



CHARACTER FOUNDATIONS

KINEFX リギング | FUR DUDE

このレッスンでは、2足歩行キャラクタ、ファー・デュード (Fur Dude) のリギングとアニメーションを行い、その後ファーを追加します。既存のジオメトリから始め、スケルトンを描画し、ジオメトリをキャプチャしてから、アニメーションリグのリグコントロールを構築します。その後、歩行サイクルのキーフレームを設定し、サーフェスにファーを追加します。

このレッスンでは、Houdini の新しい SOP ベースのプロシージャリギングツール **KineFX** を使用します。主にリターゲットワークフローに使用するツールですが、キャラクタやクリーチャのリギングに使えるツールも含まれています。ただし、ツールはどれも発展の過程にあります。このレッスンで体験していただくのは、現時点で可能なことです。今後のリリースで、KineFX やアニメーションのワークフローはさらに拡張され、洗練されていく予定です。

レッスンの目標

クリーチャ「ファー・デュード」のリギングとアニメーション、
ファーの追加を行います。

学習内容

- **KineFX** ジョイントを使用してスケルトンを作成する方法
- 変形およびリジッドジオメトリをスケルトンにキャプチャする方法
- キャプチャリグをデジタルアセットにラップする方法
- コントロールを追加して、アニメーションリグを構築する方法
- 歩行サイクルをアニメートする方法
- クリーチャにファーを追加する方法
- Solaris と Karma を使用してレンダリングする方法

使用する機能とソフトウェア

Houdini 19.5+ の機能を前提として、書かれています。

このレッスンの手順は、
以下の Houdini 製品で実行可能です。

Houdini Core	✓
Houdini FX	✓
Houdini Indie	✓
Houdini Apprentice	✓
Houdini Education	✓

ドキュメントバージョン 2.00 | 2022 年 11 月
© SideFX Software



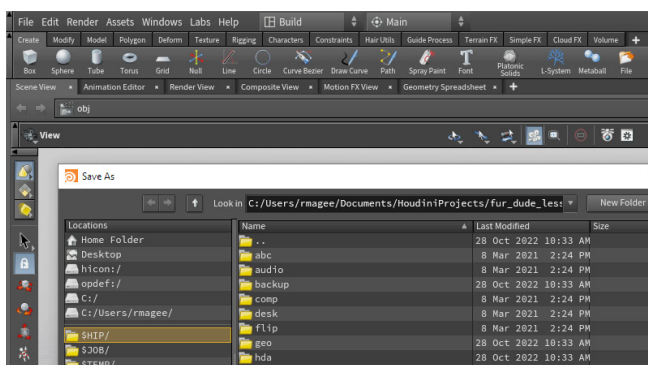
パート1

スケルトンの描画

シーンファイルを開き、ファー・デュードのジオメトリを確認してから、Skeleton ツールを使用してジョイントを配置します。このツールは、ジョイントを作成して名前を付けたり、アニメートしたいキャラクタに合わせて調整するために使えます。

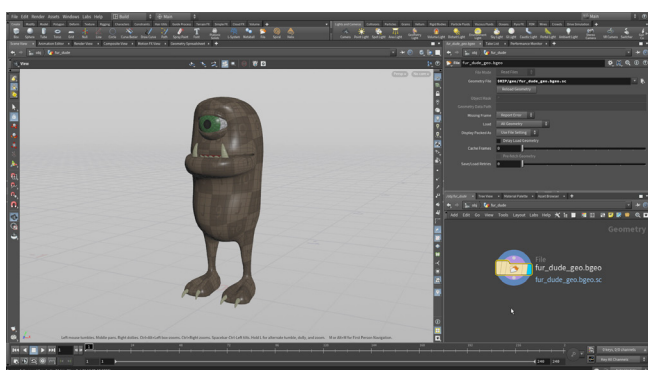
プロジェクトファイル

SideFX.com のファー・デュードのチュートリアルページ(このドキュメントを入手した場所です)から、*furdude_lesson_start* ディレクトリをダウンロードします。名前を *furdude_lesson* に変更し、**Houdini Projects** ディレクトリに配置してください。



01 **File > Set Project** を選択します。先ほどダウンロードした *furdude_lesson* ディレクトリに移動し、**Accept** を押します。これにより、先ほどコピーしたプロジェクトディレクトリとそのサブフォルダに、このショットに関連するファイルがすべて配置されるようになります。

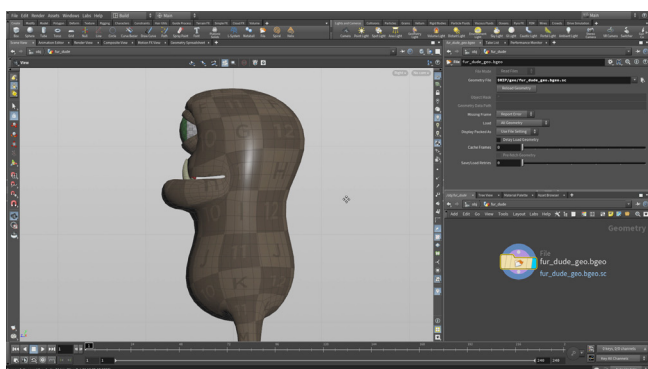
File > Open を選択すると、新しい *furdude_lesson* ディレクトリが表示されます。*furdude_start.hip* という名前のファイルを開きましょう。**File Save As...** を選択し、*furdude_01.hip* にファイル名を変更します。**Accept** をクリックして保存します。こうしておく、後でもう一度レッスンをやりたくなったときに、手つかずのスタートファイルに戻れます。



02 シーンを開くと、*fur_dude_rig* という名前のオブジェクトが1つ表示されます。これから、**KineFX** ツールセットを使用して作成したボーンに、このジオメトリをキャプチャしていきます。

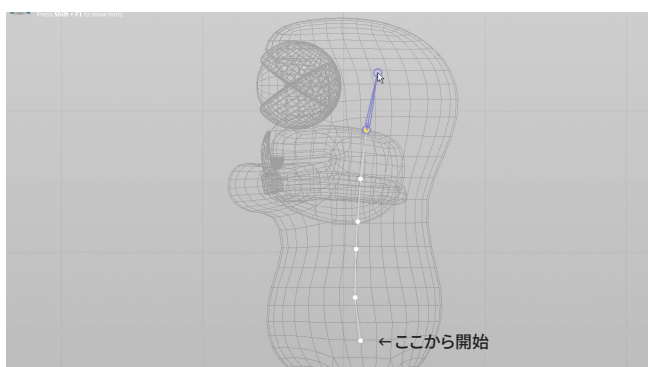
ノードを**ダブルクリック**して、ジオメトリレベルに入ります。このレッスンプロジェクトでは、ディスクから *fur_dude_geo.bgeo* ファイルをインポートしている **File** ノードを確認できます。

このジオメトリには、プリミティブカラーやグループなどの情報が保存されています。情報を確認するには、ノードを **MMB クリック**して、リストされたアトリビュートやグループを表示します。これらのグループは、後でジオメトリをキャプチャする際に使用します。



03 Scene View にカーソルを移動し、**スペースバー + B** を押して4画面ビューにします。右上のアイコンから、**Link Ortho Views** をオンにします。こうすると、Top、Front、Right のビューをすべて同時にパン、ズームできるようになります。

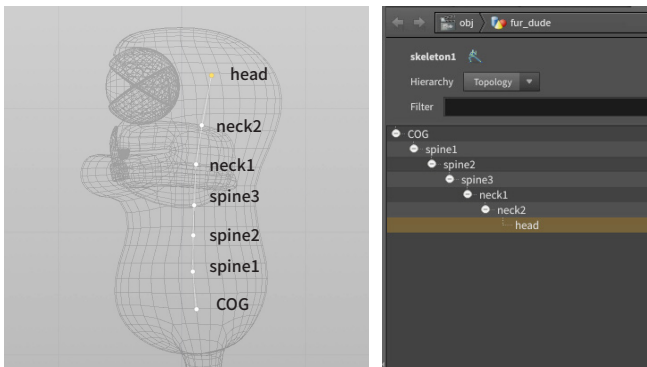
Right ビューにカーソルを移動し、再度**スペースバー + B** を押します。ジョイントの描画には、このビューがやりやすいはずです。



04 ネットワークビューで **Tab > Skeleton** を選択し、File ノードの横に新しいノードを配置します。そのノードに **Display フラグ** を設定します。File ノードに **Template フラグ** を設定して、skeleton ノードの作業時に、ジオメトリがグレーのワイヤーフレームとして表示されるようになります。

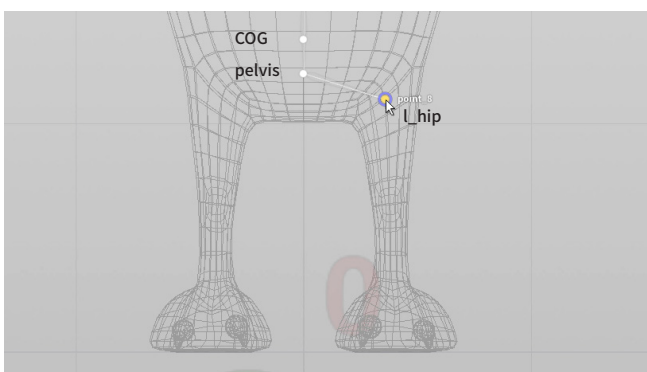
Handle ツールが選択されていることを確認します。上部のバーで、**Joint Placement** を **Freehand** に設定します。こうすると、ジオメトリとは関係なく、コンストラクション平面での描画が可能になります。脚のすぐ上に最初のジョイントを配置したら、図のように、上方向に6つのジョイントを配置します。

MMB クリックして、ジョイントの描画を終了します。



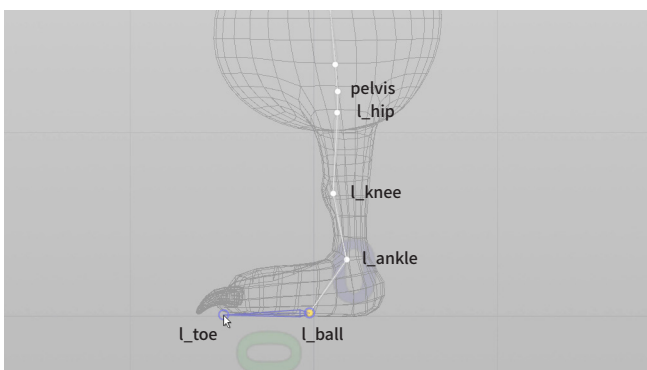
05 上部のバーで、**Mode** を **Modify** に設定します。これで、ジョイントを編集できるようになります。最初のジョイントをクリックして、上部のバーで **Name** を **COG** に設定します。

パラメータエディタのタブ領域で、**+**(プラス)記号をクリックします。**New Pane Tab Type > Animation > Rig Tree** と選択します。すると、スケルトンジョイントを表示したペインが表示されます。2つ目のジョイントをダブルクリックして、**spine1** と名前を付けます。Scene View または **Rig Tree** を使用して、残りのジョイントに図のように名前を付けます。



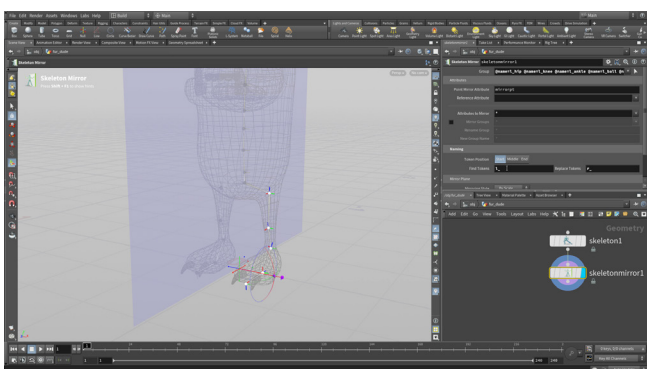
06 上部のバーで、**Mode** を **Create** に戻します。デフォルトでは、選択したジョイントの端から描画を始めることになります。**MMB** クリックして、その動作を停止します。Right ビューで **COG** ジョイントをクリックし、その下に **pelvis** ジョイントを描画します。

Scene View で、**スペースバー + B** を押して 4 面ビューに戻ります。Front ビューで、キャラクターの左側に **hip** ジョイントを描画します。



07 Right ビューに戻り、脚の最後の4つのジョイントを図のように描画します。

Modify モードに戻し、**Rig Tree** を使用してジョイントの名前を変更します。または、ジョイントを選択し、上部のバーで名前を変更することもできます。**pelvis** の後、すべてのジョイントに「**L**」の接頭辞を付けるのは、これらのジョイントを左脚に使用するためです。



08 Scene View で、**Tab > Skeleton Mirror** を選択します。このコマンドは、すべてのジョイントのミラーコピーを作成します。パラメータエディタに移動して、**Group** の横の矢印をクリックします。脚のジョイントのみを選択し、Enter を押します。脚のみがミラー化されました。

Naming で、**Find Tokens** を **L**、**Replace Tokens** を **r_** に設定します。これで、右脚のジョイントに適切な名前 (接頭辞 **r_**) が付けられます。

File > Save を選択して、ここまでの作業を保存します。



KINEFX とオブジェクトレベルのリギング

Houdini の KineFX ツールは、ジオメトリ (SOP) レベルでのジョイントベースのワークフローを提供します。Houdini の他のキャラクターワークフローはボーンベースで、この場合、作業は主にオブジェクト (OBJ) レベルで行うことになります。

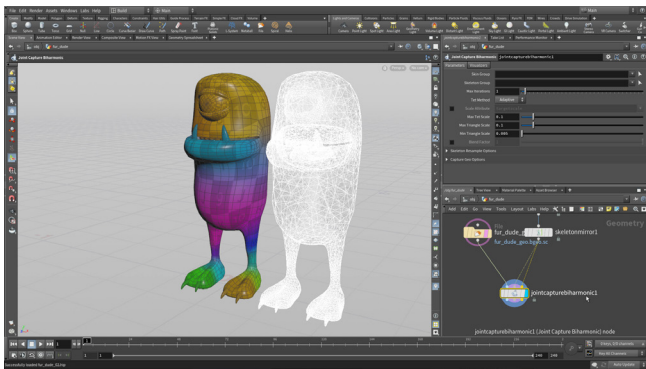
KineFX ワークフローでは、ジョイントは基本的にカーブ上のポイントにすぎません。このように扱うことで、SOP レベルのツールでリグを操作する可能性が大きく開けました。ここでは、キャラクターやクリーチャのリギング専用設計されたツールについて学習します。



パート2

ジオメトリのキャプチャ

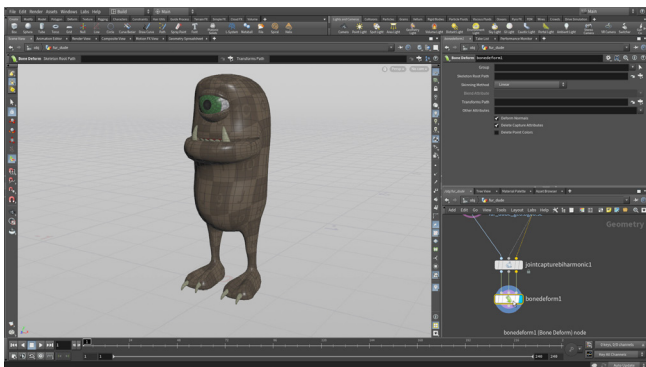
キャラクターのリギングには、ジオメトリをスケルトンジョイントにキャプチャする操作が含まれます。このようにすることではじめて、ジョイントの回転によってジオメトリが変形したり曲がるようになります。Houdini は、最初のキャプチャで素晴らしい結果を得られる Biharmonic (重調和) キャプチャ手法を使用しており、リグをすぐにテストできます。その後、キャプチャウェイトをペイントして結果を微調整し、キャラクターを動かせるようにします。



01 **スペースバー + B** を使用して、Scene View をパースビューに変更します。ネットワークビューを少し大きくして、作業スペースを確保します。

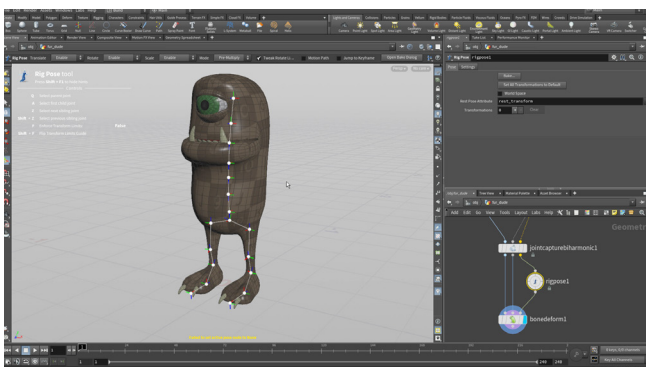
ネットワークビューで **Tab > Joint Capture Biharmonic** を選択し、このノードを skeleton ノードの下に配置します。 **fur_dude_geo** を **jointcapturebiharmonic** ノードの **1** つ目の入力に接続します。 **skeletonmirror** ノードを **jointcapturebiharmonic** ノードの **2** つ目と **3** つ目の入力に接続し、 **Display フラグ** を設定します。

ジオメトリにキャプチャウェイトが表示されるようになりました。後でこれらを微調整およびペイントして、ジオメトリの変形をセットアップします。

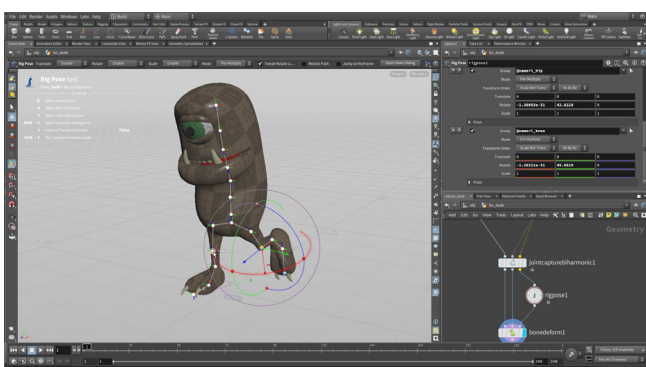


02 ネットワークビューで、 **Tab > Bone Deform** を選択します。その **3** つの入力に、 **jointcapturebiharmonic** の **3** つの出力を接続します。 **bonedform** に **Display フラグ** を設定します。

このレッスンでは、おそらく変形する必要のない歯、つめ、目をキャプチャします。これらは後ほど分離し、別の方法でキャプチャしていきます。



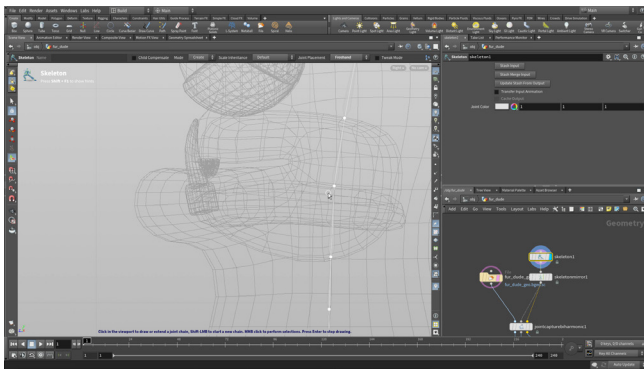
03 次に、 **Tab > Rig Pose** を選択してノードを配置します。それを、 **jointcapturebiharmonic** と **bonedform** の 3 つ目の入力を接続している 3 つ目のライン上に移動して、チェーンに追加します。ここでリグにアニメーションを付けます。 **rigpose** を使用してジョイントを回転させたり、キーフレームを設定することができます。



04 **rigpose** ノードを選択して、Scene View で **Handle** ツールがアクティブなことを確認します。さまざまなジョイントを **選択** および **回転** して、変形をテストします。これは後でリセットできるので、自由に試してください。

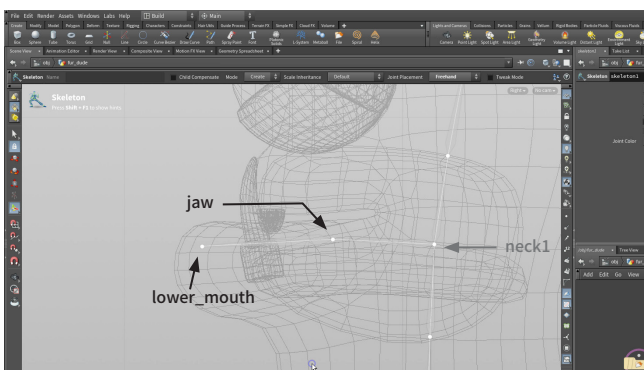
パート3 ボーンの追加

口の領域にもっとボーンが欲しいですね。Houdini のプロシージャルネットワークなら、前に戻ってジョイントを追加できます。また、Biharmonic (重調和) キャプチャを含む他のすべてのノードは、変更を反映して更新されます。クリーチャのリグの最初のセットアップを柔軟に行うことができます。

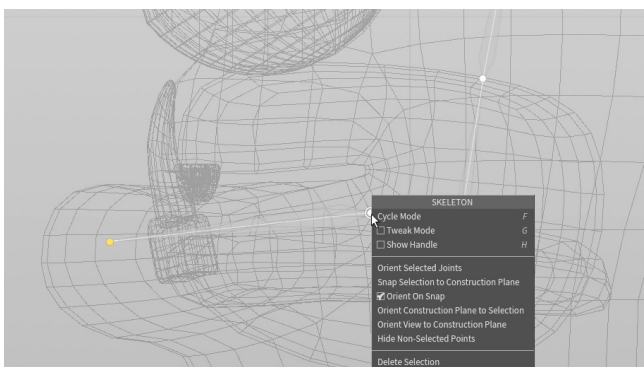


01 ネットワークビューで、**skeleton** ノードに **Display フラグ**を設定し、元の **File** ノードに **Template フラグ**を設定します。Scene View で **スペースバー + B** を2回押して、Rightビューに移動します。

skeleton ノードに **Display フラグ**を設定します。**skeleton** ノードを選択して、**Handle** ツールをオンにします。**Mode** を **Create** に設定します。スケルトンにさらにジョイントを追加できるようになります。KineFX ならこの時点でボーンを追加でき、ネットワーク内のその他のノードがプロシージャルなおかげで、後からの変更も可能です。



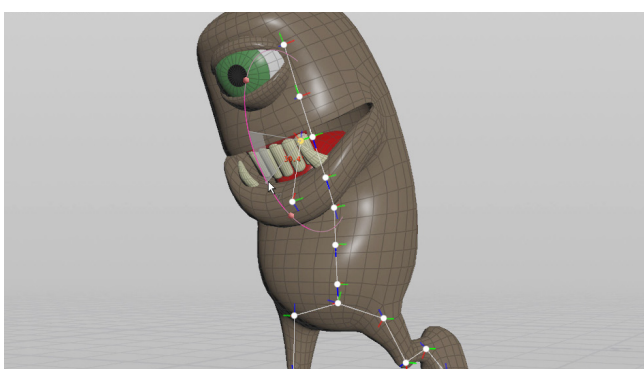
02 **neck1** ジョイントをクリックして描画を開始したら、顎および下唇をクリックして、2つのジョイントを作成します。ジョイントを配置できたら、**MMB クリック**して描画を終了し、**Mode** を **Modify** に戻します。ジョイントをクリックして名前を **jaw** と **lower_mouth** に変更します。**Rig Tree** ビューで名前を変更することもできます。



03 ジョイントの位置は **Modify** モードで編集します。上部のバーで **Tweak Mode** を **オン**にすると、ジョイントをクリック&ドラッグで移動できます。1つのジョイントを移動すると、子のジョイントもすべて移動するため、子を元の位置に戻す作業が発生します。この動作を回避するには、**Child Compensate** を **オン**にします。

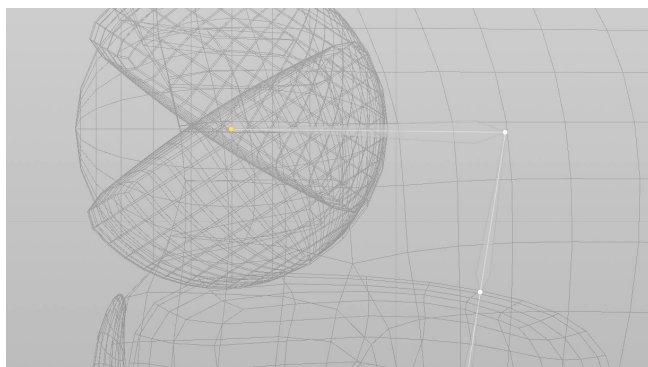
ジョイントで **RMB クリック**すると、ジョイントの分割、親子の解除、コピー、ペーストなどのオプションが表示されます。このスケルトンでは使いませんが、こうしたオプションは知っておくと便利です。

ジョイントをミラー化することもできますが、このネットワークでは別のノードを使用します。



04 **bonedeform** ノードに **Display フラグ**を設定し、**File** ノードの **Template フラグ**をオフにします。ジオメトリが再構成され、新しいボーンからキャプチャが実行されます。**rigpose** ノードを選択して、**Handle** ツールを選択します。

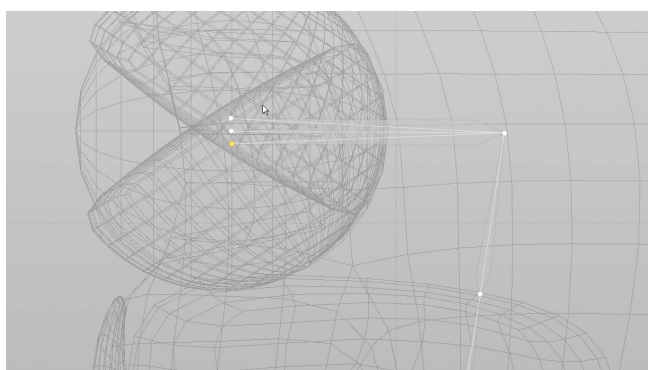
新しい **jaw** ジョイントをクリックし、下に回転させます。回転はしますが、上唇と下唇が一緒に動きます。下唇のみが影響を受ける方が良いですね。チュートリアル後のセクションでキャプチャウェイトをペイントし、これを修正します。



05 skeleton ノードに **Display フラグ** を再度設定し、元の **File** ノードに **Template フラグ** を設定します。

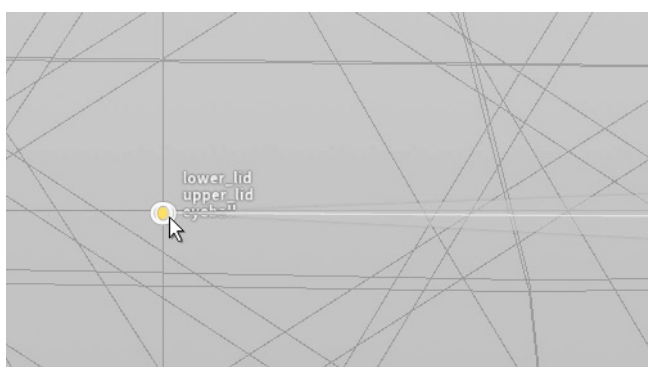
skeleton ノードを選択して、**Handle** ツールをオンにします。**Mode** を **Modify** に設定します。**head** ジョイントをクリックし、**Tweak Mode** を使用して、眼球の高さと揃うまで下げます。

Mode を **Create** に設定します。**head** ジョイントをクリックしたら、眼球の中央でクリックして新しいジョイントを作成します。**Modify** モードに切り替えて、**Name** でこのジョイントに **eyeball** と名前を付けます。



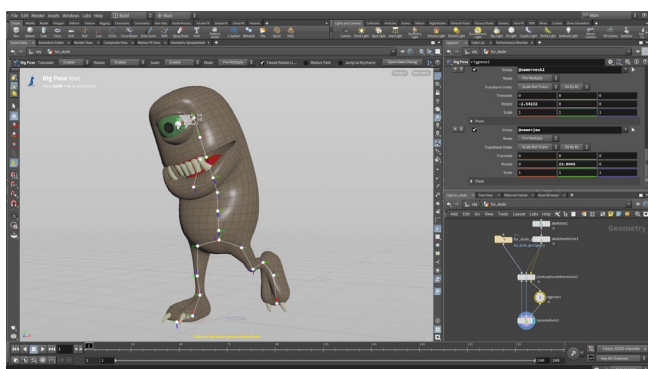
06 **Mode** を **Create** に設定します。**head** ジョイントをクリックしたら、**eyeball** ジョイントの上をクリックして新しいジョイントを作成します。**MMB クリック** して選択を解除します。再度 **head** ジョイントを選択して、**eyeball** ジョイントの下をクリックして新しいジョイントを作成します。

