CHARACTER FOUNDATIONS

KINEFX RIGGING | FUR DUDE

In this lesson, you will rig, animate and add fur to a two legged character named Fur Dude. Starting with existing geometry, you will draw the skeleton, capture the geometry then build rig controls for an animation rig. You will then keyframe a walk cycle and add fur to the surface of the creature.

This lesson uses **KineFX**, Houdini's new SOP-based procedural rigging tools. While these tools are used primarily for retargeting workflows they also include tools for rigging characters and creatures. These tools are still evolving and this lesson offers a taste of what will be currently possible. In future releases, you will see the KineFX and animation workflows expanded and refined.

LESSON GOAL

To rig, animate and add fur to the fur dude creature.

WHAT YOU WILL LEARN

- How to build a skeleton using **KineFX** joints
- How to capture deforming and rigid geometry to the skeleton
- How to wrap up the capture rig into a digital asset
- How to add controls and build an animation rig
- How to animate a walk cycle
- How to add fur to the creature
- How to render using Solaris and Karma.

LESSON COMPATIBILITY

Written for the features in Houdini 19.5+

The steps in this lesson can be completed using the following Houdini Products:

Houdini Core	✔
Houdini FX	✔
Houdini Indie	✔
Houdini Apprentice	✔
Houdini Education	✔

Document Version 2.00 | Nov 2022

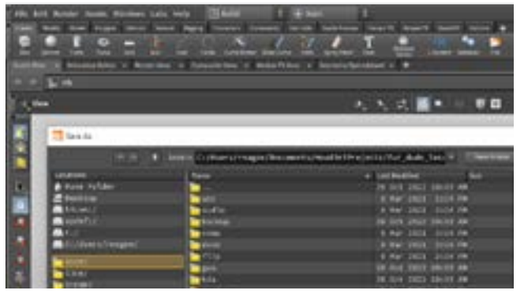
PART ONE

Draw the Skeleton

Start by opening the scene file and reviewing the Fur dude geometry then place joints using the Skeleton tool. This tool will let you create, name and adjust joints to line up with the character you want to animate.

PROJECT FILES

Go to the **Fur Dude tutorial** page on **SideFX.com** where you got this document and download the *furdude_lesson_start* directory. Rename it *furdude_lesson* then put it into the **Houdini Projects** directory.



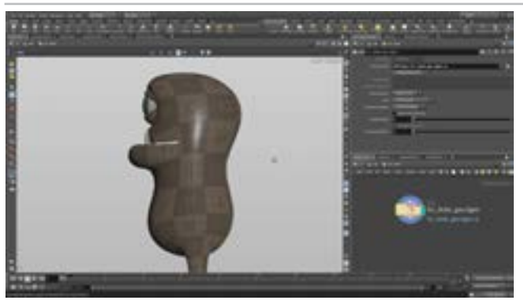
01 Select **File > Set Project.** Find the *furdude_lesson* directory that you downloaded earlier and press **Accept**. This makes this project directory and its sub folders the place for all the files associated with this shot.

Select **File > Open.** You should be looking into the new *furdude_lesson* directory. Open the file name to *furdude_start.hip*. Select **File Save As...** and rename the file *furdude_01.hip*. Click **Accept** to save. This way you can go back to the start file later if you want to redo the lesson.

02 The scene opens with a single object called *fur_dude_rig*. You are going to capture this geometry to bones created using the **KineFX** toolset.

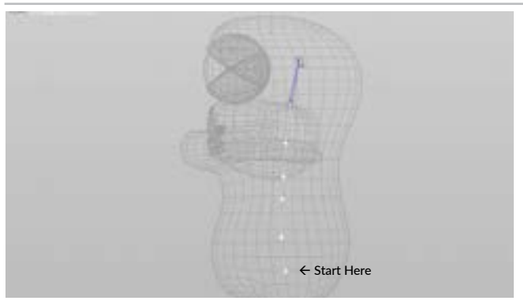
Double-click on the node to dive down to the geometry level. Here you can see the **File** node which is importing the *fur_dude_geo.bgeo* file from disk.

This geometry has some information stored with it such as primitive colors and groups. To see this you can **MMB-click** on the node to see Attributes and Groups listed. You will use the groups later on to help capture the geometry.



03 Move your cursor over the Scene View and press **spacebar-b** to go to a four view layout. From the icon in the top right, turn on **Link Ortho Views**. Now you can pan and zoom in the top, front and right views and they are all synced up.

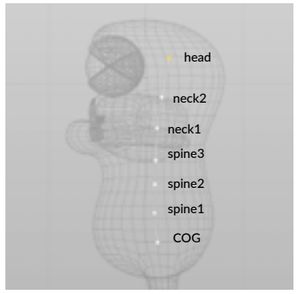
Move your cursor over the *Right* view and press **Spacebar-b** again. This is a good view for drawing joints.



04 In the Network view, press **tab > Skeleton** and place the new node next to the *File* node. Set it's **Display Flag**. Set the file node's **template flag** so that as you are working with the skeleton node the geometry is visible as a grey wireframe.

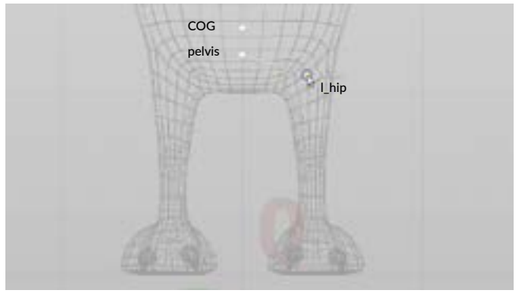
Make sure the Handle tool is selected. In the top bar, set **Joint Placement** to **Freehand**. This will draw on the construction plane without taking the geometry into account. Click to place your first joint just above the leg and then place six more joints as shown in this picture.

MMB-click to stop drawing joints.



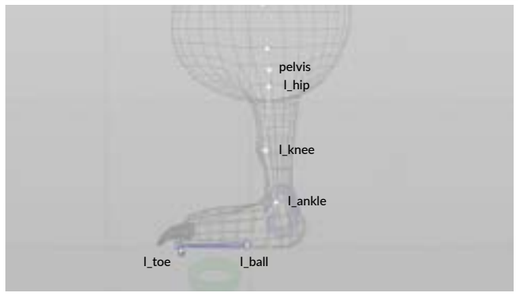
05 In the top bar, set **Mode** to **Modify**. Now you can edit the joints instead of drawing them. Click on the first joint and in the top bar set its **Name** it *COG*.

