する ことで、特定のアニメーションの影響を受ける 特定のボディパーツを除外することができ ます。上半身と下半身を使い分けるなど、 モーションを組み合わせることもできます。

アバターマスクを作成するには、Project ウィンドウを右クリックし、「Create」> 「Avatar Mask」を選択します。Inspector には、キャラクターが使用するアニメーション のタイプに応じて、2 つのタイプが表示され ます。

基本の歩行アニメーション

ヒューマノイドは、キャラクターの視覚的な ガイドを提供し、キャラクターがヒューマノ イドアニメーションタイプを持つ場合にのみ 使用されます。ヒューマノイドセクションで ボディパーツをクリックすると、そのパーツが 除外されます(除外されたパーツは赤くなり ます)。



ヒューマノイドアバターマスクの設定

*	Avatar Mask (Avatar Mask)			ଡ ‡ Open
► Hur	manoid			
▼Tra	nsform			
Use	e skeleton from	TrmatureAvatar		\odot
		Import skeleton		
٩				
Us				
~	▼ Geometry			
~	Armature_Mesh			
~	V Skeleton			
	▼ Left_Foot			
_	✓ Left_Toes			
	Left_ToesEr	id		
	Right_UpperLeg			
	Right_LowerLeg			
	Right_Foot			
	Right Tees			
	Right_Toese	:na		
	= Chost			
	▼ Left Shoulder			
		rm		
÷	v Left Low	erArm		
, i	⊳ Left Ha	and		
	► Neck			
~	Right_Upper	Arm		
~	▼ Right_Lov	verArm		
~	► Right_F	land		
	Toggle All	Collapse All	Expand All	

「Transform」セクションは、キャラクターの 骨格構造のツリーレイアウトを提供し、 ジェネリックアニメーションタイプと共に 使用されます。キャラクターのアバターを 追加し、そのアバターからスケルトンを ロードする必要があります。

ツリーを展開してすべてのボーンを表示し、 トグルボタンでオン/オフを切り替えることが できます。ベースアニメーションをオーバー ライドするように設定するアイテムを選択し ます。未選択のアイテムが除外されます。 つまり、ベースアニメーションがそれらの ボーンに使用されます。

上の例では、キャラクターの腕だけが選択 されているので、新しいアニメーションが 腕を動かし、ベースアニメーションがキャラ クターの残りの部分を動かします。

ジェネリックアバターマスクの設定

Animator に新しいレイヤーが追加され、腕が上がった状態でのジャンプアニメーションが使用されます。 遷移は Base Layer と同様です。アバターマスクが、歯車の設定アイコンをクリックしてアクセスできる Mask スロットに適用されます。

このレイヤーはウェイトでコントロールできます。ウェイト 0 は Base Layer への影響がないことを意味します。 ウェイト 1 は完全な強度を持ち、Base Layer のボーンを完全にオーバーライドすることを意味します。 ウェイトを変化させて面白い結果を得たり、アニメーションとアニメーションの間の滑らかなブレンドを 実現したりすることができます。



新規レイヤーを追加し、アバターマスクでコントロール

括弧内のパラメーター(レイヤーインデックス, レイヤーウェイト)を調節することで、C# スクリプトでレイヤーの ウェイトをコントロールできます。以下のコード例では、Arms レイヤーのインデックスが 1 で、Shift を押した ときのウェイトが 1、Shift を離したときのウェイトが 0 であることを示しています。



アニメーションレイヤーのウェイトをコントロールするコード

再生モード中、Shiftを押しながら歩くと、スクリプトは2つのアニメーションを組み合わせます。





ブレンドツリー

2 つ以上のアニメーションを、入力パラメーターに基づいて動的にブレンドする必要がある場合は、ブレンド ツリーを使用します。Unity が作成した Starter Assets – ThirdPerson パッケージは、ブレントツリーを 使用して三人称視点のキャラクターのモーションをコントロールします。

Animator Controller ウィンドウを右クリックし、コンテキストメニューで「Create state」>「From New Blend Tree」を選択してブレンドツリーを追加します。

これをダブルクリックしてブレンドツリーの設定を開きます。ブレンドツリーを閉じたい場合は、左上のボタンをクリックします。

➤ Animator	
۲	Base Layer 🔷 Blend Tree 🔷
+	
\$	

ベースレイヤーをクリックしてブレンドツリーを閉じる。

ブレンドツリーノードをクリックして、ブレンドしたいアニメーションを追加します。

下の例では、休止、歩き、走りの 3 つのアニメーションをブレンドしています。これは、W または上向き矢印を 押すと 0.0 から 1.0 になる Float 入力によって引き起こされます。その数字に基づいてアニメーションが ブレンドされます。



ブレンドツリーの設定

ブレンドツリーは、1 つの入力に基づくもの (1D ブレンド)と、2 つの入力に基づくもの (2D ブレンド)があります。例えば、WASD キーを押すと、キャラクターの動きを水平方と垂直の両方向でコントロールできます。

	Base Layer Bl						Blend	Tree				0:.
							Blend) Simple Dire			
									Horizontal	▼ Vert	tical	
+•				ſ	le	dle						
					Blend Tree							
						'alk_N						
					Blend Tree							
0		Blend Tree			v	Valk_L						
			Idle •		Blend Tree			-	D. V	D		
0			Walk_L •				Motion		Pos X	POS Y	1	
			Walk_R • Walk_N •		~ ~ ~	waik_R	Walk	UN (
0	Horizont	•	0		Blend Tree		= 🔺 Walk					
vertical	Vertical -		NaN		Wa	Walk N	A Walk	(JR 0 (N) 0		0	1	
				×	Bland Tree		- walk		9 0	L	+ +	
					Biend mee							
							Adjust Tim	ositions le Scale	Select			÷.
		Base Layer Bi	Base Layer Blend Tree Blend Tree Blend Tree Horizonta Vertical	Base Layer Blend Tree Blend Tree Blend Tree Blend Tree Walk, N Walk, N Walk	Base Layer Blend Tree + Base Layer Blend Tree Blend Tree Blend Tree Walk N Walk N Walk R Walk R Walk R Walk N Walk N Wal	Base Layer Blend Tree Base Layer Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk N Walk N Blend Tree Walk N Blend Tree Blend Tree	Base Layer Blend Tree Idle Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Blend Tree	Base Layer Blend Tree Base Layer Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Walk_L Blend Tree Walk_R Walk_R Walk_N Blend Tree Compute f Blend Tree Compute f Compute f Compu	Base Layer Blend Tree Base Layer Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Walk_L Blend Tree Walk_R Walk_R Walk_N Blend Tree Walk_R Blend Tree Compute Positions Adjust Time Scale	Base Layer Blend Tree Bend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Walk_L Blend Tree Walk_L Blend Tree Walk_R Walk_R Walk_R Walk_N Blend Tree Compute Positions Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Compute Positions Blend Tree Compute Positions Compute Positions Compute Positions Compute Positions Compute Positions Compute Positions Compute Positi	Base Layer Blend Tree Bend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Walk_L Blend Tree Walk_R Walk_R Walk_R Blend Tree Compute Distance Walk_N Blend Tree Compute Distance Compute Distance Select	Base Layer Blend Tree Bend Tree Blend Tree Blend Tree Blend Tree Blend Tree Blend Tree Walk_N Blend Tree Walk_L Blend Tree Walk_R Walk_N Blend Tree Walk_R Blend Tree Walk_N Blend Tree Walk_N Blend Tree Compute Descentional Compute Descentional Compute Descentional Compute Descentional Compute Descentional Compute Descentional Compute Descentional Compute Descentional Compute Descentional Blend Tree Walk_N Blend Tree Walk_N Blend Tree Walk_N Blend Tree Walk_N Blend Tree Compute Descentional Compute Descenti

シンプルな 2D ブレンドツリー

2D ブレンドツリーには、異なる方向に動かすために 2 つのパラメーターが必要です。X 位置 (水平方向)と Y 位置 (垂直方向)の最大値を設定できます。

上の画像は、Walk_N のアニメーションの再生速度を -1 に設定して再利用することで、後ろ向きに歩く アニメーションを作成していることを示しています。

コードでは、WとSキーまたは上下矢印のキーにより入力された軸の値を Animator の Float パラメーター (Input GetAxis("Vertical"))に渡せるようになりました。これは、1D ブレンドシェイプのブレンドをコント ロールします。



コードによるブレンドツリーのコントロール

2D ブレンドには、2 つのパラメーターを入力する必要があります。



2D ブレンドコード

キャラクターコントローラー

Character Controller コンポーネントはプレイアブルキャラクターをコントロールするために使用され、 登り勾配の制限や段差の高さの制限といった多くのプロパティを備えています。一般的に、Character Controller コンポーネントは、リジッドボディ物理演算を使用しない、三人称視点または一人称視点の プレイヤーコントロールに使用されます。

Starter Assets - ThirdPerson パッケージの他にも、Unity が作成したもうひとつのパッケージ、Starter Assets - FirstPerson を Unity Asset Store からぜひダウンロードしてください。どちらもパッケージも、 シーンのプロトタイプを素早く作成できるよう、設定済みのキャラクターコントローラーを提供しています。



Starter Assets - ThirdPerson (三人称)

Starter Assets - FirstPerson (一人称)

ここからは、プレイアブルキャラクター用に自分のキャラクターコントローラーを設定する主な手順を説明 します。

- 3 つのアニメーションクリップ(前に歩く/走る、休止する、後ろに歩く/走る)を含む新規ブレンドツリーを 作成します。「Automate Thresholds」のチェックが外れていることを確認してください。しきい値を、 walk backward(後ろに歩く)は -1、idle(休止)は 0、walk forward(前に歩く)は1 に設定します。
- 2. 結果は以下のようになります。
 - a. Wキーを押して縦軸を1にし、キャラクターを前に歩かせる。
 - b. S キーを押して縦軸を -1 にし、キャラクターを後ろに歩かせる。
 - c. A キーを押して横軸を -1 にし、キャラクターを左へ歩かせる。
 - d. D キーを押して横軸を1にし、キャラクターを右に歩かせる。

これでブレンドツリーは、垂直軸と水平軸に正しい動きを合わせるようになります。
3. モーションをコントロールするために、「Vertical」という Float パラメーターを作成し、下記の画像の ように、1D ブレンドのパラメーターとして設定します。



プレイアブルキャラクター向けにツリーアニメーションをブレンドする

これには、シンプルなキャラクターコントローラーの動きを使って、キャラクターの動きをコントロールする スクリプトが必要になります。



プレイヤーの動きと回転をコントロールする基本的な動きのスクリプト

- 4. スクリプトと Character Controller コンポーネントをキャラクターに追加します。
- Step Offset を調整して、キャラクターが段差を歩けるようにします。キャラクターが非常に急な段差を 登る必要がある場合は、この数値を高くします(ただし、高くしすぎると、小さい壁なら踏み越えてしまう 可能性があります)。
- Slope Limit プロパティを調整して、キャラクターが傾斜面を歩けるようにします。デフォルト値は 45 度 ですが、キャラクターが急斜面を登れるように、この値を高く設定することができます。90 度以上に 設定すると、キャラクターが建物の壁を歩けるようになります。

- 7. キャラクターの中心に合わせてキャラクターコントローラーを上げるには、Center Y の値を1に設定 します。
- 8. 高さを 2 に設定するとキャラクターの身長に合わせることができます。キャラクターの身長が高い 場合は、この値を大きくします。
- 9. 幅を調整するには、半径の値を変更します。横幅が大きいキャラクターの場合は 1 以上、細い場合は 0.3 以下に設定します。



プレイヤーのキャラクターに追加されたキャラクターコントローラーコンポーネント

Cinemachine は、高度なカメラコントロール、プロシージャルアニメーション、スマートなカメラ動作を提供 することで、ゲームやシミュレーション用のダイナミックカメラシステムの作成プロセスを簡素化する Unity アセットです。詳細については、本書の後半の Cinemachine について取り上げたセクションで説明します。 ひとまず、Unity Registry メニューの「Window」>「Package Manager」から Cinemachine をダウン ロードしてください。

🖬 Package Manager		ذ ت ا :
+ Packages: Unity Registry Sort: Name		
All Services		
Alembic		Cinemachine Remove
Analytics		2.9.7 - May 08, 2023 (Release)
Android Logcat		From Linky Sector by Decomposition
Animation Rigging		com unity regions of pointy reconsidered inc.
Apple ARKit XR Plugin		
AR Foundation		
Authentication		Vescription version History Dependencies Samples
Build Automation		
𝔗 Burst		New starting from 2.7.1: Are you looking for the Cinemachine menu? It has moved to the GameObject menu,
CCD Management		MODITANT NOTE: If you are upproving from the langey deset Store variation of Cinemachine, delate the Cinemachine seed from your project REFORE installing
Cinemachine	2.9.7 🗸	this version from the Package Manager.
Cloud Code		
Cloud Diagnostics		
Cloud Save		
Code Coverage		
Collections		

Package Manager の Cinemachine パッケージ

「GameObject」>「Cinemachine」>「Virtual Camera」からプレイヤー追従カメラを追加します。 「Follow」フィールドと「Look At」フィールドにキャラクターを追加し、値を調節して Scene ビューのカメラを 設定します。



追従カメラの Cinemachine バーチャルカメラ設定

これで、ゲームをプレイする際に、WASD または矢印キーでキャラクターを操作できるようになります。



キャラクターコントローラーが設定されたプレイアブルキャラクター



三人称用のスターターアセットパック

Unity でアニメーションを 作成する

Unity では、どんなオブジェクトにも直接アニメーションを付けることができます。「Window」>「Animation」>「Animation」からAnimation ビューを開きます。

FBX ファイルに埋め込まれたアニメーションは読み取り専用です。編集するにはクリップを複製する必要が ありますが、クリップを複製すると、そのアニメーションタイプを変更できなくなるので注意してください (アニメーションタイプを変更するには、FBX ファイル内の元のクリップにアクセスする必要があります)。

アニメーションを選択し、Ctrl + D または「Edit」>「Duplicate」で複製します。



アニメーションの複製

複製したアニメーションをキャラクターのアニメーターコントローラーに追加して編集します。Idle ノードを クリックし、新しいアニメーションファイルを「Motion」フィールドにドラッグします。

≙ :	Inspector		a :
Auto Live Link	Idle Tag		≠ i
	Motion	🔺 ldle	0
	Speed	1	
Run_N			✓ Parameter
	Motion Time		Parameter
	Mirror		Parameter
T A	Cycle Offset	0	Parameter
	Foot IK		
	Write Defaults	~	
Idle	Transitions		Solo Mute
	= Idle -> Run_N		