Modify モードに切り替えて、**Name** でジョイントにそれぞれ **upper_lid** と **lower_lid** と名前を付けます。アニメーション時にはこれらのジョイントでまぶたを回転させますが、ジョイントは目の中央に配置しておく必要があります。



07 ジオメトリの **Template フラグ** をオフにします。**upper_lid** ジョイントを選択し、**Tweak Mode** を使用して **eyeball** ジョイントに重なるようにドラッグします。**lower_lid** でもこの作業を繰り返して、**eyeball** ジョイントに重ねます。

これで、3つすべてのジョイントが同じ場所に揃いました。後で、それぞれ個別にジオメトリをアタッチして、独立してアニメートできるようにします。



08 **bonedeform** ノードに **Display フラグ** を設定します。ジオメトリが再構成され、新しいボーンを使用してキャプチャされます。

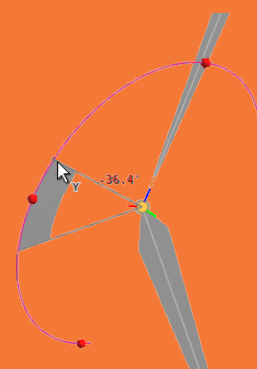
ジオメトリはまだ適切にアタッチされていないので、新しいジョイントにポーズを付けることはできません。いくつかの手順を行い、体と舌にキャプチャウェイトをペイントした後で、ようやく可能になります。



ジョイントの向き

ジョイントベースのシステムでは、各ジョイントはポイントです。ジョイントの向きが非常に重要なのは、それによって、フォワードキネマティクスでジョイントがどう回転するかが決まるからです。

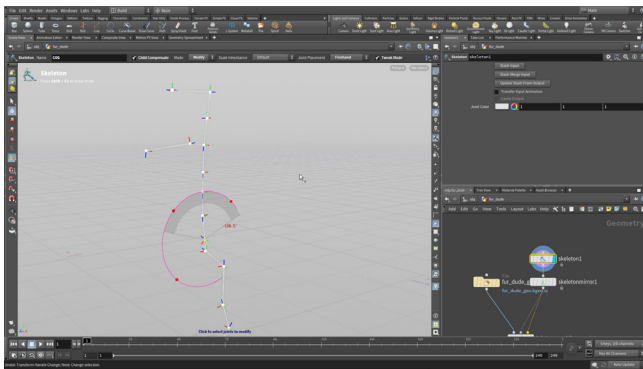
KineFX に含まれるツールを使用すると、チェーンに沿って次のジョイントの方を向くようにすることができます。また、**Child Compensate** オプションを使って手動でジョイントを回転すると、チェーン内の他のジョイントに影響を与えずに、ジョイントの向きを変更できます。



パート4

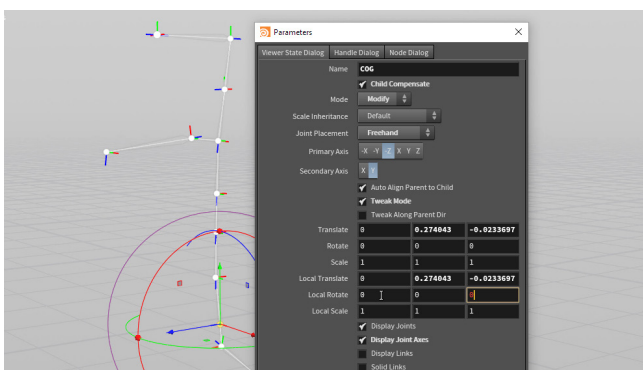
ジョイントの向き

キャラクターをアニメートするときは、ジョイントの向きがリグの操作に大きく影響します。このパートでは、一部のジョイントの向きを手動で変更します。その後、Orient Joints ノードを使用して、残りすべてのジョイントの向きをマイナス Z 軸基準に変更します。後の工程でリグを評価する際、必要に応じてここに戻って向きを微調整してください。



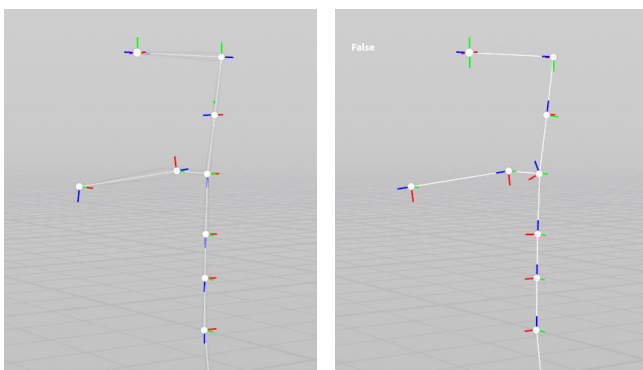
01 もう一度、*skeleton* ノードに **Display フラグ**を設定します。Scene View で **RMB クリック**して **Display Joint Axes** を選択し、ジョイントの向きが確認できるようにします。**COG** ジョイントを選択して **RMB クリック**し、**Show Handle** を選択します。これで、ジョイントを回転させると、リグ全体が一緒に動きます。**Ctrl + Z** を押して元に戻しましょう。

Child Compensate チェックボックスをオンにすると、残りのスケルトンに影響を与えることなく、ジョイントを回転できます。**Ctrl** キーを使用して、45度のインクリメントに拘束します。



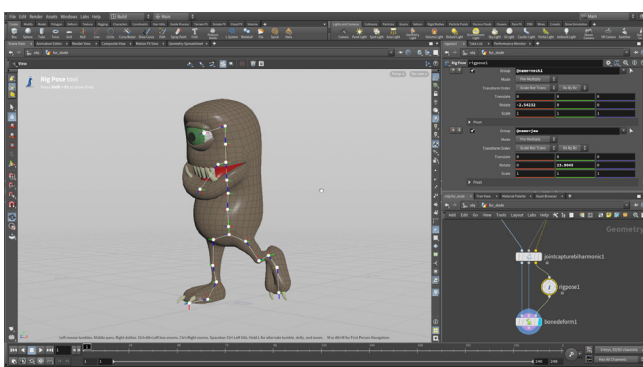
02 Scene View で、**P** を押してこのジョイントのパラメータを表示します。ワールドで COG の向きを変更するために、**Rotate** と **Local Rotate** を **0, 0, 0** に設定します。

何も無い空間でクリックして COG ジョイントの選択を解除したら、**pelvis** ジョイントをクリックします。**Rotate** と **Local Rotate** を **0, 0, 0** に設定します。



03 Skeleton ジョイントの後に、**Orient Joints** ノードを挿入します。このノードは、デフォルトではプラス Z 軸を基準にジョイントの向きを変更します。**Orient Group** の横の矢印をクリックして、Scene View ですべてのジョイントを選択します。**Ctrl** を押しながら **COG** ジョイント、**pelvis** ジョイント、**neck1** ジョイントを選択し、選択から除外します。**Enter** を押します。

これで、選択解除した3つのジョイント以外、すべてのジョイントがプラス Z 軸を基準とした向きになります。



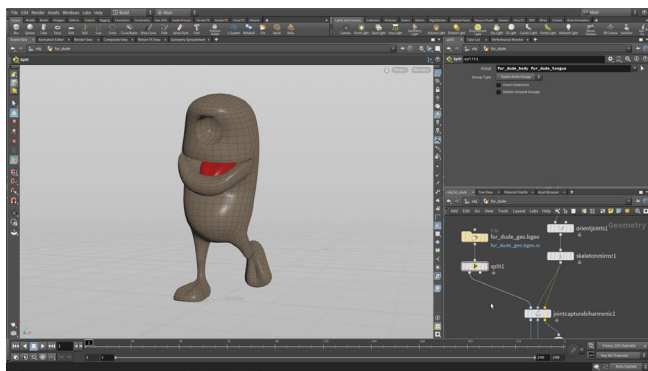
04 **Bone Deform** ノードに Display フラグを再度設定します。他のすべてのノードが更新され、新しいジョイントの向きが受け入れられます。

ジョイントの向きを変える前後では、違いはないように見えますが、アニメーション時のキャラクターのポーズ決めや操作に影響してきます。

パート5

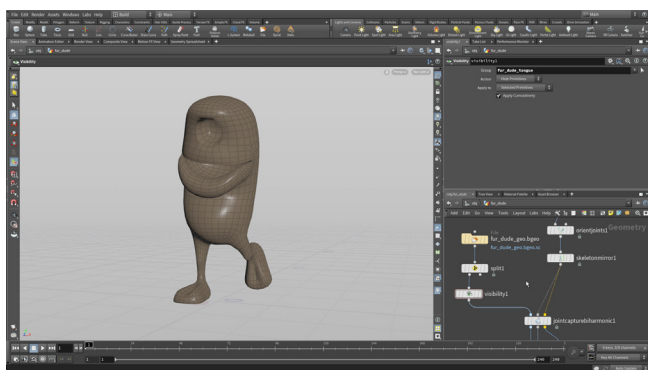
キャプチャジオメトリの取り付け

現在はスケルトンジョイントを使用してジオメトリをキャプチャし、キャラクターの各ポイントにキャプチャウェイトを割り当てています。これをさらに詳細にコントロールするには、そのジョイントの影響を延長するジョイントにカーブを取り付けます。このようにすると、かなり素早く素早く目的を達成することができます。



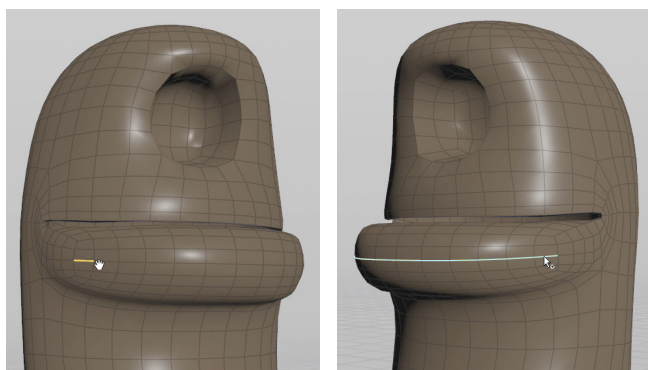
01 ネットワークビューで、**Tab > Split** を選択して、**File** ノードと **jointcapturebiharmonic** ノードの間にノードを配置します。split ノードの1つ目の出力を、**jointcapturebiharmonic** ノードの1つ目の入力に接続します。

Group フィールドのプルダウンメニューをクリックして、**fur_dude_body** グループと **fur_dude_tongue** グループを選択します。これで、この2つは split ノードの1番目の出力に送られ、眼球、歯、つめなど残りのパーツは2番目の出力に送られるようになります。このジオメトリにウェイトをペイントしてから、残りのパーツを別の方法を使ってスケルトンにバインドします。

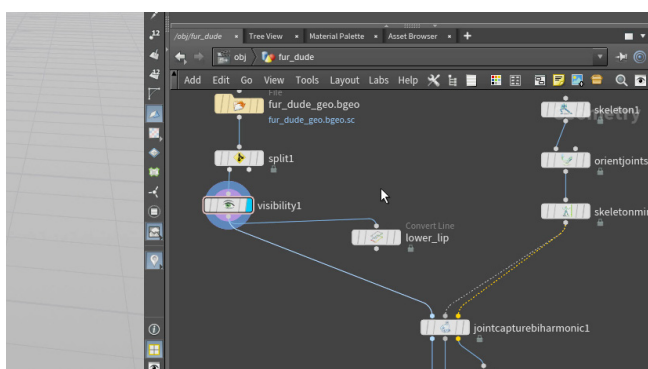


02 ネットワークビューで、**Tab > Visibility** を選択して、**jointcapturebiharmonic** ノードと **bonedeform** ノードの間にノードを配置します。**Group** フィールドのプルダウンメニューをクリックして、**fur_dude_tongue** グループを選択します。

このノードは選択したジオメトリを非表示にします(削除はしません)。つまり、Paint Capture Weights ツールを使用する際に重要となるポイント番号とプリミティブ番号が変更されません。この処理でジオメトリを非表示にするたびに、これらの情報が変更されてしまは大変です。

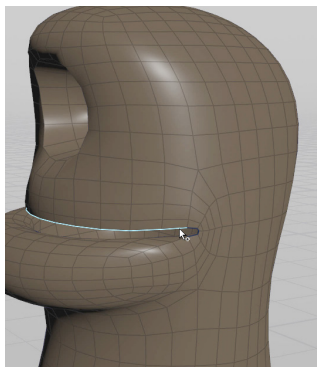
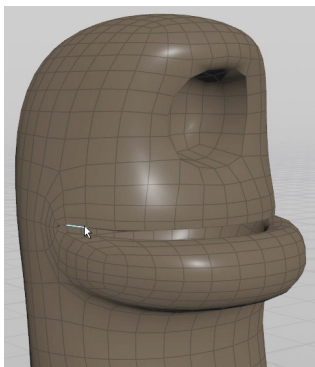


03 **S** を押して **Select** ツールにしたら、**3** を押してエッジ選択を有効にします。下唇の左側中央でエッジを選択したら、**Shift + A** を押して反対側のエッジを選択します。



04 **Tab > Curve from Edges** を押すと、選択されたエッジがジオメトリから抽出されます。このノードの名前を **lower_lip** に変更します。

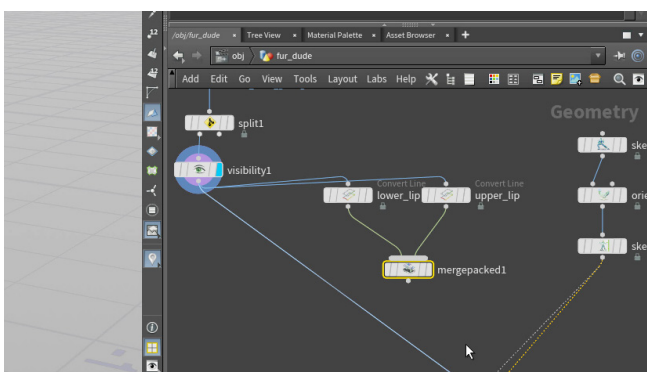
curvefromedges ノードが **File** ノードと **jointcapturebiharmonic** ノードの間に配置されています。それを横に移動して、**File** ノードの出力を **jointcapturebiharmonic** ノードの1つ目の入力に再接続します。これにより、**curvefromedges** ノードが横に分岐します。



05 **S**を押して**Select** ツールにしたら、**3**を押してエッジ選択を有効にします。上唇の左側でエッジを選択したら、**Shift + A**を押して反対側のエッジを選択します。

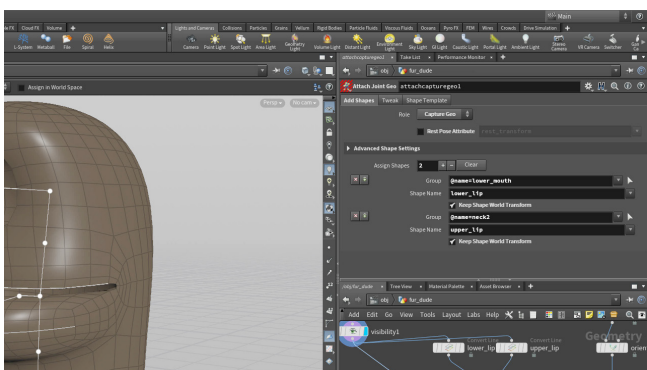
Tab > Curve from Edges を押すと、選択されたエッジがジオメトリから抽出されます。このノードの名前を **upper_lip** に変更します。

このノードを、もう片方のノードで行ったのと同じように分岐させます。



06 **Tab > Merge Packed** を選択し、このノードを抽出した2つのカーブのノードの下に配置します。その後、その2つのノードの出力を **mergepacked** ノードに接続します。

Tab > Attach Capture Geo を押します。このノードをネットワークに追加したら、**skeletonmirror** ノードを1つ目の入力に接続し、**mergepacked** ノードを3つ目の入力に接続します。



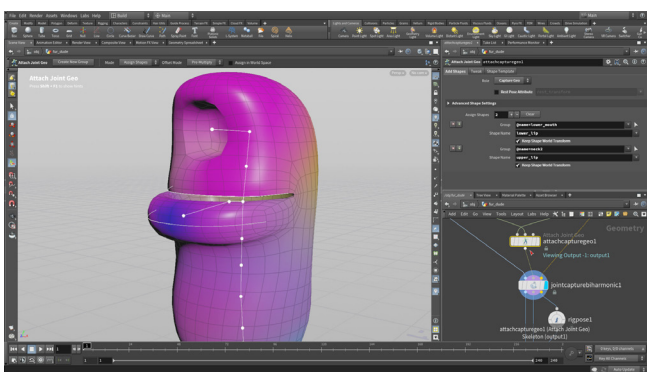
07 **add_capture_geometry** ノードの出力を、**jointcapturebiharmonic** ノードの中央の入力に接続します。**Advanced Shape Settings** で、**Assign Shapes** の+(プラス)記号ボタンを2回クリックします。1つ目は次のように設定します。

- **Group** を **@name=lower_mouth** にする
- **Shape Name** を **lower_lip** にする

2つ目は次のように設定します。

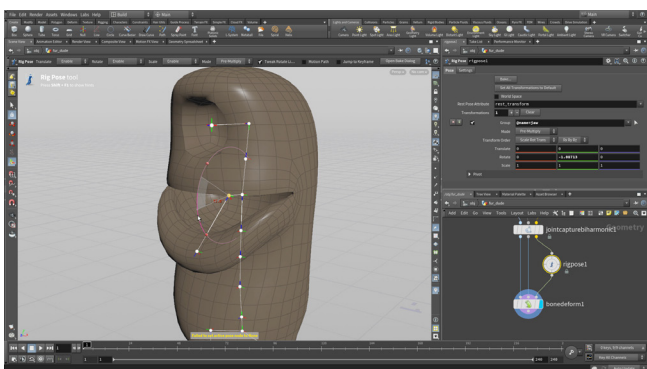
- **Group** を **@name=neck2** にする
- **Shape Name** を **upper_lip** にする

どちらの形状でも、**Keep Shape World Transform** をオンにします。



08 **jointcapturebiharmonic** に **Display フラグ** を設定します。ここで **add_capture_geometry** ノードを **バイパス** して、キャプチャウェイトの違いを表示します。

このノードがオンのとき、取り付けられたカーブは、関連付けられたジョイントへのジオメトリのキャプチャを補助します。これにより、キャラクターのセットアップをより細かくコントロールできるようになります。



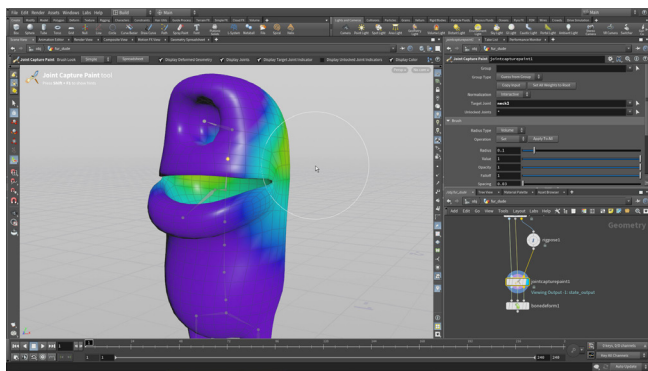
09 **bonedeform** ノードに **Display フラグ** を設定します。**rigpose** ノードをクリックしたら、パラメータエディタで **Transformations** の横の **Clear** ボタンをクリックします。これでジョイントがリセットされます。**Handle** ツールをアクティブにして、Scene View で **jaw** ジョイントをクリックします。それを下に **回転** させて唇を下げます。

この操作で、腹部のパーツがまだ変形していることがわかります。これを修正するには、キャプチャウェイトをペイントして、キャプチャウェイトを各種ジョイントに割り当て直す必要があります。

パート6

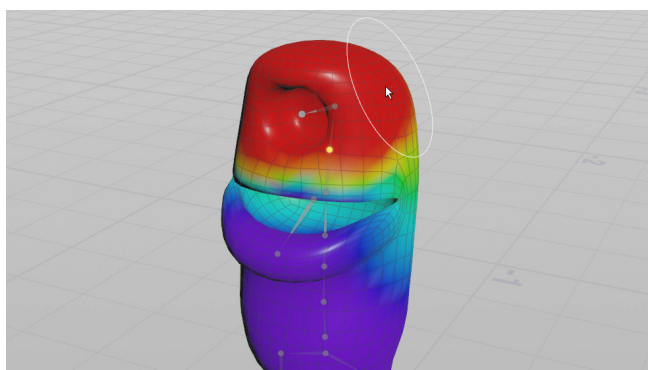
キャプチャウェイトのペイント

Biharmonic (重調和) キャプチャは、スケルトンの各種ボーンに関連付けられたキャラクターのジオメトリに、ウェイトを追加します。次は、新しいノードを追加して、ブラシワークフローによってキャプチャウェイトを調整します。このクリーチャーのレッスンで目指すのは、上唇が下唇のジョイントの影響を受けないようにすることと、足の領域のウェイトを微調整することです。



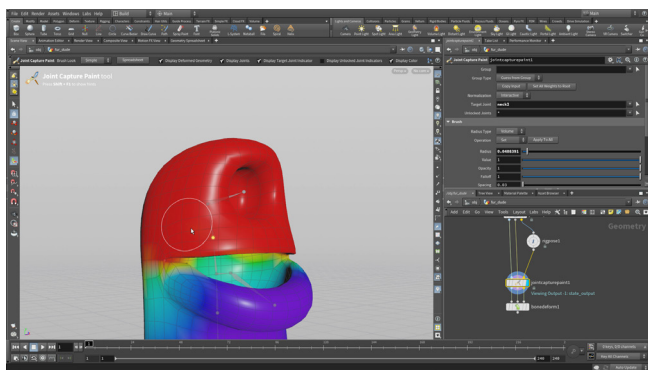
01 ネットワークビューで、**Tab > Joint Capture Paint** を押して、ノードを **bonedeform** の真上に配置し、3つすべてのコネクタを接続します。**Display フラグ**を設定してから、**Target Joint**の横のプルダウンメニューをクリックして、**neck2** ジョイントを選択します。

Scene View には、カーソルに大きい円形のペイントアイコンが表示され、これを使ってウェイトをペイントします。頭部の領域をペイントすると、ジオメトリのこの部分が **neck2** ジョイントにキャプチャされます。



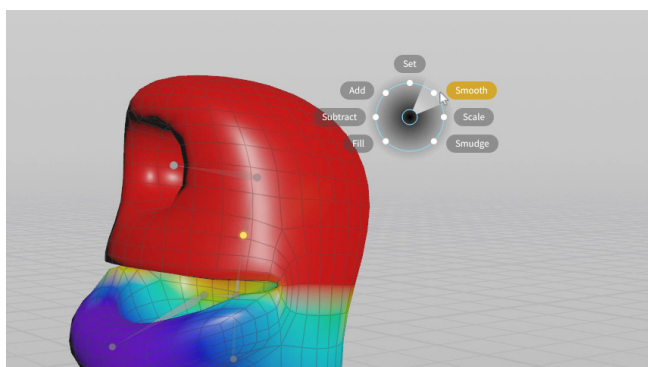
02 頭部と上唇の領域を neck2 ジョイントにキャプチャする必要があります。影響の度合いが最も強い領域のジオメトリは、赤くなります。

頭の上部では太いストロークを使用します。その後、マウスの **スクロールホイール** を使用してブラシの半径を小さくするか、パラメータエディタの **Brush** タブに移動して半径を変更します。



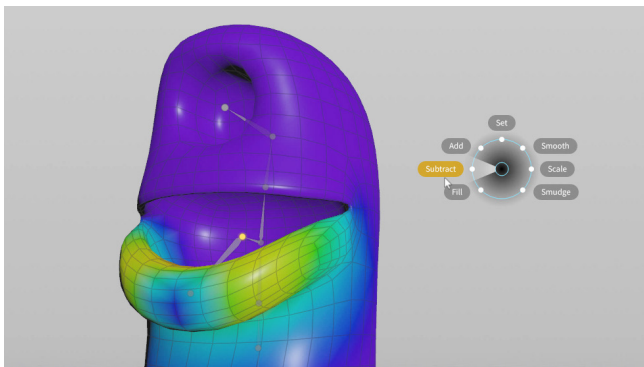
03 上唇をペイントして、この領域に対する **neck2** ジョイントの影響を強くします。ペイントするのは上唇のみで、下唇には及ばないようにします。間違えたら、**Ctrl + Z** でストロークを取り消しましょう。タンブルして口の中が見えるようにして、上部もペイントします。

オペレーションコントロールツールバーには、**Display Deformed Geometry**、**Display Joints**、**Display Color** のオプションが表示されています。これらのオン/オフを切り替えることで、キャプチャウェイトを評価しながらペイントできます。



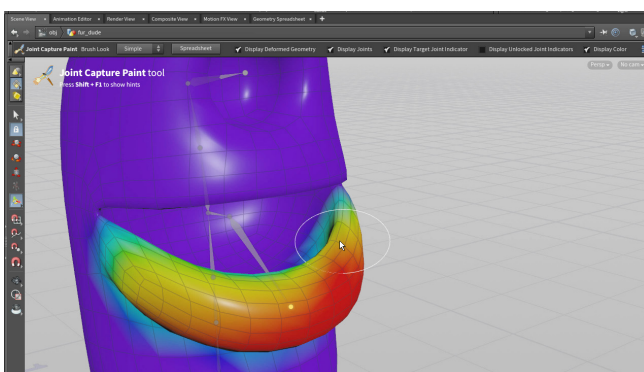
04 **F** を押して **Smooth** を選択し、この部分のペイントを周りの領域のキャプチャウェイトに馴染ませます。ブラシの半径を少し大きくして、滑らかにしましょう。

完了したら、**rigpose** ノードをクリックします。**Handle** ツールをアクティブにして、Scene View で **jaw** ジョイントをクリックします。それを下に回転させて唇を下げます。今回は、上唇は動かず、下唇およびその下のパーツが回転しているのが分かります。



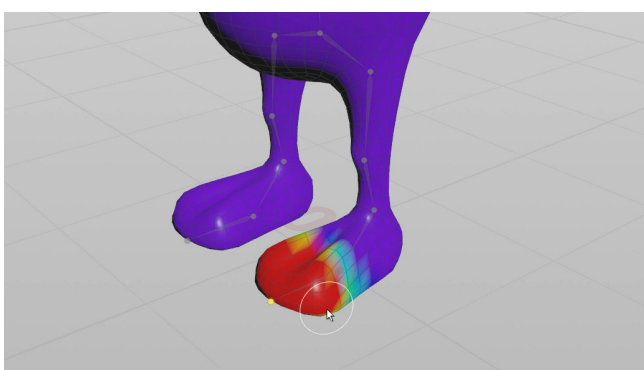
05 *jointcapturepaint* ノードを選択します。Capture Region フィールドのプルダウンメニューをクリックして、**jaw** ジョイントを選択します。F を押し **Subtract** を選択し、頭頂や目の領域に対する jaw ジョイントの影響をなくします。

この方法を使用すると、腹部に対する jaw ジョイントと lower_mouth ジョイントの影響もなくすることができます。そうすると、口が動くとき、腹部があまり影響を受けなくなります。

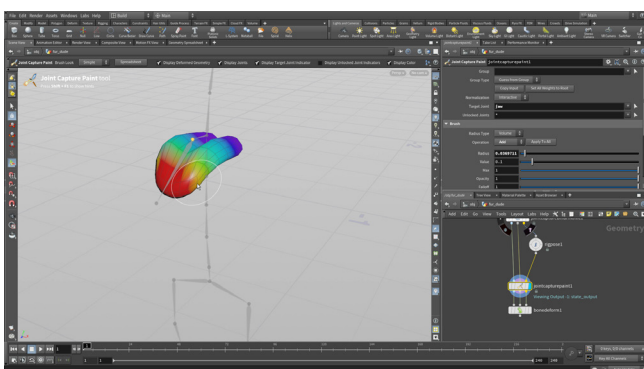


06 時々前に戻って、jaw をテストしながら、調整します。余分な領域に jaw が影響しなくなったら、完了です。再度スムーズをかけて、馴染ませてもよいでしょう。