In the Parameter pane, click on the **+ sign** in the tab area. From this choose **New Pane Tab Type > Animation > Rig Tree**. This brings up a pane that shows your skeleton joints. You can double click on the second joint and name it *spine1*. Now either use the Scene View or the **Rig Tree** to name the rest of the joints as shown.



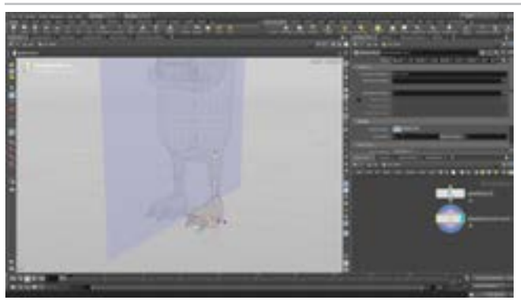
06 In the top bar, set **Mode** back to **Create**. By default it will try to draw from the end of any selected joint. **MMB-click** to stop this drawing action. Now click on the *COG* joint in the Right view then draw a *pelvis* joint just below it.

In the Scene View, press **spacebar-b** to go back to a four view In the Front view, draw a *hip* joint on the left side of the character.



07 Now go back to the Right view and draw the final four joints for the leg as shown here.

Go back to **Modify** mode then rename the joints using either the **Rig Tree** or by selecting and naming the joints in the top bar. After the *pelvis*, all the other joints will have a "*l_*" prefix since these joints will be used for the left leg.



08 In the Scene View, press **tab > Skeleton Mirror**. This creates a mirror copy of all the joints. Go to the Parameter pane and click on the arrow next to **Group**. Select only the leg joints and press enter. Now only the leg is being mirrored.

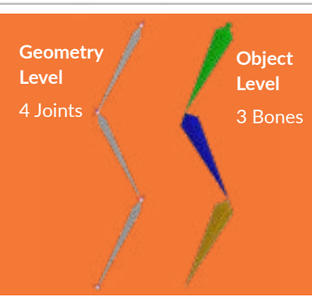
Under naming, set **Find Tokens** to *l_* and **Replace Tokens** with *r_*. Now you have the right leg joint properly named.

Select **File > Save** to save your work so far.

KINEFX VS OBJECT LEVEL RIGGING

The KineFX tools in Houdini offer a joint-based workflow that takes place at the geometry [SOP] level. Houdini's other character workflow is bone-based and in this case you work primarily at the object [OBJ] level.

In the KineFX workflow, joints are basically just points on a curve and this opens up lots of opportunities to use sop-level tools to manipulate rigs. Here you are going to learn about tools designed specifically for rigging characters and creatures.



PART TWO
Capture the Geometry

Rigging a character involves capturing geometry to the skeleton joints in such a way that rotating the joints deforms and bends the geometry. Houdini uses a biharmonic capture method that gives great results with your first capture so that you can start testing out your rig right away. Later you will paint capture weight s to refine the results to work with your character.

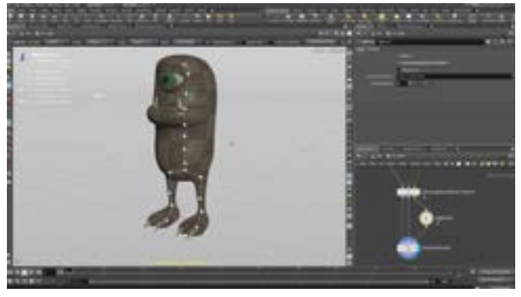
01 Change your Scene view to the perspective using **spacebar-b**. Make your Network View a bit bigger so you have room to work.

In the Network view, press **tab > Joint Capture Biharmonic** and place this node down under the skeleton nodes. Wire *fur_dude_geo* into the *jointcapturebiharmonic* node's **first input**. Wire *skeletonmirror* into the *jointcapturebiharmonic* node's **second and third input** and then set its **Display flag**.

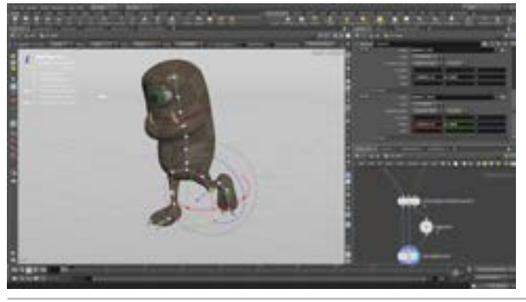
You can now see capture weights on the geometry. You will refine and paint these later to set up the deformation of the geometry.

02 In the Network view, press **tab > Bone Deform**. Wire the **three outputs** of *jointcapturebiharmonic* into the **three inputs**. Set the **Display flag** on *bonedeform*.

In this lesson, you are capturing teeth, claws and the eye which you probably don't want to deform. You will split these out later to capture them using a different method.



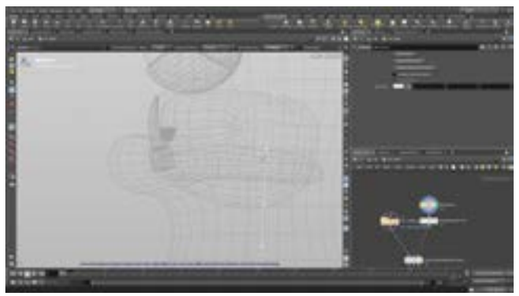
03 Next press **tab > Rig Pose** and place down the node. Move it over the third line connecting *jointcapturebiharmonic* and the third input on *bonedeform* to add it into the chain. This is where animation goes on the rig and the rigpose will be used to rotate joints and can also be used to set keyframes.



04 Select the *rigpose* node and make sure the **Handle** tool is active in the Scene View. **Select** and **rotate** various joints to test out the deformation. You can reset this later so feel free to explore.

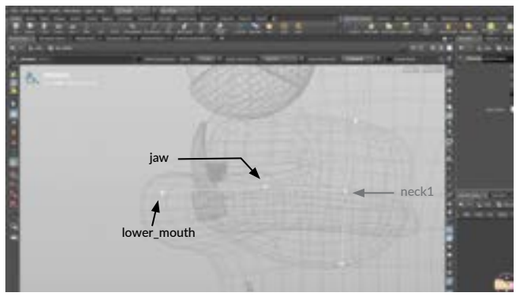
PART THREE
Add More Bones

It would be nice to have more bones in the mouth area. The procedural networks in Houdini let you to go back and add the joints and all the other nodes including the biharmonic capture will update to reflect the changes. This gives you flexibility when first setting up your creature's rig.