新しいクリップを Motion スロットにドラッグしてアニメーションクリップを入れ替える

キャラクターを選択すると、Animation ビューから直接アニメーターコントローラー内のアニメーションに アクセスできるようになります。これらは左パネルのドロップダウンメニューから選択できます。複製された クリップは編集可能ですが、埋め込まれたクリップは読み取り専用です。



修正するには埋め込まれたアニメーションを複製する必要がある。

同じドロップダウンメニューから、「Create New Clip」を選択して、新しいアニメーションクリップを作成する こともできます。



キーフレームの表示(左上)とアニメーションオプションの変更(右上)

キーフレームをクリックして選択します。選択したキーフレームは削除、移動、複製、編集が可能です。修正する には「Record」ボタンをオンにしてください。

Animation ウィンドウの右上にある 3 つの点のアイコンをクリックして、オプションを変更します。Timeline を 秒単位またはフレーム単位で表示したり、サンプルレートを 30fps から 60fps に変更したり、その逆も可能 です。

新しいクリップで独自のフォワードキネマティック(FK)アニメーションを作成できます。新しいクリップは、 キーフレームが設定されていない空の状態で作成されます。「Record」ボタンを押してキーフレームの設定を 開始します。



キーフレームが打たれていない新しいクリップ。レコードを押してキーフレームを打ち始める。

キーフレームを設定するには、Hierarchy でボーンを選択し、Inspector で回転を右クリックして「Add Key」を選択します。これで最初のキーフレームが設定されます。



Rotation フィールドの横を右クリックしてボーンにキーを追加する。

ボーンには、キーフレームとキーフレームの間でアニメーションが付くので、ボーンを一定時間同じ位置に とどめたい場合に便利です。



ボーンを回転さえてキーフレームを打つ。

例えば、脚を2秒間静止させたい場合、0に キーを追加し、2秒後に移動してもうひとつ キーを追加します。これにより、脚が静止 した状態になります。これ以降のポイントで 回転を始めることができます。

Scene ビューで回転ツールを使い、ボーン を回転させて新しいキーフレームを作成 します。 「Add keyframe」ボタンから、選択したフレームにアニメーションのすべてのボーンに対してキーフレームを 追加することができます。

Animation				a∶⊡×
Preview 🧿 🖽 🔰 🕨 📕 🛏	30		0:20 0:25 0:30	
New Animation 🔻 💠 🔸	+ +			
Add key	/frame.		•	<u> </u>
Left_UpperLeg : Rotation		•	•	•
🕨 🙏 Left_LowerLeg : Rotation		♦	•	+
A Chest : Rotation			•	•
🕨 🦊 Left_Shoulder : Rotation		•	•	•
Right_Shoulder : Rotation		•	•	•
A did Decements				
Add Property				

「Add keyframe」オプションですべてのボーンに対して特定のフレームに新しいキーフレームを追加できる。

特定のボーンにのみキーを打ちたい場合、リストからそのボーンを選択し、「Filter by selection」ボタンを クリックします。これで、「Add keyframe」ボタンを押すと、それらのボーンのキーだけが追加されます。これは、 手動で各ボーンの回転を右クリックし、キーを追加するよりも速くできます。



選択項目で絞り込むと、選択したボーンだけにキーを追加できる。

X 軸、Y 軸、Z 軸の Rotation、Transform、Scale の値を個別に手動で調整したり、特定の値を入力したり することもできます。

🕒 Ar	nima	tion																			a : t
Previ			144	I		Þ	₩	1		6	0				0:15						
wave										+ +	Į.										
																					•
												•									•
					J. R			-60	0.143			•									•
					J.R			139	9.26			•									•
									149			•									۰
		_	_	Ad	d Pr	oper	y														

X、Y、Z 軸にキーフレームを追加

Animation ウィンドウのボタンで **Dopesheet** ビュー (キーフレームを表示するビュー)から Curves ビューに変更し、個別のキーフレームのカーブをクリックして編集したり動かしたりできます。



下部のボタンで Dopesheet (キーフレーム)と Curves の画面を切り替えることができる。

右クリックでカーブの形を変え、開始位置と終了位置の間のイージングを調整したり、ベジエハンドルを 使ってカーブを設定したりできます。



カーブの編集

デフォルトのカーブには、キーフレーム間の動きにイーズインとイーズアウトが追加される **Clamped Auto** が 設定されています。ゆっくりと始まり、中盤でスピードアップし、終盤で再びスピードダウンするという、流れる ような滑らかな動きを提供します。



リニア接線を選択することでキーフレームからカーブを削除できる

キーフレーム間のイージングがない ようにモーションからカーブを削除する には、キーフレームを右クリックし、 両方の接線を選択し、Linear に設定し ます (これは、左または右からの直線的 なフローだけにカスタマイズすること もできます)。

アニメーションは、あらゆるオブジェクト とそのコンポーネントにアニメーション を付ける他にも、アイテムのオン/オフ、 マテリアルの値やライトの強度の変更、 カメラの視野の調節にも使用できます。 アニメーションの設定は、Inspectorでは赤色で表示され、レコードモードを終了すると青色で表示されます。

Inspector		a :
Main Camera		Static 🔻
Tag MainCamera	 Layer Default 	
🔻 🙏 🛛 Transform		0 ‡ :
Position	X 3.003401 Y 1.888924	Z 2.382371
Rotation	X 15.162 Y -103.373	Z 0
Scale	čə X 1 Y 1	
🔻 💶 🖌 Camera		0 ‡ ;
Render Type	Base	
Projection		0
Projection	Perspective	
Field of View Axis	Vertical	
Field of View		60
Clipping Planes	Near 0.3	
	Far 1000	
Physical Camera		
Rendering		0 i
Stack		9
Environment		0
▼ Output		6

アニメーションが付いた項目は、この画像の位置と回転の値のように、赤または青に変わる。

Animation ウィンドウの便利なショートカット

Animation ウィンドウのショートカットを表示するには、まず「Edit」>「Shortcuts」を開きます。リストから 「Animation」を選択します。

Category	Command	Shortcut	
All Unity Commands	Show Curves	С	
Binding Conflicts	Ripple (Clutch)	2	
Main Menu	Frame All	А	
3D Viewport	Previous Frame	1	
Analysis	Play Animation	Space	
Animation	Toggle Ripple	Shift+2	
Camera	Key Selected	K	
Curve Editor	Next Keyframe	Alt+.	
Grid	Key Modified	Shift+K	
Help	First Keyframe	Shift+,	
Hierarchy View	Next Frame		
Overlays	Previous Keyframe	Alt+,	
ParticleSystem	Last Keyframe	Shift+.	
Profiling			

Windows の便利なアニメーションショートカット

選択されたキー/カーブをフレーミングする	F
すべてのカーブをフレーミングする	A
ズームイン/ブームアウト	マウスホイール (または) ドラッグ Alt + 右クリック
縦にズーム	ドラッグ Alt + shift + 右クリック
横にズーム	ドラッグ Alt + ctrl + 右クリック

Windows 向けの Animation ウィンドウのズームコントロール

Windows で Animation ウィンドウを操作するショートカット

プレビューモードとレコードモード



プレビューモードでは、手動でキーフレームを設定する必要があるプロパティに候補キーを設定できます。 レコードモードでは、クリップにすべての変更が自動的に追加されます。

キーフレームに特定の値を設定する

time	16		
value	3.039258		
		1 · ·	

Curves ビューで、1 つ以上のキーフレームを選択した状態で Enter を押します。ポップアップダイアログが 表示され、特定の値と時間の値を入力することができます(注:これにより接線が修正されるわけではあり ません)。

選択項目で絞り込む

Project C Animation		
Preview 🔘 🛏 I 🕨 🕨 🍽	0	0:00 0:15
Idle 🔻		

≻PlayerArmature : Animator.Chest Front-Back	0.01957 •	(* (((* (* (* (* (* (* (* (* (* (* (* (*
≻PlayerArmature : Animator.Chest Left-Right	-0.0054 •	*******
≻PlayerArmature : Animator.Chest Twist Left-Right	-0.008C •	*******
≻PlayerArmature : Animator.Head Nod Down-Up	0.11357 •	** ** * ** * * * * ** ** ** **
≻PlayerArmature : Animator.Head Tilt Left-Right	0.0049!	************
≻PlayerArmature : Animator.Head Turn Left-Right	-0.0782 •	**** *** ** * * **** * * ***
≻PlayerArmature : Animator.Jaw Close	1.0286 •	•
≻PlayerArmature : Animator.Jaw Left-Right	-7.0577	•
右は、「Filter by selection」ボタンのアイコンで	್ಕ.	

© 2024 Unity Technologies

これにより、Scene ビューで選択されたオブジェクトのプロパティのみを表示します。これを無効にすると、 アニメーションクリップのすべてのプロパティが表示されます。

アニメーションイベント

C# スクリプトで実行する関数をトリガーするイベントを、アニメーションから呼び出すことができす。これらの 関数は、ゲームオブジェクトにアタッチされた、どのスクリプトにも記述することができます。

下の例では、左足または右足が地面に接触した時に水しぶきを発生させ、足音のサウンドを再生する 2 つの カスタム関数があります。



アニメーションイベントによりトリガーされるカスタム機能

Animation ウィンドウでは、アニメーションの正しい位置にイベントのタイミングを合わせることができます。 イベントアイコンをクリックしてイベントを追加します。



新しいイベントを追加するイベントアイコン



Inspector では、トリガーする関数をスクリプトから選択するよう求められます。

アニメーションにイベントを追加

アニメーションを再生すると、足が水たまりに当たった際に水しぶきの効果が発生し、歩行サイクルに合わ せて足音が流れます。



アニメーション再生中にイベントにより発生するコード

上級者向けのアニメーション 機能

読み取り専用アニメーションのイベント

イベントは、FBX ファイル内に埋め込まれた読み取り専用のクリップに追加することもできます。クリップが FBX ファイルから複製されていない場合、Animation ウィンドウで編集することはできません。

FBX ファイルのアニメーションセクションにある Event セクションを開きます。 画面下部の Preview ウィンドウを見ながら、クリップのタイミングに移動できます。 「Add Event」をクリックします。



新しいイベントを追加するイベントアイコン

ここでは、スクリプトに関数名を手動で入力する必要があります。整数、浮動小数点数、文字列など、必要な パラメーターを追加するオプションもあります。

A Walk_N	0 :
Length 0.967	
0:00 10:05 10:	io , , , 10:15 , , 10:20 , , , 10:25 , , 🖻
Start 0	End 29
Loop Time	
Loop Pose	~
Cycle Offset	0
Additive Reference Pose	
Pose Frame	
► Curves ▼ Events	1:33[0:50[0:67 _ ↓]0:83[1:00
Function	RightFootImpact
Float	0
Int	0
String	
Object	None (Object)
▶ Mask	
Motion	
P Import Messages	Revert Apply

読み取り専用クリップにイベントを追加

ルートモーション

多くのクリップは in-place アニメーションで、歩くアニメーションが再生されている間、キャラクターは1か所に 留まります。スクリプトを使うことでキャラクターの動きをコントロールできます。

ー部のクリップには、3D 空間でキャラクターを動かす移動キーフレームが含まれています。例えば、歩行 アニメーションでは、クリップが再生される間、キャラクターは前方に進みます。しかし、クリップがループして 最初に戻ると、キャラクターも原点にジャンプしてしまい、おかしなことになってしまいます。

≻ ✓ Animator		8	규는	:
Controller	🖫 Controller			\odot
Avatar	🕆 ArmatureAvatar			\odot
Apply Root Motion	\checkmark			
Update Mode	Normal			▼
Culling Mode	Cull Update Transforms			▼
Clip Count: 7 Curves Pos: 312 Quat: 318 Euler: 1 Curves Count: 3129 Constant: 226	I Scale: 306 Muscles: 0 Generic: 0 PPtr: 0 67 (72.5%) Dense: 672 (21.5%) Stream: 190 (6.1%)			

Animator コンポーネントに、ルートモーションを適用するオプションがあります。このボックスにチェックを 入れると、クリップがループする際、元の位置にジャンプして戻るのではなく、現在の位置からクリップを 続けるようになります。これにより、キャラクターは自然に前へ歩くようになります。

ルートモーションは、キャラクターのキーフレームとタイミングを合わせて動かすため、効果的と言えます。 足が地面に接触すると、キャラクターは自然な形で前進し、コードでキャラクターを動かした際に起こりうる、 足が滑ってしまう問題を回避できます。

ルートモーションは、巧妙なマルチパンチコンボを繰り出すボクサーのキャラクターや、足が前後左右に 素早く動くような複雑な動きに適しています。

ジャンプアニメーションを含むクリップは、Unity の Timeline を使ってキャラクターの位置をコントロール できるカットシーンアニメーションに便利です。実際のゲームプレイでのジャンプアニメーションでは、キャラ クターが空中に浮かないように、キャラクターの Y 位置をクリップにベイクします。

Root Transform Rotation	1	- sector
Bake Into Pose	✓	loop match 🔴
Based Upon	Original	•
Offset	0	
Root Transform Position	(Y)	
Bake Into Pose		loop match 🔴
Based Upon (at Start	Original	•
Offset	0.2	