次は、スケルトンの別のジョイントにウェイトをペイントします。Biharmonic (重調和)のおかげで、脚や足に問題はないはずですが、**rigpose** を使用してリグをテストし、改善点があればウェイトをペイントして微調整してください。



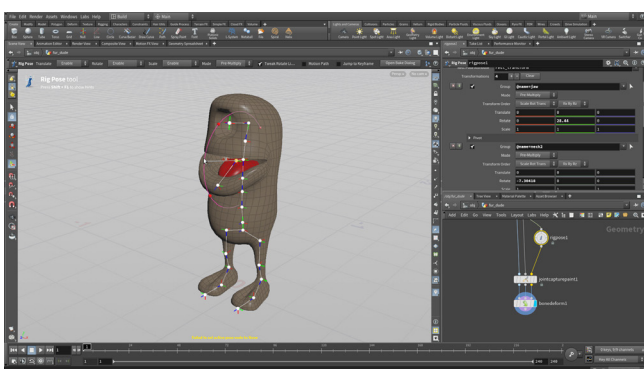
07 toe ジョイントを選択して、足の先端が左右の toe ジョイントにキャプチャされていることを確認します。後ほど、リバースフットのセットアップを構築して、ジオメトリをつま先にアタッチします。



08 visibility ノードで **Apply to Non-Selected Primitives** に設定します。すると、舌だけが表示されます。

デフォルトで適切にウェイトが付いているはずですが、必要に応じてウェイトをペイントできます。*jointcapturepaint* ノードを選択して、**neck1** ジョイントと **jaw** ジョイントのウェイトをペイントします。

完了したら、**visibility** ノードを再度**バイパス**します。



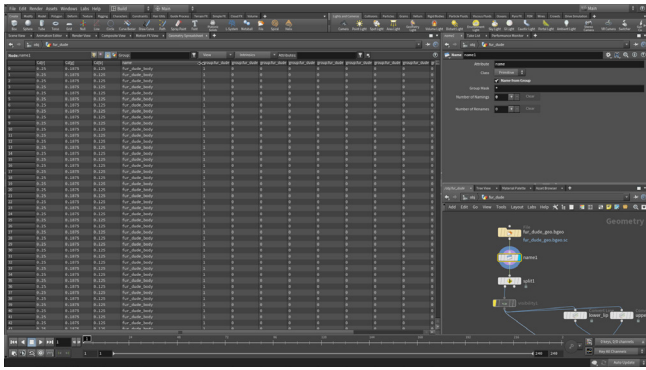
09 visibility ノードに **Bypass フラグ**を設定します。

完了したら、**rigpose** ノードをクリックします。**Handle** ツールをアクティブにして、Scene View で **jaw** ジョイントをクリックします。それを下に回転させて唇を下げます。すべてのパーツの動きに満足したら、リジッドジオメトリをキャプチャする準備の完了です。

パート7

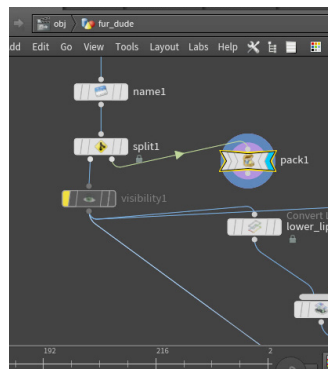
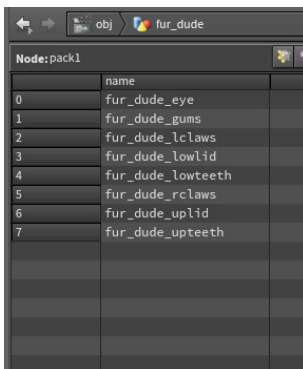
リジッドジオメトリのキャプチャ

パート5で、目、歯、つめのジオメトリを分割しました。ここでは、このジオメトリをパック化し、その後 Capture Packed Geometry ノードを使用して、各パーツをジョイントに割り当てます。KineFX では、親子化はジオメトリレベルでは使用できません。この操作は、各オブジェクトをスケルトンの子にすることに相当します。



01 ネットワークビューで、**Tab > Name from Groups** を選択して、ノードを **File** ノードと **split** ノードの間に配置します。**Group Mask** を * に変更します。これで、すべてのグループが name アトリビュートになります。

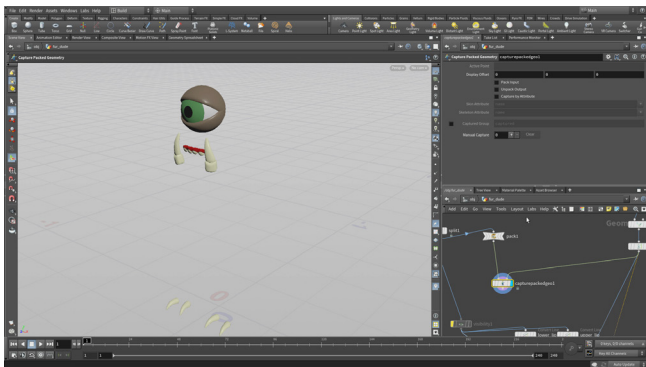
これを表示するには、**Geometry Spreadsheet** タブをクリックして、左上の **Primitives** ボタンをクリックします。下にスクロールすると、name アトリビュートがあること、グループ名がすべてのジオメトリの値として使用されていることを確認できます。



02 ネットワークビューで **Tab > Pack** を選択し、**split** ノードの右側にノードを配置します。**split** ノードの2番目の出力を **pack** ノードに接続して、**Display フラグ**を設定します。すると、Scene View に目、歯、つめのパーツが表示されます。

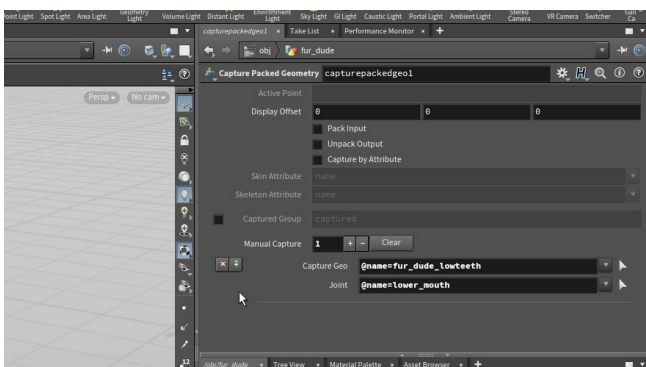
Path Attribute チェックボックスを **オフ** にして、**Name Attribute** チェックボックスを **オン** にします。**Transfer Attributes** を **name** に設定します。

ネットワークビューの右上の矢印を使用して、**Split** を選択します。**Geometry Spreadsheet** に、8つのパックプリミティブが表示されます。これらは、リジッドジオメトリとしてスケルトンにキャプチャできます。



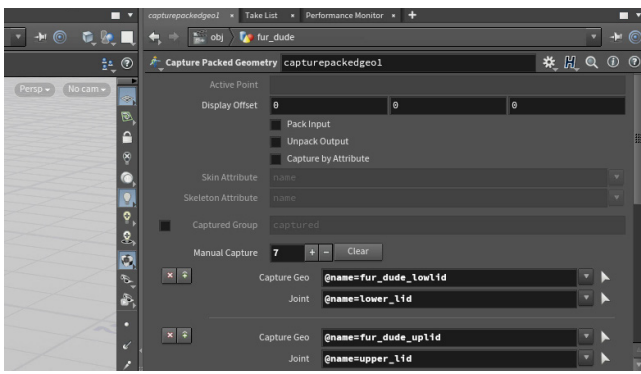
03 ネットワークビューで **Tab > Capture Packed Geometry** を選択して、そのノードを **pack** ノードの下に配置します。pack ノードの出力を、**capturepackedgeo** ノードの1つ目の入力に接続します。次に、skeletonmirror ノードの出力を、**capturepackedgeo** ノードの2つ目の入力に接続して、**capturepackedgeo** ノードに Display フラグを設定します。

これだけではまだ何も起こりません。ボーンとキャプチャしたいジオメトリを関連付ける必要があります。



04 パラメータエディタで、**Manual Capture** の横にある **+** (プラス) 記号をクリックします。**Capture Geo** の横の矢印をクリックして、Scene View で **下の歯** を選択します。ワンクリックで済むのは、下の歯はパックグループの1つだからです。**Enter** を押して確定します。

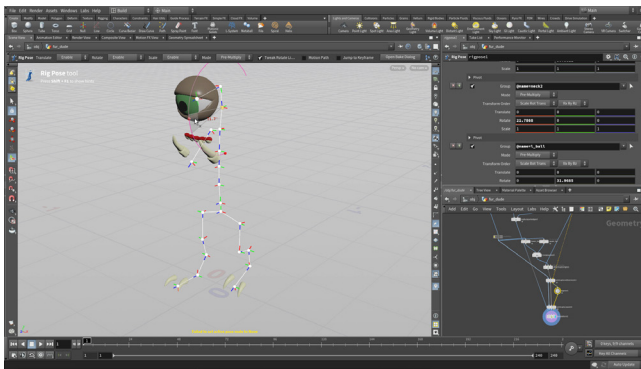
次に、Joint の横の矢印をクリックします。ジオメトリが消え、スケルトンを表すラインとポイントが表示されます。**lower_mouth** ジョイントを選択したら、カーソルを Scene View 上に置いたまま **Enter** を押して確定します。



05 + (プラス)記号をクリックして、**Capture Geo** の横の矢印をクリックします。Scene View で、Shift を押しながら **fur_dude_upteeth** と **fur_dude_gums** を選択します。**Enter** を押して確定します。次に、**Joint** の横の矢印をクリックします。**neck2** ジョイントを選択して、**Enter** を押します。

この手順をさらに 2 回繰り返して、以下を関連付けます。

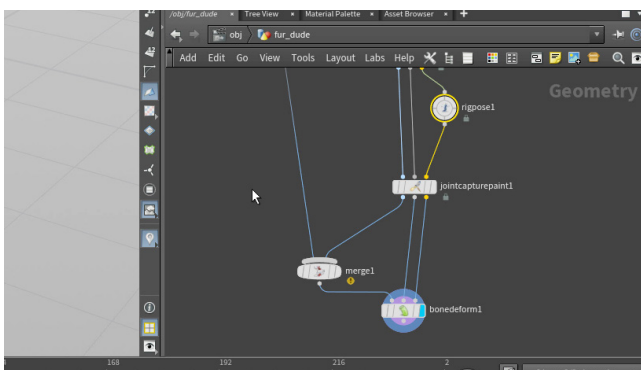
- **fur_dude_rclaws** と **r_toe**
- **fur_dude_lclaws** と **l_toe**
- **fur_dude_eye** と **eyeball**
- **fur_dude_uplid** と **upper_lid**
- **fur_dude_lowlid** と **lower_lid**



06 **capturepackedgeo** ノードの出力を、**bonedform** ノードの 1 つ目の入力に接続します。**bonedform** ノードに Display フラグを設定します。

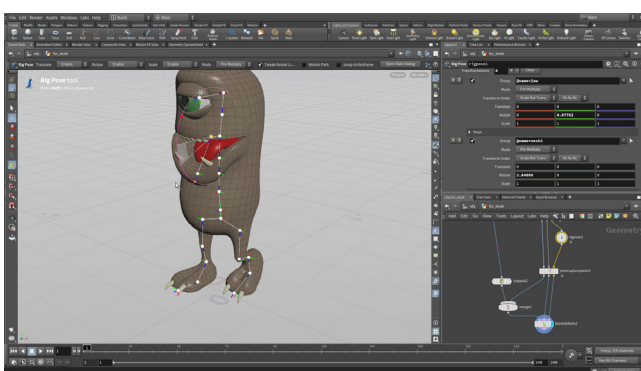
rigpose ノードを選択したら、Scene View でジョイントを選択して回転します。ジオメトリがジョイントにキャプチャされ、変形がないことを確認できます。

目とまぶたのジョイントは重なっているため、ジョイントをクリックしたら、小さいポップアップメニューを使用していずれかを選択します。必要なジョイントにたどり着くまで、数回クリックする必要がある場合もあります。選択したジョイントを使用して、目とまぶたを別々に回転します。



07 ネットワークビューで **Tab > Merge** を選択して、ノードを **capturepackedgeo** ノードと **bonedform** ノードの間に配置します。**capturelayerpaint** ノードを merge ノードに接続します。**merge** ノードで、**capturelayerpaint** の横の青い上向き矢印を押して、入力の順番を変更します。

これですべてが変形するようになりました。しかし、パックジオメトリには色が付いていて、体と舌はグレーです。merge ノードにも問題があります。一方には Primitive アトリビュートとしてカラー (Cd) が設定されているのに対し、もう一方には Point アトリビュートとしてカラーが設定されています。



08 ネットワークビューで **Tab > Unpack** を選択し、**capturepackedgeo** ノードと merge ノードの間にノードを配置します。**Iterations** を **2**、**Transfer Attributes** を *** ^Cd** に設定します。***** ですべてのキャプチャアトリビュートが取り込まれます。また **^Cd** により、元のカラーアトリビュートが除去されません。これで、どちらもポイントカラーを使用するようになり、リグが正しく表示されるようになりました。

rigpose ノードを選択し、Scene View でジョイントを選択して回転します。すべてのキャプチャされたジオメトリと一緒に動くのを確認できます。

作業内容を保存します。



ネットワークの統合

ジオメトリのキャプチャ、ウェイトのペイント、アニメーション用のジオメトリの準備のために作成したネットワークは適切に動作しますが、変更を加えると更新に時間がかかります。

より効率的なリグを作成するには、ネットワークを統合し、キャプチャウェイトをジオメトリに格納した単一ファイルにします。このファイルを Bone Deform SOP に接続すると、最初にジオメトリをキャプチャするのに使用したボーンと同じボーンのスケルトンであれば、効率的に変形されるようになります。



furdude_capt

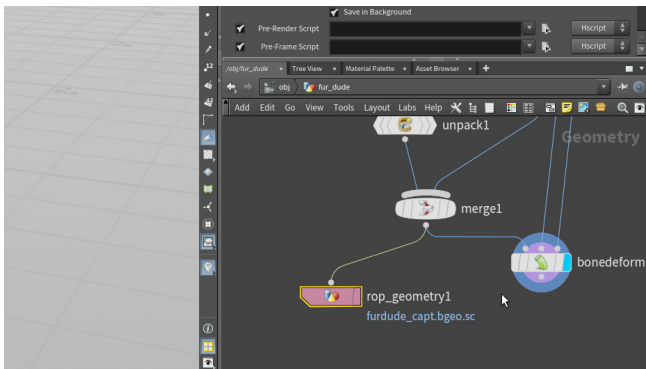


furdude_skel

パート 8

キャプチャリグのデジタルアセットの作成

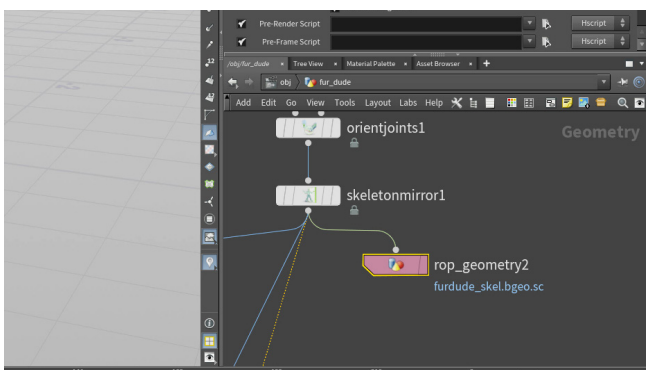
キャプチャしたジオメトリとスケルトンを、デジタルアセットにラップして、アニメーションリグの土台として使用します。まず、キャプチャウェイトやスケルトンとともにジオメトリをエクスポートして、デジタルアセットファイルに埋め込みます。こうすると、キャラクターのアニメートで効率の良いリグが得られます。



01 ネットワークビューで **Tab > ROP Geometry Output** を選択し、**merge** ノードの下にノードを配置します。**merge** ノードの出力を **rop_geometry** ノードの入力に接続します。パラメータエディタで、**Output File** を次のように設定します。

\$HIP/geo/furdude_capt.bgeo.sc

Save to Disk をクリックして、ジオメトリを **geo** ディレクトリに保存します。このジオメトリは、レッスンの最初にインポートしたジオメトリと同じように見えますが、変形を可能にするキャプチャトリビュートなどの重要な情報が含まれています。



02 ネットワークビューで **Tab > ROP Geometry Output** を選択し、**skeletonmirror** ノードの横にノードを配置します。**skeletonmirror** ノードの出力を **rop_geometry** ノードの入力に接続します。パラメータエディタで、**Output File** を次のように設定します。

\$HIP/geo/furdude_skel.bgeo.sc

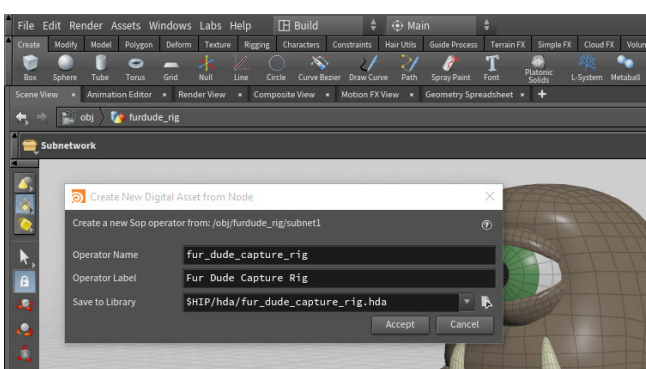
Save to Disk をクリックして、ジオメトリを **geo** ディレクトリに保存します。このジオメトリはスケルトンを表し、アニメーションリグアセットの構築に使用できます。



03 オブジェクトレベルに戻ります。オブジェクトの名前を **fur_dude_capture** に変更します。その **Display フラグ** をオフにします。

Create シェルフの **File** ボタンをクリックします。**\$HIP** をクリックして、**geo** ディレクトリに移動し、**furdude_capt.bgeo.sc** ファイルを選択します。**Enter** を押して原点に配置したら、新しいオブジェクトノードの名前を **furdude_rig** に変更します。**ダブルクリック** してこのオブジェクトの中に入ります。**File** ノードを **Alt ドラッグ** して、2つ目のノードを作成します。**Geometry File** を **\$HIP/geo/furdude_skel.bgeo.sc** に変更します。

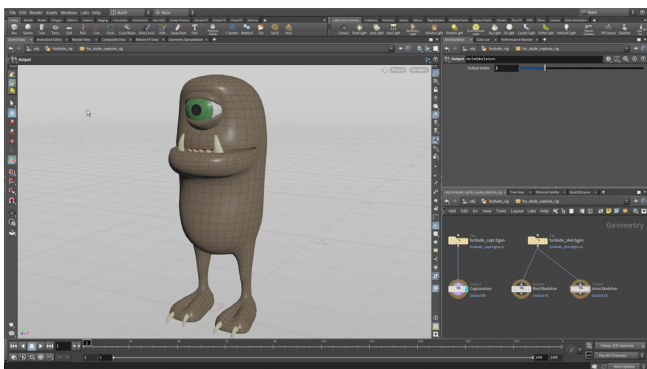
それを **furdude_skel.bgeo** という名前に変更し、**furdude_capt.bgeo** ノードに **Display フラグ** を設定します。



04 2つの **File** ノードを選択し、**Assets** メニューから **New Digital Asset from Selection** を選択します。次のように設定します。

- **Operator Name** を **fur_dude_capture_rig** にする
- **Operator Label** を **Fur Dude Capture Rig** にする
- **Save to Library** を **\$HIP/hda/fur_dude.hda** にする

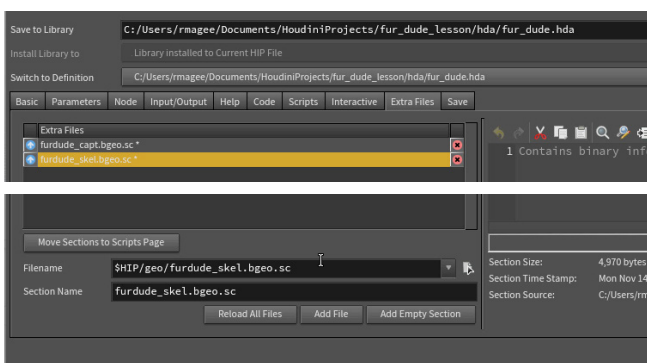
Accept をクリックして、**HDA** ファイルを作成します。Edit Operator Type Properties ウィンドウが表示されます。**Maximum Outputs** を **3** に設定し、**Accept** をクリックします。サブネットを **fur_dude_capture_rig** という名前に変更します。



05 このノードをダブルクリックして、サブネットの中に入ります。**Tab > Output** を選択して、output ノードを **fur_dude_geo_capt File** ノードの下に配置します。**File** ノードを **output** ノードに接続します。名前を **CaptureGeo** に変更します。

Alt ドラッグ操作を 2 回行い、新しい Output ノードを 2 つ作成します。2 つ目の名前を **RestSkeleton** に変更し、**Output Index** を **1** に設定します。3 つ目の名前を **AnimSkeleton** にして、**Output Index** を **2** に設定します。**fur_dude_skel.bgeo File** ノードを、2 つ目および 3 つ目の **output** ノードに接続します。

CaptureGeo output ノードに **Display フラグ** を設定します。

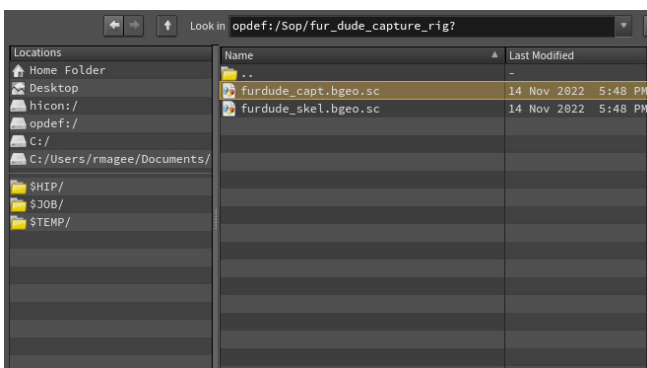


06 Asset メニューから、**Edit Asset Properties > Fur Dude Capture Rig** を選択します。Edit Operator Type Properties ウィンドウが開きます。**Extra Files** タブをクリックします。

左下の **Filename** の横にある選択ボタンをクリックして、**\$HIP/geo/fur_dude_capt.bgeo.sc** に移動します。**Accept** をクリックします。**Add File** をクリックします。

この手順を **fur_dude_skel.bgeo.sc** ファイルでも繰り返します。これで、これらのファイルがデジタルアセットファイル内部に配置できました。完全なパッケージとして他の人と共有しやすくなります。

Accept をクリックして完了します。

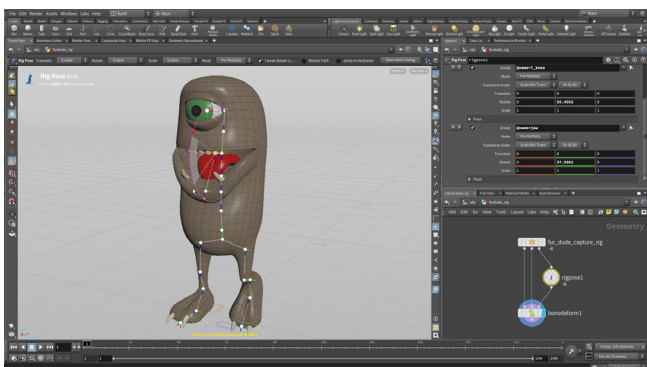


07 最初の file ノードで、**Geometry File** の横にある**選択**アイコンをクリックします。**Locations** サイドバーで **opdef:/** を選択し、**Sop** ディレクトリをダブルクリックしたら、**fur_dude_capture_rig** フォルダをクリックします。**fur_dude_capt.bgeo.sc** ファイルを選択し、**Accept** を押します。これにより、以下の opdef エクスプレッションが作成されます。

`opdef:/Sop/fur_dude_capture_rig?fur_dude_capt.bgeo.sc`

この手順を **fur_dude_skel.bgeo.sc File** ノードでも繰り返します。

Assets メニューから **Lock Asset > Fur Dude Capture Rig** を選択します。**Save Changes** をクリックして、このアセットのコンテンツを保護します。これらのファイルは、必要に応じて後からロックを解除して、更新できます。



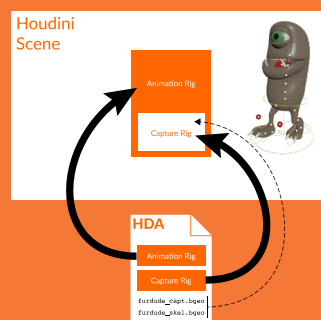
08 1 つ上のレベルに移動すると、3 つの出力があるキャプチャリグを確認できます。**Tab > Bone Deform** を選択し、このノードを下に配置します。**fur_dude_capt_rig** の 3 つの出力を、**bonedform** ノードの 3 つの入力に接続します。3 つ目のチェーンの中央に、**rigpose** ノードを追加します。**bonedform** ノードに **Display フラグ** を設定します。

rigpose ノードを選択して、**Handle** ツールがアクティブなことを確認します。すべてのスケルトンジョイントが再度表示されます。スケルトンに**ポーズ**を付け、変形が前と同じように動作しているかをテストします。

HDA

このようなノードのネットワークは、Houdini デジタルアセット (HDA) として保存されます。これはディスク上の 1 ファイルで、簡単に共有できます。キャプチャリグはディスクから参照されるアセットです。次のセクションで、内部にキャプチャリグをネストしたアニメーションリグを構築します。どちらのファイルも、bgeo ファイルとともに単一の HDA ファイル内に保存されます。

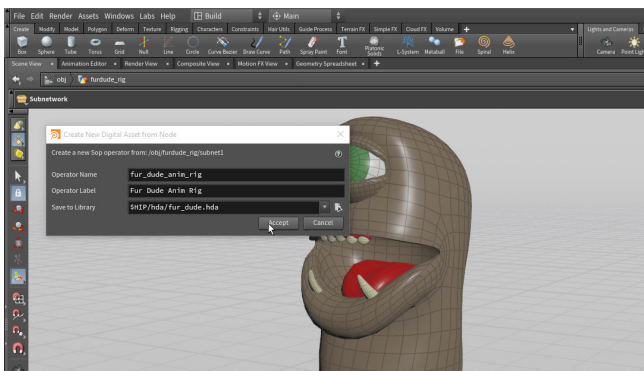
ディスク上の HDA ファイルが変更されると、アニメータがシーンファイル内で作業しているアセットのインスタンスが更新されます。



パート9

アニメーションリグアセットの作成

このパートでは、内部にキャプチャリグをネストした2つ目のデジタルアセットを作成します。この新しいアセットをアニメートして、キャラクターの最終的なモーションを作成します。また、この新しいアセットに、アニメーションをサポートするインバースキネマティクスや Aim 拘束などのリギングツールを追加していきます。コントロールを追加するたびにテストしていきます。このためには、ロックされたテストリグをセットアップし、2つ目の Scene View ペインに表示させます。

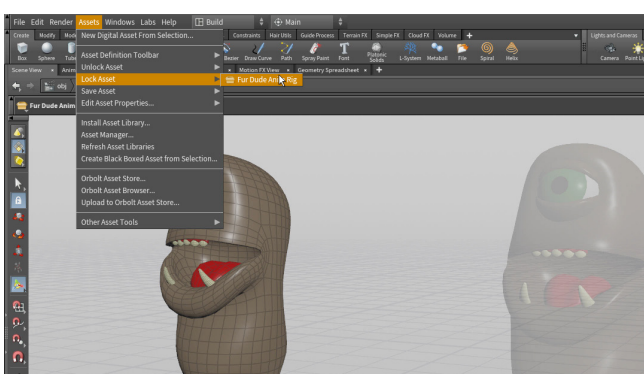


01 3つのノードを選択し、**Assets** メニューから **New Digital Asset from Selection** を選択します。次のように設定します。