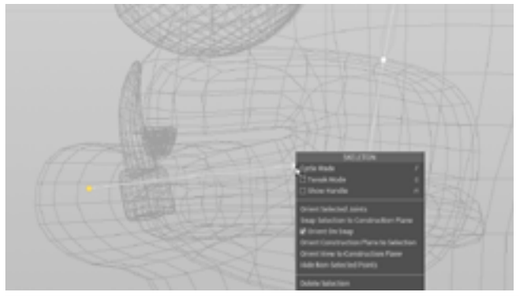


01 In the Network view, set the **Display flag** on the skeleton node and the **Template flag** on the original *File* node. Go to the Right view in the Scene View using **spacebar-b** twice.

Set the **Display Flag** on the *Skeleton* node. Select the *skeleton* node and turn on the **Handle** tool. Set **Mode** to **Create**. This will allow us to add more joints to the skeleton. With KineFX, you can add bones and the procedural nature of the other nodes in our network will allow these changes to be accepted.



02 Click on the *neck1* joint to start drawing then click two joints at the jaw and lower mouth. When you have these joints in place **MMB-click** to stop drawing and change **Mode** back to **Modify**. Now either click on the joints and rename them *jaw* and *lower_mouth* or rename them using the **Rig Tree** view.



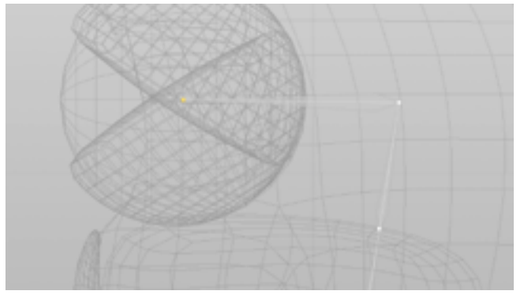
03 You can edit the joint positions in **Modify** mode. Turn **On** the **Tweak** checkbox in the top bar and now you can **click drag** on joints to move them. As you move a joint, all of its children will also move and you may have to adjust them back into place. To avoid this, you can turn on **Child Compensate**.

If you **RMB-click** on a joint, you will see options for splitting, unparenting, copying and pasting joints. You won't need to do those things for this skeleton but it is good to know they are there.

You can even mirror joints here but you are using a different node for that in this network.

04 Set the **Display flag** on the *bonedeform* node and turn off the **Template flag** on the *file* node. The geometry will be reconfigured then captured using the new bones. Select the *rigpose* node and select the **Handle** tool.

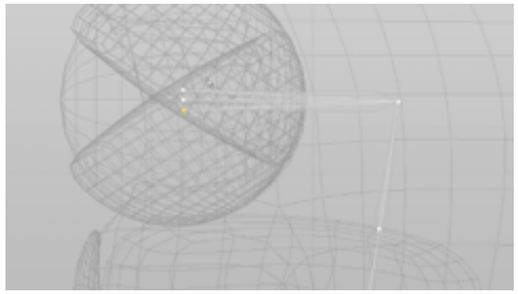
Click on the new *jaw* joint and rotate it down. This works but it manipulates both the lower lip and upper lip. It would be better if it only affected the lower lip. To fix this you will paint the capture weights in a later section of the tutorial.



05 Set the **Display flag back to** the skeleton node and the **Template flag** on the original *File* node.

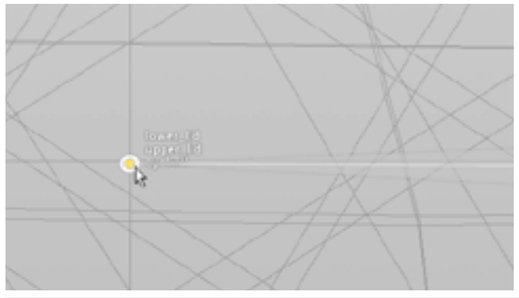
Select the *skeleton* node and turn on the **Handle** tool. Set **Mode** to **Modify**. Click on the **head** joint and using **Tweak Mode,** lower it until it aligns with the eyeball.

Set **Mode** to **Create**. Click on the *head* joint and then click a new joint right in the center of the eyeball. Switch to **Modify** mode and **Name** this joint *eyeball*.



06 Set **Mode** to **Create**. Click on the *head* joint and then click a new joint above the eyeball joint. **MMB-click** to deselect then again click on the *head* joint and then click a new joint below the *eyeball* joint.

Switch to **Modify** mode and **Name** these joints *upper_lid* and *lower_lid*. You will use these joints to rotate the eyelids during animation but they need to also be at the center of the eye.



07 Turn off the Template flag on the geometry. Select the *upper_lid* joint and use **Tweak Mode** to drag it down on top of the *eyeball* joint. Repeat with the *lower_lid* to overlap the *eyeball* joint.

You now have all three joints in the same spot. Later you will attach geometry to each of them individually and then you will be able to animate them independently.

08 Set the **Display flag** on the *bonedeform* node. The geometry will be reconfigured then captured using all the new bones.

You are not able to pose the new joints yet because the geometry hasn't be attached to them in a rigid manner yet. That will come a couple steps down the line after you paint capture weights on the body and the tongue.

JOINT ORIENTATION

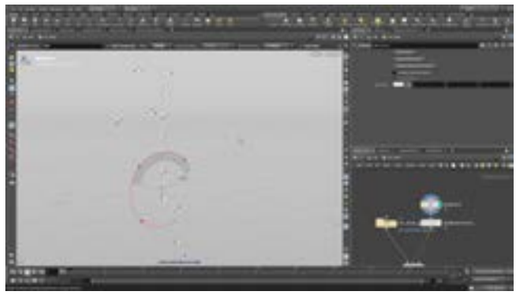
With a joint based system where each joint is a point, the joint orientation is critical because it defines how the joint will rotate during forward kinematics.

KineFX has tools that allow you to point along the chain to the next joint or you can rotate each joint by hand using the **child compensate** option to make sure that re-orienting the joints doesn't affect the other joints in the chain.