ルートモーションのジャンプクリップ用に Y の位置をベイク

Animator コンポーネントにルートモーションを適用

スクリプトを使うことでキャラクターのジャンプをコントロールできます。Rigidbody.AddForce を使用して 最適なジャンプ効果を生み出しましょう。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
 The second secon
public class JumpScript : MonoBehaviour
 {
                  private Rigidbody rb;
                  public float jumpForce = 12f;
                  private Animator animator;
                  private
                   // Start is called before the first frame update
                  ✤ Unity Message | 0 references
                  void Start()
                    {
                                    rb = GetComponent<Rigidbody>();
                                    animator = GetComponent<Animator>();
                   }
                  // Update is called once per frame
                   ✤ Unity Message | 0 references
                  void Update()
                   {
                                     if (Input.GetButtonDown("Jump"))
                                     {
                                                      animator.SetTrigger("Jump");
                                                      rb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
                                     }
```

コードでリジッドボディを使用してキャラクターのジャンプをコントロール

ブレンドシェイプ

ブレンドシェイプは、他の 3D ソフトウェアツールではモーフターゲットと呼ばれることもあり、キャラクターの メッシュ形状に対して頂点を変形させるもので、元のメッシュ形状とブレンドすることができます。

ブレンドシェイプは FBX ファイルにベイクされ、Inspector で微調整できます。下の画像にあるスライダーを 調整することで、ブレンドのウェイトを設定することができます。0 では影響なし、1 では完全にオーバーライド します。複数のスライダーを混ぜて、新しいブレンドモーフを作ることができます。



Inspector のブレンドシェイプスライダー

フェイシャルリグのブレンドシェイプは、『The Heretic』の Gawain のような、より上級者向けのキャラクターに よく使用されます。顔の点をリアルタイムでドラッグして、さまざまな表情を作り出すことができます。これは、 顔の変形に合わせてリアルなシワを追加するリンクルマップを使用するデジタルヒューマンシェーダーと組み 合わされています。





Unity の デモ『The Heretic』のデジタルヒューマンキャラクターの顔に使用されているブレンドシェイプ







ブレンドシェイプにより作成されたさまざまな表情

© 2024 Unity Technologies

衣服のアイテムにブレンドシェイプを追加して、Unity で二次的な動きを作り出すこともできます。この種の ブレンドシェイプは、Blender や Maya などの 3D ソフトウェアで作成できます。



ネクタイの動きアニメーションを付けるために Blender で作成されたシェイプキー





ブレンドシェイプにより 4 つの羽のパーツにアニメーションを付けることができる

ヒューマノイド アニメーションタイプ

ヒューマノイドアニメーションは、人型のキャラクターの大半で機能する必要があるアニメーションに使用 されます。ヒューマノイドアニメーションでは、フォワードキネマティクス (FK)を使って Unity で直接キーフ レームを打つことはできません。代わりに、他のソフトウェアでアニメーションを作成し、Unity にインポート することができます。ヒューマノイドアニメーションは、後のセクションでも説明されるアニメーションリギングで コントロールできます。

ヒューマノイドアニメーションタイプは、ツイストボーンやリーフボーンをサポートするなどの機能により、 一般的なものよりも複雑になっています。ヒューマノイドアニメーションは、ジェネリックアニメーション タイプのように名前付きのボーンに頼るのではなくボーンの位置に対して参照位置を使うため、ほとんどの 人型のキャラクターで機能します。

ヒューマノイドアニメーションタイプを設定するには、ターゲットを最適に照合できるよう、キャラクター モデルが T または A のポーズであることを確認してください。FBX モデルの Rig セクションで、ドロップ ダウンリストから「Humanoid」を選択し、このモデルからアバターを作成します。「**Apply**」をクリックして アニメーションタイプを変更します。

	Model Ri	g Animation	Materials		
Animation Type		Iumanoid			
Avatar Definition		Create From TI	his Model		
The avatar can be con	figured after	settings have be	een applied.		
				Cor	nfigure
Skin Weights		Custom			•
Max Bones/Vertex		•			4
Min Bone Weight	•)			0.001
Strip Bones		·			
Optimize Game Obje	ects				
				Revert	Apply
Asset PostProcess	ors				

キャラクターのヒューマノイドリグの設定

「Avatar Definition」フィールドで「Create from this Model」のオプションを選択すると(上下の画像を 参照)、ヒューマノイドへの変換プロセスにより、選択されたキャラクターの骨格構造が分析されます。ボーンは 主要な位置で識別され、主要な参照位置のスロットに配置されます。例えば、上腕の参照位置では、この キャラクターの上腕のボーンが使用されます。ボーンの名前はもはや関係なく、アニメーションはスケルトンに アニメーションを付けるために参照位置を使用することができます。

変換に成功すると、Rig セクションにメッセージが表示されます。

		Model	Rig	Animation	Materials			
Anima	tion Type		Hu	manoid				•
Avata	r Definition		Cre	eate From Th	nis Model			•
						~	Conf	igure
Q	A Biped was det for this avatar. default Biped se	tected. De Translatio etup.	efault I n DoFs	Biped mapping s have been a	and T-Pose I ctivated. Use	nave be Configu	en conf ire to m	igured odify
Skin V Strip E Optim	Veights Bones Ize Game Obje	ects	Sta	andard (4 Bo	nes)			•
						R	evert	Apply

ヒューマノイドコンバージョンに成功

変換に問題がある場合、Unity は見つからなかったボーンをハイライト表示します。この場合、手動でボーンを 合わせる必要があるかもしれません。キャラクターの骨格が複雑すぎる場合、ヒューマノイドに変換できない 場合があります。

その場合は「Configure」をクリックし、ヒューマノイドアニメーションタイプに合わせてボーンを手動で 設定してください。





ボーンはアバターの主要な参照位置で識別される。



ヒューマノイドアニメーションタイプのキャラクターにジェネリックアニメーションを使用すると、キャラクターは 以下の画像のような設定ポーズになります。

ジェネリックアニメーションがヒューマノイドアニメーションタイプのキャラクター上で再生されると、設定ポーズをとる。

ヒューマノイドアニメーションタイプの変換プロセスは、キャラクターが T または A ポーズの時に最もよく機能 します。これらのポーズでは、Unity が想定する位置にボーンが配置されるからです。キャラクターがアニメー ションのポーズをとっている場合、ボーンが通常とは異なる位置にある場合があり、変換が難しくなります。 T ポーズのヒューマノイドアバターを作成し、このアバターを使ってインポートするアニメーションクリップ ファイルを変換します。

例えば、Mixamo からアニメーションクリップをインポートする場合、変換プロセス中に「Avatar Definition」 フィールドで「**Copy From Other Avatar**」オプションを選択し、T ポーズのキャラクターからアバターを選択 します。

Inspector*						a :
Ninja_idle Imp						
						Open
	Model	Rig	Animation	Materials		
Animation Type		Hu	manoid			
Avatar Definition		Co	py From Oth	er Avatar		
If you have already cr you can copy its Avata With this option, this m	eated an ar definitio Iodel will r	Avatai on. iot cre	r for another n ate any avata	nodel with a r r but only imp	ig identical to t port animations	his one,
Source		*	NinjaAvatar			⊙
Skin Weights		Sta	andard (4 Bo	nes)		
Strip Bones		~				
					Revert	Apply
Asset PostProcess	sors					

アニメーションをヒューマノイドリグに変換し、T ポーズのキャラクターからアバターを使用

T ポーズキャラクターのアバターは、アニメーションクリップファイルと同じ骨格構造を持っていることが 重要なので、アニメーションの作成やダウンロードに使用しているソフトウェアから、アニメーションクリップと 併せて T ポーズモデルもインポートしてください。

これらのクリップをヒューマノイドに変換するために、T ポーズをとった別のキャラクターのアバターを使うのは、 問題が生じるのでやめましょう。

クリップがヒューマノイドアニメーションタイプに変換されると、元のキャラクターだけでなく、ヒューマノイド アニメーションタイプを使用する他の二足歩行のキャラクターの大半でも機能するようになります。



骨格構造の異なる3体のキャラクターで機能する、インポートされた歩行アニメーション

ヒューマノイドクリップは、他のクリップに合わせて変形や回転のオフセットを調整するなど、ジェネリック クリップよりも多くの設定があります。

以下の画像のように、「loop match」オプションがすべての設定に対して緑になっています。これは、最初と 最後のキーフレームが一致し、シームレスなループが作成されていることを意味します。Transform の場合、 これはキャラクターのアニメーションがその場で行われていることを意味し、アニメーションにキーフレームが 打たれたルートモーションがないことを意味します。このタイプのクリップは、コードを使ってキャラクターを 動かす際に便利です。

Loop Time Loop Pose Cycle Offset	マ マ 0	loop match 🍈	
Root Transform Rotation Bake Into Pose Based Upon (at Start)	Body Orientation	loop match 🍐	
Offset Root Transform Position (Y)	0		
Bake Into Pose Based Upon	Center of Mass	loop match 🍮 👻	
Offset Root Transform Position (XZ)	-1		
Bake Into Pose Based Upon	Center of Mass	loop match 🅚	

アニメーション設定とオフセットの使用

このオプションが赤い場合は、キーフレームが一致していないことを意味し、アニメーションがループした際に 動きがカクカクしてしまいます。すべてのクリップがループするように設計されているわけではありませんが、 この赤はアニメーションにキーフレームが設定されていることを意味します。これはルートモーションと いい、ルートモーションのあるヒューマノイドアニメーションを使用している場合は、Animator コンポー ネントで「Apply Root Motion」オプションに必ずチェックを入れましょう。

🔻 ≻ 🗹 Animator	0 ‡	:
Controller	🖫 Ninja_RangeOfMotion	\odot
Avatar	🖈 NinjaAvatar	\odot
Apply Root Motion	 Image: A set of the set of the	
Update Mode	Normal	▼
Culling Mode	Cull Update Transforms	▼
Clip Count: 11 Curves Pos: 763 Qu Generic: 0 PPtr: 0 Curves Count: 7554 Stream: 1032 (13.79	at: 673 Euler: 90 Scale: 763 Muscles: 14 4 Constant: 5902 (78.1%) Dense: 620 (8.29 %)	%)

ルートモーションクリップを使用するためにルートモーションを適用

リジッドボディ物理演算を持つキャラクターは、ルートモーションファイルとの競合を引き起こす可能性があります。これらの競合の対処方法として、「Bake Into Pose」オプションを使用してモーションをクリップにベイクします。



ルートモーションを使ったループと、ポーズにベイクするオプション

「Root Transform Rotation」と「Root Transform Position (Y)」(上下にジャンプするモーション)の設定で、 このオプションにチェックを入れてください。ただし、前に歩く動きと後ろに歩く動きを維持するために、「Root Transform Position (XZ)」の設定で「Bake Into Pose」のオプションをチェックしないでください。

Animation Rigging

ビデオゲームでは、ゲームプレイ中に環境やキャラクターの位置が変化します。ワールド内でキャラクターを リアルに動かすために必要なすべてのバリエーションを予測することは難しいでしょう。だからこそ、実行時に ゲームの変数や特定のニーズに基づいて既存のアニメーションを調整できることが鍵であり、時間の節約にも なります。

Animation Rigging パッケージにより、Unity 内のあらゆるタイプのキャラクターおよびオブジェクトにイン バースキネマティクス (IK) を作成することができます。ジェネリックとヒューマノイドの両方のアニメーション タイプで、Timeline エディター内でも機能します。

インバースキネマティクス(IK)は、アニメーションやロボット工学で使用される技術で、キャラクターの 手や足といった、エンドエフェクターの望ましい位置に基づいて、骨格構造における関節の位置と回転を 決定し、より自然でリアルな動きを実現可能にします。ゲーム内でキャラクターの手をコントロールする 場面を想像してください。インバースキネマティクス(IK)を使用すると、手が希望の位置にくるように腕の 各関節を手動で動かすのではなく、単純に手を伸ばしたい位置を指定するだけでこれを実現できます。 IK システムは、その位置に手を伸ばすために、肩、肘、手首の関節をどのように調整するかを自動的に 計算し、より自然で流れるような動きを実現します。

「Window」>「Package Manager」を開き、左上のドロップダウンメニューから「Unity Registry」を選択 して Animation Rigging パッケージをインストールしましょう。パッケージをインストールしたら、サンプル パッケージをダウンロードすることをおすすめします。サンプルパッケージには、各リギングコンストレイントの サンプルシーンが含まれており、プロジェクトにどのように実装するかを確認することができます。

Package Manager			: 🗆 ×
+ Packages: Unity Registry Sort: Name (asc)			
All Services			
Ads Mediation		Animation Diaging	
Advertisement Legacy			ove
Al Navigation		1.2.1 - January 12, 2023 Release	
Alembic	2.3.3	From Unity Registry by Unity Technologies Inc.	
Analytics		com.unity.animation.rigging	
Android Logoat	132		
Animation Dissing	121.4	Description Version History Dependencies Samples	
Animation Rigging			
Apple ARKIt XR Plugin	5.0.7	Animation Diracing Constraint Samples 207.58 //R	mport