- **Operator Name** を **fur_dude_anim_rig** にする
- **Operator Label** を **Fur Dude Anim Rig** にする

Save to Library では、ブラウザボタンをクリックします。\$HIP をクリックし、HDA ディレクトリをダブルクリックします。fur_dude.hda ファイルを選択し、**Accept** をクリックします。これで、**\$HIP/hda/fur_dude.hda** に設定されました。

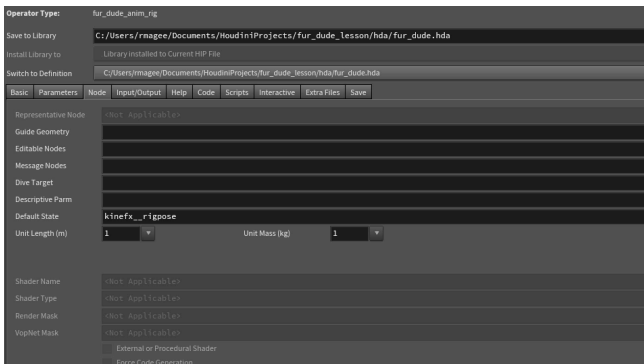
Accept をクリックし、Edit Operator Type Properties ウィンドウで再度 **Accept** をクリックします。これで、新しいアセット定義が同じ HDA ファイルに追加されました。サブネットを **fur_dude_anim_rig** という名前に変更します。



02 オブジェクトレベルに移動します。**fur_dude_rig** ノードを **Alt ドラッグ** して新しいジオメトリノードを作成し、**test_rig** と名前を付けます。T を押して、テストリグを左に動かします。

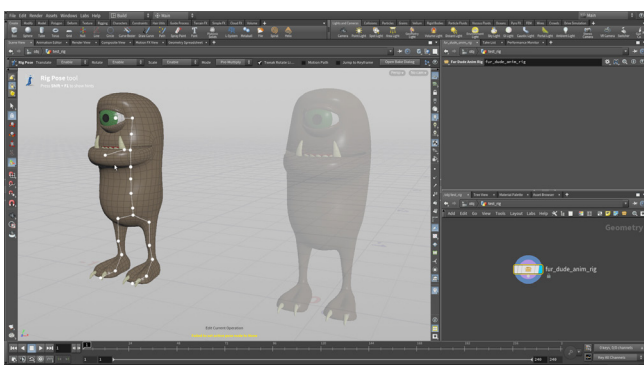
ダブルクリックして、このノードに入ります。**fur_dude_anim_rig** ノードを選択し、**Assets** メニューから、**Lock Asset > Fur Dude Anim Rig** を選択します。

1つ上のレベルに戻ります。これで2つのバージョンのリグができました。**test_rig** はロックされ、完成したアセット内でどのように振舞うかを確認できます。現在は相互作用するものが何もありません。これは後ほど解決します。ネットワークエディタでクリックして、**Ctrl + 1** を押して Quick マークを設定します。



03 **fur_dude_rig** オブジェクトに移動したら、さらに **fur_dude_anim_rig** に戻ります。ネットワークエディタでクリックして、**Ctrl + 2** を押して Quick マークを設定します。これで、これらのネットワーク間を素早く行き来できるようになりました。

rigpose を選択し、**Transformations** の横にある **Clear** ボタンをクリックします。**Asset** メニューに移動し、**Save Asset > Fur Dude Anim Rig** を選択します。**Asset** メニューから、**Edit Asset Properties > Fur Dude Anim Rig** を選択します。**Node** タブをクリックして、**Default State** を **kinefx__rigpose** に設定します。**Accept** をクリックします。



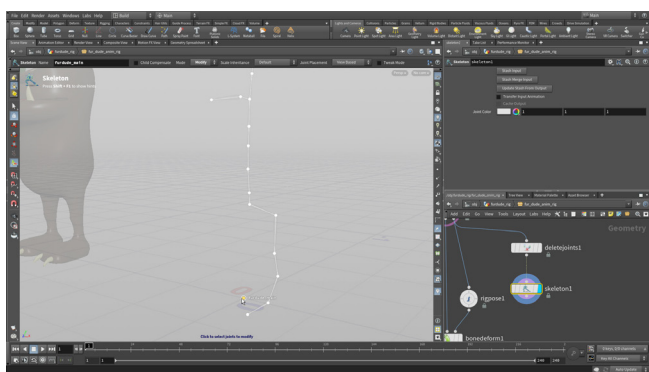
04 ネットワークビューで、**1** を押して **test_rig** に戻ります。左側の Scene View で、ツールバーを展開して **Handle** ツールをクリックします。

これで、**test_rig** にジョイントが表示されます。ジョイントをクリックしても変更は不可能です。これは、パラメータがアセットにプロモートされていないためです。ここから、このキャラクターにアニメーション可能なインターフェースを構築していきます。

パート 10

コントロールジョイントの追加

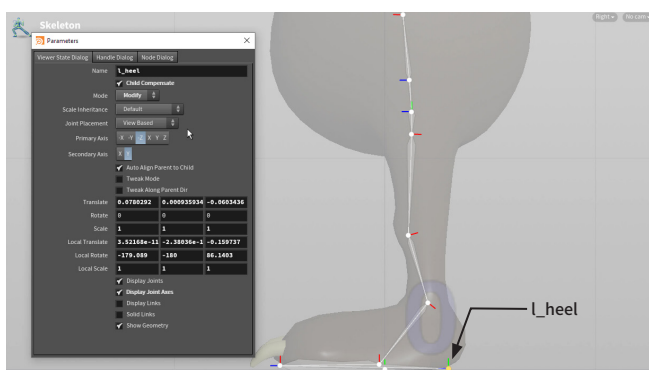
コントロールリグの柔軟性を高めるため、スケルトン全体のルートジョイント、リバースフットのセットアップ用にかかとジョイント、眼球のターゲットとなる Look At ポイントなど、複数のジョイントを追加します。これらのジョイントには、元の静止スケルトン内のジョイントと同じ名前を付けます。こうしておくと、キャラクタのモーションを駆動するために使用できます。



01 ネットワークビューで、**2** を押して `fur_dude_anim_rig` に戻り、**Display** フラグを設定します。`fur_dude_capture_rig` ノードに **Template** フラグを設定します。

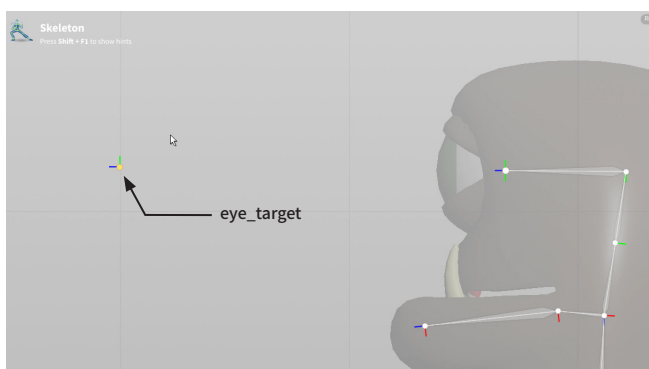
Delete Joints ノードを `fur_dude_capture_rig` ノードの 3 つ目の出力から分岐させます。Group の横にある矢印をクリックして、ファー・デュードの右脚のジョイントを選択します。**Enter** を押します。

Skeleton ノードを追加します。**Mode** を **Create** に変更します。**グリッドスナップ** をオンにして、原点にポイントを追加します。**Modify** モードに変更して、新しいジョイントの名前を `fur_dude_main` に変更します。



02 Right ビューに移動します。**Create** モードに戻り、**MMB** クリックしてメインジョイントの選択を解除します。**L_toe** ジョイントをクリックして、かかとがあるべき場所に新しいジョイントを追加します。

Modify モードに変更して、新しいジョイントの名前を `L_heel` に変更します。**P** を押して、**Child Compensate** をオンにし、**Rotate** を **0, 0, 0** に設定します。



03 目の領域に移動します。**Create** モードに戻し、**MMB** クリックして現在の選択を解除します。目の前に新しいジョイントが追加されました。

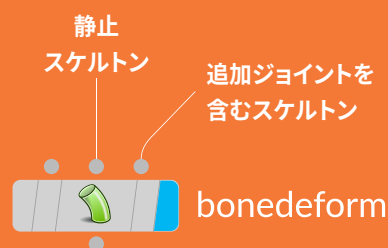
Modify モードに変更して、新しいジョイントの名前を `eye_target` に変更します。**P** を押して、**Rotate** を **0, 0, 0** に設定します。

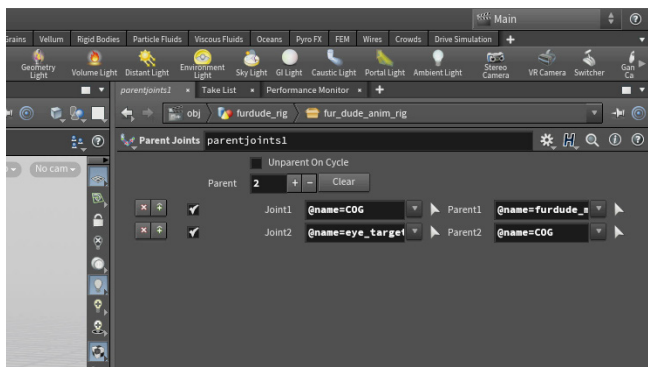


リグにジョイントを追加する

これらの追加ジョイントは右端のノードストリームに追加されました。一方、中央の静止スケルトンは元のジョイントのままです。追加のジョイントを `bonedform` ノードに接続しても、「ファントム」ジョイントは無視されます。元のジョイントだけが、リグの最終出力を決定します。

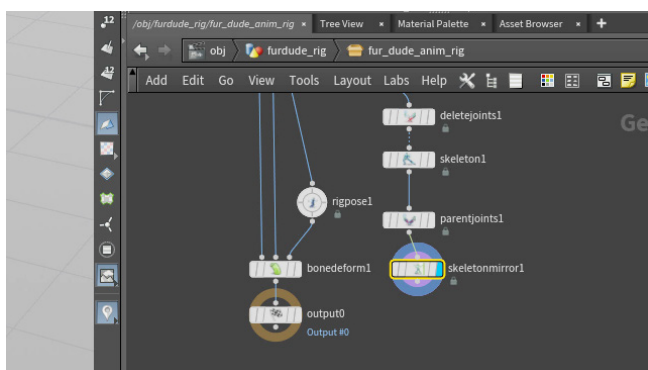
これらの追加ジョイントを `bonedform` ノードの中央の静止スケルトン入力に接続すると、エラーになります。これらのジョイントは、入力ジオメトリに、対応するキャプチャウェイトを持っていないからです。





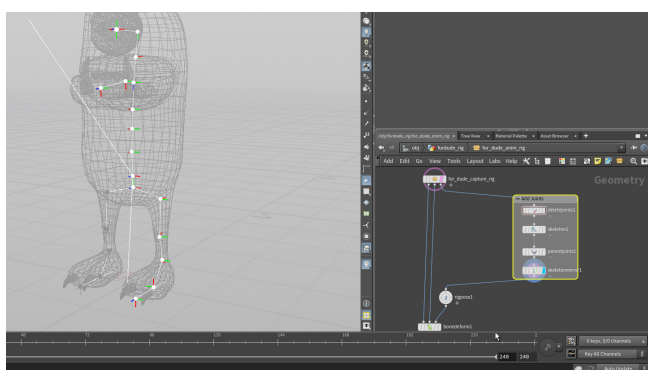
04 Parent Joints ノードを追加します。**+(プラス)**記号を2回クリックします。**Joint1**の横にある矢印をクリックして、Scene Viewで**COG** ジョイントをクリックします。(カーソルを Scene View に置いたまま) **Enter** を押して確定します。次に、**Parent1**の横にある矢印をクリックして、**fur_dude_main** ジョイントを選択します。

2つ目のエントリでは、**Joint2**の横にある矢印をクリックして、Scene Viewで**eye_target** ジョイントをクリックします。(カーソルを Scene View に置いたまま) **Enter** を押して確定します。次に、**Parent2**の横にある矢印をクリックして、**COG** ジョイントを選択します。



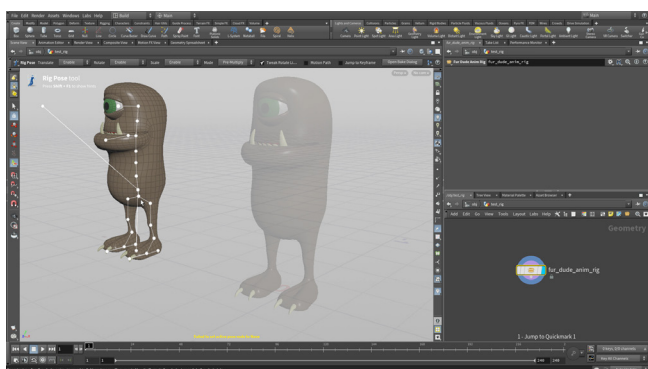
05 Skeleton Mirror ノードをチェーンに追加します。これにより、すべてのジョイントのミラーコピーが作成されます。パラメータエディタに移動して、**Group**の横の矢印をクリックします。左脚のジョイントのみを選択して(新しい**L_heel** ジョイントを含む)、**Enter** を押します。脚のみがミラー化されました。

Namingで、**Find Tokens**を**L_**、**Replace Tokens**を**r_**に設定します。これで、右脚のジョイントに適切な名前が付けられます。



06 skeletonmirror ノードを**rigpose** ノードに接続します。**bonedeform** ノードに**Display** フラグを設定します。足のつめが反転しているように見えます。そのスケルトンジョイントに戻り、**L_toe** を選択し、**P**を押してパラメータを表示します。**Rotate**を**0, 0, -90**に設定します。

ジョイントを追加するのに使用した4つのノードを選択し、**Add Network Box** ボタンをクリックします。ノードを囲むようにボックスを配置します。ボックスのタイトルをクリックして、**Add Joints** と入力します。



07 Assets メニューから、**Save Asset > Fur Dude Anim Rig** を選択します。変更がアセット定義に保存され、**test_rig** が更新されます。まだ**test_rig** を編集することはできません。いずれのパラメータもトップレベルにプロモートされていないからです。

次は、メインコントロールをセットアップし、パラメータをプロモートして、**test_rig** に生命を吹き込んでいきます。



テストリグの役割

現在作業しているリグは、ロックされていないアセットです。アセット内のすべてのジョイントを操作できます。ただし、アセットがアニメータに渡されたときにアニメータが扱えるのは、キャラクターのトップレベルにプロモートされたパラメータのみです。

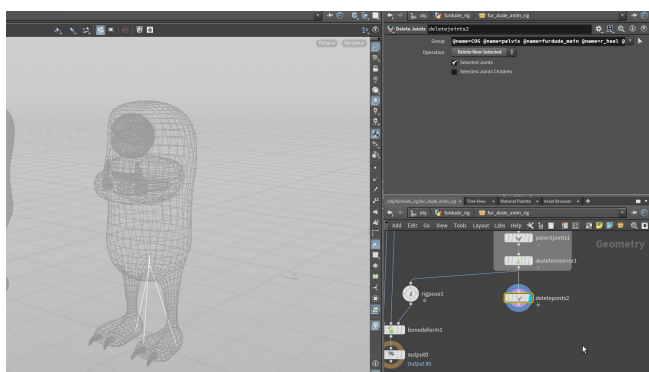
test_rig はアセットの2つ目のロックバージョンであり、パラメータをプロモートしたりコントロールを構築するまで、アセットを操作することはできません。テストリグが素晴らしいのは、アニメーションを付けられる状態になっているかどうかを確認できるツールだからです。テストリグを操作できれば、アニメータはキャラクターにポーズを付けることができません。



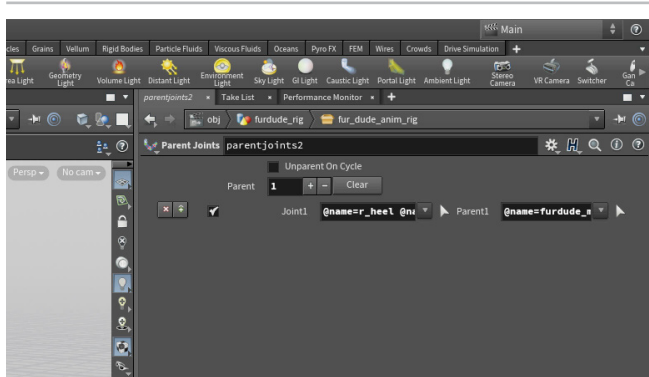
パート 11

メインコントロール

キネマティクスを追加するためには、COGの下に足がある、現在の階層を切り離す必要があります。いくつかのジョイントを切り離し、再度親子化して、希望通りの階層を構築します。Bone Deform を適切に動作させるには、この再親子化を別に行ってから、結果を元のスケルトン階層にブレンドすることが重要です。



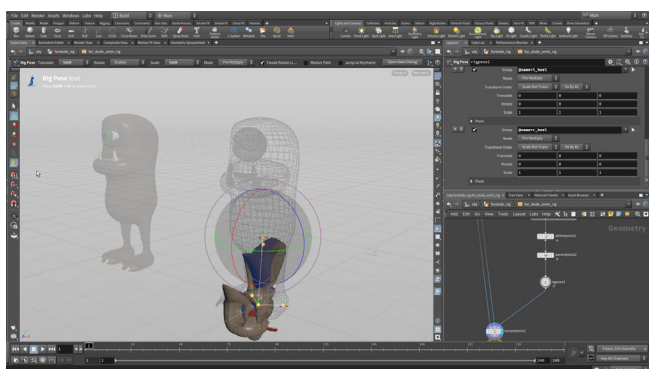
01 Delete joints ノードを *skeletonmirror* から分岐させます。Group の横にある矢印をクリックして、Scene View で *furdude_main*、*COG*、*pelvis*、*l_heel*、*r_heel* ジョイントを選択します。Enter を押し、Operation を Delete Non-Selected に設定します。



02 deletejoints ノードの後に Parent Joints ノードを追加します。+(プラス)記号をクリックして、ジョイントリストを追加します。Joint1 の横にある矢印をクリックして、Scene View で 2 つの heel ジョイントを選択します。Enter を押します。

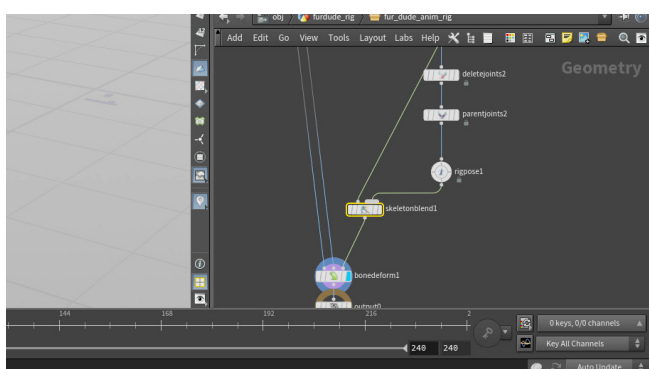
Parent1 の横にある矢印をクリックして、Scene View で *furdude_main* ジョイントを選択します。Enter を押します。

heel ジョイントが *COG* から切り離され、脚の下部で IK を駆動するのに使用できるようになりました。*COG* ジョイントに親子化されている *pelvis* ジョイント以外、すべてのジョイントは *furdude_main* ジョイントに親子化されています。



03 parentjoints ノードを rigpose ノードに接続します。このノードからの既存のジョイントをすべてクリアして、Display フラグを設定します。*furdude_main* ジョイント、*COG* ジョイント、*pelvis* ジョイント、2 つの heel ジョイントをクリックします。

ここで *bonedform* ノードに Display フラグを設定すると、ジョイントの多くがなくなっているため、想像とは違う表示になります。これは、Skeleton Blend を使用して修正できます。



04 ネットワークエディタに Skeleton Blend ノードを追加します。*skeletonmirror* ノードを左側の入力に、*rigpose* ノードを右側の入力に接続します。その後、*skeletonblend* ノードを *bonedform* ノードの 3 つ目の入力に接続します。

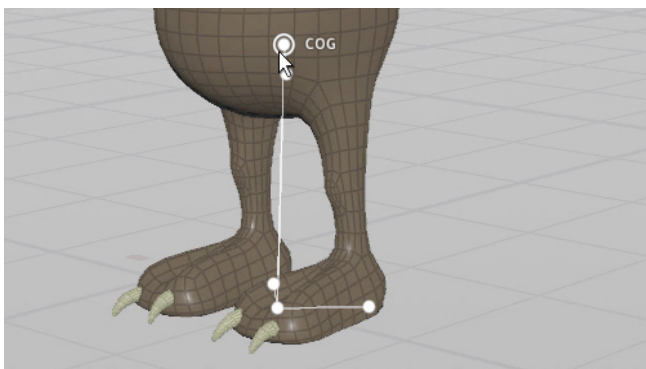
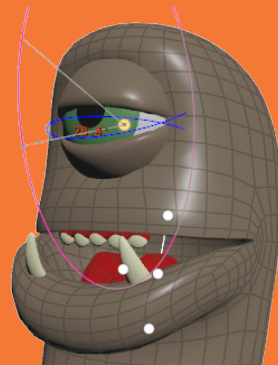
skeletonblend のパラメータエディタで、World Space チェックボックスをオンにして、weight1 を 1 にします。



コントロールジオメトリ

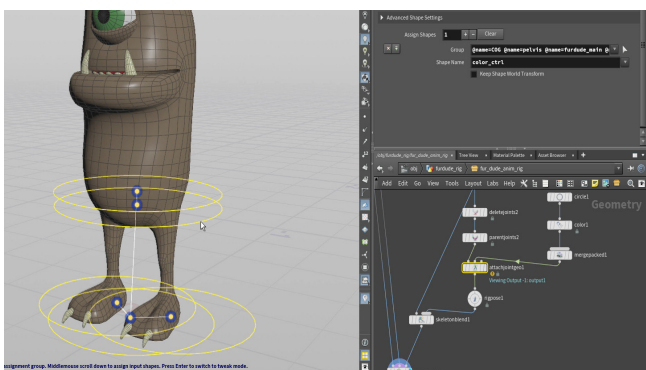
ジョイントを rigpose に追加するとき、アセットのトップレベルにプロモートすると、ジョイントを選択してアニメートできるようになります。Attach Control Geometry ノードを使って、ジオメトリをリグのさまざまなパーツに割り当てれば、ジョイントの選択や操作が簡単になります。好きな形状を作成して、コントロールに使用できます。

コントロールジオメトリの用途としては、眼球や、重なり合った2つの lid (まぶた) ジョイントなどが挙げられます。コントロールジオメトリを使用すれば、それらのジョイントを簡単に選択して、その部分のリグをセットアップできます。ここでは、メインコントロールに使用します。



05 Assets メニューから **Save Asset > Fur Dude Anim Rig** を選択します。これにより現在の設定が保存されます。ネットワークビューで、**1** を押して **test_rig** に戻ると、rigpose にリストされている5つのジョイントのみが表示されます。

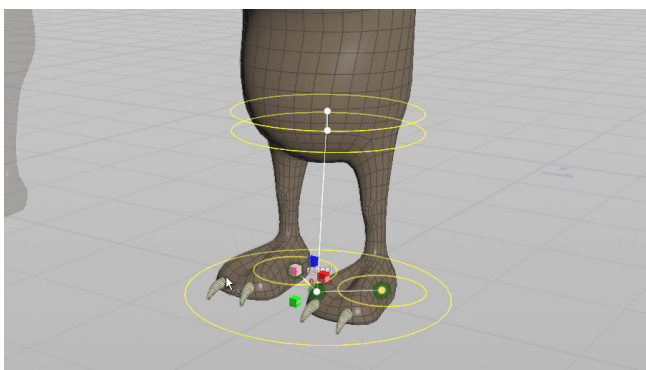
パラメータはまだプロモートされていないため、ジョイントの選択や移動はできません。



06 ネットワークビューで、**2** を押して **fur_dude_anim_rig** に戻ります。Circle ノードをネットワークに追加します。Orientation を ZX に設定します。Uniform Scale を 0.2 に設定します。Divisions を 36、Arc Type を Open Arc に設定します。Color ノードを追加して、Color を黄色に設定します。

さらに Merge Packed ノードを追加して、Name 1 を circle_ctrl に設定します。

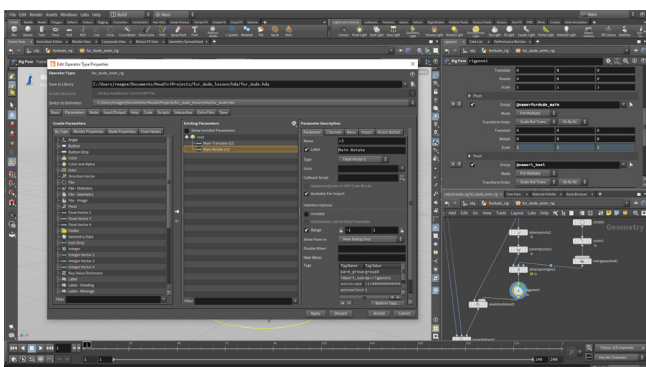
parent ノードと rigpose ノードの間に Attach Joint Geometry ジオメトリを配置します。その2つの入力に mergepacked ノードを接続します。このノードに Display フラグを設定し、Scene View で Enter を押します。表示されているジョイントをすべて選択したら、G を押し、スクロールホイールを使用して circle_ctrl ジオメトリを見つけます。ジオメトリがすべてのジョイントに割り当てられます。



07 上部のオペレーションコントロールツールバーで、Mode を Tweak Shapes に変更します。

COG と pelvis ジョイントを選択し、G を押してトランスフォームハンドルを表示します。E を押してスケールハンドルにしたら、中央のハンドルをクリック & ドラッグして、これらのコントロールが少し小さくなるまで3方向すべてでスケールします (パラメータエディタで約 0.67)。

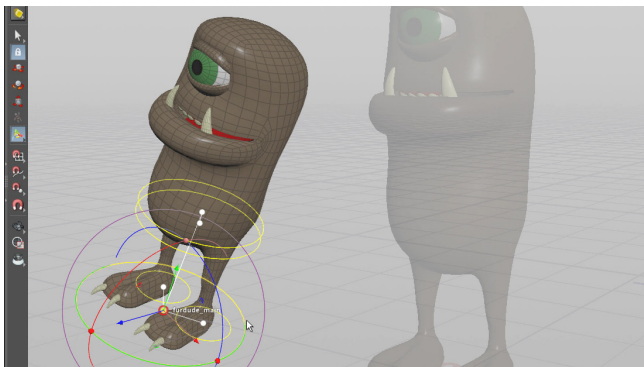
2つの heel ジョイントを選択し、G を押してトランスフォームハンドルを表示します。E を押してスケールハンドルにしてから、中央のハンドルをクリック & ドラッグして、かかとのコントロールがかなり小さくなるまで3方向すべてでスケールします (パラメータエディタで約 0.3)。



08 rigpose ノードを選択し、コントロールジオメトリを使用して4つのジョイントを選択します。これをテストリグで動作させるには、パラメータをプロモートする必要があります。

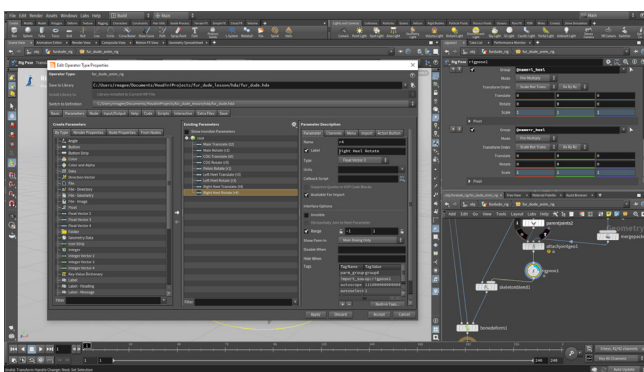
Assets メニューから、Edit Asset Properties > Fur Dude Anim Rig を選択します。Parameters タブをクリックします。

rigpose ノードで、furdude_main に移動し、Scale で RMB クリックして Lock Parameter を選択します。Translate をドラッグして、root の下のパラメータリストに移動します。Label を Main Translate に設定します。Rotate パラメータも同じようにして、Main Rotate と名前を付けます。Accept をクリックして終了し、結果をアセットに保存します。