PART FOUR

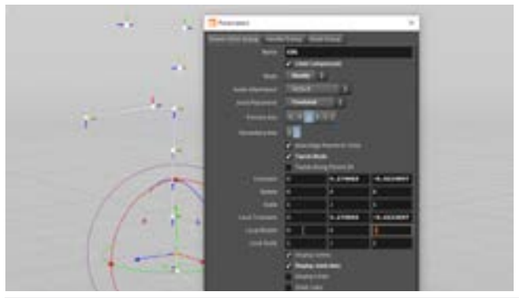
Joint Orientations

When you animate the character, the orientation of the joints plays a significant role in how you manipulate the rig. At this point, you will orient some of the joints by hand and then use a Re-orient joints node to point all the other joints down the -z axis. Sometimes when you evaluate your rig at a later stage, you may need to come back and tweak the orientations.



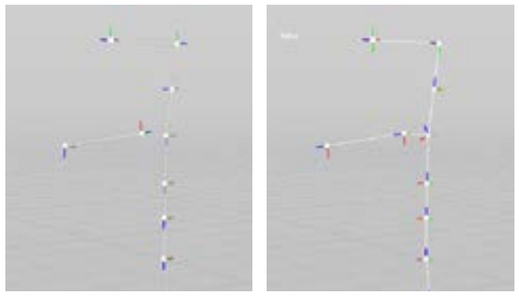
01 Go back and set the **Display flag** on the *skeleton* node. In the Scene View , **RMB-click** and select **Display Joint Axes** to see the joint orientations. Select the *COG* joint and **RMB-click** and select **Show Handle.** Now if you rotate the joint the whole rig goes with it. Press **Ctrl-Z** to undo.

Click on the **Child Compensate** checkbox and now you can rotate the joint without changing the rest of the skeleton. You can use the **Ctrl** key to constrain to 45 degree increments.



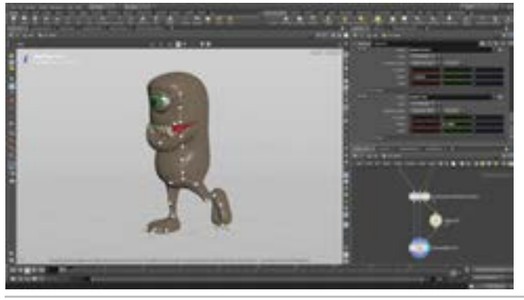
02 In the Scene View, **press p** to bring up the parameters for this joint. To orient the COG with the world, Set **Rotate** and **Local Rotate** to **0, 0, 0**.

Click in empty space to deselect the COG joint then click on the *Pelvis* joint. Set **Rotate** and **Local Rotate** to **0, 0, 0**.



03 After the Skeleton joint, insert a **Orient Joints** node which will by default point the orientations along the positive Z axis. Click on the arrow next to **Orient group** and in the Scene view, select all the joints. **Press Ctrl** and select the *COG*, *pelvis* and *neck1* joints to remove them from the selection. Press **Enter**.

Now all of the joints are oriented along positive Z except for the three deselected joints.



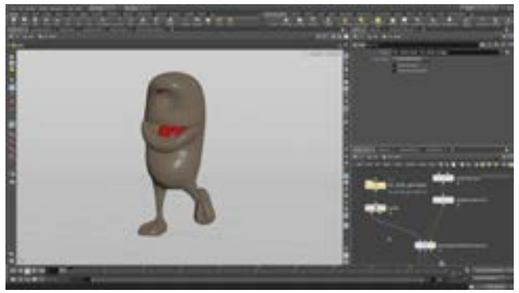
04 Set the display flag back to the **Bone Deform** node to allow all the other nodes to update and accept the new joint orientations.

The results don't look any different than before you oriented the joints but it will affect how you pose and manipulate the character when you are animating.

PART FIVE

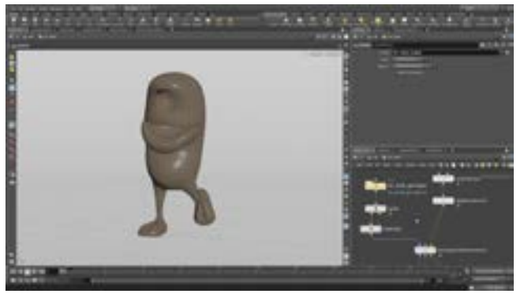
Attach Capture Geometry

Right now you are using the skeleton joints to capture geometry and assign capture weights to each point on your character. To manage this with more control, you can attach curves to the joints that will extend the influence of that joint. This provides a method for achieving your goals as quickly as possible.



01 In the Network View, press **tab > Split** and place the node between the *File* node and the *jointcapturebiharmonic* node. The first output of the *split* node should be wired into the first input of the *jointcapturebiharmonic* node.

Click on the pull down menu in the **Group** field and choose the *fur_dude_body* and *fur_dude_tongue* groups. These are now feeding the first output of the split node and all the remaining pieces such as the eyeball, teeth and claws are going out the second output. You will use a different method to bind those to the skeleton after painting weights on this geometry.



02 In the Network View, press **tab > Visibility** and place the node between the *jointcapturebiharmonic* node and the *bonedeform* node. Click on the pull down menu in the **Group** field and choose the *fur_dude_tongue* group.

This node hides the selected geometry but does not remove it. This ensures that point numbers and primitive numbers don't get changed which is important when working with the paint capture weight tool. You don't want to change this information every time you hide some geometry during this process.

03 **Press S** to get the **Select** tool and **3** to go to Edge selection. Select an edge in the middle of the lower lip on the left then press **Shift-A** and select an edge on the other side.

04 Press **tab > Curve from Edges** and the selected edge will be extracted from the geometry. Rename this node *lower_lip*.

The *curvefromedges* node gets placed in between the *file* and *jointcapturebiharmonic* nodes. Move it to the side and then reconnect the *file* node output to the first input of the *jointcapturebiharmonic* node. This branches the *curvefromedges* node off to the side.

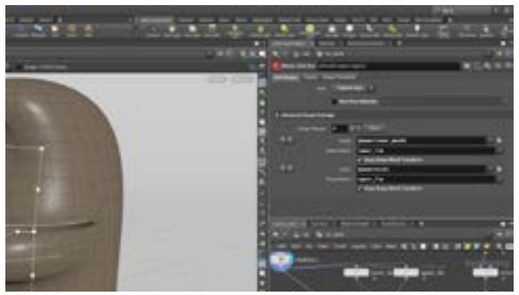
05 **Press S** to get the **Select** tool and **3** to go to Edge selection. Select an edge on the upper lip on the left then press **Shift-A** and select an edge on the other side.

Press **tab > Curve from Edges** and the selected edge will be extracted from the geometry. Rename this node *upper_lip*.

Branch this node off the same as you did with the other one.