AR Foundation		Import Constraint Samples unit uniques as as as a set of and	npore
Authentication		usage of animation rigging constraints.	
Build Automation			
🔗 Burst			

Package Manager から Animation Rigging をインストール

サンプルは Samples フォルダーにあります。再生モードに入り、Hierarchy でその構造を確認することで、 サンプルがどのように機能するかをプレビューすることができます。



それぞれのコンストレイントがどのように機能するかを示すシーンの例

コンストレイントは定義済みのスクリプトに基づいており、Unity 向けの IK ソリューションをより簡単に 設定できます。また、コンストレイントはモジュール化されています。キャラクターのスケルトンのすべての ボーンをコントロールする単一の IK リグを使用する代わりに、動物やクリーチャーのような二足歩行以外の キャラクター含む、あらゆるキャラクタータイプに合わせて個々のコンストレイントを組み合わせることが できます。IK リグの一部を独立してオフにすることもできます。

コンストレイントには「Component」>「Animation Rigging」からアクセスできます。

Scripts	>		
Cinemachine	>		
UI	>		
Visual Scripting	>		
Animation Rigging	>	Setup	>
Event	>	Blend Constraint	
Input	>	Chain IK Constraint	
XR	>	Damped Transform	
UI Toolkit	>	Multi-Aim Constraint	
Add	Ctrl+Shift+A	Multi-Parent Constraint	
Navigation	>	Multi-Position Constraint	
		Multi-Referential Constraint	
		Multi-Rotation Constraint	
		Override Transform	
		Twist Chain Constraint	
		Twist Correction	
		Two Bone IK Constraint	

リギングコンストレイント

IK リグの設定

キャラクターを選択し、「Animation Rigging」>「Rig Setup」を開きます。これにより、Animator コンポー ネントと Rig Builder コンポーネントが追加されます。このルートキャラクターゲームオブジェクトに Rig Builder コンポーネントを追加すると、そのルートの子となる Rig ゲームオブジェクトが自動的に作成され、 デフォルト(変更可能)で Rig 1 と命名されます(下の画像を参照)。

「Animation Rigging」>「Bone Renderer Setup」からボーンレイアウトを表示することもできます。これは 骨格のレイアウトを視覚的に表現するものですが、Rigging パッケージを機能させるために必須なものでは ありません。このガイドの残りの例では、キャラクターを見やすくするために、Bone Renderer コンポーネントを 省いています。



忍者キャラクターの IK Rig 設定

Rig コンポーネントのモジュラーを追加するには、Rig 1と命名された空のゲームオブジェクトを追加します。 すると、コンポーネントメニューから追加するコンストレイントを選択できるようになります。

Two Bone IK Constraint は、腕、脚、指などへのインバースキネマティック(IK) コンストレイントを追加するのに 使用されます。手を Hierarchy から **Tip** スロットにドラッグします。



Tip オブジェクトを Two Bone IK Constraint に追加

3 つの点のオプションアイコンをクリックし、「Auto Setup from Tip Transform」を選択します。これにより、 接続されたボーンが自動的に見つかり、ターゲットオブジェクトとヒントオブジェクトが正しい位置に追加 されます。



Tip Transform の Auto Setup が Two Bone IK Constraint を自動的に設定

また、ターゲットとヒントオブジェクトを視覚化するエフェクターも追加されます。ターゲットオブジェクトは 手の位置と回転をコントロールし、ヒントオブジェクトは肘の回転をコントロールします。

エフェクターにはさまざまなものがあり、サイズ、位置、回転、色を変更できます。



ターゲットとヒントを可視化するのに使用するエフェクターのタイプ



キャラクターが箱を運ぶ用に位置付けされた Two Bone IK Constraint

両腕に Two Bone IK Constraint を使うことで、 3D 空間に配置することができるようになり ました。アニメーションは腕に影響を与えない まま、体の他の部分は依然としてコントロール します。これは、キャラクターが壁に寄りかかっ たり、ドアを開けたり、箱を運んだりするときに 便利です。 すべてのコンストレイントには、ターゲットへの全体的な影響力を調節するためのウェイトスライダーが あります。強度が1の場合、ターゲットは手の位置を完全にコントロールします。強度が0の場合、手には 何の影響もありません。また、位置ウェイトと回転ウェイトを独立してコントロールすることもできます。

再生モードまたはレコードモード中に、ターゲットを移動または回転させて、キャラクターの腕の位置に 影響を与えることができます。

Multi-Aim Constraint を使って、キャラクターの頭をターゲット(動いているオブジェクトなど)に向ける ことができます。ターゲットはそれらのオブジェクトの親となり、キャラクターは動きながらそれらを見る ようになります。



キャラクターが動いているターゲットを見るように Multi-Aim Constraint を使用



World Up Type フィールドで「Scene Up」オプションを使用して正しい回転を確保する Multi-Aim Constraint コンポーネント

頭の回転が不自然に見える場合は、頭の Aim Axis と Up Axis の向きを確認してください。ローカル空間に いることも確認してください。頭は Y 軸方向を向き、「Up Axis」フィールドが - X に設定されている必要が あります。



頭は Y のプラスをエイム軸に、X のマイナスをアップ軸に使っている。

また、頭の回転を -90 から 90 までに制限したい場合など、最小回転量と最大回転量を設定することも できます。

複数リグ

「Animation rigging」>「Rig setup」から、一体のキャラクターに対していくつでもリグを作成することができます。Rig Builder コンポーネントに追加される 2 つ目のリグが作成されます。Rig 1 のコンストレイントは Rig 2 とは独立して計算されることに注意してください。複数のリグは、2 つ以上のリグを互いに独立させたい 場合に便利です。

フルボディコントロールリグ



『The Heretic』に登場するキャラクターの Gawain は、フルボディコントロールリグの 一例で、Multi-Parent、Multi-Position、 Multi-Aim などのさまざまなコンストレ イントを使用して各ボディパーツをコント ロールしています。

フルボディコントロールリグの例

このタイプの設定の利点は、どんな状況でもキャラクターをコントロールできることです。コンストレイントの いくつかは IK で、キャラクターを完全にコントロールします。他のいくつかはオーバーライドで、基礎となる アニメーションに影響を与えることなく、ボーンの位置と回転を調整します。

特定のピボットからボーンの位置をコントロールできる Multi-Referential Constraint を使えば、足や手の 位置を厳密に決めることが可能です。足の位置決めの場合、足の指と母指球を使って正確な位置を設定して います。



Multi-Referential Constraint コンポーネントを使用したボーンピボットの駆動

各ポイント(この場合は足のかかと、母指球、つま先)にターゲットポイントを設定する必要があります。そして、 これらを右脚のターゲット(各ポイントのピボットによってコントロールされるターゲット)と併せてコンストレ イントに追加します。

再生モードまたはレコードモード中に、駆動ピボットをドロップダウンから設定できます。ターゲットポイントを 選択し、回転させると、そのピボットポイントから足が回転するのがわかります。これについては、Unite セッションで詳細に説明しています。

足の IK

足の IK は、リアルなアニメーションを作るための重要なコンセプトであり、大抵のアニメーターはこのことを よく理解しています。足の IK は、足が滑ったり、地面に沈んだりしてしまう問題を減少させます。例えば、キャラ クターが椅子に座っているときなどに、足を正しい位置に保つことができます。

まず、以下の例は腰コントロール付きの新しいリグです。IK は Hierarchy に表示された順番に計算される ため、このコンストレイントを2つの足のコンストレイントの上に置くことが重要です。つまり、初めに股関節の 動きを計算し、次に足の位置を計算します。もしこれが逆だと、腰が動く前に足が計算されてしまいます。



腰と脚の IK

Inspector		a :	Inspector	a :
		Static v	RightLegIK	Static 🔻
			Tag Untagged Layer Default	
Tag Untagged	Layer Delault		Transform	a ± :
🔻 🙏 🛛 Transform		07‡ :		
Position	X O Y O	Z O	Position X 0 Y 0	2 0
Rotation	x 0 Y 0	7 0		7 1
Scale	80 X 1 Y 1	7 1		
JCale			🔻 🖻 🛛 Two Bone IK Constraint	07‡ €
Versition	Constraint	07‡ :	Weight	— 1
Weight		—• 1	Root kightUpperLeg (Transform	
Constrained Object	Hips (Transform)	\odot	Mid KightLowerLeg (Transform) 💿
Constrained Axes			Tip ARightFoot (Transform)	
Source Objects		1	▼ Source Objects	
			Target	m) 💿
= 🙏 HipsTarget (Tra	nsform) 💿 ———	-•1	Hint ARightKneeHint (Transform)	
		+ -	▼ Settings	
▼ Settinas			Maintain Target Offset None	
Maintain Offset			Target Position Weight	
Offset	X 0 X 0	7 0	Target Rotation Weight	
Oliset		2 0	Hint Weight	— 1

腰と脚の IK の設定

両足には Two Bone IK Constraint が使用され、キャラクターが座る際、膝が正しく曲がるようになっています。この際、足はずっと地面についています。

これにより、再生モード中、腰のターゲットの位置を変更することで、キャラクターを椅子に座らせることができるようになります。必要に応じて、左右の脚のヒントを使って膝の位置を調整することができます。



足を地面につけたまま座る、IK を使った簡単なアニメーション

アニメーションでの IK のマージ

通常、IK コンポーネントはボディパーツを完全にコントロールするため、そのパーツではアニメーションが 実行されなくなります。

しかし、Override Transform コンポーネントは、IK コントロールでアニメーションをマージすることができ、 別のボディパーツに対するアニメーションの動作を止めることなく、既存のアニメーションを少し調整するの に便利です。

これを設定するには、各ボーンの位置にターゲットを設定する必要があります。最適な方法は、キャラクターの スケルトンのボーン(例えば背骨)を右クリックし、空のゲームオブジェクトを作成することです。

ターゲットをエフェクターに割り当てて視覚化し、リグセクションにドラッグダウンすることができます。 Override Transform コンポーネントは、これらそれぞれのターゲットに追加され、修正するボーンは 「**Constrained Object**」フィールドに、ターゲット自体は「**Override Source**」フィールドに追加されます。

Inspector							а	:
💬_ 🗸 SpineOverrid	e					Sta	tic	•
Tag Untagged		▼ La	aye	er Default				•
🔻 🙏 🛛 Transform						0	Ę.	:
Position	х	0	Y	0	z	0		
Rotation	Х	0	Y	0	z	0		
Scale ୪ହ	х	1	Y	1	Z	1		
🔻 💿 🛛 Override Trans	for	m				0	ᅶ	:
Weight					-	1		
Constrained Object	Å	Spine (Tra	ns	form)				\odot
▼ Source Objects								
Override Source	Å	SpineTarg	et	(Transform	ו)			\odot
Override Position		0		0		0		
Override Rotation		0		0		0		
▼ Settings								
Space	Pi	vot						•
Position Weight					-	1		
Rotation Weight					-	1		
							_	

オーバーライドは、アニメーションがボーンをコントロールしている間もボーンを調整できる。

これにより、アニメーションの再生中でも、これらのターゲットにアニメーションを付け、背骨の回転、肩の調整、 首の位置、脚の回転などの調整を行うことができます。このプロセスの良い例として、こちらの動画をご覧 ください。

BUILDING A TWIST CHAIN CONSTRAINT

- We're proposing a bidirectional twist chain constraint driven by two effectors.
- We'll be limiting this constraint to animating rotation only.



セッションを 視聴



フリーフォームアニメーションのリギング: 進化するアニメーションパイプライン

Dave Hunt は、この GDC 講演で Unity がどのように Animation Rigging パッケージにフリーフォームアニメー ションを実装し、アニメーターがリギングスペシャリストの 助けを借りることなく、コントロールリグの構造を変更し ながら非破壊的にモーションコンテンツを保持できるように したかを紹介しました。

Animation Rigging を使用するメリット

- ー 使いやすい IK システムで、チームメンバー全員が利用できる
- ー 一体のキャラクターに複数のコンストレイントがあっても、再生パフォーマンスが効率的
- プレビューとアニメーションのためのリアルタイムの IK アップデートは、イテレーションとキーフレーム 作成を効率化する
- ー モジュール式リグシステムにより、二足歩行および非二足歩行キャラクター用のカスタム IK リグ設定を 作成できる
- ー モジュラーシステムにより、さまざまなタイプのキャラクター用のリグを効率的に作成できる
- ー 各モジュラーリグセクションのウエイトにより、システムをカスタマイズできる
- ー 同じリグ設定を異なるキャラクターにコピーできる
- ー コードはオープンソースで、開発者が変更できる
- ー Timeline は、精巧な結果を達成するために、複数のコンストレイントをオーバーライドすることができる

アニメーションカット シーンへの Timeline の 活用

Timeline は、ビデオエディターのような見た目と機能を持つアニメーションエディターで、Unity で直接 リニアアニメーションシーケンスを作成することができます。シーン内のすべてのオブジェクトのアニメー ションを1か所からコントロールでき、他のすべてのオブジェクトやキャラクターに対する、アニメーションの 発生タイミングもコントロールできます。



URP 3D Sample コックピット環境の Timeline の例


ジェネリックとヒューマノイドの両方のアニメーションタイプに対して機能し、Animation Rigging パッケージも サポートしています。

Package Manager の Cinematic Studio コレクションには、アニメーションシーケンスの作成に便利な パッケージが多数用意されています。「Window」>「Package Manager」からインストールできます。

Package Manager		: 🗆 ×
+ ▼ Packages: Unity Registry ▼ Sort: Name (asc) ▼ Filters ▼ Clear Fil		: •
All Services		
▼ Features	Cinomatic Studia	loctali
⊛ 2D	Cilientatic Studio	liistali
8 packages	Release	
3D Characters and Animation	From Unity Registry by Unity Technologies Inc.	
4 packages	com unity feature cinematic	
3D World Building	containty neutro contentatio	
4 packages	Description Packages Included	
⊛ AR		
5 packages	Create films, TV shows, and other animated linear content. Organize and assemble your assets in struct	ured sequences. Automate complex
Cinematic Studio A packages	camera movements. Export data for round-trip and review, and output your final cut. Work with common	n film industry formats.
o packayes		
© Engineering		l .