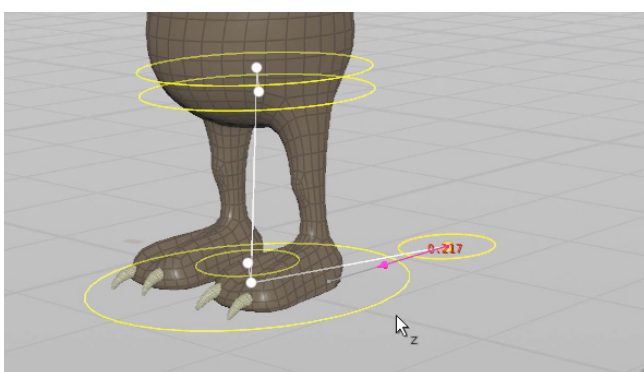
09 ネットワークビューで、**1**を押して **test_rig**に戻ります。テストリグが更新され、新しいコントロールが表示されています (**Handle** ツールがアクティブになっている場合)。コントロールジオメトリを使用して **furdude_main** ジョイントを選択すると、トランスフォームハンドルが表示され、それを使ってリグを動かすことができます。

動かした後は**取り消し**て、元の位置に戻します。



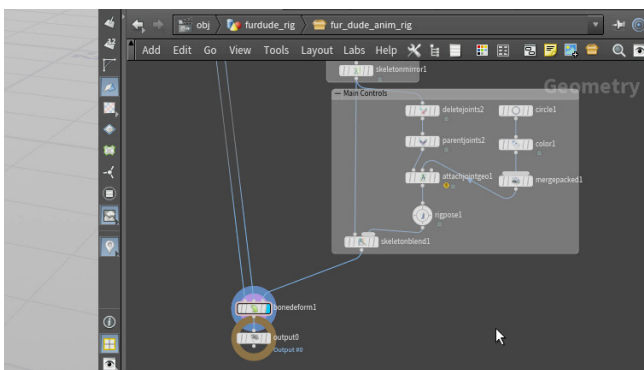
10 ネットワークビューで、**2**を押して **fur_dude_anim_rig**に戻ります。**Assets** メニューから、**Edit Asset Properties > Fur Dude Anim Rig** を選択します。**COG**、**l_heel**、**r_heel** ジョイントの **Translate** と **Rotate**、**pelvis** ジョイントの **Rotate** をドラッグします。いずれの場合も、各ジョイントの使用しない **Scale** や **Translate** パラメータ (**pelvis** の場合) はロックします。

Accept をクリックして終了し、結果をアセットに**保存**します。



11 ネットワークビューで、**1**を押して **test_rig**に戻ると、新しいコントロールが表示されるようになりました。コントロールジオメトリを使って **COG** ジョイントと **heel** ジョイントを選択し、パーツをトランスフォームさせてみましょう。

終了したら**取り消し**て、すべてのパーツを元の位置に戻します。



12 ネットワークビューで、**2**を押して **fur_dude_anim_rig**に戻ります。メインコントロールのセットアップに使用したノードを選択し、**Add Network Box** ボタンをクリックします。ノードを囲むようにボックスを配置します。ボックスのタイトルをクリックして、**Main Controls** と入力します。

Assets メニューから、**Save Asset > Fur Dude Anim Rig** を選択します。この操作はリグに影響しませんが、アセットを最新の状態に保つことができます。また、シーンファイルも**保存**しましょう。



ネットワークの整理

ひと手間かけて、ノードを整理させたり、**ネットワークボックス**を追加しておきましょう。よく整理されたネットワークは、後で作業しやすく、他のスタッフにも意図が伝わりやすいものです。

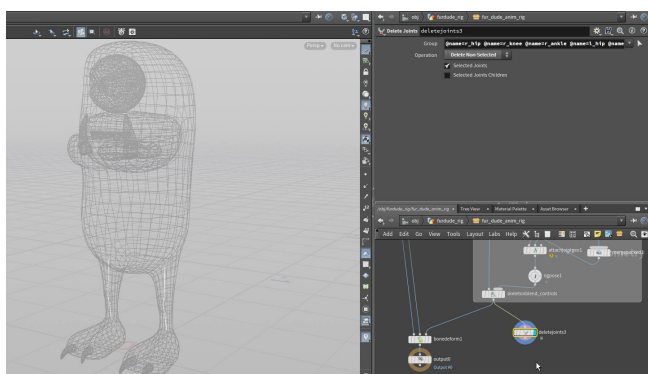
また、各ノードにはコメントを追加して、ネットワークに表示させることができます。**ステッキーノート**を使用すれば、ノードの大きいブロックについての説明も付加できます。チームでネットワークを作成する場合には、コミュニケーションが常に重要です。

This part of the network organizes the main controls such as the root, the COG and the heel joints.

パート 12

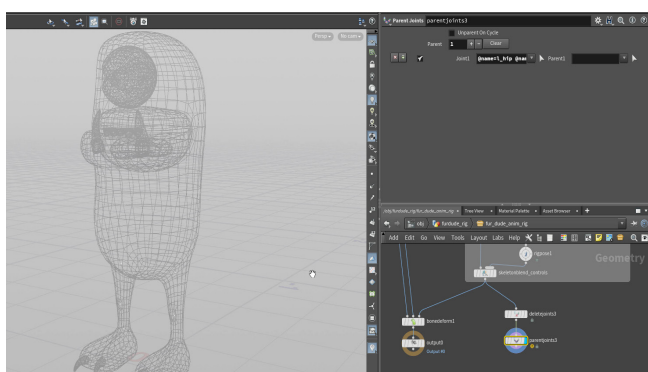
脚のインバースキネマティクス

キャラクターのアニメーションでは、脚にインバースキネマティクスを設定して、足あるいは腰の動きに合わせて膝が適切に曲がるようになります。このパートでも、メインのスケルトンのジョイントを利用し、KineFX を使ってセットアップしていきます。結果も、前のパートと同じようにオリジナルの階層にブレンドして戻します。

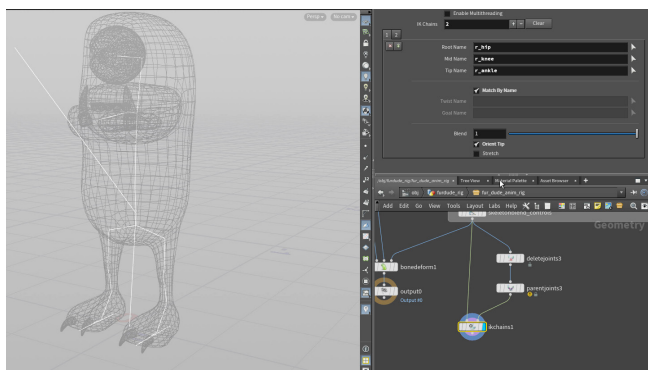


01 *skeletonblend* ノードの名前を *skeletonblend_controls* に変更します。これらのノードはよく使用するので、識別しやすくしましょう。

skeletonblend_controls ノードから **Delete Joints** ノードを分岐させ、**Display フラグ**を設定します。Group の横の矢印をクリックし、Scene View で左右の *hip*、*knee*、*ankle* の各ジョイントを選択します。**Enter** を押し、**Operation** を **Delete Non-Selected** に設定します。

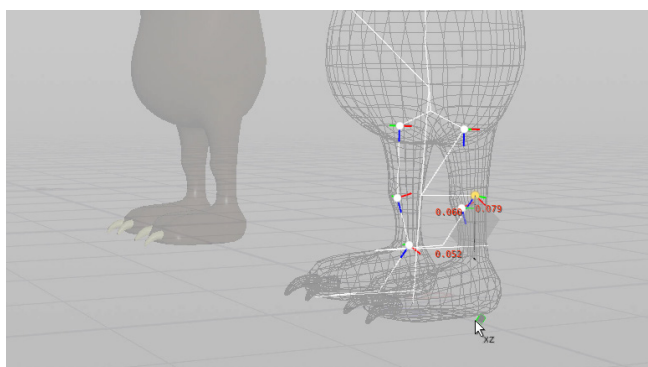


02 *Parent Joints* ノードを分岐させ、**Display フラグ**を設定します。**+(プラス)**記号をクリックしてジョイントを追加し、**Joint1** を * に設定します。**Parent1** は空のままにします。これで、すべてのジョイントの接続が解除されました。各ジョイントを個別に使用できます。



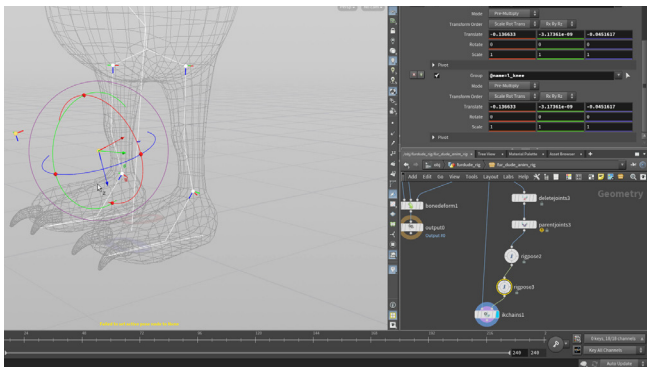
03 *IK Chains* ノードをネットワークエディタに配置します。その1つ目の入力に *skeletonblend_controls* を接続し、2つ目の入力に *parentjoints* ノードを接続します。**Display フラグ**を設定します。

パラメータエディタで **+(プラス)**記号をクリックし、**Root Name** の横の矢印をクリックします。**L hip** ジョイントを選択して、(Scene View にカーソルを置いたまま)Enter を押します。**Mid Name** を **L knee**、**Tip Name** を **L ankle** に設定します。矢印を使用してジョイントを選択してもよいですし、名前を入力してもかまいません。**Match by Name** を **オン**にし、**Blend** を **1** に設定したら、**Orient Tip** を **オン**にします。**+(プラス)**記号をもう一度クリックして、右脚についても同じ手順を繰り返します。



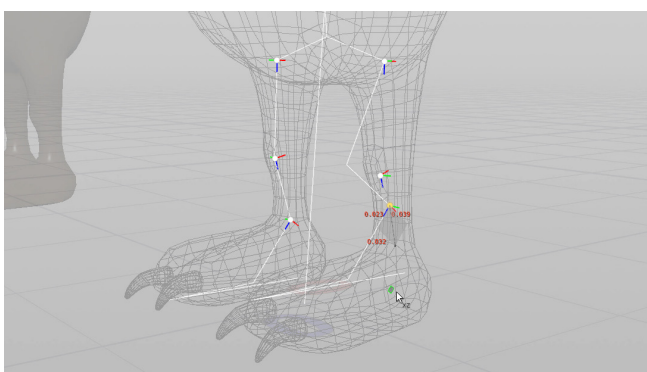
04 *Rig Pose* ノードを *parent* ノードと *ikchains* ノードの間に追加します。左右の *ankle* ジョイントをクリックして動かし、インバースキネマティクスが機能していることを確認します。**Match by Name** を **オン**にしたので、*ankle* ジョイントは、IK チェーンのエンドエフェクタとして動作しています。

足首を動かすと、時々、膝が逆方向に曲がります。これは、*knee* ジョイントがツイストエフェクタとして使用され、適切に配置されていないことが原因です。脚の前方に移動するようにならなければなりません。

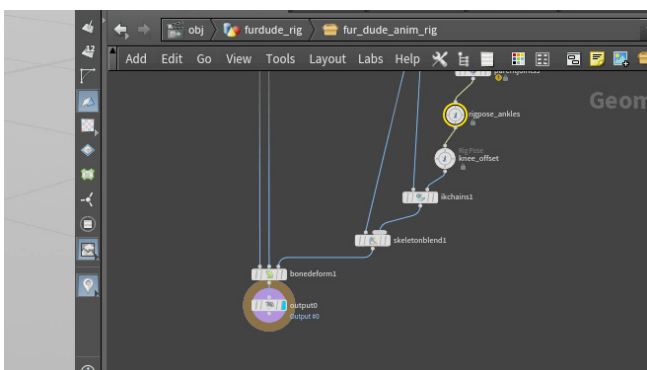


05 Rig Pose ノードを最初の *rigpose* ノードと *ikchains* ノードの間に追加します。Shift キーを押しながら両方の *knee* ジョイントを選択し、それらをキャラクターの前方に動かします。これで、IK での膝の向きは正常になるはずですが。

1 つ目の *rigpose* ノードの名前を *rigpose_ankles*、2 つ目の *rigpose* ノードの名前を *knee_offset* にします。

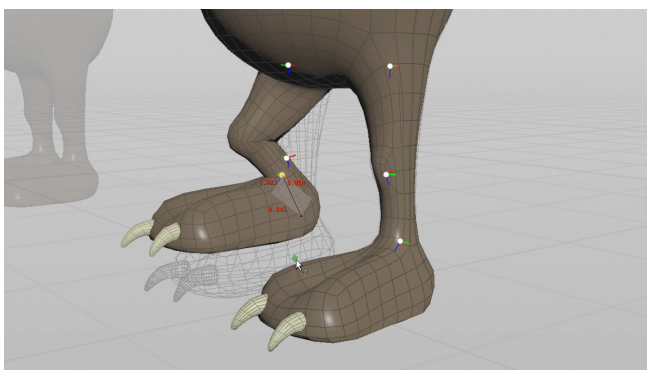


06 *rig_ankles* ノードに戻り、左右の ankle ジョイントをテストします。膝をオフセットしたので、逆方向に曲がることはありません。また、*knee_offset* ノードを選んで膝を動かし、IK チェーンのツイストエフェクタとして動作していることも確認します。



07 *Skeleton Blend* ノードを追加して、1 つ目の入力に *skeletonblend_controls* ノード、2 つ目の入力に *ikchains* ノードを接続します。この新しい *Skeleton Blend* ノードの出力を Bone Deform ノードの 3 つ目の入力に接続します。

このノードの名前を *skeletonblend_ik* に変更し、**World Space** チェックボックスを**オン**にして、**weight1** を **1** に設定します。



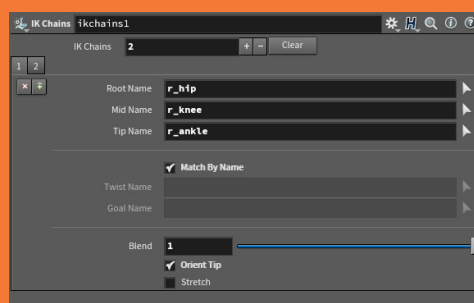
08 *rigpose_ankles* ノードに戻り、ankle ジョイントがキャプチャしたサーフェスに作用するかどうかをテストします。

この時点では、デジタルアセットを保存して変更をテストリグにプッシュする操作は行いません。最終的なリグでは、足首のコントロールに *rigpose_ankles* ノードは使用しないからです。以降の手順でリバースフットのセットアップを構築し、足全体のセットアップを機能させてから、脚全体のセットアップをアセットに保存します。



FK/IK のブレンド

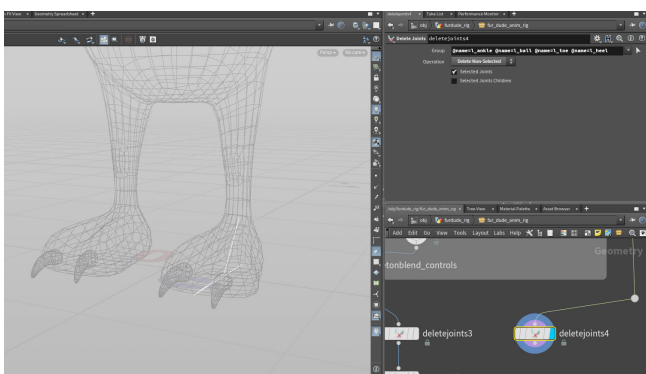
IK と FK をブレンドする方法はいくつかありますが、このレッスンでは使用しません。このキャラクターの脚には IK しか使わないからです。ブレンドするには、*rigpose* を使用しているアセットに脚のジョイントのパラメータをプロモートしてから、IK Chains ノードの **Blend** アトリビュートを使用するか、IK ソリューションと *rigpose* の回転ジョイントの間で *Skeleton Blend* を使用します。ファー・デュードに腕があれば、このセットアップをすることになるでしょう。



パート 13

リバーズフットのセットアップ

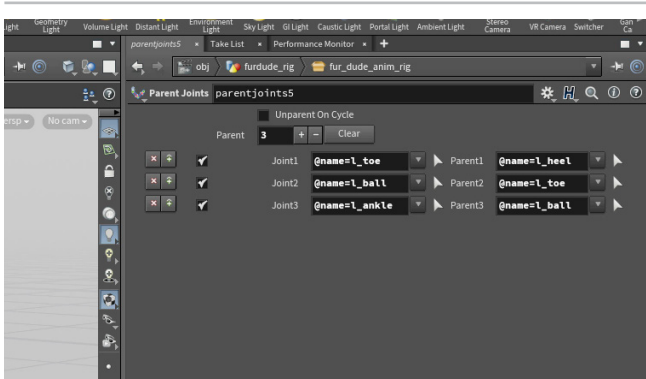
足をコントロールするために、伝統的なリバーズフットのセットアップを作成します。かかと(heel)がルートで、つま先(toe)、ボール(ball: 土踏まずから前)、足首(ankle)の親になります。KineFXなら簡単に作成でき、元のスケルトンにブレンドできます。ここでは右足を完全に再構築しますが、ジョイント名が揃っていないので、すべて正常に機能するはずですが、



01 *rigpose_ankles* ノードを削除します。足首は、リバーズフットのセットアップを使用してコントロールします。

skeletonmirror ノードから **Delete Joints** ノードを分岐させ、**Display フラグ**を設定します。Groupの横の矢印をクリックし、Scene Viewで **L_ankle**、**L_ball**、**L_toe**、**L_heel** ジョイントを選択します。**Enter** を押し、**Operation** を **Delete Non-Selected** に設定します。

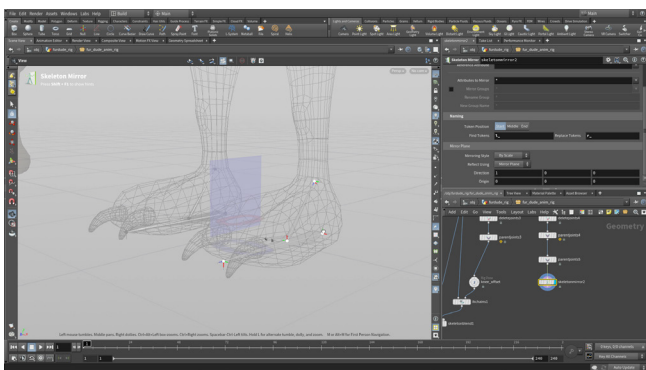
注：ワイヤーを迂回させて、ネットワークボックスの周りを通るようにしたいときは、ワイヤーを **Alt** クリックしてドットを追加します。



02 **Parent Joints** ノードを追加し、**Display フラグ**を設定します。**+**(プラス)記号をクリックしてジョイントを追加し、**Joint1** を * に設定します。**Parent1** は空のままにします。

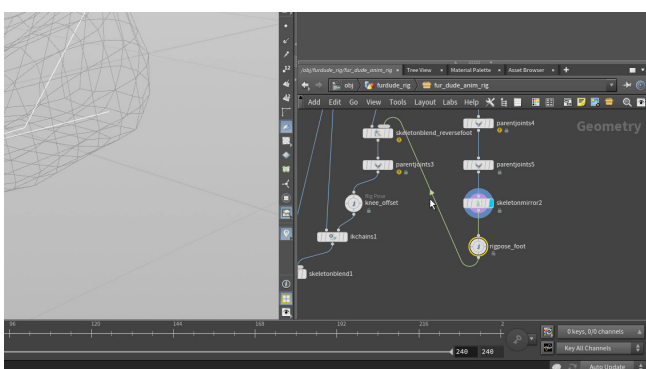
2つ目の **Parent Joints** ノードを分岐させ、**Display フラグ**を設定します。Scene Viewで **L_heel** ジョイント、**L_toe** ジョイント、**L_ball** ジョイント、**L_ankle** ジョイントをクリックします。MMB クリックして終了します。パラメータエディタでは、これらのジョイントが次の順番で表示されます。

- **Joint1** :@name=l_toe | **Parent1** :@name=l_heel
- **Joint2** :@name=l_ball | **Parent2** :@name=l_toe
- **Joint3** :@name=l_ankle | **Parent3** :@name=l_ball



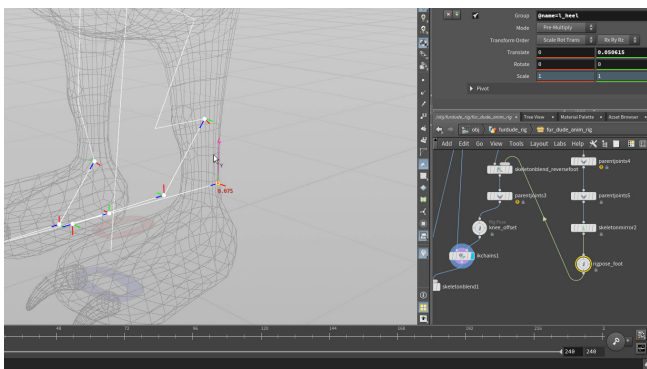
03 **Skeleton Mirror** ノードをチェーンに追加します。パラメータエディタの **Naming** で、**Find Tokens** を **L_**、**Replace Tokens** を **r_** に設定します。これで、右脚のリバーズフットを作成できます。

この階層内のすべてのジョイントは、オリジナルのスケルトンと同じ名前です。このためリグのポーズを設定する際にも、情報が正しく転送されます。



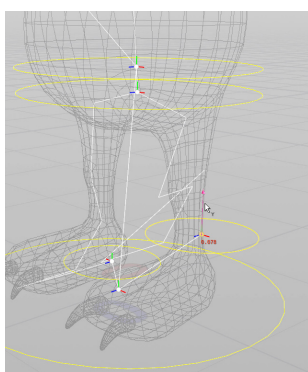
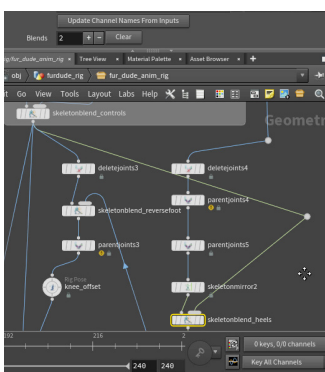
04 脚にセットアップした *deletejoints* と *parentjoints* の間に **Skeleton Blend** ノードを追加します。このノードの名前を **skeletonblend_reversefoot** に変更し、**World Space** チェックボックスを **オン** にして、**weight1** を **1** に設定します。**Group** の横の矢印をクリックし、左右の **ankle** ジョイントを選択します。**Enter** を押します。

skeletonmirror の後に **Rig Pose** ノードを追加します。*rigpose_foot* に名前を変更します。次に、*skeletonblend_reversefoot* ノードの2つ目の入力に接続します。



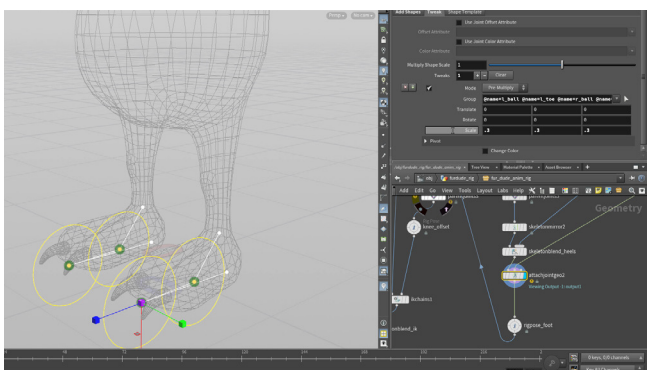
05 *ikchains* ノードに **Display** フラグを設定します。 *rigpose_foot* ノードに移動し、 *l_heel* ジョイントを選択します。そのジョイントを動かすと、足と脚のチェーン全体が動きます。 *l_toe* ジョイントを選択して、回転します。同じようにリバースフットが機能し、IKチェーンがアクティブになります。

最後は **Clear** を押して、すべてのジョイントを削除します。いくつかは後でまた追加します。かかとのジョイントについては、メインコントロールの一部として前にセットアップした、heel ジョイントを使用します。



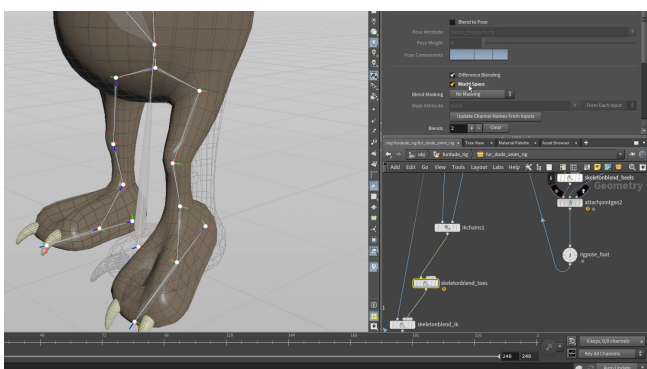
06 つい先ほど使用したリバースフットの *skeletonmirror* と *rigpose* ノードの間に **Skeleton Blend** ノードを追加します。このノードの名前を *skeletonblend_heels* に変更し、 **World Space** チェックボックスを **オン** にして、 **weight1** を **1** に設定します。その2つ目の入力に *skeletonblend_controls* ノードの出力を接続します。

Group の横の矢印をクリックし、2つの *heel* ジョイントを選択します。 **Enter** を押します。メインコントロールの *rigpose* を選択したら、左または右のかかたを動かして、セットアップ全体をコントロールしてみましょう。COG ジョイントを選択して上下に動かしても、IKチェーンは適切に機能するはずで。



07 **Attach Joint Geometry** ノードを *skeletonblend_heels* ノードと *rigpose_foot* ノードの間に配置します。その2つ目の入力に、 **Main Controls** ネットワークボックスの *mergepacked* ノードを接続します。このノードに **Display** フラグを設定し、Scene View で **Enter** を押します。

toe と *ball* ジョイントを選択し、 **G** を押してスクロールホイールを使用して *circle_ctrl* ジオメトリを見つけます。上部のオペレーションコントロールツールバーで、 **Mode** を **Tweak Shapes** に変更します。 *toe* と *ball* ジョイントを選択し、 **G** を押します。 **E** を押してスケールハンドルにしたら、中央のハンドルをクリック & ドラッグして、コントロールを約 **0.3** にスケールします。



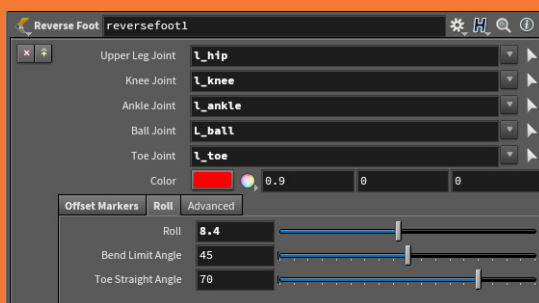
08 *bonedeform* ノードに **Display** フラグを設定します。 *rigpose_foot* ノードを使用して *l_ball* のポーズを決めます。ボール(足の土踏まずから前)を回転させると、つま先は曲がるのではなく、下を向きませ。

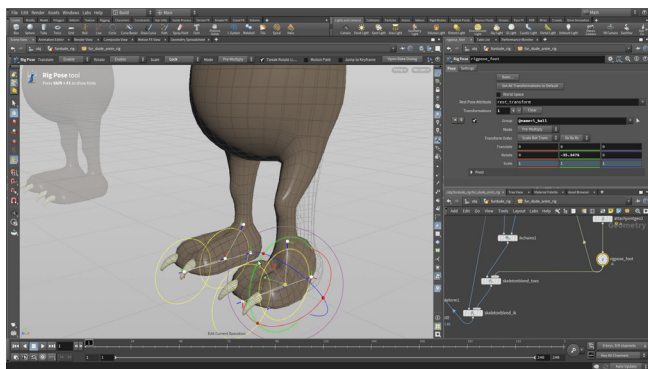


REVERSE FOOT SOP

Reverse Foot SOP は、足のコントロールや脚のキネマティクスの駆動に利用できます。 **Reverse Foot SOP** には足の回転を制御できるスライダや、部位別のコントロールがあります。ただし、このレッスンでは使用しません。