06 Press **tab > Merge Packed** and place this node down underneath the two extracted curve nodes then feed the output of those nodes into the *mergepacked* node.

Press **tab > Attach Capture Geo**. Add this node to the network then feed the *skeletonmirror* node into the first input and the *mergepacked* into the third input.



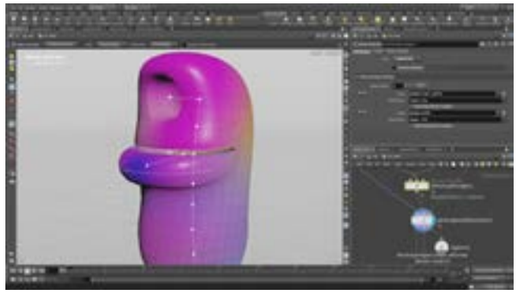
07 Feed the *add_capture_geometry* node's output to the middle input of *jointcapturebiharmonic*. Under **Advanced Shape Settings**, click the **Assign Shapes** plus sign button two times. For the first one, set:

- **Group** to *@name=lower_mouth*
- **Shape Name** to *lower_lip*

And the second one to;

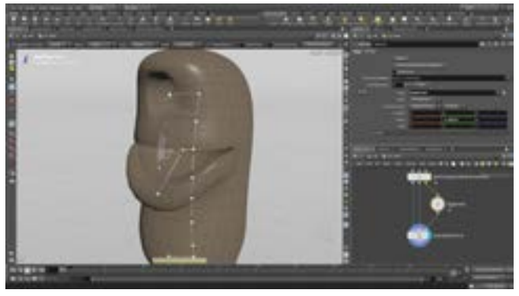
- **Group** to *@name=neck2*
- **Shape Name** to *upper_lip*

Turn **On** keep **Shape World Transform** for both Shapes.



08 Set the **Display Flag** on *jointcapturebiharmonic*. Now **bypass** the *add_capture_geometry* node to show difference in the capture weights.

When this node is turned on, the attached curves are assisting with the capture of geometry to the associated joint. This allows for more control as you set up your character.



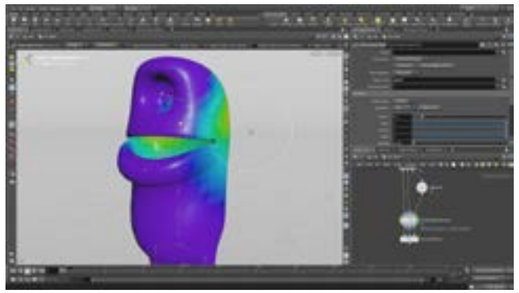
09 Set the **Display flag** on the *bonedeform* node. Click on the *rigpose* node and in the Parameter pane, click on the **Clear** button next to **Transformations**. This will reset the joints. Now with the **Handle** tool active and in the Scene View click on the *jaw* joint. **Rotate** it to lower the lips down.

You can see that this still deforms parts of the belly. To correct this, you need to paint capture weights to reassign capture weights to different joints.

PART SIX

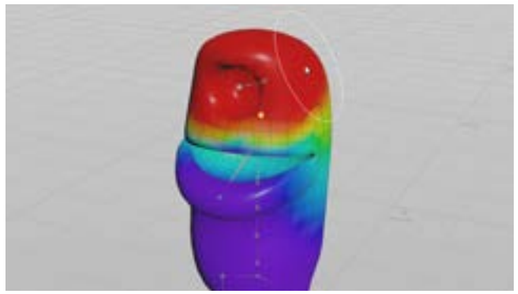
Paint Capture Weights

The biharmonic capture added weights to the character's geometry that are associated with the different skeleton bones. You can now use a new node to adjust the capture weights using a brush workflow. For this creature, the goal will be to get the top of the mouth to not be influenced by the lower mouth joint and to tweak how the feet area is weighted.



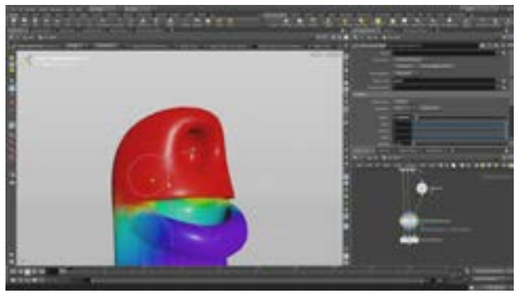
01 In the Network View, press **tab > Joint Capture Paint** and place the node just above the *bonedeform* node with all three connectors getting connected. Set its **Display Flag** then click on the pull down menu next to **Target Joint** and choose the *neck2* joint.

In the Scene view, you will also see a big round paint icon on your cursor which you can use to paint weights. Paint on the head area and this part of the geometry will be captured to the *neck2* joint.



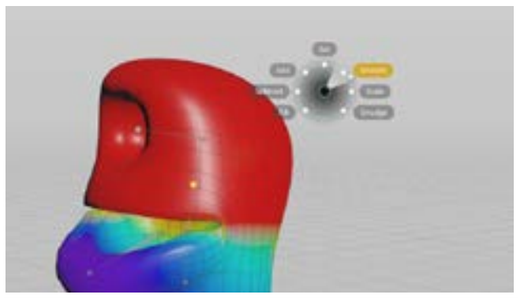
02 You want to capture the head and the upper lip area to the neck2 joint. The geometry will appear red when it is at its strongest.

Use big strokes for the top of the head then use the **Scroll wheel** on your mouse to reduce the brush radius or go to the **Brush** tab in the Parameter pane to change the radius.



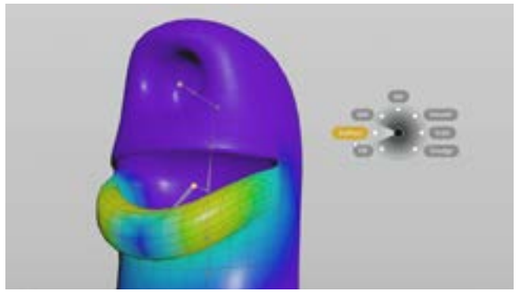
03 Paint on the upper lip to increase the influence of the *neck2* joint on this area. Make sure you are painting the top lip and not the bottom lip. If you mess up, use **Ctrl-Z** to undo your stroke. Tumble around, even inside the mouth to get the upper part of the mouth.

In the **Operation Controls** panel, you can see options to **Display Deformed Geometry**, **Display Joints** and **Display Color**. You can turn these on and off to help you evaluate your capture weights as you paint.