1 backedes		

Package Manager の Cinematic Studio パッケージ

パッケージに含まれるものは以下のとおりです。

- Timeline:カットシーンなどのプレイアブルシーケンスを作成
- Sequences:ムービー、シネマティック、その他のあらゆるリニアコンテンツを作成
- ー FBX Exporter:Unity で作成したジオメトリ、ライト、カメラ、アニメーションを 3D モデリングソフトに 送信
- ー Alembic:Unity でアニメーションを再生できるよう、アニメーションデータをファイルにベイクする
- Cinemachine:ターゲットの追跡、合成、ブレンド、ショット間のカットといった、複雑な数学と論理を 解決する高度なカメラシステム
- Recorder:ゲームプレイ映像をビデオファイルまたは画像シーケンスとして、予告編やチュートリアル などのコンテンツに最適なフレームレートでキャプチャする

アニメーションカットシーン

以前は、ゲームのカットシーンは、Autodesk 3ds Max や Maya、Blender などのソフトウェアで作成され、 カットシーン中に再生されるビデオファイルとしてレンダリングされていました。ビデオファイルは容量が 大きくで、ビデオエンコーディングを使って圧縮する必要があり、全体的な品質が低下する恐れがありました。

しかし、Unity に Cinemachine と Timeline システムが追加されて以来、ビデオ圧縮することなく、 ゲームエンジン内で直接映画品質のカットシーンを再生できるようになりました。



アニメーション短編映画『Enemies』は Timeline を使用して Unity で作成されている。

シーンに Timeline シーケンスを追加する

Timeline シーケンスは、現在のシーンの一部として含まれ、1 か所に保存される必要があります。 まず、空のゲームオブジェクトを作り、これを TimelineHolder という名前にします。「Window」> 「Sequences」>「Timeline」から Timeline ウィンドウを開きます。

Hierarchy で TimelineHolder オブジェクトを選択し、Timeline アセットを作成します。これは、アニメーションクリップを含むタイムライン上のすべてのデータを保存します。

また、タイムラインを TimelineHolder ゲームオブジェクトにバインドする Playable Director コンポーネントも 作成されます。これで、Hierarchy で TimelineHolder が選択されているとき、トラックエディターに完全に アクセスできるようになります。



Project ウィンドウで Timeline アセットをクリックすると、フレームレートを含む Timeline シーケンスの プロパティを定義できます。デフォルトのフレームレートは 60fps ですが、他のフレームレートに変更する こともできます。

また、Timeline シーケンスに特定の継続時間を設定したり、追加したクリップの数に応じて継続時間を 延長したりすることもできます。

Inspector			a :						
TimelineHolderTimeline									
			Open						
Frame Rate	Game: 60 fps	▼ 60							
Duration Mode	Based On Clips		•						
Duration	s 22.23333	f 1334							
Scene Preview	 Image: A start of the start of								

Timeline プロパティ

アニメーションクリップを追加する



Timeline ウィンドウの「+」アイコンを クリックしてトラックを作成し、 「Animation Track」タイプを選択します。

アニメーショントラックを追加

アニメーションクリップをトラックに直接ドラッグし、クリック&ドラッグで所定の位置に移動できます。トラックを 右クリックして「Add from Animation Clip」を選択し、追加したいアニメーションを選択することもできます。

		0	el a
	Сору	Ctrl+C	
	Paste	Ctrl+V	
	Duplicate	Ctrl+D	
	Delete		
	Lock	L	
	Lock selected track only		
	Mute	М	
	Mute selected track only		
	Track Offsets	>	
🖻 Console 🛛 🛤 Timeline* 🖿 Project	Add Override Track		
Preview 🛏 I ► ►I 🛏 [▶] 0	Add From Animation Clip		
_ + ▼ I+ I*I ∓ ↓ [³⁰]; ; ;	Add Signal Emitter		
	Add Signal Emitter From Signal Asset		
Adam (Animator)			

右クリックで「Add from Animation Clip」を選択し、アニメーションクリップを選択する。

ワールドのルート原点が 0 の場合、この操作を行うとキャラクターが消えてしまう可能性があることに注意 してください。これは、キャラクターが X、Y、Z のワールド位置 0 にジャンプし、そこでアニメーションを再生し 始めることを意味します。これは、キャラクターがいるべき場所から遠く離れているかもしれません。



🔻 🙏 Transform						요군:
Position	х	0	Y	0	z	0
Rotation	х	0	Y	-134.011	۷	0
Scale හැ	x s					

ルート原点が 0 の場合、キャラクターはワールド位置 0 に移動する

これを解決するには、キャラクターの現在位置に空のゲームオブジェクトを作成し、その中にキャラクターを 配置して親にします。 子オブジェクトの位置は、親オブジェクトの位置と相対関係になります。そのため、この場合の子オブジェクトは X、Y、Z 軸の Transform 位置が 0 のままでも、これらはもはやワールド空間のルート原点 0 ではなく、空の ゲームオブジェクトの現在位置になります。



Adam のキャラクターが Characters と呼ばれる空のゲームオブジェクトの子となる。

🛭 Ins	pect	tor											а	:
\bigcirc	~	Characters										Stati	ic	-
~.	Tag	Untagged 👻 Layer Default												
₹.,	Т	Transform										0		
Pos	ition				-507.155			-1.187461		z	29.059			
Rot	ation							-67.52		Z	0			
Sca			\$					1		z				

空の Characters ゲームオブジェクトの位置



Adam のキャラクターが正しい位置でアニメーションを再生する。

アニメーションクリップは、Timeline ウィンドウのトラック上にソリッドブロックとして表示され、トラックの 長さの方向にドラッグして時間を進めたり戻したりすることができます。



Timeline 上の Adam キャラクターのアニメーションクリップと Inspector のクリップオプション

Inspector Inspector		∎ Project	a	. :
Idle				9
Clip Timing				
Start				
End		8.333333	500	
Duration		8.333333	500	
Ease In Duration		0	0	
Ease Out Duration			0	
Clip In		0	0	
Speed Multiplier		1		
Animation Extrapolation Post-Extrapolate	H	old 🔻 Infinity		
Blend Curves In Out V Animation Playable A	~	None Hold Loop Ping Pong Continue		
Animation Clip	1	Idle		

アニメーションクリップの事後補外

各クリップには、クリップの左右にスペースがある場合、前および後の外挿方法があります。これは、再生前後に アニメーションがどのようになるかを決定します。デフォルトでは Hold に設定され、アニメーションの最初か 最後のフレームから静止ポーズを作成します。 ビデオエディターのように、クリップを右クリックして「Editing」を選択することで、アニメーションクリップを カットしたり、サイズを変更したりすることができます。ショートカットキーも表示され、より効率的な編集が 可能です。

	Copy Paste Duplicate Delete Match Content Edit is Asimatica Window	Ctrl+C Ctrl+V Ctrl+D C			
	Editing Tile	>	Trim Start Trim End	 0	
	Find Source Asset		Split	S	
	Reset Offsets		Complete Last Loop		
▼ M TimelineHolderTimeline (Ti	Export Clip To FBX		Trim Last Loop		
	Add Signal Emitter		Match Duration		1360 1390 142 [,]
	Add Signal Emitter From Signal Asset		Double Speed		
a 👁 :		Idle	Half Speed		
			Reset Duration Reset Speed Reset All		

アニメーションクリップの編集オプション

クリップの左右に他のクリップがない場合、クリップをフェードイン/フェードアウトするように調整できます。 これは、Inspector でブレンドインとブレンドアウトの値を調整することで、カクッとした動きを防ぎ、ポーズが 徐々に変わるようにすることができます。



アニメーションクリップでフェードインとフェードアウトするポイント

Inspector					a :	
Idle					•	
Clip Timing						
Start		0		0		
End		8.333333		500		
Duration		8.333333		500		
Ease In Duration	s	1.87	f	112.2		
Ease Out Duration	s	1.8	f	108		
Clip In		0		0		
Speed Multiplier		1				

フェードインとフェードアウトの継続時間を調整

2 つ目のアニメーションを追加すると、それらをブレンドすることができます。2 つ目のアニメーションを 1 つ目の アニメーションにドラッグして、ブレンドオーバーラップを作成します。



Timeline で 2 つのアニメーションを 1 つにブレンド



また、Speed Multiplier 設定を使って クリップの再生速度を変更するオプションも あります。再生速度が1以外に設定されて いる場合、アニメーショントラックの上に 点線が表示されます。ブレンドカーブを 「Manual」に設定し、カーブをクリックして 2 つのクリップ間のイージング方法を 変更します。下部にプリセットが用意されて いるほか、ベジエハンドルを調整して独自の カーブを作成することもできます。

ブレンドのインのカーブを調整

トラックオーバーライドを追加する

アバターマスクを使って、Animator ウィンドウと同じように 2 つ以上のアニメーションをマージすることが できます。3 つの点のトラックオプションアイコンをクリックし、「Add Override Track」を選択します。

	Сору	Ctrl+C
	Paste	Ctrl+V
	Duplicate	Ctrl+D
	Delete	
	Lock	L
	Mute	M
	Track Offsets	>
Console	Add Override Track	
Preview H4 IA D DI DH [D] 416	Add From Animation Clip	.1
	Add Signal Emitter	
🐴 ≻ Adam (Animator) 💿 💿 ∓ a 🗢 🗄	Add Signal Emitter From Signal Asset	

アニメーショントラックにオーバーライドトラックを追加

アバターマスクを追加するトラックをクリックします。このトラックに配置したアニメーションは、ベースト ラックのボーンを上書きするので、面白くユニークなアニメーションを作成することができます。



オーバーライドトラックでアバターマスクを使用して、2 つのアニメーションをマージします。

ルートモーションを含まないその場アニメーションを使用する場合は、オーバーライドトラックを使用して、 キャラクターの位置、回転、またはスケールをコントロールできます。

アニメーショントラックには、好きなだけオーバーライドトラックを追加できますが、2 つのオーバーライド トラックが同じボーンまたはボーンをコントロールする場合、リストの一番下にあるオーバーライドトラックが、 その上にあるすべてのオーバーライドトラックをオーバーライドします。 キャラクターの位置を調節しながら、無限ループにキーフレームを追加するには、新しいオーバーライド トラックの「**Record**」をクリックします。レコーディング中、トラックは赤になります。

E	Console	🛱 Tim	eline										
Ρ	review 🖊 🛛	•	M	₩ [I		712			TimelineHolderTimeline (TimelineHold	der)			
								ŧ			dirette	a	
v	🐴 ≻ Adam (A	Animat	or)		0						Wa	king	
				1	i r (

オーバーライドトラックを使ってキャラクターの動きを記録

右クリックして「Convert to Clip Track」を選択し、キーフレームをアニメーションクリップに変換されます。 これにより、速度、ループ、ブレンドスタイルなど、アニメーションクリップに使用できるすべてのオプションに アクセスできます。

アニメーションクリップをダブルクリックして、Animation ウィンドウでキーフレームを開きます。ここから、 この eBook の前半でも説明したように、キーフレームを調整したり、イベントを追加したりすることができます。

Ctrl + D を押すか、「Edit」>「Duplicate」からアニメーションクリップを複製することができます、1 つ目の クリップから 2 つ目のクリップに移行する際、キャラクターの位置が原点にジャンプして戻ることがわかります。 これを解決するには、2 つ目のクリップを選択し、「Match Offsets to Previous Clip」を選択します。これで、 ルートモーションを使うアニメーションのように動作し、キャラクターは原点に戻ることなく、前のクリップの 最後の位置から動きを継続します。また、現在のクリップの右側に次のクリップがあれば、オフセットを 合わせることもできます。

キーフレームの打ち方

Timeline では、このウィンドウから直接アニメーションを作成することもできます。ジェネリックアニメーションタイプでは、フォワードキネマティクス (FK)を使って各ボーンの回転を調整できます。ボーンの位置をより細かく調整するために、オーバーライドを使用することもできます。

アニメーションをカスタマイズできるオーバーライドを使い、キーフレームによりベースアニメーションを 調整することもできます。ここでカーブにアクセスし、キーフレーム間のイージングを微調整することもできます。



既存のアニメーション向けのジェネリックリグキーフレームオーバーライドとカーブの調整

ヒューマノイドアニメーションは、フォワードキネマティック(IK)でキーフレームを打つことでアニメーションを 付けることはできません。代わりに、Animation Rigging パッケージでインバースキネマティック(IK)を使用して ターゲットの位置をアニメーションで動かすことができます。この eBook の Animation Rigging のセクション をご覧ください。

シーンでリグのアップデートを確認するには、まず、Rig Builder コンポーネントを含む親ゲームオブジェクトに 初期トランスフォームキーフレームを設定する必要があります。オーバーライドトラックを追加し、ターゲットの 位置と回転を調整すると、Scene ビューでリアルタイムに更新されるのを確認できます。



IK リグの調整をリアルタイムでアニメーション化

トラックタイプ

アクティベーショントラック

アクティベーショントラックを使うと、アニメーション再生中でそのブロックが継続する間は、オブジェクトを オンにできます。

🛢 Console 🛛 🛤 Timeline* 🖿 Project	
Preview ₩ I ► ► ► ► ► [►] 200	TimelineHolderTimeline (TimelineHolder)
+- 1+ 1+ 1+	10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
🗹 🕄 Sphere 💿 🕴 🗎 💿 🗄	Active

ブロックの継続時間中オブジェクトを切り替えるアクティブブロック

オーディオトラック

オーディオトラックを使うと、アニメーションの特定のポイントにサウンドファイルや音楽を配置できます。 複数のオーディオトラックを使って、複雑なオーディオスコアを組み合わせることもできます。

E Console 🔛 Timeline 🖿 Project		a:
Preview #4 4 ▶ ▶ ₩ [▶] 327 - 6		
+ • at it it.	1 ⁰ 30 160 190 1120 1180 1210 240 270 1900 339 180 1420 450 480 510 1840 1570 160	
🔺 🗠 Adam (Animator) 💿 💿 🕴 a 👁 🚦	Watking 00	
💵 🖬 Audio Player (Audio Source) 💿 🌭 🖡 a 👁 🗄		
💵 🛋 None (Audio Source) 💿 🏷 🕴 👁 🗄	Footstep	
📲 📢 None (Audio Source) 💿 🍾 👫 a 👁 🗄		ander Baller
💶 None (Audio Source) 💿 🍡 ∔ a 👁 🗄	Effect.	

シーケンスに深みのあるサウンドトラックを生み出すために複数のオーディオトラックを使用。

オーディオソーススロットが空の場合、サウンドと音楽は 2D 空間で再生されます。Audio Source コンポー ネントを持つオブジェクトをこのスロットに配置すると、3D サウンドを 3D ワールドの特定の部分から再生 するように設定でき、サラウンドサウンドオーディオスコアを作成できます。