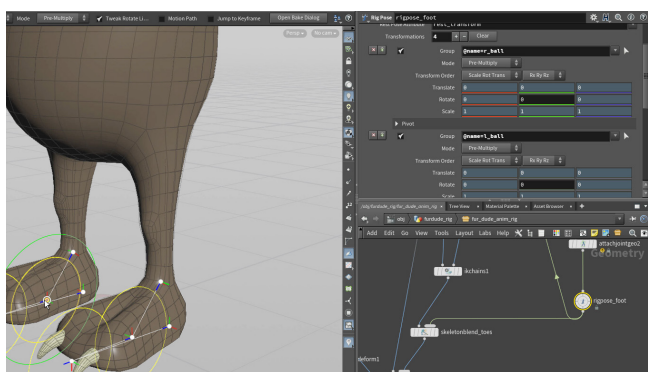
ここでは手作業でリバースフットソリューションを作成して、ジョイントをどのように操作すれば必要なコントロールが得られるかを学びます。



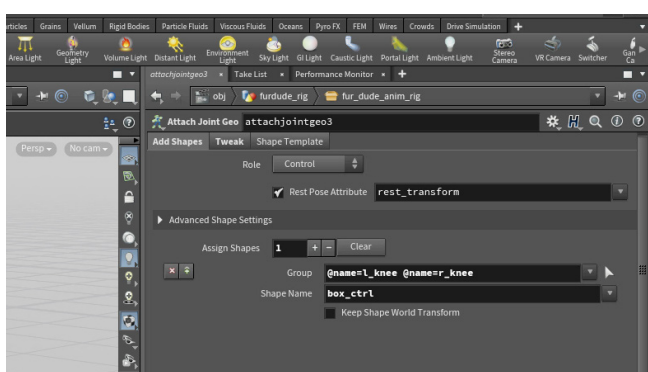


09 *ikchains* と *skeletonblend_ik* ノードの間に **Skeleton Blend** ノードを追加します。このノードの名前を **skeletonblend_toes** に変更し、**World Space** チェックボックスを**オン**にして、**weight1** を **1** に設定します。その2つ目の入力に **rigpose_foot** ノードの出力を接続します。**Group** の横の矢印をクリックし、**toe** と **ball** ジョイントを選択します。**Enter** を押します。

これで **ball** ジョイントを回転すると、つま先が正しい方向を向くようになりました。

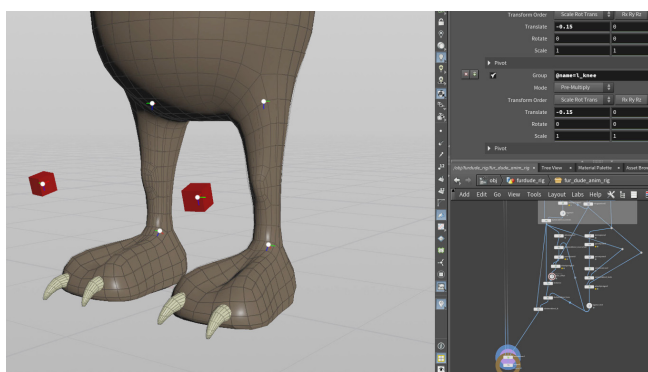


10 *rigpose_feet* ノードを選択します。このノードには**左右の ball** と **toe** のジョイントのみがリストされていることを確認します。すべてのジョイントに対して、**Translate**、**Rotate**、**Scale** パラメータ上で **RMB クリック > Lock Parameter** を選択します。その後、それぞれのジョイントの **Rotate Y** で **RMB クリック > Unlock Parameter** を選択します。これで、これら4つのジョイントのいずれかを選択したときに、**Rotate Y** のみが可能になりました。



11 Main Controls ネットワークボックスに戻ります。**Box** ノードをネットワークに追加します。**Uniform Scale** を **0.02** に設定します。次に **Color** ノードを追加して、**赤色** に設定します。このノードを **mergedpacked** ノードに接続して、**Name 2** を **box_ctrl** に設定します。

parentjoints ノードと *knee_offset* ノードの間に **Attach Joint Geometry** ノードを配置します。その2つ目の入力に **mergedpacked** ノードを接続します。**Mode** を **Tweak Shapes** に設定したら、**knee** ジョイントを選択します。**G** を押してスクロールホイールを使用し、**box** ジオメトリを見つけます。



12 *knee_offset* ノードに移動します。2つの **knee** ジョイントのみがリストされていることを確認します。その両方のジョイントに対して、**Rotate** と **Scale** パラメータ上で **RMB クリック > Lock Parameter** を選択します。この2つのジョイントは、移動によってコントロールできるようになりました。

両方の膝の **Translate** を **-0.15, 0, -0.05** に設定します。

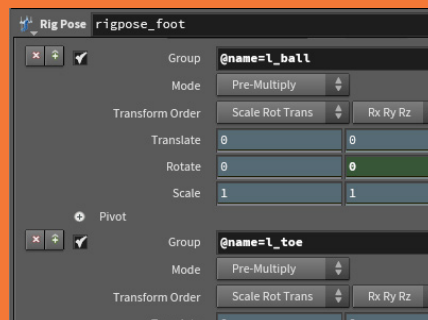
背骨と頭部のジョイントを定義するのに使用したこれらすべてのノードを囲む**ネットワークボックス**を追加し、**Leg Controls** と名前を付けます。



RIG POSE の役割

ここまで、Rig Pose ノードを使用してリグをテストしてきました。アニメーションコントロールリグを構築するときは、このノードを使って、トップレベルにプロモートされるパラメータをセットアップしたり、デジタルアセットの使用時に表示されるジョイントを定義することもできます。

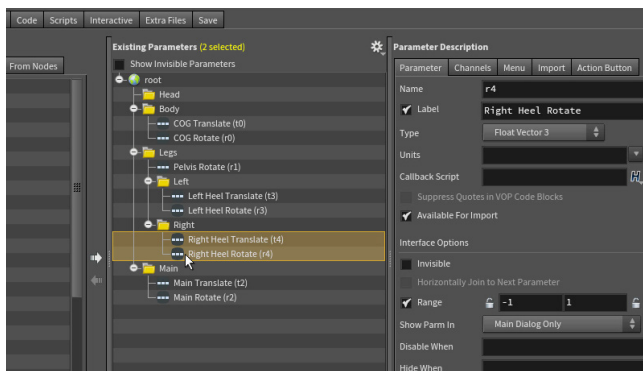
こうしたノードのセットアップでは、不要なパラメータを誤って追加して、トップレベルに余計なジョイントが追加されてしまうことがよくあります。また、X ボタンを間違えてクリックして、必要なパラメータを削除してしまう可能性もあります。



パート 14

脚と背骨のコントロールのプロモート

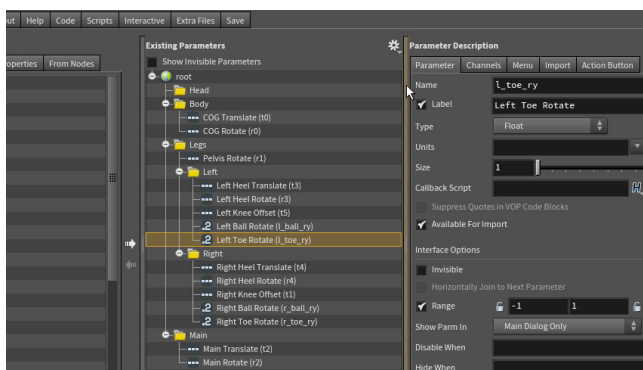
脚と背骨のすべてのコントロールをアニメータが利用できるようにするには、パラメータをアセットのトップレベルにプロモートする必要があります。この手順を行わないと、アニメータに必要なコントロールを渡すことができません。また、アニメータに使用してほしくないパラメータを非表示にしておくことも可能です。



01 Assets メニューから、Edit Asset Properties > Fur Dude Anim Rig を選択します。Parameters タブをクリックします。

フォルダを4つ作成します。それぞれ **Head**、**Body**、**Legs**、**Main** と名前を付けます。**COG** パラメータを **Body** フォルダ、**Main** パラメータを **Main** フォルダにドラッグします。

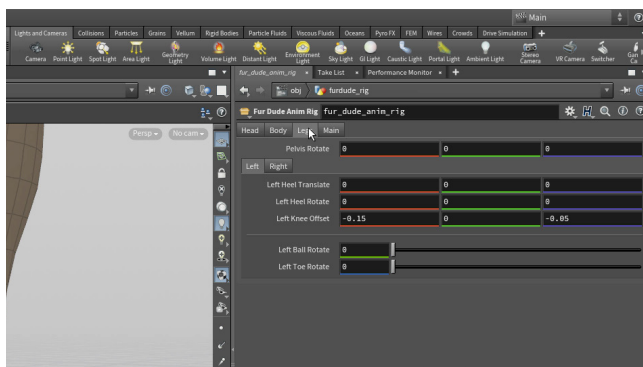
Legs フォルダの中に2つのフォルダを作成し、それぞれ **Left** および **Right** と名前を付けます。**Left Heel** パラメータを **Left** フォルダに、**Right Heel** パラメータを **Right** フォルダに追加します。



02 **knee_offset** ノードから、**L_knee** の Translate 値を **Leg** フォルダの **Left** フォルダにドラッグします。名前を **Left Knee Offset** に変更します。Channel タブをクリックして、デフォルトを **-0.15, 0, -0.1** に設定します。

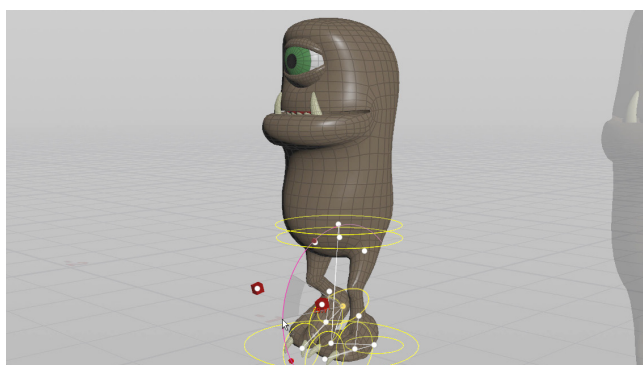
rigpose_foot ノードで、**L_ball** ジョイントから **Rotate Y** をドラッグし、名前を **Left Ball Rotate** に変更して、**Range** を **0, 30** に設定します。同様に **L_toe** ジョイントから **Rotate Y** をドラッグし、名前を **Left Toe Rotate** に変更して、**Range** を **0, 20** に設定します。**Knee Offset** と **Ball Rotate** パラメータの間に **Separator** をドラッグします。

右膝および右足についても同じ手順を繰り返します。



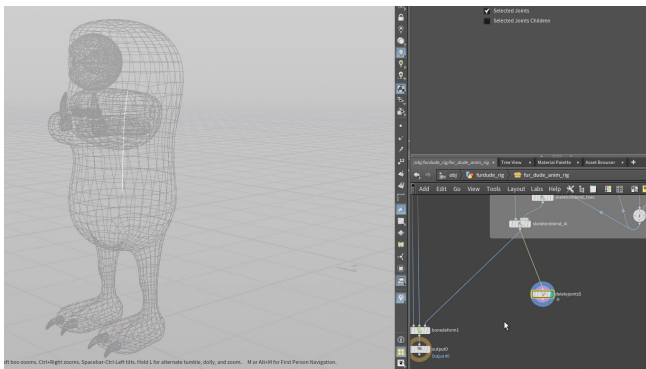
03 **Accept** をクリックし、新しいパラメータとコントロールをアセットに保存します。1つ上のレベルに移動して、**fur_dude_rig** にレイアウトされたアセットパラメータを確認します。

これらのパラメータは、作業対象のロックしていないリグに設定されています。これらのパラメータをそのままテストするのではなく、**test_rig** を使ってテストしましょう。



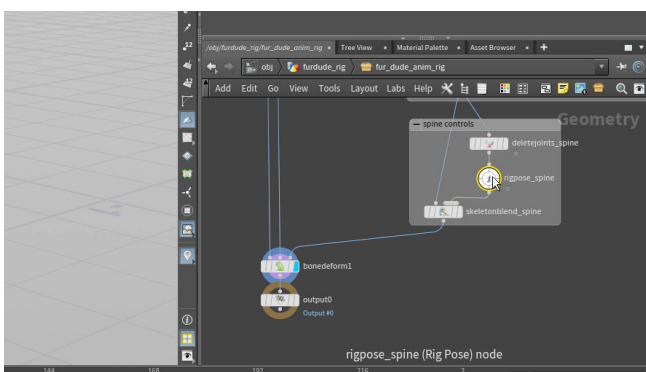
04 ネットワークビューで、**1** を押して **test_rig** に戻ると、新しいコントロールが表示されるようになりました。

Scene View のテストリグのさまざまな動作を、ハンドルやフローティングパネルのパラメータを使って試します。



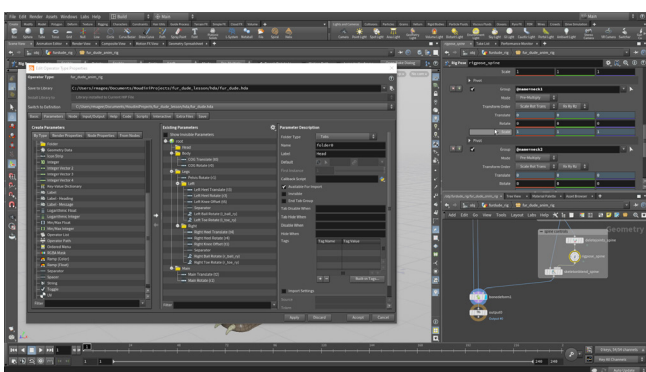
05 *skeletonblend_ik* ノードから **Delete Joints** ノードを分岐させて、その **Display フラグ**を設定します。名前を *deletejoints_spine* に変更します。

Group の横の矢印をクリックし、Scene View で *spine1*、*spine2*、*spine3*、*neck1*、*neck2*、*jaw* ジョイントを選択します。**Enter** を押し、**Operation** を **Delete Non-Selected** に設定します。



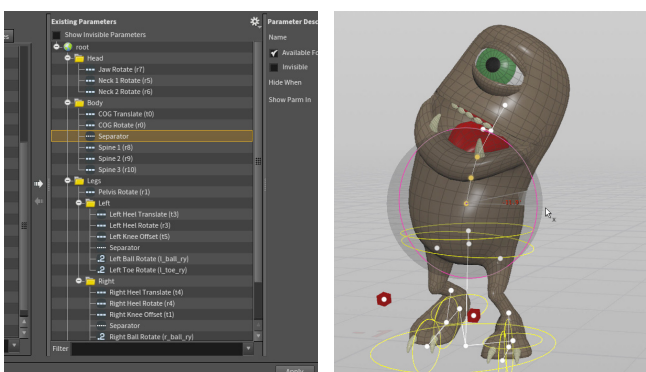
06 **Rig Pose** ノードを追加して、名前を *rigpose_spine* に変更します。*skeletonblend_ik* と *bonedeform* ノードの間に **Skeleton Blend** ノードを追加します。ノードの名前を *skeletonblend_spine* に変更し、**World Space** チェックボックスを **オン** にして、**weight1** を **1** に設定します。その2つ目の入力に *rigpose_spine* ノードの出力を接続します。

背骨と頭部のジョイントを定義するのに使用したこれらすべてのノードを囲む **ネットワークボックス**を追加し、**Spine Controls** と名前を付けます。



07 **Display フラグ**を設定したら、Scene View で **S** キーを押しながらすべてのジョイントを選択します。これらのジョイントがリグポーズのリストに追加されます。

これらすべてのジョイントについて、**Translate** および **Scale** パラメータをロックします。ここでは回転のみを使用します。



08 **Assets** メニューから、**Edit Asset Properties > Fur Dude Anim Rig** を選択します。**Parameters** タブをクリックします。

neck1、*neck2*、*jaw* を **Head** フォルダにドラッグして、**Neck 1 Rotate**、**Neck 2 Rotate**、**Jaw Rotate** と名前を付けます。*spine1*、*spine2*、*spine3* を **Body** フォルダにドラッグして、**Spine 1**、**Spine 2**、**Spine 3** と名前を付けます。COG パラメータと背骨パラメータの間に **Separator** を追加します。

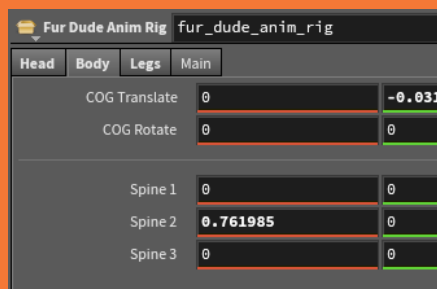
Accept をクリックします。これで新しいコントロールがリグに保存されました。*test_rig* でコントロールを試してみましょう。



背骨のコントロール

このリグでは、ジョイントの回転を使用して背骨をセットアップしました。これは **フォワードキネマティクス** と呼ばれるものです。リグの他のパーツと同じく、メインのスケルトンのジョイントをベースに、**Rig Pose** でセットアップして、トップレベルにプロモットします。

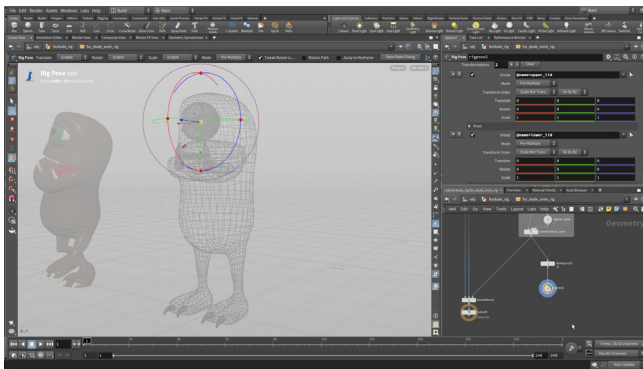
これらのパーツにはコントロールジオメトリは使用しませんでした。Scene View でジョイントを簡単に選択できるからです。すべてのジョイントにコントロールジオメトリが必要なわけではありません。



パート 15

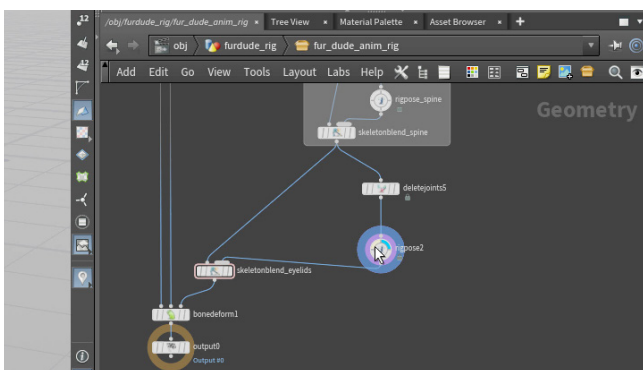
目のコントロール

次は、コントロールジオメトリを使ってまぶた(eyelid)をセットアップし、重なり合ったジョイントを選択しやすくします。また、VOP と呼ばれる Houdini の別のセクションを使用して、目のターゲットジョイントを眼球の Look At としてセットアップします。これらのパーツのリギングが終了したら、適切なパラメータをキャラクターのアセットに再度プロモートします。

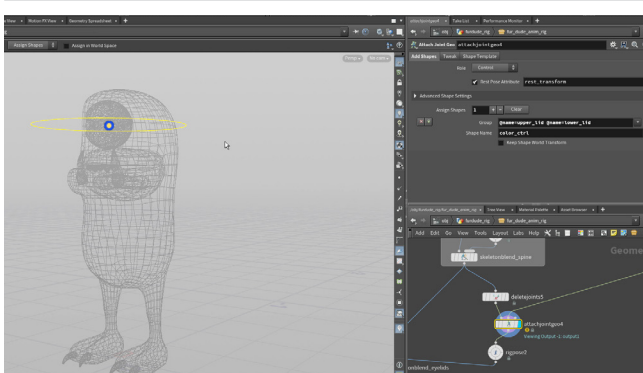


01 *skeletonblend_spine* ノードから *Delete Joints* ノードを分岐させて、**Display フラグ**を設定し、*deletejoints_eyelids*と名前を付けます。**Group**の横の矢印をクリックし、**Rig Tree**で *upper_lid*と *lower_lid*を選択します。カーソルを Scene View に移動して **Enter**を押したら、**Operation**を **Delete Non-Selected**に設定します。

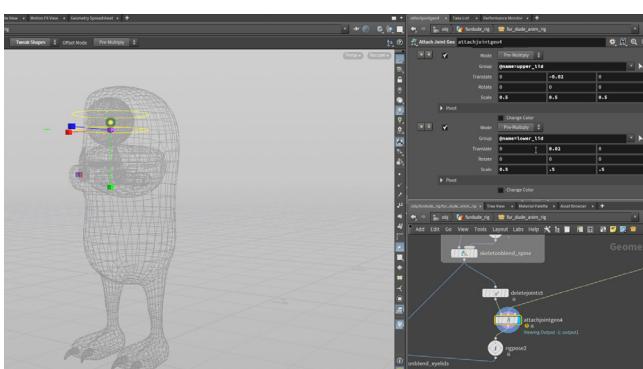
Rig Pose ノードを追加して、名前を *rigpose_eyelids*に変更します。**Display フラグ**を設定したら、Scene View で **S** キーを押しながらすべてのジョイントを選択します。これらのジョイントがリグポーズのリストに追加されます。



02 *skeletonblend_spine* と *bonedform* ノードの間に *Skeleton Blend* ノードを追加します。このノードの名前を *skeletonblend_eyelids*に変更し、**World Space** チェックボックスを **オン**にして、**weight1**を **1**に設定します。その2つ目の入力に *rigpose_eyelids* ノードの出力を接続します。



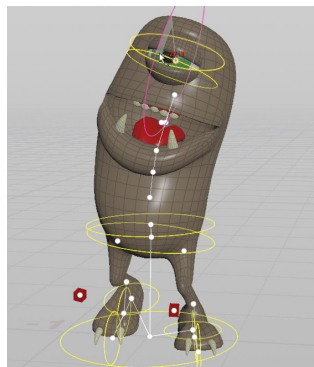
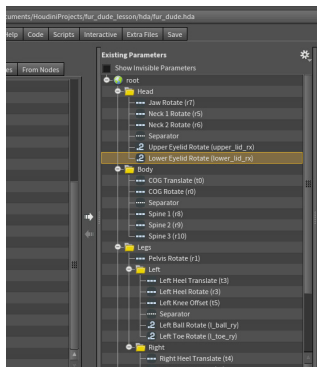
03 *deletejoints* ノードと *rigpose* ノードの間に *Attach Joint Geometry* ジオメトリを配置します。その2つ目の入力に *mergepacked* ノードを接続します。このノードに **Display フラグ**を設定し、Scene View で **Enter**を押します。**Mode**が **Assign Shapes**に設定されていることを確認します。まぶたのジョイントを2つとも選択したら、**G**を押して、スクロールホイールを使用して *circle_ctrl* ジオメトリを見つけます。



04 上部のオペレーションコントロールツールバーで、**Mode**を **Tweak Shapes**に変更します。

eyelid ジョイントを選択し、**G**を押してトランスフォームハンドルを表示します。**E**を押してスケールハンドルにしたら、中央のハンドルをクリック&ドラッグして、これらのコントロールが少し小さくなるまで3方向すべてでスケールします(パラメータエディタで約0.5)。

upper_eyelid ジョイントを選択して、上に **0.02** 移動します。次に、*lower_eyelid* ジョイントを選択して、下に **0.02** 移動します。

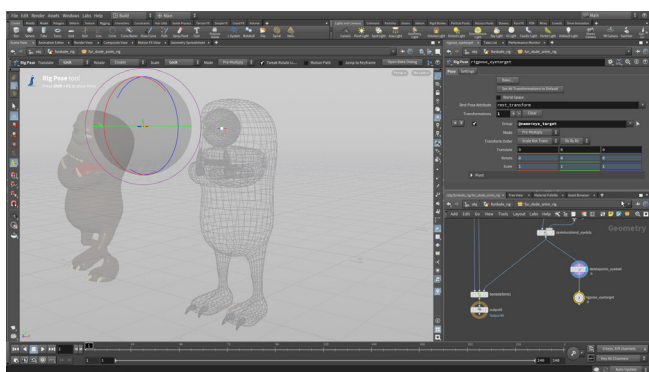


05 Assets メニューから、Edit Asset Properties > Fur Dude Anim Rig を選択します。Parameters タブをクリックします。

これらすべてのジョイントについて、Translate、Rotate、Scale パラメータをロックします。Rotate X パラメータをロック解除します。rigpose_eyelids ノードで、Rotate X を upper_lid から Head フォルダにドラッグします。Range を -10, 30 に設定します。次に、Rotate X を lower_lid から Head フォルダにドラッグします。Range を -20, 20 に設定します。

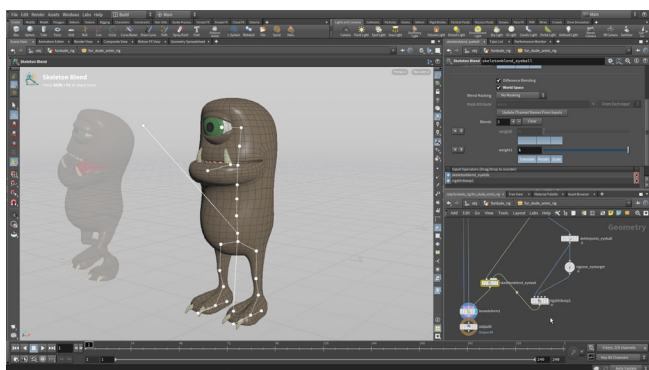
Separator を追加して、目のコントロールを頭部のコントロールから分割します。

Accept をクリックします。これで新しいコントロールがリグに保存されました。test_rig でコントロールを試してみましょう。



06 skeletonblend_eyelids ノードから Delete Joints ノードを分岐させて、Display フラグを設定し、deletejoints_eyes と名前を付けます。Group の横の矢印をクリックし、Scene View で eyeball および eye_target ジョイントを選択します。Enter を押し、Operation を Delete Non-Selected に設定します。

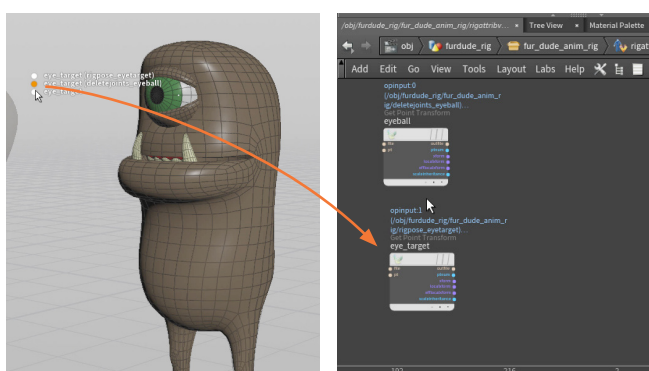
Rig Pose ノードを追加して、名前を rigpose_eyetarget に変更します。Display フラグを設定し、Scene View で eye_target ジョイントを選択します。そのジョイントがリグポーズのリストに追加されます。Look At 拘束でコントロールするので、eyeball ジョイントは必要ありません。eye_target ジョイントについて、Rotate および Scale パラメータをロックします。