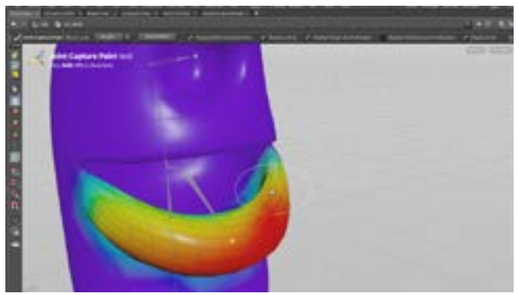
04 **Press F** and choose **Smooth** to smooth out what you have painted here with other capture weights. Make your brush radius a bit bigger to create more smoothing.

Once this is finished, click on the *rigpose* node with the **Handle** tool active and in the Scene View click on the *jaw* joint. Rotate it to lower the lips down. You can see that now the top lip isn't moving but other parts of the head are being rotated.



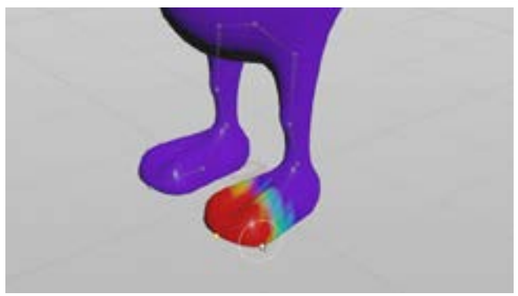
05 Select the *jointcapturepaint* node. Click on the pull down menu in the **Capture Region** field and choose the *jaw* joint. **Press F** and choose **Subtract** to take away influence of this joint on the top of the head and eye area.

You can also use this method to remove the influence of the jaw and the lower_mouth joints on the belly. That way when the mouth moves the belly isn't affected too much.

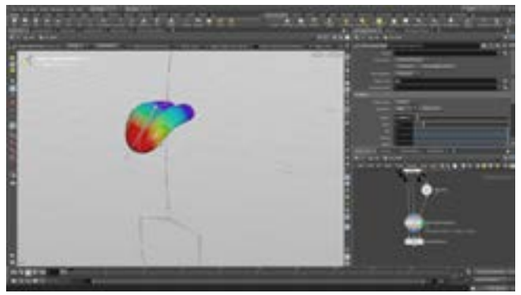


06 From time to time, go back to the and test out the jaw. Once you have it no longer influencing the head area then you are good to go. You can also go back to smoothing to clean things up once you are finished.

You can now paint weights for other joints in the skeleton. The Biharmonic should have done a good job on the legs and feet but you can test the rig using *rigpose* then paint weights to refine the results.



07 Select the toe joints and make sure that the ends of the feet are captured to those joints. Later you will use a reverse setup and having the geometry attached to the toes .



08 On the visibility node set **Apply to Non-Selected Primitives** . Now you only see the tongue.

It should also be weighted fine by default but you can paint weights if you need to. Select the *jointcapturepaint* node and paint weights for the *neck1* and *jaw* joints.

When you are finished then **Bypass** the *visibility* node again.

09 On the visibility node set **Bypass Flag**.

Once this is finished, click on the *rigpose* node with the **Handle** tool active and in the Scene View click on the *jaw* joint. Rotate it to lower the lips down. If you are happy with how all the parts are working then you are ready to capture the rigid geometry.

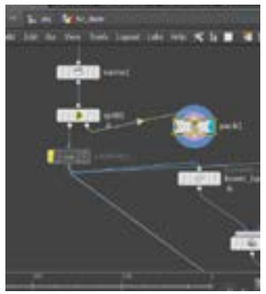
PART SEVEN

Capturing the Rigid Geometry

Earlier you split out the eyes, teeth and claw geometry. Now you are going to pack this geometry then assign each part to a joint using the capture packed geometry node. This is the equivalent of parenting each object to the skeleton since parenting isn't an option at the geometry level when using KineFX.

01 In the Network View, press **tab > Name from Groups** and place the node between the *File* node and the *split* node. Change the **Group Mask** to *. This will turn all of the groups into name attributes.

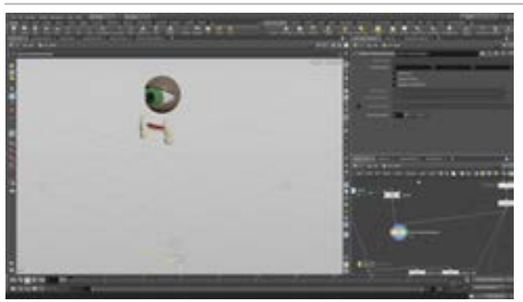
To see this, click on the **Geometry Spreadsheet** Tab and click on the **Primitives** button at the top left. Scroll down to see that there is a name attribute and the group names are used as the values for all of the geometry.



02 In the Network View, press **tab > Pack** and place the node down to the right of the *split* node. Wire the second output of the *split* node into the *pack* node and set its **Display Flag**. You can see the body parts in the Scene View.

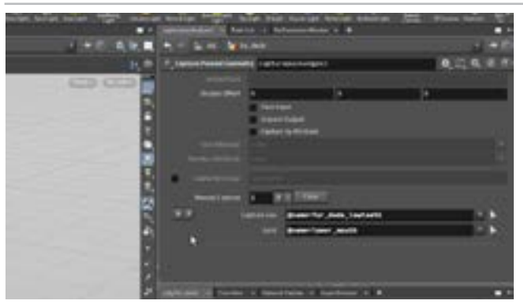
Turn the **Path Attribute** checkbox to **Off** then the **Name Attribute** checkbox to **On** and set **Transfer Attributes** to **name**.

Using the arrow in the top right of the Network view, choose **Split** In the **Geometry Spreadsheet,** you can see the 8 packed primitives. These can now be captured to the skeleton as rigid geometry.



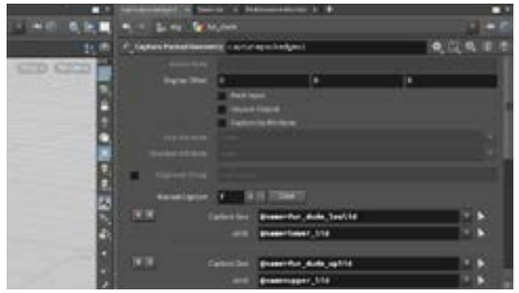
03 In the Network View, press **tab > Capture Packed Geometry** and place the node down under the *pack* node. Wire the output of the *pack* node into the first input of the *capturepackedgeo* node. Next wire the output of the skeletonmirror node into the second input of the *capturepackedgeo* node then set the Display flag on the *capturepackedgeo* node.