Inspector				а	:
Ambience					9
Clip Timing					
Start		0.45	27		
End		26.95	1617		
Duration		26.5	1590		
Ease In Duration		1.23	73.8		
Ease Out Duration		0	0		
Clip In		0	0		
Speed Multiplier		1			
Blend Curves					
Out		uto 🔻			
Audio Playable Asset					
Clip	5	Ambience			
Loop					
Volume		•		0.3	
The final Volume is 0.3 Calculated from: Clip: 0.3 Track: 1					

「Ease In Duration」および「Ease Out Duration」の設定により、トラックをフェー ドインおよびフェードアウトすることも できます。トラックに表示されている斜めの 線がフェードの継続時間を示しています。 Inspector から音量を調節することもでき ます。

音のフェードイン/フェードアウトおよび音量の調整

コントロールトラック



コントロールトラックを使うと、シーン内の パーティクルエフェクトを有効化できます。 例えば、アニメーションの特定のポイントで 爆発を発生させることができます。また、 パーティクルトラック終了後に起こる アクションを選択することもできます。無効 にする、アクティブ設定でループを続ける、 Revert 設定で休止状態に戻す、後の時点で 再び有効化するまで待機させるなどです。

🖬 Console 📙 Timeline 🖿 Pr	roject			
Preview H I > > > > > [>]	215			TimelineHolderTimeline (TimelineHolder)
+-	4	II +	141	
🔺 ≻ Adam (Animator)	۲	۲		Walking
😚 Control Track				SmallExplosionEffect 0.80x ¥
(Ohere)				

コントロールトラックを使用してパーティクル効果をコントロールする



コントロールトラックを使い、現在のタイムラインから他のタイムラインを調整できます。複数の Timeline シーケンスを設定した時間に開始させたい場合は、TimelineHolder ゲームオブジェクトをコントロール トラックにドラッグして、適切な時間に起動させることができます。



コントロールトラックを使用して複数のタイムラインをコントロール

プレイアブルトラック

プレイアブルトラックでは、独自のカスタムスクリプトを作成してこのトラックに追加し、シーケンス中に オブジェクトやキャラクターを動かすことができます。これに関する説明は、Unity ブログの投稿でご覧ください。

シグナルトラック

シグナルトラックの機能は、Animation ウィンドウのイベントとよく似ています。ゲームオブジェクトのコン ポーネントにアクセスするシグナルイベントや、オブジェクトにアタッチされたスクリプトのカスタムパブリック 関数を設定することができます。

キャラクターが歩くのに合わせて足音を出すには、左足用と右足用の 2 つのパブリック関数でスクリプトを 作成します。



足音を再生するスクリプト

サウンドを再生する Audio Source コンポーネントと併せてスクリプトをキャラクターにアタッチします。

次に、タイムラインにシグナルトラックを追加します。足音を出したい最初のポイントを右クリックし、Signal Emitter を追加します。すると、Signal Asset がまだ存在しないため、黄色の警告メッセージが表示されます。 Inspector からこのアセットを作成できます。

Inspective	ector				а	:
) S	Signal Emit	er				0
•						
Time			0.7666667	f 46		
Retroa	active					
Emit C	Once					2.1
Emit S	Signal	N	one			•
	Your	oroject c	ontains no Sign	al assets		
		C	reate Signal			
🔻 🗭 Sig	gnal Receiv	ver				
Recei	ver Compo	nent 😭	Adam			0
S	Signal		Reactio	n		ľ.
						_

新しいシグナルアセットを作成



ドロップダウンメニューから、キャラクターにアタッチされたすべてのコン ポーネントにアクセスでき、複数のタスクを行うことができます。スクリプトを 選択すると、PlayLeftFootSound 関数と PlayRightFootSound 関数が 見つかり、左右の足音用に複数のシグナルを追加できます。



Signal Asset からカスタム機能を選択



Cinemachine トラック

Cinemachine では、複雑なシネマティックカメラの設定を効率的な方法で行うことができます。Cinemachine トラックでは、Cinemachine で使用されているバーチャルカメラを操作して、ショットのタイミングを合わせたり 調整したりできます。

メインカメラを空のスロットにドラッグすると、メインカメラに Cinemachine Brain コンポーネントを作成 できます。これは、メインカメラがバーチャルカメラの設定に合わせるために使用されます。

作成およびフレーミングされたバーチャルカメラは、このトラックに追加されます。1 つのカメラが次のカメラに 続くと、2 つのカメラの間を直線的にカットすることになります。



2 つのカメラショット間の Cinemachine ストレートカット

あるカメラブロックを別のカメラブロックにドラッグすると、クロスオーバーブレンドとなり、最初のショットと次のショットの間でカメラが動きます。

E Console	🛱 Timeline*					
Preview 🖊	I4 ► ► ₩ [▶]	81	🔹 💮 Timelii	neHolder2Timeline (Timeli	ineHolder2)	
+ -		II+ I≛I Ŧ		⁶⁰		
🛋 ≻ Adam	(Animator) O				Walking	
🚈 🛎 Main	Camera (Cinemachine	• I a • I	VCam1			VCam2

2 つのカメラショット間の Cinemachine ブレンド

上記のスクリーンショットでは、Cinemachine バージョン 2 が使用されています。Cinemachine バージョン 3 の 新機能に関する詳細情報は、こちらのブログ記事をご覧ください。



トラックグループ

トラックが増えるにつれ、タイムラインの操作がより複雑になることがあります。

トラックグループを使うと、特定のトラックを一括りにして整理することができます。

_					_								
B	Console	🛱 Tii	meline										
Pr	eview 🖊	►	· •	▶ [►] [27	'5			▼ ⑦ Timeline	eHolderTi	meline (TimelineHolder)	
	+ •				 +	1±1	Ŧ				and and a	300	
v	Adam W	alking					• -	ł					
	🛋 ≻ Ad	lam (Ani	mator)	\odot	0	∓ a	۲				Walking		
	📕 🗣 Ad	lam (Sig	nal Re	ceiver)	\odot	Ŧ : a							
	✓ ⊕ vc	am1			0	Ŧ a	۲				Active		
	😚 Cont	trol Trac				∓ a	۲					SmallExplosionEffect	<u>* </u>
	🚰 🛎 Ma	ain Cam	era (Ci	nemach	i 💿	∓ a	۲		VCam1			VCam2	

トラックグループにまとめられたトラック

トラックグループを最小化すると、タイムラインにスペースができ、操作しやすくなります。

	Project	🖬 Con	sole	Ħ Tin																	
Pre	view 🖊	4 ▶	. ►I	▶ [▶]	224				pitFlyth	rough											
	+ -														1 ²¹⁰⁰	li					
•	k Markers						:														
v	PlayerS						+														
	≪ ≻Co	ockpit (Ar	nimato													++-					
	🔺 ≻ Pla	ayerShip	Cart (A													R	corded				
	🔺 ≻ Pla	ayerShip	UpVec																		
	🗹 🎲 Ну	perSpac	eTunn	el																	
	HyperS						÷														
		ating Cap					÷														
•	Asteroid	EatingW					+														
							+														
	CapitalS	Ships					+	_													

最小化されたトラックグループ

シーケンス

Timeline で作業する際は、Sequences パッケージを使うと、大きなプロジェクトが扱いやすくなります。 再生中に特定のポイントでシーンを動的にロードできるので、シーンの扱いが容易になります。

Sequences パッケージの詳細な使い方に関しては、こちらの Unity Learn チュートリアルをご覧ください。

上級者向け シミュレーション

Unity の Built-In 3D Physics エンジンはアルタイムで物理演算イベントのシミュレーションを行います。 重力に反応するオブジェクトや、髪の毛、毛皮、布、マント、カーテン、旗などのオブジェクトのリアルな動きの シミュレーションを行うために使用できます。

リジッドボディ物理演算

ソリッド物理演算を追加するには、Rigidbody コンポーネントと、オブジェクトが地面に埋もれないよう Mesh Collider コンポーネントをオブジェクトに追加します。

ソリッド物理演算とは、仮想環境におけるオブジェクト間の物理的相互作用のリアルなシミュレーションを行うことを指します。 質量、重力、摩擦、衝突などの特性のシミュレーションが行われ、現実世界を模倣したオブジェクトの挙動を実現できます。

🔠 🛛 Bo_mesh (Mesh Fi	lter)	0 7 i
🕨 🔣 🖌 Mesh Renderer		0 ‡ i
🔻 🔀 🖌 Mesh Collider		Ø ∓ :
Convex	~	
ls Trigger		
Provides Contacts		
Cooking Options	Everything	
Material	<u>≜</u> Staff	
Mesh	⊞bo_mesh	
Layer Overrides		
Layer Override Priority	0	
Include Layers	Nothing	
Exclude Layers	Nothing	
🔻 🔥 Rigidbody		0 ≓ :
Mass	1	
Drag	0	
Angular Drag	0.05	
Automatic Center Of Mass	✓	
Automatic Tensor	~	
Use Gravity		
Is Kinematic		
Interpolate	None	
Collision Detection	Discrete	
▼ Constraints		
Freeze Position		
Freeze Rotation		
🔻 Layer Overrides		
Include Layers	Nothing	
Exclude Layers	Nothing	

物理演算を実現する Rigidbody コンポーネントが取り付けられた杖のゲームオブジェクト

Rigidbody コンポーネントは、質量と抗力を設定して、現実世界の物体が落下するときに空気とどのように 相互作用するかのシミュレーションを行うことができます。特定のレイヤーを含めたり除外したりでき、 オブジェクトが設定されたレイヤー上の他のオブジェクトとしか衝突しないようにすることができます。 「Freeze Position」と「Freeze Rotation」の設定から、さらに細かくシミュレーションをコントロールする こともできます。

オブジェクトが重力に従うよう、「**IsKinematic**」プロパティのチェックは入れないでおきましょう。「IsKinematic」 がチェックされていると、オブジェクトは重力に従いませんが、衝突のような物理ベースの相互作用は依然 として処理します。

コライダーはデフォルトでは物理コライダーに設定されています。これにより、オブジェクトが他のオブジェ クトを通り抜けるのを防げます。しかし、「Is Trigger」オプションをチェックすることでコライダーをトリガー コライダーとして設定することができます。トリガーコライダーは物理的な側面を無効にし、オブジェクトが 他のオブジェクトを通過することを可能にしますが、他のオブジェクトと交差したときにはトリガーを返します。 これはコードを書く際にとても便利です。

Physic Material アセットを Mesh Collider に追加して、オブジェクトが現実世界のオブジェクトを 再現するように動作させることができます。例えば、よく跳ねるゴムボールの場合は、伸縮性のある Physic Material を Collider コンポーネントの Material フィールドに追加する必要があります。

新しい Physic Material を作成するには、Project ウィンドウで右クリックして「Create」>「Physic Material」を 選択し、希望の結果になるように値を調整します。



Inspector		а:	🔻 🏭 🖌 Mesh Collider		፼ ≓ :
Staff (Physic Materia		Ø ≓ ÷ Open	Convex Is Trigger		
Dynamic Friction Static Friction Bounciness Friction Combine	0.45 0.45 0.1 Average		Provides Contacts Cooking Options Material Mesh	Everything 올 Staff 땐 bo_mesh	• ©
Bounce Combine	Multiply	*	Layer Overrides Layer Override Priority Include Layers Exclude Layers	0 Nothing Nothing	* *

木製の杖に Physic Material を定義し、それをメッシュコライダーに割り当てている

Material	Dinamic Friction	Static Friction	Bounciness	Friction Combine	Bounce Combine
Bouncy	0.3	0.3	1	Average	Maximum
Ice	0.1	0.1	0	Multiply	Multiply
Max Friction	1	1	0	Maximum	Average
Metal	0.15	0.15	0	Minimum	Average
Rubber	1	1	0.5	Maximum	Average
Wood	0.45	0.45	0	Average	Average
Zero Friction	0	0	0	Multiply	Average

さまざまなタイプのオブジェクトに利用できる便利な物理演算プリセット

囹

物理演算の設定は「Edit」>「Project Settings」から調整できます。ここから重力の量を変更できます。 デフォルトでは、現実世界の重力と同じ -9.81 に設定されていますが、例えば、キャラクターが空間に浮いて いるようにしたい場合は、これを変更することができます。



Physics 設定は Project Settings ウィンドウからアクセス可能

下部のレイヤー衝突マトリックスを使って、どのレイヤーが相互作用できるかを定義します。

「Component」>「Physics」から、さまざまな物理演算コライダー、ジョイント、力を追加できます。

Component	Services	Animation Rigging	Jobs	Tutorial	Window H	elp
Mesh		>				
Effects		>				
Physics		>	R	igidbody		1
Physics 2	2D	>	C	haracter	Controller	
Audio		>	В	ox Collid	er	
Video		>	S	phere Co	ollider	
Renderir	ng	>	C	Capsule C	ollider	
Tilemap		>	N	Aesh Coll	ider	
Layout		>	v	Vheel Co	llider	
Playable	S	>	Т	errain Co	llider	
Miscella	neous	>	C	loth		
Scripts		>		rticulatio	n Padu	
Cinemad	hine	>		lingo loir	n bouy	
UI		>		inge Jon	1L +	
Visual Se	ripting	>	r c	nring loi	L nt	
Animatio	on Rigging	g >	5	baractor	loint	
Event		>		Configura	ble leint	
Input		>		onngura	bie Joint	
XR		>	C	onstant l	orce	
UI Toolk	it	>				
Add		Ctrl+Shift+A				

Physics コンポーネント



Constant Force は、風や物を投げるシミュレーションなど、物理的な物体に特定の方向への力を加えたいときに便利です。

また、コードを使って Rigidbody コンポーネントに力を加えることもできます。例えば、マウスクリック時に 力を加えて、特定の速度でナイフを前に飛ばすことができます。



プレイヤーがマウスを左クリックした際にオブジェクトに力を加えるコード

Rigidbody コンポーネントは Y 移動軸上でフリーズされているため、ナイフは地面に落ちることなく、 コードで設定された力の速度で前方に飛びます。



投げたナイフが Y 軸にコンストレイントされている



キャラクターの動きをコントロールするために Rigidbody コンポーネントを使用することもできます。これは、 ゲームワールドの物理オブジェクトに反応するリアルなモーションを作成するのに便利です。

これを行うには、コードで Rigidbody コンポーネントにアクセスし、MovePosition または MoveRotation コマンドを使用します。