07 Rig Attribute VOP ノードを追加します。1つ目の入力に deletejoints_eyes ノードを、2つ目の入力に rigpose_eyetarget ノードを接続します。

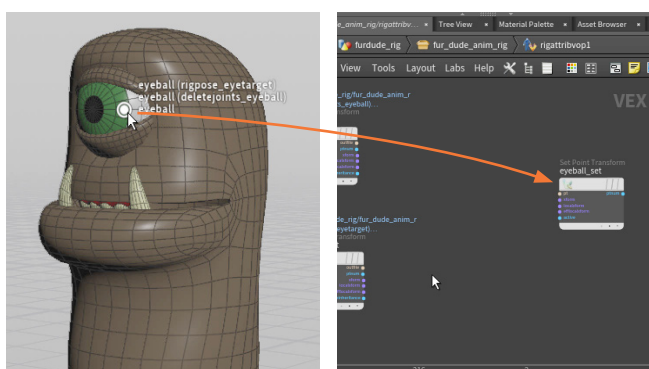
skeletonblend_eyelids と bonedform ノードの間に Skeleton Blend ノードを追加します。このノードの名前を skeletonblend_eyeball に変更し、World Space チェックボックスをオンにして、weight1 を 1 に設定します。その2つ目の入力に rigattributevop ノードの出力を接続します。

bonedform ノードに Display フラグを設定します。

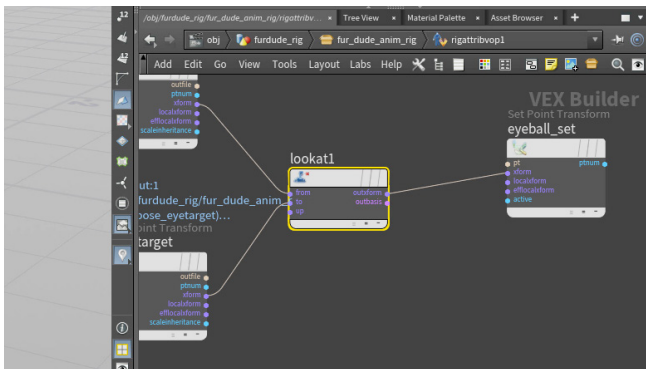


08 rigattributevop ノードをダブルクリックして、中に入ります。Scene View で eyeball ジョイントをクリックし、eyeball (deletejoints) バージョンをネットワークエディタにドラッグします。これにより Get Point Transform ノードが配置されます。これは First Input からの eyeball ジョイントにフォーカスするノードです。

eye_target ジョイントをクリックし、eyetarget (ripose_eyetarget) バージョンをネットワークエディタにドラッグします。これにより Get Point Transform ノードが配置されます。これは Second Input からの eye_target ジョイントにフォーカスするノードです。

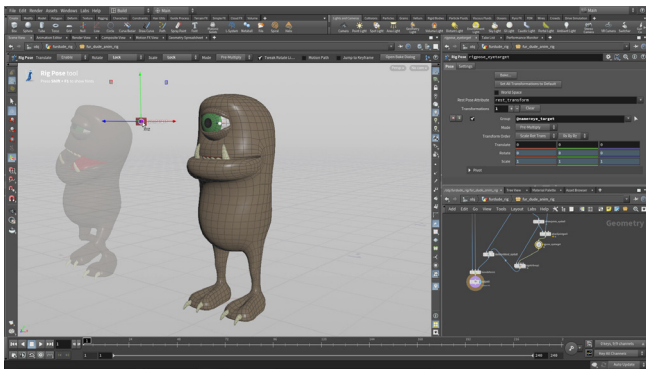


09 eyeball ジョイントをクリックし、eyeball バージョンをネットワークエディタにドラッグします。Set Point Transform ノードが配置されます。これは eyeball ジョイントにフォーカスするノードです。



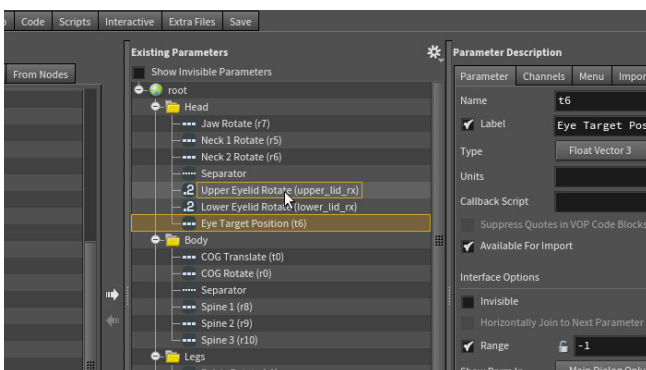
10 Tab > Look At (KineFX) を選択し、ノードを中央に配置します。eyeball **getpointtransform** ノードの **xform** 出力を、**lookat** ノードの **from** 入力に接続します。eye_target **getpointtransform** ノードの **xform** 出力を、**lookat** ノードの **to** 入力に接続します。**lookat** ノードの **outxform** 出力を、**eyeball_set** ノードの **xform** 入力に接続します。

眼球のジオメトリが反転しています。**lookat** ノードを選択し、**Look At Axis** を **Z** に設定すると、最初にリグをセットアップしたときに設定した方向と一致します。



11 **Attach Control Geometry** ノードを **deletejoints_eyes** ノードと **rigpose_eyetarget** ノードの間に配置します。その2つの入力に、先ほどのネットワークから **mergepacked** ノードを接続します。このノードに **Display** フラグを設定して、**Handle** ツールに移動します。**Mode** が **Assign Shapes** に設定されていることを確認します。3D ビューで **eye_target** ジョイントを選択したら、**G** を押し、スクロールホイールを使用して **square_ctrl** ジオメトリを見つけます。

bonedform ノードに **Display** フラグを設定します。**rigpose_target** をクリックします。**eye_target** ジョイントを選択して動かすと、それに合わせて眼球の向きが変わります。**Undo** を実行して元の位置に戻します。

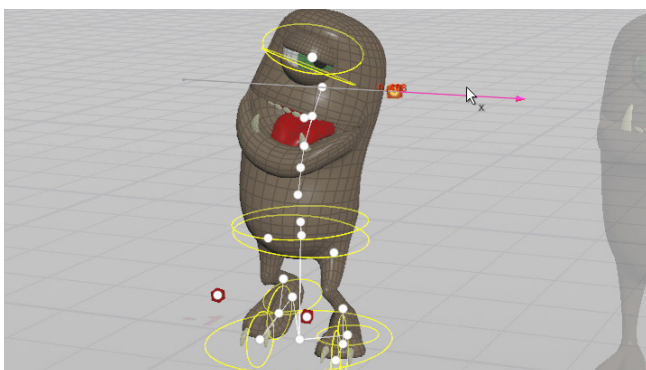


12 目を定義するのに使用したすべてのノードを囲む **ネットワークボックス** を追加し、**Eye Controls** と名前を付けます。

Assets メニューから、**Edit Asset Properties > Fur Dude Anim Rig** を選択します。**Parameters** タブをクリックします。

rigpose_eyetarget ノードで、**Translate** パラメータを **eye_target** から **Head** フォルダの **目 (eye)** のセクションにドラッグします。**Eye Target Position** と名前を付けます。

Accept をクリックします。これで新しいコントロールがリグに保存されました。



13 **test_rig** でコントロールを試してみましょう。

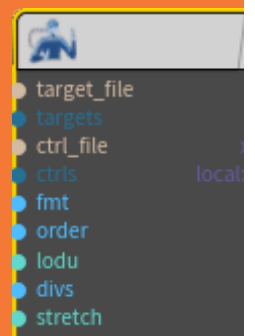
このコントロールリグのすべてのパーツが完成しました。歩行サイクルのアニメーションを付け始められます。テストリグのコピーをもう1つ作成し、そのネットワークを使ってアニメートします。

このデジタルアセットは、複数のシーンファイルに複数のインスタンスを作成できます。後戻りして変更を加えた場合、すべてのアセットが更新されます。これが、デジタルアセットリグを使用したパイプラインの利点です。



VOP でのリギング

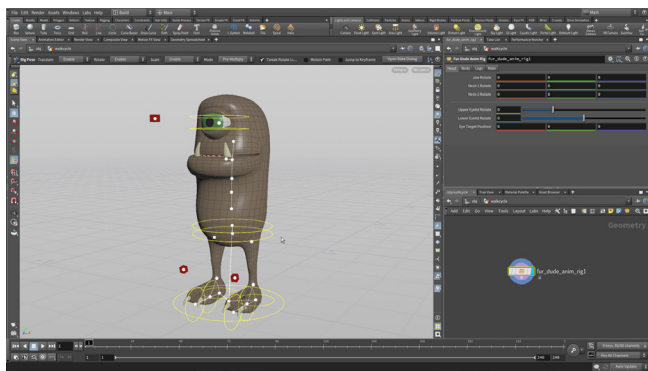
Rig Attribute VOP は、このレッスンで紹介したもの以外にもさまざまなソリューションを提供します。IK チェーンはこの方法で構築でき、前に使用した IK チェーン SOP にはその1つが含まれています。VOP を使用すると、Curve Solver、Realistic Shoulder、リバースフットなどもセットアップできます。Scene View から VOP ネットワークにジョイントをドラッグできるこれまでにないワークフローは、ワークフローを高速化します。



パート 16

リグのアニメーション

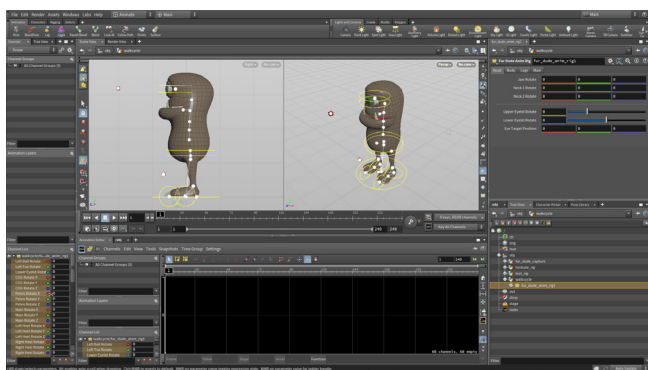
ファー・デュード(Fur Dude)にキーフレームを設定して、歩行サイクルを作成します。Channel List を使用してチャンネルをピン留めし、動きをブロッキングするなど、新しいツールも使用します。簡単な歩行サイクルを完成させて、動くファー・デュードを確認しましょう。ここでの目標は、基本のキーフレームワークフローを計画し、KineFX リグのアニメーション方法を理解することです。



01 オブジェクトレベルに移動します。**Tab > Geometry** を選択してノードを配置します。名前を walkcycle に変更します。すべてのオブジェクトの **Display フラグ** をオフにします。

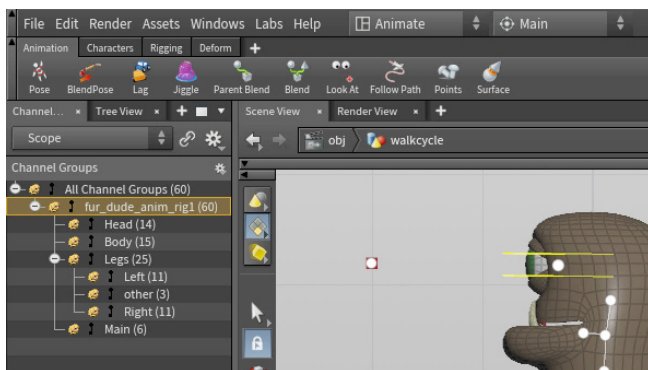
ダブルクリックして walkcycle のの中に入り、ネットワークビューで **Tab > Fur Dude Anim Rig** を選択します。**Enter** を押して、原点に配置します。これは新しいロックバージョンの Fur Dude リグで、ゼロからアニメートしていきます。

この方法では、シーンに別バージョンのキャラクタアセットを配置します。このシーンファイル(または任意のシーンファイル)には、複数のバージョンを配置でき、いずれもディスク上の同じアセット定義を参照します。



02 Desktop メニュー(ここでは Build)から、**Animate** を選択します。キーフレームワークフローに対応したパネルが表示されます。場合によっては、walkcycle オブジェクトに戻る必要があります。

左側の **Channel List** は、キャラクタのアニメーションのブロッキングで重要な役割を果たします。**アニメーションエディタ**では、アニメーションカーブを表示および編集できます。このレッスンでは、動きをブロッキングするまでにとどめ、カーブの編集は行いません。



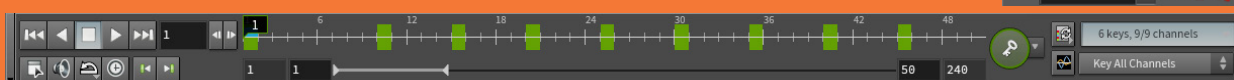
03 ネットワークビューで、walkcycle オブジェクト内の **fur_dude_anim_rig** を選択します。パラメータエディタで、右上のボックスアイコンをクリックします。**Parameters and Channels > Create Nested Channel Groups** を選択します。ポップアップウィンドウで **Close** をクリックします。アセットのパラメータがリストされ、フォルダ別に整理されています。

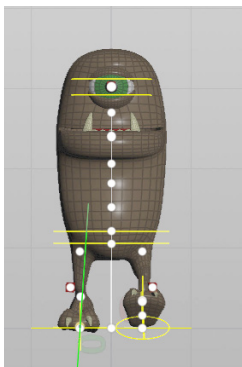
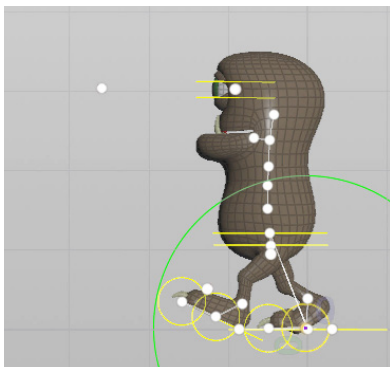
fur_dude_anim_rig チャンネルグループの横の **ピン**アイコンをクリックして、これらのチャンネルをピン留めします。タイムラインが**フレーム 1**に設定されていることを確認したら、**K**を押して、すべてのチャンネルにキーフレームを設定します。



チャンネルの動作

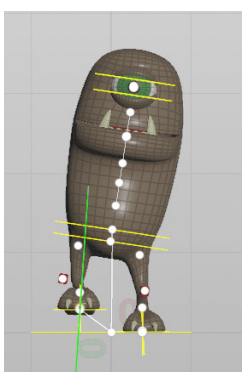
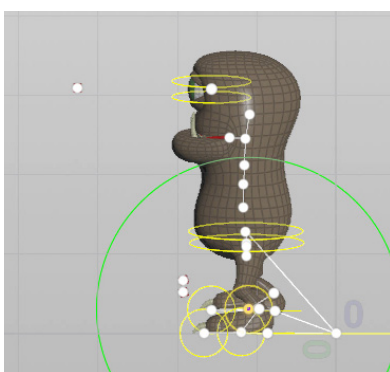
ジョイントを選択すると、そのチャンネルがチャンネルリストに読み込まれます。**K**を押すと、該当のジョイントにキーフレームを設定できます。ブロッキングのためにチャンネルを読み込んだままにしたい場合は、チャンネルをピン留めするか、チャンネルグループとしてまとめてピン留めします。チャンネルグループはアセットから直接作成できます。グループは、キャラクタ用の UI をどのように構築したかに基づいて編成されます。独自のグループを構築して、特定のチャンネルをピン留めすることも可能です。





04 タイムラインを 10 で開始し、50 で終了するように設定します。
フレーム 10 に移動します。すべてのチャンネルをピン留めした状態のまま、**K** を押し、もう 1 つキーフレームを設定します。先にキーフレームを設定してから、ポーズを付けることをお勧めします。ポージングによって、キーフレームを設定したフレームで値が更新されます。

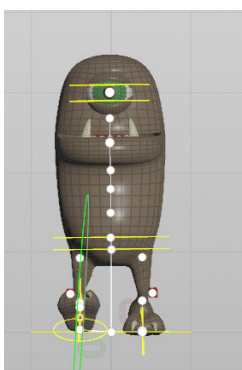
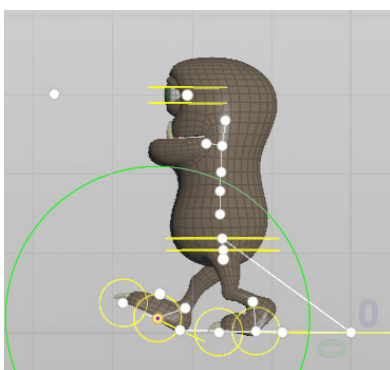
左の heel を前に動かし、上に回転させたポーズを作成します。**COG** を少し下げ、**右の ball** を前方に回転させます。



05 フレーム 15 に移動し、**K** を押します。

左のかかと を回転して平らにします。ただし、移動はしません。左足の動きに合わせて **COG** を動かします。左足に合わせて、**右の heel** を上に移動します。**右の ball** を回転して元に戻し、足を平らにします。

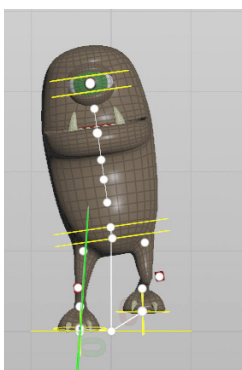
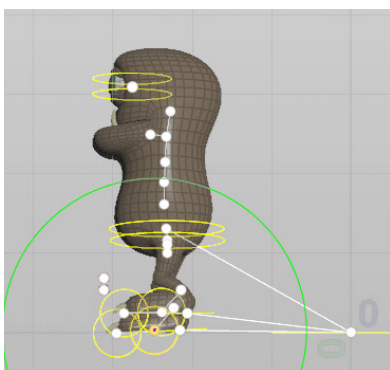
COG を左足の方に少し回転させます。**背骨** の 3 つのジョイントをいっぺんに回転させて、この傾きを強調してもよいでしょう。



06 フレーム 20 に移動し、**K** を押します。

右の heel を前に動かし、上に回転させたポーズを作成します。**COG** を少し下げ、**左の ball** を前方に回転させます。つまり、フレーム 10 とは逆のポーズです。

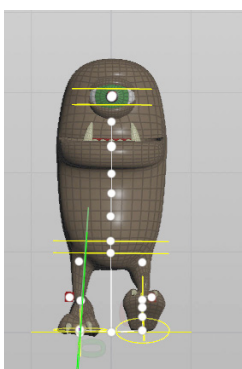
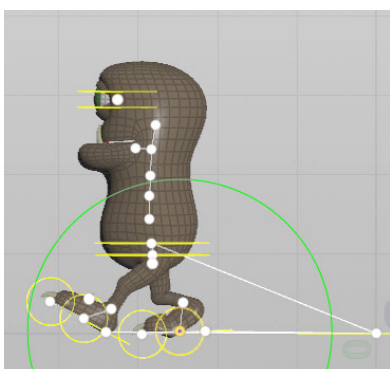
COG と **背骨** のジョイントを回転して中央に戻します。



07 フレーム 25 に移動し、**K** を押します。

右の heel を回転して平らにします。ただし、移動はしません。右足の動きに合わせて **COG** を動かします。左足に合わせて、**左の heel** を上に移動します。**左の ball** を回転して元に戻し、足を平らにします。

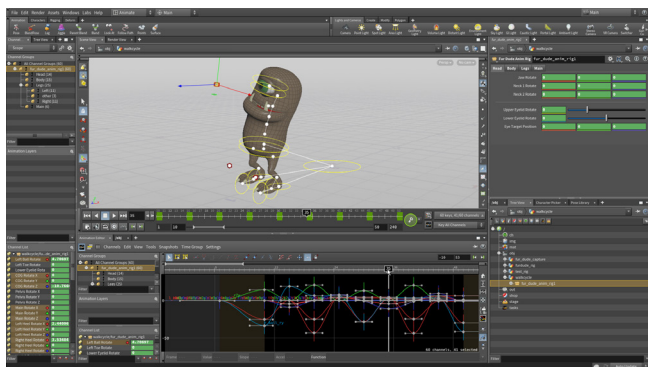
COG を右足の方に少し回転させます。**背骨** の 3 つのジョイントをいっぺんに回転させて、この傾きを強調してもよいでしょう。



08 フレーム 30 に移動し、**K** を押します。

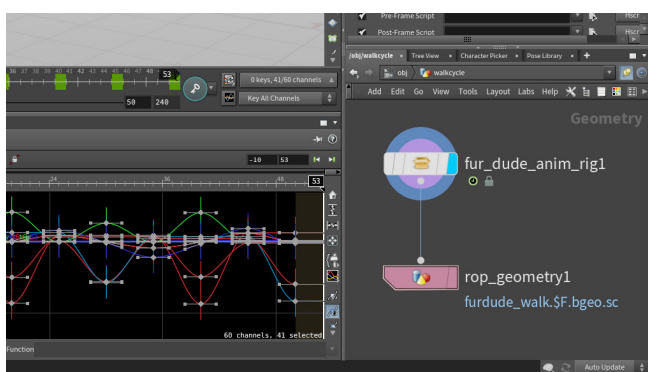
左の heel を前に動かし、上に回転させたポーズを作成します。**COG** を少し下げ、**右の ball** を前方に回転させます。

COG と **背骨** のジョイントを回転して中央に戻します。



09 このパターンをフレーム **50** まで続けます。同じポーズを繰り返すと、前に進む歩行サイクルを作成できます。

この時点で、戻ってポーズを調整したり、動きを整えることができます。キーフレームをさらに追加して、追従するアクションを作成してもよいでしょう。目の動きやまばたきをアニメートしたり、もちろん、50 フレーム以上の長いアニメーションにしてもかまいません。

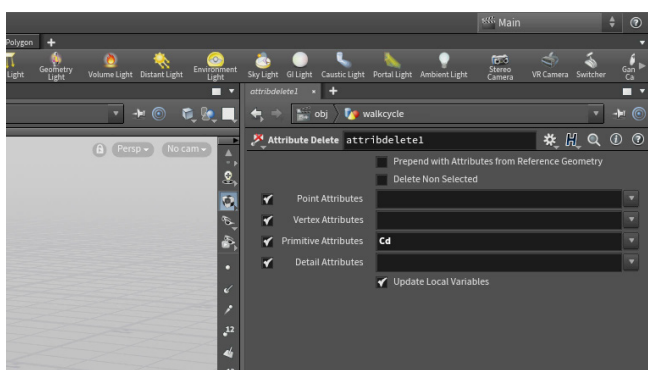


10 **fur_dude_anim_rig** ノードの出力を **ROP Geometry** ノードに接続します。これで、ファー・デュードのジオメトリのキャッシュをエクスポートできるようになります。**Output File** を次のように設定します。

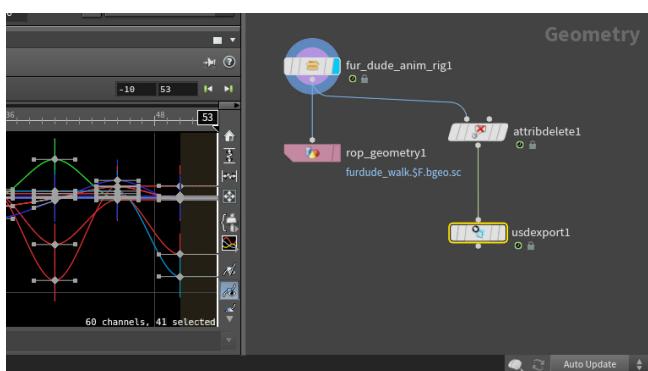
`$HIP/geo/furdude_walk.$F.bgeo.sc`

次に、**Valid Frame Range** を **Render Frame Range** に設定します。**Start/End/Inc** パラメータで **RMB** クリックし、**Delete Channels** を選択します。**Start** を **1**、**End** を **50** に設定します。

Save to Disk ボタンをクリックして、キャッシュをディスクに保存します。これを使って後ほどファーを追加します。



11 **fur_dude_anim_rig** ノードから **Attribute Delete** ノードを分岐させます。**Primitive Attributes** で **Cd** を選択します。これで、全ボディパーツから、カラーが取り除かれます。



12 **attributedelete** ノードの出力を **USD Export** ノードに接続します。これで、ファー・デュードのジオメトリを USD フォーマットでエクスポートできるようになります。**Output File** を次のように設定します。

`$HIP/usd/furdude_walk.usd`

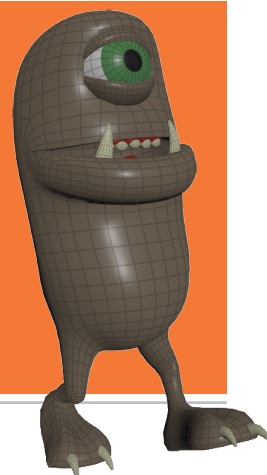
次に、**Valid Frame Range** を **Render Frame Range** に設定します。**Start/End/Inc** パラメータで **RMB** クリックし、**Delete Channels** を選択します。**Start** を **1**、**End** を **50** に設定します。

Save to Disk ボタンをクリックして、キャッシュを USD ファイルに保存します。このファイルは、後のレンダリングプロセスで使用します。



アニメーションのキャッシュ化

Houdini のワークフローはプロシージャルなため、アニメーションのキャッシュ化は絶対に必要なわけではありません。このネットワークのアニメーションは、任意のネットワークで参照してグルーミングすることもできますし、Solaris で参照するために USD に変換することも可能です。キャッシュ化のメリットは、アニメーションを固定したり、統合したディスク上のファイルを使用できることです。これは、プロダクションにとって非常に好都合なアプローチです。Solaris では、ディスクを参照する USD ファイルの方が効率的でもあります。Houdini はディスクからファイルを参照するので、いつでも自由にアニメーションを変更したり、新しいシーケンスを出力できます。また、それらを自動的に取得するようになります。



パート 17

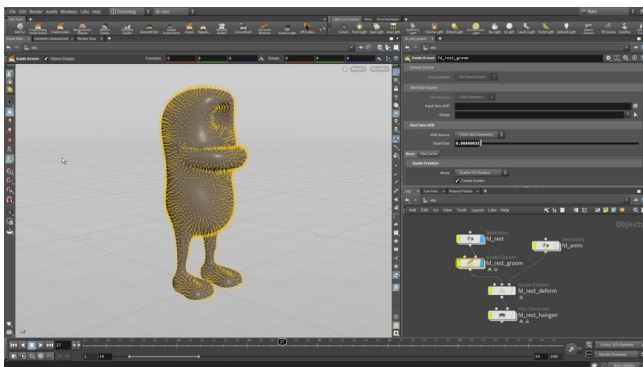
ファーの追加とグルーム

このキャラクターをファー・デュード(Fur Dude)と名付けたのには理由があります。さまざまなグルーミングツールを使用して、もじやもじやヘアを追加し、形状を整えていこうというわけです。グルーミング用に設計されたデスクトップを使用して、縮毛、クランプ(束)、ヘアのダイナミクスを追加し、Fur Dude の歩行に合わせてシミュレートします。完成したら、レンダリング用にエクスポートします。



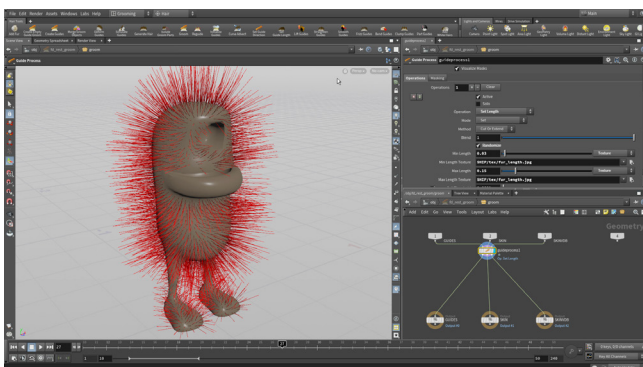
01 Grooming デスクトップに切り替えます。4つの既存のオブジェクトをネットワークボックスで囲み、そのボックスに **Rig & Animate** と名前を付けます。

ネットワークエディタで、**Tab > File** を選択します。ノードを配置したら、ダブルクリックして中に入ります。**Geometry File** の横の **ブラウズ** ボタンをクリックして、**\$HIP/geo** に移動します。**furdude_walk.\$F.bgeo.sc** を選択し、**Accept** を押します。**Blast** ノードを追加し、**Group** を **fur_dude_body** に設定します。体に集中できるように、**Delete Non-Selected** チェックボックスを **オン** にします。**Display フラグ** を設定したら、オブジェクトレベルに移動して、名前を **fd_anim** に変更します。そのオブジェクトを **Alt** ドラッグしてコピーを作成し、**fd_rest** と名前を付けます。その中に入り、**Geometry File** を **\$HIP/geo/furdude_walk.1.bgeo.sc** に変更します。