Nothing has happened yet - you need to associate the bones and the geometry you want to capture.



04 In the Parameter pane, click on the **plus sign** next to **Manual Capture**. Click on the **arrow** next to **Capture Geo** and in the Scene view, select the *lower teeth*. One click is all you need because the lower teeth are one of your packed groups. Press **Enter** to accept.

Now click on the **arrow** next to Joint. The geometry disappears and the skeleton is shown as lines and points. Select the *lower_mouth* joint and make sure your cursor is over the Scene View then press **Enter** to accept.



05 Click the plus sign and then the **arrow** next to **Capture Geo** and in the Scene view, shift select the *fur_dude_upteeth* and *fur_dude_gums* . Press **Enter** to accept. Now click on the **arrow** next to **Joint**. Select the *neck2* joint and press **Enter**.

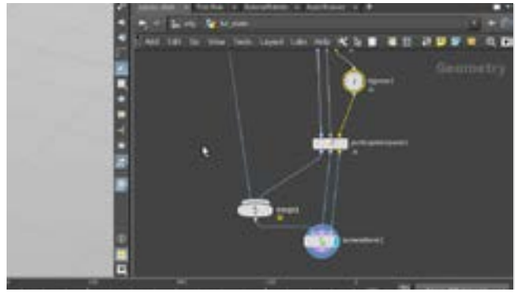
Repeat these steps two more times to associate the following:

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

06 Wire the output of the *capturepackedgeo* **into the first input of the** *bonedeform* **node**. Set the Display flag on the *bonedeform* node.

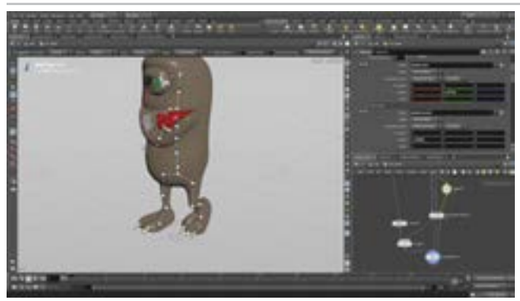
Select the *rigpose* node and then select and rotate joints in the Scene view. You can see that the geometry is being captured to the joints without any deformation.

The eye and eyelid joints overlap therefore when you click a little menu pops up letting you choose - you may need to click a few times to get the joint you need. You can then use them to rotate the eye and the eyelids separately.



07 In the Network View, press **tab > Merge** and place the node down in-between the *capturepackedgeo* and *bonedeform*. Wire the *capturelayerpaint* node into the *merge* node. On the *merge* node, press the blue up arrow next to *capturelayerpaint* to reorder the inputs.

Now everything is deforming but the body and tongue are grey while the packed geometry has its colors. There is also an error on the merge node because the one side has color (Cd) as a primitive attribute and the other side has it as a point attribute.



08 In the Network View, press **tab > Unpack** and place the node down between the *capturepackedgeo* and the *merge* nodes. Set **Iterations** to **2** and **Transfer Attributes** to * **^Cd**. The * grabs all the capture attributes and the **^Cd** make sure that you do not remove the original color attribute. Now both sides are using point colors and the rig looks correct.

Now you can select the *rigpose* node and then select and rotate joints in the Scene view. You can see that all the captured geometry is now working together.

Save your work.

FLATTENING THE NETWORK

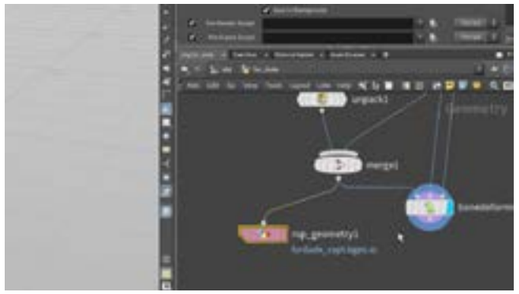
The network that you have created to capture the geometry, paint weights and prepare the geometry for animation works well but can take a little time to update when changes are made.

To create a more efficient rig, you will flatten the network into a single file that has capture weights stored in the geometry. This file will deform efficiently when fed into a bone deform sop when accompanied by a skeleton with the same bones that were used to capture the geometry in the first place.

PART EIGHT

Create Capture Rig Digital Asset

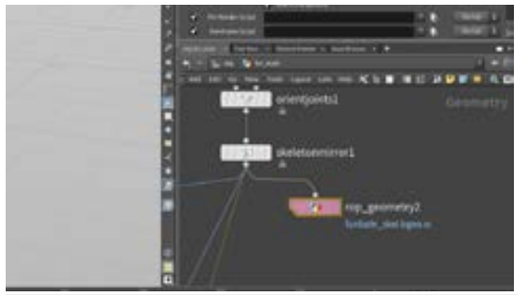
The captured geometry and the skeleton can now be wrapped up into a digital asset that can be used as the foundation for the animation rig. To start, you will export the geometry with its capture weights and the skeleton then embed these into the digital asset file. This will help make the capture rig more efficient when animating the character.



01 In the Network View, press **tab > ROP Geometry Output** and place the node down under the *merge* node. Wire the output of the *merge* node into the input of the *rop_geometry* node. In the Parameter pane, set **Output File** to:

$HIP/geo/furdude_capt.bgeo.sc

Click **Save to Disk** to save the geometry to your *geo* directory. This geometry looks like the geometry you imported at the beginning of the lesson but now it includes important data such as capture attributes that allow it to deform.



02 In the Network View, press **tab > ROP Geometry Output** and place the node down next to the *skeletonmirror* node. Wire the output of the *skeletonmirror* node into the input of the *rop_geometry* node. In the Parameter pane, set **Output File** to:

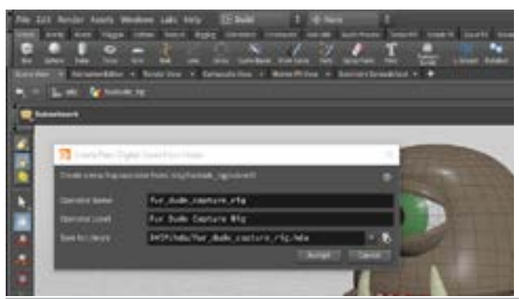
$HIP/geo/furdude_skel.bgeo.sc

Click **Save to Disk** to save the geometry to your *geo* directory. This geometry represents your skeleton and can be used to build up an animation rig asset.