```
oublic class BasicMoveNoRootMotion : MonoBehaviour
   private Animator animator;
  private Rigidbody rb;
  private Vector3 rotateY;
  public float walkSpeed = 2.5f;
  public float rotateSpeed = 150.0f;
   void Start()
   {
       animator = GetComponent<Animator>();
       rb = GetComponent<Rigidbody>();
  void Update()
   {
       rb.MovePosition(rb.position + Input.GetAxis("Vertical") * Time.deltaTime * walkSpeed * transform.forward);
       if (Input.GetAxis("Vertical") != 0)
           if (Input.GetAxis("Horizontal") > 0)
           {
               rotateY = new Vector3(0, rotateSpeed * Time.deltaTime, 0);
           if (Input.GetAxis("Horizontal") < 0)</pre>
               rotateY = new Vector3(0, -rotateSpeed * Time.deltaTime, 0);
           rb.MoveRotation(rb.rotation * Quaternion.Euler(rotateY));
       animator.SetFloat("Vertical", Input.GetAxis("Vertical"));
```

```
リジッドボディによるキャラクターの動きのコントロール
```

ジャンプを行うには、Rigidbody に力を追加します。



Rigidbody に力を追加してジャンプを行う

クロス物理演算

岔

クロス物理演算は、クロス(布)の動きのシミュレーションを行い、十分なレベルのサブディビジョンを持つ メッシュに適用できます。メッシュは、シミュレーションが行われるマテリアルがリアルな動きをするのに 十分な頂点を持つ必要がありますが、パフォーマンスコストを増加させるほど多くしてはいけません。

Cloth コンポーネントをオブジェクトに追加し、設定でマテリアルタイプを定義します。革、シルク、コットン などを再現できるよう、「Stretching Stiffness」や「Bending Stiffness」のプロパティを調整してください。



クロスオブジェクトへのコンストレイントの設定

囹

Inspector			а	
🙀 🗹 ribbon_mesh		Sta	atic	
Tag Untagged	 Layer Default 			
🔻 🙏 🛛 Transform		0	규 <mark>브</mark>	
Position	X 0 Y 0	0		
Rotation	X 0 Y 0	0		
Scale 🗞	X 1 Y 1			
🕨 🗟 🗹 Skinned Mesh Ren	derer	Ø	규는	
🔻 🎁 🖌 Cloth		0		
	* *			
Stretching Stiffness	1			
Bending Stiffness	0			
Use Tethers	~			
Use Gravity	 Image: A start of the start of			
Damping	0			
External Acceleration	X 0 Y 0	0		
Random Acceleration	X 0 Y 0	0		
World Velocity Scale	0.5			
World Acceleration Scale				
Friction	0.5			
Collision Mass Scale	0			
Use Continuous Collision	~			
Use Virtual Particles	 Image: A start of the start of			
Solver Frequency	120			
Sleep Threshold	0.1			
Capsule Colliders				
Sphere Colliders				
Virtual Particle Weights				

クロスは、地面に落ちないよう、原点で コンストレイントが設定されていなくては なりません。赤いピンの付いた T シャツの アイコン (左の画像参照)をクリックすると、 Cloth Constraints メニューが開きます。 「Max Distance」を 0 に設定し、動かし たくない箇所に赤い点をペイントします。 黒い点の箇所はコンストレイントがなく、 クロスシミュレーションにより動きます。

クロスプロパティとクロスコンストレイントのボタン



点と点の間のクロスの動きに特定の距離を定義

また、頂点が互いに離れることができる 最大距離を特定の数値で設定することも できます。これは、スクリーンショットの忍者 のキャラクターが着ている革の衣装の ような、重いマテリアルを定義するのに便利 です。オレンジから緑の間の色は、クロスに 異なるタイプの動きを与えるために、互いに 離れていく点の特定の距離を定義し、 グラデーションツールを使用してペイント されています。



間違えた場合は、「Max Distance」ボックス のチェックを外すと、点を上描きしてコンスト レイントされていない黒い点に戻せます。

点のコンストレイントを削除

M

Cloth オブジェクトは、シミュレーション中にキャラクターの他の部分と交差するので、クロスがキャラクターを 突き抜けて移動しないようにコライダーを配置する必要があります。Cloth はカプセルとスフィアのコライダー のみをサポートしています。

複数のカプセルとスフィアを使って正確な形状を定義し、それらを Cloth コンポーネントのカプセルと スフィアのコライダー配列に追加することで、複雑な形状を作り上げることができます。



ボーンにコライダーを追加

Hierarchy 内の適切なボーンにカプセルまたはスフィアを追加し、配置します。コライダーはボーン上に 留まり、キャラクターにリアルで正しい動きを提供するのに役立ちます。コライダーを配置したら、右クリッ クして「**Remove Component**」を選択し、その上の Renderer コンポーネントを削除します。



両脚にコライダーを設定したら、それぞれのオブジェクトの Cloth コンポーネントに追加します。

Skinned Mesh Renderer コンポーネントにカプセルを追加



クロス物理演算のリアルなシミュレーション

ラグドール物理演算

ラグドール物理演算は、キャラクターが地面に落ちたり倒れたりする際に、脚や腕がバタバタする動きの シミュレーションを行いします。Unity には、キャラクターのスケルトンに物理コンポーネントを追加する クイック設定システムがあります。

キャラクターを選択し、「GameObject」>「3D Object」>「Ragdoll」をクリックします。すると、設定ウィンドウ が開きます。キャラクターは T または A ポーズをとっているはずです。適切なスロットにボーンをドラッグして 「Create」をクリックします。



Ragdoll 設定ウィンドウ

これで、Rigidbody コンポーネントと、ラグドールスタイルの物理演算を実現する Character Joint コンポーネントがリストの各ボーンに追加されました。

伀

🔻 裔 🛛 Character Joint						9 ∓±	
Edit Angular Limits	9	۹.					
Connected Body	6	UpperChes	t (F	ligidbody)			\odot
Anchor	х	0		0	Ζ	0	
Axis	х			0	Z	0	
Auto Configure Connected	~						
Connected Anchor	х	-0.13423		-0.015940	Ζ	0.19117	89
Swing Axis	х	0		0	Ζ	1	
Twist Limit Spring							
Low Twist Limit							
High Twist Limit							
Limit	10)					
Bounciness	0						
Contact Distance	0						
Swing Limit Spring							
Swing 1 Limit							
Limit	50)					
Bounciness	0						
Contact Distance	0						
Swing 2 Limit							
Enable Projection							
Projection Distance	0.	1					
Projection Angle	18	30					
Break Force	In	finity					
Break Torque	In	finity					
Enable Collision							
Enable Preprocessing							
Mass Scale	1						
Connected Mass Scale							

Character Joint コンポーネントを修正して、 お好みのラグドール効果を得ることが できます。Rigidbody コンポーネントの Mass プロパティに設定した値もシミュ レーションの効果に影響します。コンポー ネントをコピーして、リストになかったスケ ルトンの他のボーンに貼り付けて、独自の カスタムラグドール設定を作成することも できます。

ラグドール物理演算を扱う Character Joint コンポーネント



再生中のアニメーションは、ラグドールのシミュレーションを上書きするため、再生モードでラグドール物理 演算を見るには、必ず Animator コンポーネントを非アクティブにしてください。

忍者キャラクターの Ragdoll シミュレーション

例えば、キャラクターが撃たれたときにラグドール効果を再生するように、スクリプトで Animator コンポー ネントを非アクティブにすることができます。しかし、アニメーションからラグドールに直接移行すると、結果が 奇妙なものになるでしょう。キャラクターの物理演算が予期せぬ作用を見せることがあります。

この問題を解決するには、Animator がオフに切り替わるまで重力による影響を受けないよう、各ボーンの Rigidbody コンポーネントを **isKinematic** に設定します。そして、isKinematic を false に戻すと、ラグドール 効果が問題なく再生されます。



Animator コンポーネントをオフにする際、Rigidbody を isKinematic に切り替える (これは親のゲームオブジェクトに配置されている)

毛皮と髪

Unity の HD レンダーパイプライン(HDRP)は、リアルな髪や毛皮を作成するための洗練された Shader Graph ベースのヘアシェーダーを備えています。また、ラインベースレンダラーを使用した独自のヘア エフェクトを作成することもできます。Unity のデモ動画『Lion』で実際にご覧ください。



この『Lion』デモでは、Wētā Digital、SpeedTree、Ziva、SyncSketch、Unity エディターのアーティストツールで作成されたリアルタイムのコンテンツを見ることができる

Package Manager の High Definition RP セクションにある Materials Samples フォルダーをインストール すれば、ヘアシェーダーのサンプルシーンを見ることができます。

Package Manager			: 🗆 ×
All Services			
		High Definition PD	
			Remove
		14.0.8 · August 15, 2023 Release	
Editor Coroutines		From Unity Registry by Unity Technologies Inc.	
		comunity.render-pipelines.mgn-definition	
		Description Version History Dependencies Samples	
		Additional Post-processing Data 35.34 MB	Import
		Adda additional post-processing data to your project, such as Lens Dirt textures.	
		Procedural Sky 22.38 KB	Import
		Adds the deprecated procedural sky of HDRP.	
JetBrains Rider Editor		Darticla Svetam Sharlar Samilae - 2.28.48	Import
		Adds Particle System Shader samples to your project. Open the	import
		SampleScene to see examples of various lit and unlit particle	
		enects.	
		Material Samples 43.62 MB	Import
		Adds Material Samples to your project. Open the MaterialSamples	
		scene to see examples of materials based on the lit shader with effects such as subsurface scattering, displacement, anisotropy and	
S Mathematics		more. This scene requires Text Mesh Pro to render text	
Memory Profiler	1.0.0	explanations.	

Material サンプルにはヘアのサンプルシーンが含まれる

ヘアシーンは Scenes フォルダーにあります。ストランド(束)ベースとカードベースのヘアレンダリング 機能があり、マテリアルオプションは Inspector にあります。



HDRP Materials Samples フォルダーにあるリアルなへアシェーダーオプション

また、こちらの Unity Learn チュートリアルに従って、重力や力と連動する独自のダイナミックなへアシステムを 作成することもできます。



Unity Learn チュートリアル「ヘアシミュレーションを始めよう」では、キャラクターやオブジェクトに独自のヘアシミュレーションを追加する方法を学ぶことが できる

Alembic

Alembic パッケージは Alembic ファイルを Unity で使用することを可能にします。このパッケージを使用 すると、水やその他の流体、髪や布の動きなどの複雑なアニメーションを、その高度でリアルなシミュレー ションを計算するパフォーマンスコストなしで作成できます。Alembic は、再生モードで再生されるファイルに 頂点の動きをベイクします。

Package Manager で Alembic パッケージをインストールし、alembic.abc ファイルを Unity にインポート します。

Package Manager		: 🗖
+ ▼ Packages: Unity Registry ▼ Sort: Nan		
All_Services		
Addressekies	1 01 10	Alamhic
Addressables		
Ads Mediation	1.0.0	2.3.4 · November 03, 2023 Release
Advertisement Legacy	4.4.2	From Unity Registry by Unity Technologies Inc.
AI Navigation		com.unity.formats.alembic
Alembic	2.3.4 🗸	
Analytics	5.0.1	Description Version History Dependencies
Android Logcat	1.3.2	
Animation Rigging	1.2.1 🗸	Import and export Alembic files (.abc), record and stream animations in the Alembic format directly in Unity. NOTE: Supported build platforms: only Windows 64, macOS X, and Linux 64.
Apple ARKit XR Plugin	5.0.7	The Alamhic format is commonly used in animation to transfer facial cloth and other simulation between annihilations in the form of haked data
AR Foundation	5.0.7	
Authentication	3.1.0	
Build Automation		
𝚱 Burst	1.8.8 🗸	

Unity Package Manager の Alembic インポーターと再生パッケージ

シミュレーションは通常、Maya、Blender、Marvelous Designer などの他のソフトウェアで作成され、.abc alembic ファイルの拡張子を使用してエクスポートされます。Alembic はフレームごとに頂点位置を保存 するため、ファイルサイズが非常に大きくなります。シミュレーションの複雑さにもよりますが、ファイルサイズは 1 GB 以上になることが予想されます。そのため、リアルタイムのゲームプレイよりも、アニメーションの シーケンスに最適と言えます。



alembic ファイルを Project ウィンドウにドラッグしてインポートし、シーンに追加します。

Blender で作成し、Unity にインポートした Alembic 流体シミュレーション

Alembic ファイルを再生するには、Alembic トラックを使用してオブジェクトを Timeline シーケンスに 追加します。

Inspector					а	:
waterSimulation						0
-						
Clip Timing						
Start	s	0	f	0		
End	s	1.666667	f	100		
Duration		1.666667	f	100		
Clip In		0	f	0		
Speed Multiplier		1				
Animation Extrapolation Post-Extrapolate	N	one 🔻 Infinity				
Alembic Shot Asset						
Script		AlembicShotAsset				•
Stream Player	ŧ	waterSimulation (A	lem	bic Stream Pla	0	•
Preview H4 I4 D DI DH [b] 49		TimelineHolderTimeline	Time	lineHolder)		
	10	¹²⁰	lanno	180 240		
ଦିନ Alembic Track 🕴 🕴 🔍	:	AlembicS	ShotA	sset	۲	

Timeline での Alembic トラックを使った Alembic ファイルの再生

パーティクル

パーティクルは、火、煙、水、魔法攻撃などの効果を作り出すのに使われます。シーン内に何千ものパーティ クルをレンダリングしても、フレームレートにほとんど影響を与えないので、使い勝手は非常に良いです。 Unity Asset Store では、さまざまな種類のパーティクル効果を見つけることができます。Unity でパーティ クル効果を作成する方法については、eBook「Unity での上級者向けビジュアル効果制作の完全ガイド」を ご覧ください。



Unity eBook をダウンロード

AI キャラクター

ノンプレイアブルキャラクター (NPC) や群衆のシーンのアニメーションを作る場合は、Unity の Al Navigation システムを活用すれば、各キャラクターのアニメーションに何時間も費やす必要がなくなります。 このシステムにより、シーン内の何百ものキャラクターをコントロールし、建物やオブジェクトを通り抜けたり、 互いにぶつかったりしないようにすることができます。

ナビゲーションシステムを始めるには、休止状態と歩行状態の間を遷移する、-1 から 1 に変化する変数に よりコントロールされるキャラクターのブレンドツリーを作ります。

Blend Tree		Wa • Blend Tree	Iking Idle	
Walkin Idi Walkin AlMoven 0	g	• Blend Tree Wa • Blend Tree	Iking	your
Inspector Blend Tree				a:
Blend Type 1D				•
Parameter	AlMovement	>	~	
-1				1
Motion		Threshold	1	∲ ©
= 🔺 Walking	0	-1	1	
= Aldle	0	0	1	
😑 🔺 Walking	0	1	1	
Automate Thresholds				+
Compute Thresholds	Select			•
Adjust Time Scale	Select			*

NPC キャラクター向けのブレンドツリーアニメーター



Package Manager から AI Navigation パッケージをインストールします。

🖬 Package Manager		: 0	×
+ - Packages: Unity Registry - Sort: Name (asc) -			
All Services			
2D Tilemap Editor	1.0.0		
2D Tilemap Extras	3.1.2	Al Navigation Remove	
Adaptive Performance	4.0.1	1.1.5 · October 03, 2023 Release	
Addressables	1.21.19	From Unity Registry by Unity Technologies Inc.	
Ads Mediation		com.unity.ai.navigation	
Advertisement Legacy	4.4.2		
AI Navigation	1.1.5 🗸	Description Version History Dependencies Samples	
Alembic	2.3.2 🗸	Hinh-Javal NavMach components for huilding and using NavMachae at runtime and at adit time	
Analytics	5.0.1	high rever territical components for building and baing territistics at functione and at due time.	