02 タイムスライダを少し進めましょう。片方のオブジェクトはファー・デュードの静的バージョンで、もう一方はアニメートされています。**Add Fur** ボタンをクリックします。**fd_rest** ジョイントを選択して、**Enter** を押します。

次は **fd_anim** ジョイントを選択して、**Enter** を押します。**fd_rest_anim**、**fd_rest_deform**、**fd_rest_hairgen** の **Display フラグ** をオフにします。**fd_rest** および **fd_rest_groom** の **Display フラグ** をオンにします。

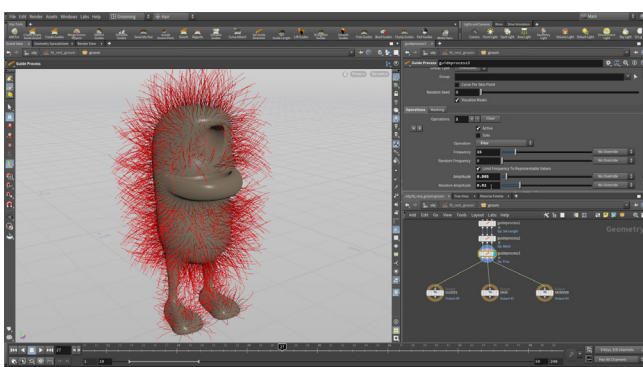


03 **fd_rest_groom** ノードを選択し、**Hair Tools** シェルフで **Set Guide Length** をクリックします。Randomize ボタンをオンにします。**Min Length** を **0.03** に設定し、右側のメニューから **Texture** を選択します。

ファイルブラウザボタンを使用して **\$HIP** を選択したら、**tex** ディレクトリに移動して **fur_length.jpg** を選択します。

Max Length を **0.15** に設定し、**Texture** を再度選択します。右側の矢印を使用して、**fur_length.jpg** を選択します。

これで、目、唇、足の裏がマスクアウトされ、残ったファーの長さはランダムになります。

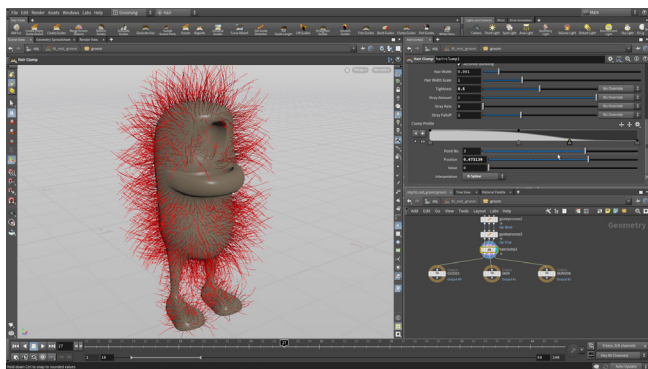


04 **Hair Tools** シェルフで **Bend Guides** をクリックします。**Angle** を **45** に設定して、ガイドをいくらか曲げます。

次は **Frizz Guides** ツールをクリックします。次のように設定します。

- **Frequency** を **15** にする
- **Amplitude** を **0.005** にする
- **Random Amplitude** を **0.02** にする

これで、ヘアをレンダリングしたとき、真っすぐすぎません。モジャモジャ感を増したい場合は、縮毛を追加します。

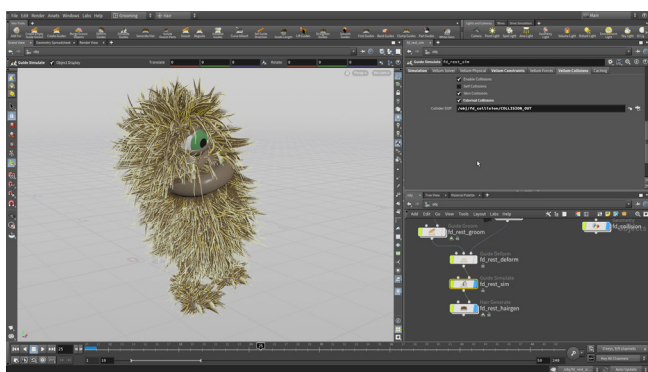


05 Hair Tools シェルフで **Clump Guides** をクリックします。 **Clump Size** を 0.02、**Tightness** を 0.5 に設定します。

Clump Profile を変更します。束の根本から中間では高い値を維持し、ヘアの先端付近で徐々に細めます。

オブジェクトレベルに移動し、**fd_rest** と **fd_rest_groom** の Display フラグを **オフ** にします。そして、**fd_anim**、**fd_rest_sim**、**fd_rest_hairgen** の Display フラグを **オン** にします。

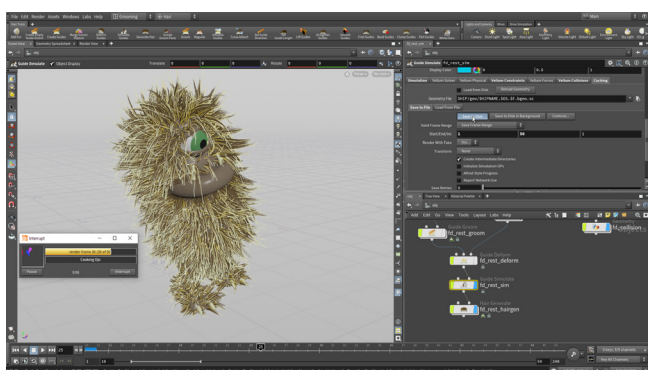
fd_rest_groom ノードを選択し、**Density** を **20000** に設定します。



06 **fd_rest deform** ノードを選択し、Hair Tools シェルフで **Simulate Guides** をクリックします。**fd_rest_sim** ノードで、**Vellum Constraints** タブに移動し、**Bend** の **Stiffness** を **5** に設定します。

fd_anim ノードを **Alt** ドラッグしてコピーを作成し、**fd_collision** と名前を付けます。このノードの中に入り、**blast** ノードで **Delete Non Selected** を **オフ** にします。**Group** に舌、上の歯、歯茎を追加したら、Null ノードを追加して **COLLISION_OUT** と名前を付けます。

オブジェクトレベルで、**fd_rest_sim** を選択します。**Vellum Collisions** で **External Collisions** を **オン** にして、**Collider SOP** を **../fd_collision/COLLISION_OUT** に設定します。



07 Caching タブをクリックして、**Valid Frame Range** を **Save Frame Range** に設定します。**Start/End/Inc** パラメータで **RMB** クリックし、**Delete Channels** を選択します。**Start** を **1**、**End** を **50** に設定します。**Save to Disk** をクリックして、シミュレーションを実行します。

ここで **Load from Disk** チェックボックスを **オン** にします。フレーム毎にヘアを計算するのではなく、キャッシュを使用してファーが定義されるようになります。



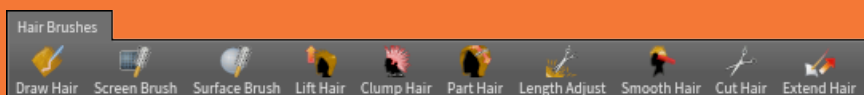
08 **fd_rest_hairgen** ノードを選択します。**Distribution** の **Density** を **1000000** に設定します。Guide Interpolation にスクロールし、**Clump Crossover** を **0.25** に設定して、束同士が少し重なるようにします。これで、ふさふさのヘアを持つファー・デュードの見た目になります。

これらは、Solaris と呼ばれる Houdini のライティングコンテキストでレンダリングするヘアではありません。代わりに、ガイドヘアを取り込んで、レンダリング時にヘアプロシージャルを使用してレンダリングします。



ヘアブラシ

グルーミングデスクトップには、キャラクターのサーフェス上でインタラクティブに使用できるヘアブラシツールもあります。ヘアを長くしたり、滑らかにしたり、カットしたり、伸ばすことができます。ファー・デュードのグルーミングには使いませんが、後でこれらを使用して、最終的なルックに磨きをかけてもよいでしょう。



パート 18

ショットの設定とレンダリング

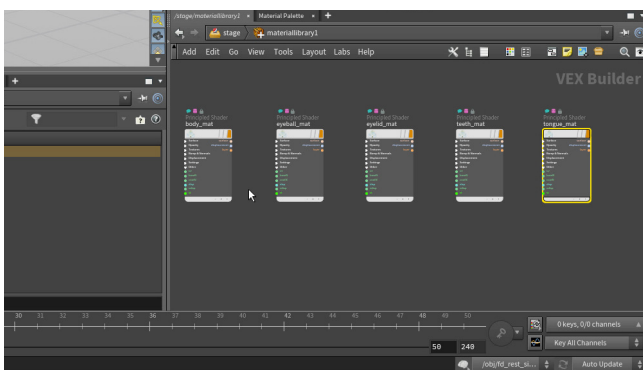
ショットのレンダリング時には、USD ファイルを Solaris ステージで参照し、背景を追加します。Solaris は Houdini のコンテキストで、LOP ノードを使用して USD シーングラフをセットアップします。次に、ファールを読み込んで、カメラとライトを配置します。その後、Karma レンダラを使ってショットのプレビューレンダリングを作成してから、アニメーションシーケンスをレンダリングします。



01 デスクトップを **Solaris** に変更します。パスバーで **Stage** を選択します。ネットワークビューで **Tab > Reference** を選択してからクリックし、**Reference** ノードを追加します。

Reference File の横にある **File Pattern** をクリックして、**furdude_walk.usd** を指定します。ノードの名前を **furdude** に変更します。**Primitive Path** を **/char/`@sourcename`** に設定し、ノード名を使用して、**char** というグループに配置されるようにします。**シーングラフツリー** で **char**、**furdude** の順に展開すると、名前の付いたプリミティブをすべて確認できます。

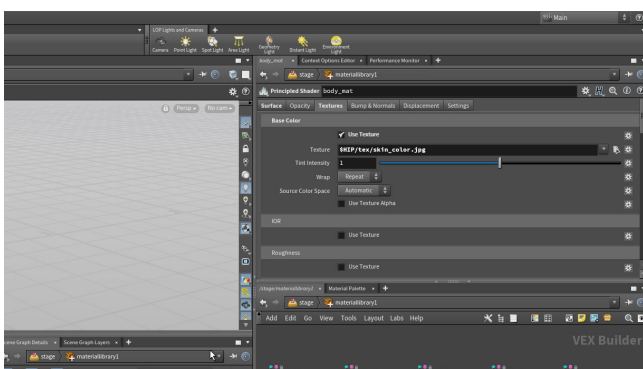
Scene View で、ビューをホームする **スペースバー + H** のような表示ツールを使用して、歩行サイクルがよく見えるようにします。



02 **Tab > Material Library** を選択します。それを **reference** ノードの出力に接続し、**Display フラグ** を設定します。

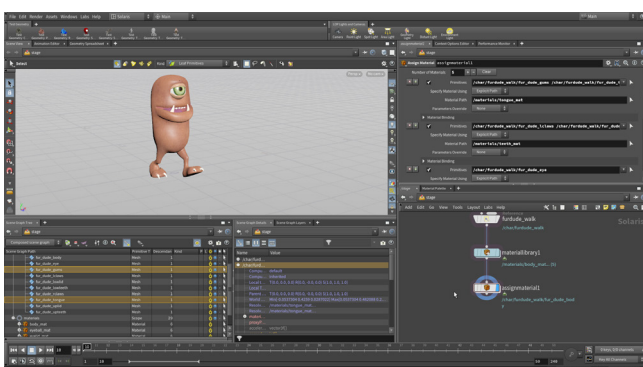
Material Palette ペインに移動します。**/stage/materiallibrary** の横にある矢印をクリックし、このエリアを開きます。パレットの左側のマテリアルギャラリーをスクロールして、**Principled Shader** マテリアルを **materiallibrary** 作業エリアにドラッグします。

ネットワークビューに移動して、このマテリアルを **Alt** ドラッグしてさらに4つ作成します。5つのマテリアルの名前を、それぞれ **body_mat**、**eyeball_mat**、**eyelid_mat**、**teeth_mat**、**tongue_mat** に変更します。マテリアルは、**シーングラフツリー** でも確認できます。



03 **furdude_body_mat** で、**Surface** タブの **Base Color** を **1, 1, 1** に設定します。**Textures** タブをクリックし、**Base Color** の **Use Texture** をクリックしてオンにしたら、**Texture** の横のボタンを使用してファイルウィンドウを開きます。左側のリストで **\$HIP** をクリックしたら、**tex** フォルダをクリックして開き、**skin_color.jpg** をワンクリックして選択します。**Accept** をクリックし、テクスチャをマテリアルに割り当てます。次に、**Roughness** を **0.5**、**Reflectivity** を **0** に設定します。

同じ手順で、**eye_color.jpg** と **eye_lid.jpg** をそれぞれのマテリアルに割り当てます。**tongue_mat** を赤に近いピンクに、**teeth_mat** を黄色がかった白に設定します。



04 ステージレベルに戻ります。**Material Library** の後に **Assign Material** ノードを追加します。シーングラフから **fur_dude_body** を **Primitives** フィールドにドラッグしたら、**Material Path** の横にある矢印をクリックして、**body_mat** を選択します。横にある **+** (プラス) 記号をクリックして、4つの新しいエントリを追加します。次のように割り当てます。

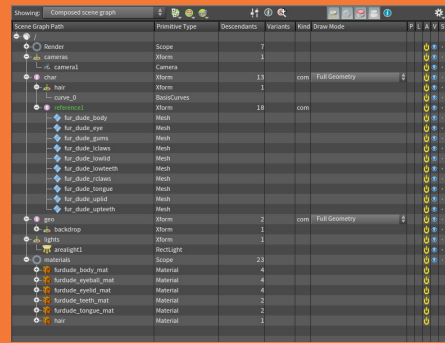
- **fur_dude_eye > eyeball_mat**
- **fur_dude_lowid/uplid > eyelid_mat**
- **fur_dude_lowteeth/upteeth/claws > teeth_mat**
- **fur_dude_tongue/gums > tongue_mat**



USD シーングラフ

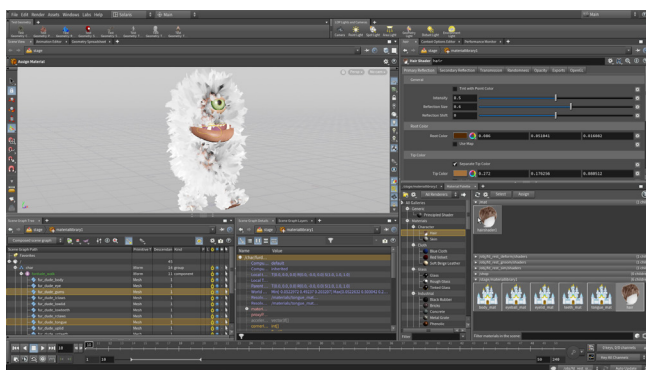
Solaris で作業するとき、LOP ノードを使用して追加するジオメトリやマテリアルは、シーングラフに追加され、USD に変換されます。ライトやカメラを追加すると、それらも USD シーングラフの一部になります。

アーティストの皆さんは、Houdini でライティングやレンダリングを行ううえで、USD を完璧に理解する必要はありません。しかし、プロジェクトのパイプラインの観点から考えると、USD はショット管理に便利なツールとなりますはずです。



05 ネットワークビューで、**Tab** を押して **SOP Import** と入力します。クリックしてノードを配置します。**hair** と名前を変更します。**Import Path Prefix** を **/char/\$OS** に設定します。SOP Path の横のノードアイコンをクリックして、**fd_rest_hairgen** ノードに移動します。

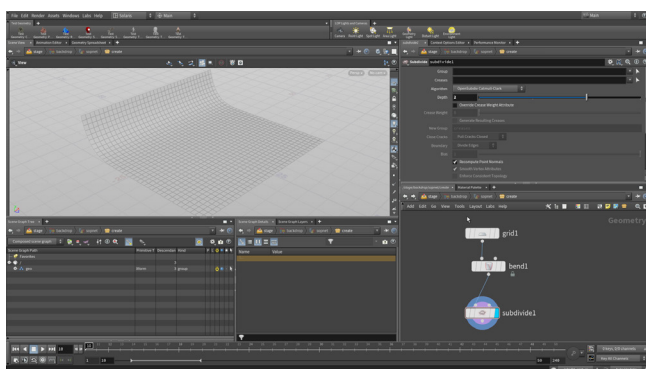
Merge ノードを **furdude** と **materiallibrary** ノードの間に追加します。**hair** ノードをそれに接続します。



06 **Material Palette** ペインに移動します。**/stage/materiallibrary** を開きます。**Hair** マテリアルを **materiallibrary** 作業エリアにドラッグします。

ヘアの **Root Color** と **Tip Color** の設定はデフォルトのままにします。次に **Secondary Reflection** タブをクリックして、**Root Color** を **ダークグレー**、**Tip Color** を **ミディアムグレー** に設定します。

Assign Material ノードに戻り、**Primitives** の横にある矢印をクリックして、**シーングラフ** でファアのカーブを選択します。**Material Path** の横の矢印をクリックして、**hair** を選択します。



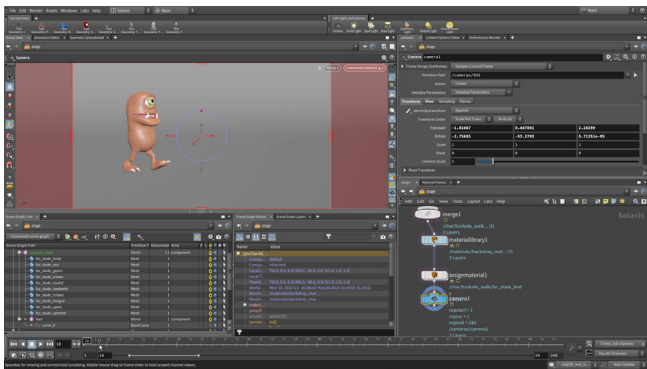
07 ネットワークビューで、**Tab** を押して **Grid** と入力します。クリックしてノードを配置します。ノード名を **backdrop** に変更し、**merge** ノードに接続します。**Import Path Prefix** を **/geo/\$OS** に設定します。**backdrop** ノードを **ダブルクリック** して、ジオメトリレベルに入ります。

Grid ノードを選択し、**Size** を **50, 50**、**Rows** および **Columns** を **10** に設定します。**Grid** ノードの出力を **RMB** クリックして、**Bend** と入力します。クリックして **Bend** ノードを配置したら、**Display フラグ** を設定します。**Bend** を **75**、**Capture Origin** を **0, 0, -10**、**Capture Direction** を **0, 0, -1**、**Capture Length** を **10** に設定します。**bend** ノードの出力を **RMB** クリックして、**Subdivide** と入力します。**Display フラグ** を設定し、**Depth** を **2** に設定します。



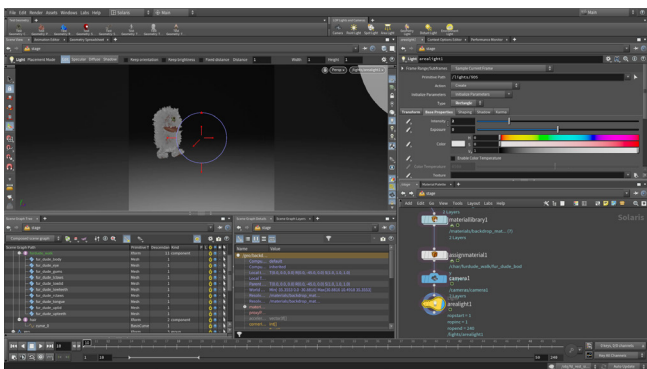
08 オブジェクトレベルに移動し、**Rotate Y** を **-45** 度に設定します。マテリアルを追加して割り当てます。デフォルトのグレーのままにしても、独自のベースカラーを追加してもかまいません。

時間範囲を **10** から **50** に設定します。ヘアが安定した挙動になるまで **10** フレームほどかかるため、このシーケンスはフレーム **10** からレンダリングします。



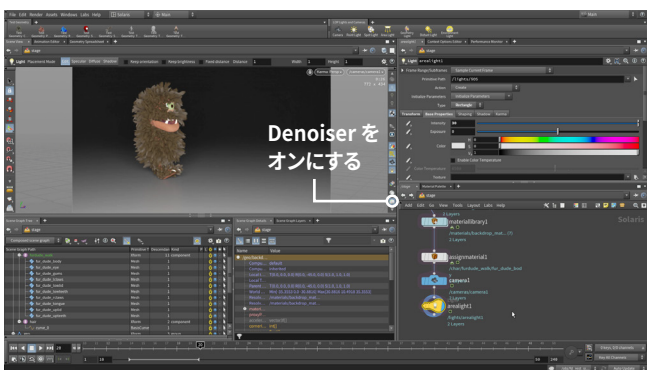
09 表示ツールを使用して、正面から **furdude** が見えるようにします。LOP Lights and Camera シェルフで、**Camera** ツールを **Ctrl** クリックします。ネットワークに camera ノードが加わり、カメラ越しにシーンビューを見られるようになります。

Lock Camera/Light to View ボタンを押し、ビュー変更に応じてカメラの位置が更新されるようにします。Scene View で **タンブル**、**パン**、**ドリー** してカメラを再配置し、ファー・デュードが左から右に動くようにします。タイムラインをスクラップし、シーケンス全体でカメラが機能していることを確認します。



10 **Lock Camera/Light to View** ボタンを **オフ** にしたら、タンブルしてファー・デュードを見下ろすようにしましょう。**LOP Lights and Camera** シェルフで、**Area Light** ツールを **Ctrl** クリックします。**arealight** ノードをチェーンの終端に追加します。

arealight ノードを選択し、**Base Properties** タブの **Intensity** を 2 に設定します。

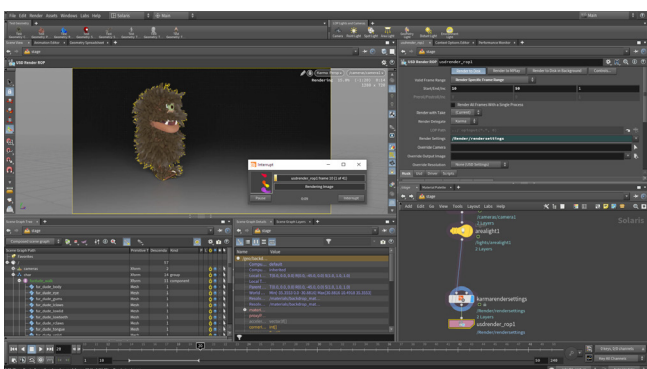


Denoiser をオンにする

11 Persp メニューで **Karma** を選択し、Scene View で Karma を使ってレンダリングします。タイムラインの別フレームに移動すると、Scene View が素早く更新されます。

Karma は USD を使用するよう設計されているので、LOP コンテキストのすべてが USD シングラフに変換されます。Houdini のこの部分からのみ Karma レンダラを使用できます。

レンダリング時によりクリーンな画像を得るには、Nvidia グラフィックカードを使用している場合には、Denoiser をオンにします。**Render** メニューから **Denoiser** をインストールし、**Display Options** バーでそれをオンにしてください。



12 **Tab > Karma** を押し、**Karma Render Settings** と **USD Render ROP** ノードを追加します。それらをチェーンの終端に接続します。**karmarendersettings** を選択して、**Image Output > Filters** タブで **Denoiser** を **nvidia Optix Denoiser** に設定します。**Output Picture** を **\$HIP/render/walk/furdude_walk_-\$F2.exr** に設定します。名前の **\$F** は、レンダリングにフレーム番号を付加するためのもので、**2** はフレーム番号のパディングです。

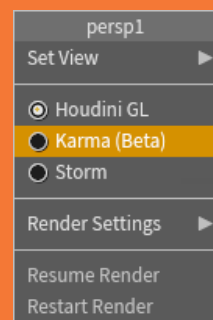
usdrender_rop を選択します。**Start/End/Inc** パラメータを **RMB** クリックし、**Delete Channels** を選択します。**Start** を **10**、**End** を **50** に設定します。**usdrender_rop** ノードを選択します。**Render to Disk** をクリックします。



KARMA レンダラ

Houdini のレンダラである Karma を使用して、シーケンスをレンダリングします。Karma は USD をレンダリングするよう設計されており、これはレンダーデリゲートと呼ばれています。最初は Scene View でレンダリングします。Scene View で **D** を押して表示オプションを表示し、レンダリングをコントロールします。Denoiser をオンにしたり、Pixel Samples を設定したり、Image Resolution を定義することができます。

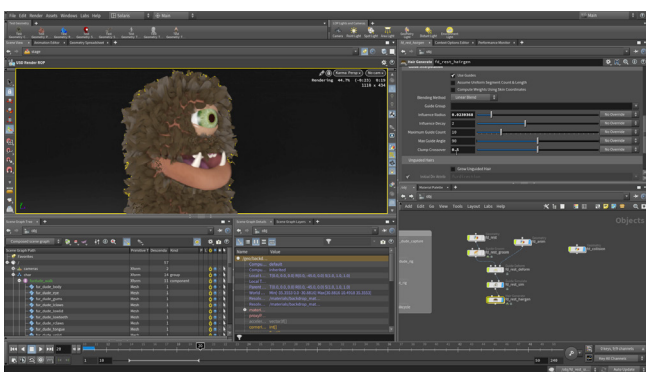
その後 **Karma LOP** をセットアップすると、そのノードのレンダリング設定を使用して最終的な出力を作成し、ディスクに保存できます。





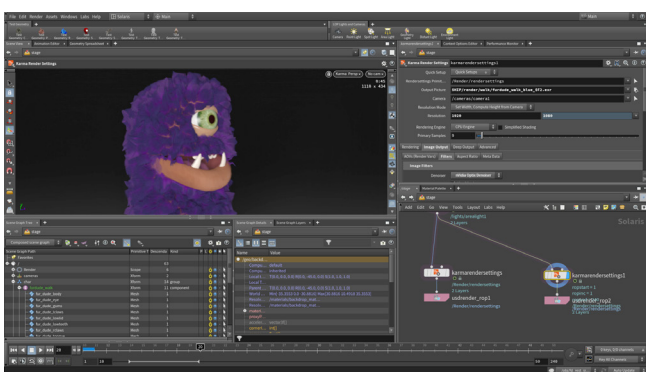
13 完了したら、**Render > Mplay > Load Disk Files** を選択し、レンダリングした画像を開いて最終的なシーケンスを確認します。

その後で別の Karma ノードを分岐させ、解像度とレンダリング設定を上げて最終的なレンダリングを行います。最初は低解像度でテストレンダリングを行い、すべてが希望通りになっていることを確認するようにしてください。



14 ヘアーやファーの設定を調整したい場合は、Scene View を LOP ネットワークにピン留めしてから、オブジェクトレベルに戻り、**Simulate Guides** ノードで **Load from Disk** チェックボックスを**オフ**にします。これで、変更はすべて最終的なレンダリングに反映されます。

ヘアーが短くなり、さらにモジャモジャになりました。好きなように調整してください。完了したら、再キャッシュ化して、**Load from Disk** を再度**オン**にします。



15 Solaris ネットワークに戻って、ファーの色を変更し、再レンダリングすることもできます。

最終レンダリングでは、**解像度**を **1920 x 1080** に上げるなど、画質の設定を変更しましょう。たとえば、**Pixel Samples** を **128**、**Light Sampling Quality** を **16** に設定するとよいでしょう。



まとめ

Houdini の **KineFX** ツールを使用して、Fur Dude キャラクターのリギング、アニメーション、レンダリングを行いました。**キャプチャリグ**を作成し、その上に**アニメーションコントロールリグ**を重ねるなど、さまざまな重要な手順を見てきました。また、キャラクターを **Houdini Digital Asset** にパック化し、一般的な**歩行サイクル**のキーフレームを設定しました。

その後、さまざまなグルーミングツールを使ってファーを追加し、**Karma** でレンダリングしました。ワークフローを一通り実行し、キャラクターのショットを作成できました。より完璧な結果を追求したい場合は、戻って手順の一部を調整し、何度もやり直してください。

冒頭で述べたように、**KineFX** ツールセットの現時点での主な用途は、本レッスンでは取り上げなかったリターゲティングとモーション編集です。**KineFX** のリギングおよびアニメーションツールは進化を続けています。このレッスンでは、今後の **Houdini** のプロシージャルリギングワークフローの可能性を、わずかながら、体験していただきました。