Android Logcat	1.3.2		
Animation Rigging	1.2.1		
Apple ARKit XR Plugin	5.0.7		
AR Foundation	5.0.7		

Package Manager にある AI Navigation パッケージ

これで、NavMesh エージェントに NPC を追加できるようになります。設定を変更することで、キャラクターが 歩く速さや回転する速さ、エージェントの半径(他のキャラクターがどれだけ近づくことができるかを決める) をコントロールすることができます。

🔻 🛵 🗹 Nav Mesh Agent		0 ‡ :	
Agent Type	Humanoid	-	
Base Offset	0		
Steering			
Speed	3.5		
Angular Speed	120		
Acceleration	8		
Stopping Distance	0		
Auto Braking	~		
Obstacle Avoidance			
Radius	0.5		
Height	2		
Quality	High Quality	-	
Priority	50		
Path Finding			
Auto Traverse Off Mesh Link	~		
Auto Repath			
Area Mask	Everything	-	

NPC コントロール向けの NavMesh エージェント設定
囹

NavMesh エージェントには、動きを指示するための NavMesh Surface コンポーネントが必要です。 「GameObject」>「AI」>「Navmesh surface」を開きます。NavMesh をベイクする前にすべてのダイナ ミックキャラクターをオフに切り替えます。これにより、「Show NavMesh」オプションがチェックされていると、 青いサーフェスが現れます。NPC が歩いて向かうターゲットオブジェクトを追加します。



シーンへの NavMesh のベイクと表示。NPC は青いエリアしか歩けない。

NavMesh Agent をコントロールするにはスクリプトが必要です。これはエージェントの速度を使って、動いているときに歩きを再生するか、止まっているときに休止状態を再生するかを Animator に指示します。



NPC で NavMesh Agent をコントロールするスクリプト

スクリプトのスロットにターゲットをドラッグします。ターゲットやキャラクターは好きなだけ持つことができます。 可能であれば、一体のキャラクターにつきターゲットを1つにすると、物事をシンプルに保つことができます。

	Area Mask	Everything	-					
	🔻 🐞 🖌 Al Move (Script)		0 ‡ :	l				
	Script	🛱 AlMove						
	Target	A Target (Transform)	0					
l								
	Add Component							

NPC は常にターゲットに向かって歩き、ターゲットに到達すると止まる。

これで、再生モード中にキャラクターがターゲットに向かって歩くようになります。動いているターゲットに アニメーションを付けて NPC に追従させることもできます。Al Navigation を使えば、群衆全体のシミュ レーションをどれほど簡単に行えるかが分かるでしょう。

アニメーションの エクスポート

チームで作業したり、DCC ツールでアニメーションを洗練させたりしたい場合、何度も Unity から他のソフト ウェアにアニメーションの付いたキャラクターをエクスポートして調整する必要があるかもしれません。

FBX モデルのエクスポート

Package Manager で、FBX Exporter がレジストリからインストールされていることを確認してください。

🖬 Package Manager		: • ×
+ - Packages: Unity Registry - Sort: Na		
All Services		
Deployment		EDV Evportor Iblatic Damage
Device Simulator Devices		
Economy	3.3.0	🚔 4.2.1 - January 10, 2023 Release
Editor Coroutines		Se installed as part of the Chematic Studio feature.
Entities	1.0.16	From Unity Registry by Unity Technologies Inc.
Entities Graphics	1.0.16	contunity.commais.tox
🔒 FBX Exporter	4.2.1 🗸	
Google ARCore XR Plugin	5.0.7 .	Description Version History Dependencies
Havok Physics for Unity	1.0.16	The Unity FBX Exporter package makes it easy to send geometry and animation to any application that supports the FBX format and send them back to Unity
High Definition RP	14.0.8	with minimal effort.
In App Purchasing	4.10.0	In particular, this round-trip workflow enables you to export Unity Scenes to FBX files, import them into Autodesk® Maya®, Autodesk® Maya LT*, or Autodesk®
Input System	1.6.3	s as max® using an artist-mentily interface, export Unity-ready rbx geometry and animation, and sarely merge your changes oack into those assets to continue your work in Unity.
iOS 14 Advertising Support	1.0.0	
🔒 JetBrains Rider Editor	3.0.24 🗸	

FBX ファイルをエクスポートするには FBX Exporter をインストールする必要がある。

キャラクターのアニメーションは、Animator ウィンドウで設定されているはずです。キャラクターを右クリックし、 「**Export to FBX**」を選択します。



FBX Export オプション

エクスポートボックスが開き、保存先 フォルダーを選択できます。デフォルトでは、 Project Assets フォルダーが選択されて います。

FBX ファイルを開くソフトウェアに応じて、 ASCII またはバイナリ形式でファイルを エクスポートできます。Blender の場合は バイナリ形を選択してください。

ファイルには、3D メッシュだけ、または アニメーションファイルだけ、あるいはその 両方を指定できます。

LOD レベルが存在する場合はすべての LOD レベルをエクスポートするか、または、 最も高い詳細レベルか最も低い詳細レベル のいずれかをエクスポートするかを選択 します。

スキニングされたジオメトリがソフトウェア に表示されるように、「Animated Skinned Mesh」にチェックを入れます。他のソフト ウェアにテクスチャも表示させたい場合は、 「Embed Textures」オプションをチェック します。

ー度エクスポートすれば、他の 3D ソフトウェアでファイルを開いて、3D メッシュやアニメーションファイルを 調整することができます。



Blender で開かれた、埋め込みアニメーション付きでエクスポートされた FBX ファイル

埋め込みアニメーションは Blender のノンリニアアニメーションウィンドウで利用できます。一度ファイルを調整 したら、FBX ファイルとしてエクスポートして Unity で開くことができます。

Unity Recorder

Unity でアニメーションを作成する場合や、ゲームに高品質な映像が必要な場合は、最終的なアニメーションを ビデオファイルとしてエクスポートします。

エディターやビルドでゲームをプレイしながら生映像をキャプチャし、同時にビデオキャプチャソフトウェアを 実行すると、コンピューターのリソースに負荷がかかり、マーケティングやビデオコンテンツ目的でゲームが 思うように動作しなくなることがあります。フレームスキップによるスタッターが生じない安定したフレーム レートを確保し、最高のレコーディング品質を提供するため、Unity Recorder は、エディター内からリアル タイムで動画や画像シーケンスを作成することができます。

Package Manager から Recorder をインストールします。

= Package Manager		: n x
 → Packages: Unity Registry ▼ Sort: Name (asc) All Services 		
Profile Analyzer	1.2.2	Recorder Remove
Python Scripting	6.0.1	4.0.1. May 17 2023 [Dalaace]
Recorder	4.0.1 🗸	4.0.1 mag 17, 2023 [Release]
Relay		com.unity.recorder
Remote Config	3.3.2	
Scriptable Build Pipeline	1.21.21	
Sequences		Description Version History Dependencies Samples
Serialization		The Recorder allows you to capture and save data from the Unity Editor during Play mode: animations, videos, images, audio, arbitrary output variables (AOVs),
🔗 Shader Graph	14.0.8 🗸	and more.
Splines	2.4.0	
Sysroot Base	2.0.7	

ビデオや画像シーケンスをエクスポートする Recorder パッケージ

シーンを設定し、レコードの準備ができたら、「Window」>「General」>「Recorder」から Recorder ウィンドウを開きます。フレームレートはセットリストから設定することも、自分で定義することもできます。 ビデオファイルの安定性のために、ビットレートは「Constant」のままにしてください。

新しいレコーディングを作るには「Add Recorder」タブをクリックし、利用可能なオプションから選択して ください。ムービー、画像シーケンス、オーディオを作成できます。

Recorder		.L.				: 🗆 ×
	S Exit Play Mode Recording Mode	TART RECORDING ✓ Manual	V	Frame Rate Playback Target FPS Cap FPS	Constant 30 ✔	*
+ Add Record	er : Animation Clip Movie Image Sequence Audio FBX Alembic Clip	No recorder selected				

新規レコーダーの作成

「Movie」オプションを選択し、エクスポート規定に合うように設定をカスタマイズしてください。ビデオは Scene ビューから作成されるので、そこに表示されているものはすべて録画されます。また、特定のカメラの 映像だけを録画したり、VR 用に 360° ビューを録画したりすることもできます。

セットリストから出力解像度を設定したり、独自の解像度を定義したりすることもでき、録画時に Scene ビューはその解像度に準拠します。シーンが異なる比率、例えば 4:3 で設定されている場合、出力として 16:9 を選択すると、録画時に Scene ビューが 16:9 に変更され、エディターのシーンとは異なるビデオが 出力されますので、ご注意ください。

出力形式では、使用するエンコーダーを選択できます。Unity Media Encoder は、H.264 MP4 と WebM の コーデックをサポートしています。Apple ProRes エンコーダーは Apple ProRes 4444 と 422 のコーデックを サポートしています。アニメーション GIF のエクスポートには Gif エンコーダーが使用されます。

オーディオ付きまたはオーディオなしでビデオをエクスポートすることもでき、ファイルはプロジェクト内の Recordings フォルダーに保存されます。

Recorder								: 🗆 ×
	S Exit Play Mode Recording Mode	TART R	RECORDING			rame Rate Playback Target FPS Cap FPS	Constant 30 ✔	*
+ Add Recorder		R	ecorder Type	Mo	vie Recor	der Settings		0 , ‡
🗸 🖺 Movie		T	▼ Input					
		S	ource	Ga	Game View			
		0	output Resolution	4K	- 2160p			
		A	spect Ratio	16:	9 (1.7778)		
			Output Format					
			ncoder	Un	ity Media	Encoder		
		С	odec	H.2	264 MP4			
			ncoding quality	Hig	j h			
		In	nclude audio	~				
		▼ Output File						
			File Name <i< td=""><td colspan="3"><recorder>_<take></take></recorder></td><td>+ Wildcards -</td></i<>		<recorder>_<take></take></recorder>			+ Wildcards -

エクスポートするビデオファイルのムービー設定

プロジェクトは、JPG、PNG、または EXR ファイルの画像シーケンスとしてエクスポートすることもできます。 これらは、カラーグレーディングやビデオ効果を加えた後などに、ビデオ編集ソフトで最終的なエンコードを 行いたい場合に便利でしょう。

「Start Recording」を押すとレコードモードに入り、シーケンスが再生されると同時にビデオが録画されます。 ターゲットフレームレートで上限が設定され、ターゲットフレームレートよりも遅く再生されることに気づくかも しれません。これはエンコードプロセスにより発生するものですが、ビデオはターゲットのフレームレートで スムーズに再生されます。

レコードモードがマニュアルに設定されている場合は、設定した時間で停止を押す必要があります。レコード モード設定で、録画の開始フレームと終了フレームを入力できます。

Timeline を使用してレコーダートラックを追加することもできます。トラックを右クリックして RecorderClip を 追加します。これはアクティブブロックとして機能します。このブロックが継続する間は録画が行われます。



Timeline を使ってレコーダーの開始ポイントと停止ポイントを設定

2D Animation

Unity の 2D プロジェクトには、似たようなツールやワークフローが採用されています。2D プロジェクトで FBX の 3D モデルに相当するものは、2D スプライトです。Unity には、2D Sprite Editor、2D Tilemap system、そして 2D Sprite Shape の他にも、2D 物理演算とライティングツールセットがあります。 Animator Controller と Timeline ツールは 2D プロジェクトでも機能します。

従来の 2D アニメーションは、2D ゲームのアートを開発する上で最も時間がかかり、大変な部分です。 2D スケルタルアニメーションは、時間を節約し、滑らかで適応性の高いアニメーションを作成し、より少ない アセットでより多くのアニメーションを作成するのに役立ちます。

PSD ワークフロー

2D Animation パッケージは、キャラクターのアートワークを Photoshop から Unity に直接インポートする ことを可能にします。 2D PSD Importerパッケージは、この目的のためにレイヤー化された PSD および PSB ファイルを扱います。キャラクターのすべてのレイヤーをスプライトとしてインポートし、アプリで描かれた通りに 配置することができます。

インポートした PSD または PSB ファイルを選択すると、スプライトのインポート設定に似たオプションが 表示されますが、2D アニメーションとリギング用のオプションが追加されています。選択すべき主要オプ ションは以下の通りです。

- Include Hidden Layers:このオプションは、非表示レイヤーを含むすべてのレイヤーを PSD または PSB ファイルから読み込みます。
- Individual sprites (Mosaic):この設定は、Texture Type が Multiple に設定されている場合にのみ利用可能です。インポートしたレイヤーからスプライトを作成し、テクスチャアトラス上に配置します。キャラクターをリギングする場合は、このオプションを有効にしてください。

- Use as Rig:このオプションは、PSD または PSB ファイルと同じレイヤー階層と位置を持つキャラクター プレハブを生成します。
- Use Layer Group: PSD または PSB ファイルからレイヤーグループを追加します。これをオンに すると、スプライトを入れ替えたパーツといった、キャラクターの一部をグループ化できます。

Layer Management セクションでは、レイヤーの階層を確認したり、インポートしたりするレイヤーを手動で 追加または削除することができます。



サンプルプロジェクト『Happy Harvest』のメインキャラクター

Sprite Editor でのリギング

Sprite Editor の Skinning Editor でキャラクタースケルトンを作成できます。まず、「Create Bone」ボタンを クリックし、メインウィンドウエリアで左クリックします。最初のクリックでボーンの中心を作成し、2回目の クリックでボーンの先端の位置をマークします。このツールは、ボーン同士を連結して入れ子にします。

「Edit Bone」ボタンを使ってボーンを微調整できます。「Split Bone」ボタンはボーンを 2 つに分割します。 これは、手足を作るのに適した選択肢です。片脚のボーンを作り、膝があるべき場所をクリックすると、 ボーンは太ももとふくらはぎの部分に分かれます。

階層が正しいことを確認するために、「Preview Pose」ボタンを選択していくつかのポーズを試してください。 ボーンの位置をリセットするには、ツールバーの「Reset Pose」ボタンを押します。

スプライトをボーン、ジオメトリ、ウェイトに接続する

スプライトをボーンに割り当てるには、ジオメトリを作成する必要があります。「Auto Geometry」ボタンを 押して開始します。小さなポップアップウィンドウが開き、ジオメトリの作成方法を定義できます。 ジオメトリをできるだけシンプルに保つために、すべてのスライダーをゼロに設定するのが良いでしょう。 「Weights」オプションを有効にして、ボーンをスプライトに自動的にバインドします。「Generate For All Visible」ボタンをクリックすると、すべてのスプライトのボーンウェイトが作成され、設定されます。個々の スプライトに対してこれを行うには、スプライトをダブルクリックします。これは、特定のスプライトのジオメトリを 微調整するのに便利です。

頂点の数やジオメトリの曲がり方を完全にコントロールするには、Geometry セクションのツールを使って メッシュを手動で編集する必要があります。

使う頂点は可能な限り少なくしてください。各頂点の位置はボーンの回転に基づいて計算される必要が あるため、頂点数が少ないほどパフォーマンスを効率化できます。また、頂点数が少ないほど、少ないポイント にウェイトを設定しやすくなるため、曲がったメッシュの見栄えがより良くなります。さらに、各ターゲットプラット フォームには理想的なゲーム頂点数があるので、ジオメトリを最適化するのが良いでしょう。



アニメーションを最適化するためには、ジオメトリをきれいにすることが重要だ。関節のような変形が顕著な部分では特にそう言える。

ウェイト

スプライトのジオメトリが整ったら、次はウェイトの設定です。ウェイトは、各頂点に対するボーンの影響を 0から1までで定義します。0はボーンが頂点に対して何の影響力も持たないことを意味し、1は頂点が ボーンに接着されたように動くことを意味します。メッシュのウェイトをうまく調整することで、リアルな 曲がり方を生み出すことができます。ウェイトの設定を誤ると、ゲームの視覚効果が崩れたり、スプライトが 歪んだりします。

ウェイトの設定を始めるには、どのボーンが特定のスプライトに影響するかを定義する必要があります。 「Bone Influence」ボタンをクリックし、スプライトを選択します。小さなポップアップウィンドウから、どの ボーンが選択したスプライトに影響を与えるかを設定できます。スプライトに影響を与える必要のあるボーン だけを設定し、残りを削除します。

スプライトリゾルバーと 2D IK

Unity サンプルプロジェクト『Happy Harvest』を見ると、複数のスプライトが同じボーンによってどのように 影響を受けているのかがわかるでしょう。例えば、手のスプライトには、横、前、後ろなどのバリエーションが あり、これらはアニメーションの過程やゲームプレイ中に入れ替わります。さらに、アニメーションをゲーム ワールドに反応させるため、2D インバースキネマティクス(IK)は、回転を計算し、一連のボーンをターゲット 位置に移動させることができます。



2D IK リゾルバーを使用して『Happy Harvest』のメインキャラクターにアニメーションを付ける様子。左の画像では、ポーンの位置と回転を修正している。 右の画像では、スプライトリゾルバーを使って手のスプライトを変更している。



まとめ

Unity でアニメーションを作成しようとしているアーティストなら、Unity ドキュメントの全文と併せ、Unity Learn、Unity ブログ、Animation フォーラムをご覧ください。

開発者およびアーティスト向けの Unity の詳細な eBook は、すべてベストプラクティスハブや Unity マニュアルのベストプラクティスセッションから見つけることができます。

Unity の Unity Product Board では、現在開発中の アニメーション機能の概要に加え、次のリリースに ついて知ることができます。機能リクエストを追加することもできます。

皆様の開発の成功をお祈り申し上げます。



unity.com