03 Go back up to the Object level. Rename the object to *fur_dude_capture*. Turn of it's **Display flag**.

Click on the **File** button on the **Create** shelf. Click on *$HIP* then go into the *geo* directory and select the *furdude_capt.bgeo.sc* file. Press **Enter** to place it at the origin then **rename** the new object node to *furdude_rig*. **Double click** to dive into this object. **Alt-drag** on the *File* node to create a second one. Set its **Geometry File** to: *$HIP/geo/furdude_skel.bgeo.sc*

Rename it *furdude_skel.bgeo* then set the **Display Flag** on the *furdude_capt.bgeo* node.



04 Select the two *File* nodes then from the **Assets** menu select **New Digital Asset from Selection**. Set the following:

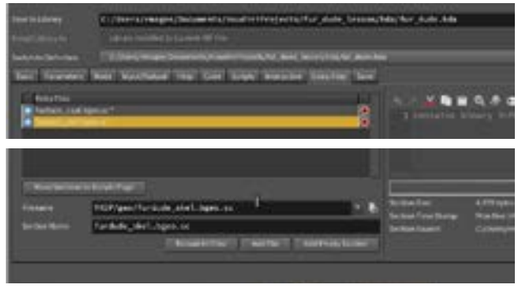
- **Operator Name** to *fur_dude_capture_rig*
- **Operator Label** to *Fur Dude Capture Rig*
- **Save to Library** to: *$HIP/hda/fur_dude.hda*

Click **Accept** to create the HDA file. An Edit Operator Type Properties window pops up. Set **Maximum Outputs** to **3** and click **Accept**. **Rename** the subnet to *fur_dude_capture_rig*.

05 **Double click** on this node to dive into the subnet. Press **tab > Output** and place an output node beneath the *fur_dude_geo_capt File* node. Wire the *File* node into the *output* node. Rename it *CaptureGeo*.

Alt-drag twice to create two new Output nodes. Name the second one *RestSkeleton* and set its **Output Index** to **1**. Name the third one *AnimSkeleton* and set its **Output Index** to **2**. Wire the *fur_dude_skel.bgeo File* node into both the second and third *output* nodes.

Set the **Display Flag** on the *CaptureGeo* output node.

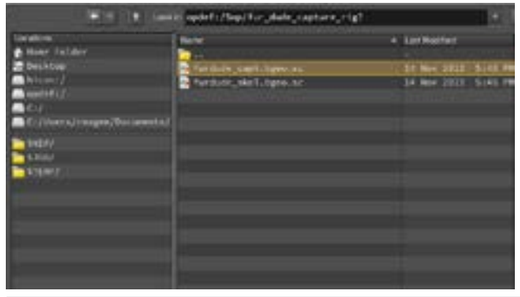


06 From the **Asset** menu, select **Edit Asset Properties > Fur Dude Capture Rig**. This opens up the Operator Type Properties window. Click on the **Extra Files** tab.

Click on the chooser button next to **Filename** in the lower left and navigate to *$HIP/geo/fur_dude_capt.bgeo.sc*. Click **Accept**. Then click **Add File**.

Repeat these steps for the *fur_dude_skel.bgeo.sc* file. Now you have placed these files inside the digital asset file which will make them easier to share with other people as a complete package.

Click **Accept** to finish.

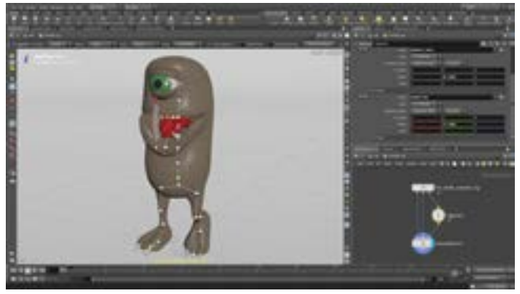


07 On the first file node, click on the **Chooser** icon next to **Geometry File** then click on *opdef:/* in the **Locations** sidebar then double click on the *sop* directory then again on the *fur_dude_capture_rig* folder. Select the *fur_dude_capt.bgeo.sc* file then press **Accept**. This creates the following opdef expression:

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

Repeat these steps for the *fur_dude_skel.bgeo.sc File* node.

From the **Assets** menu, select **Lock Asset > Fur Dude Capture Rig**. **Save Changes** to protect the contents of this asset. Later you can unlock it and update these files if needed.



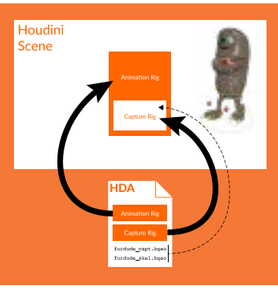
08 Go up one level where you will see the capture rig with three outputs. Press **tab > Bone Deform** and place this node underneath. Wire the three outputs of the *fur_dude_capt_rig* into the three inputs of the *bonedeform* node. Now add a rigpose node in the middle of the third chain. Set the **Display Flag** on the *bonedeform* node.

Select the *rigpose* node and make sure the **Handle** tool is active. You can again see all the skeleton joints. **Pose** the skeleton to test that the deformations are working in the same way as before.

HDA

This network of nodes was saved as a Houdini Digital Asset or HDA. This is a file on disk that is easy to share. The capture rig is an asset that is referenced off disk and in the next section, you will build the animation rig with the capture rig nested inside it. Both of them along with the bgeo files will be stored in a single HDA file.

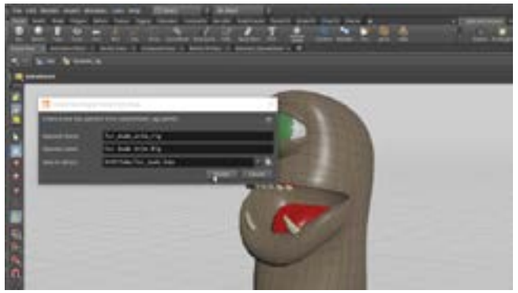
Changes made to the HDA file on disk will update instances of the asset being worked on by animators in their scene files.



PART NINE

Create the Animation Rig Asset

You are now going to create a second digital asset that has the capture rig nested inside it. This new asset will be the one that can be animated to create the final motion of the character. This new asset will contain all of the rigging tools such as inverse kinematics and aim constraints that assist with animation. In order to test these controls as you add them, you will set up a test version of the rig that is locked and visible in a second Scene view pane.

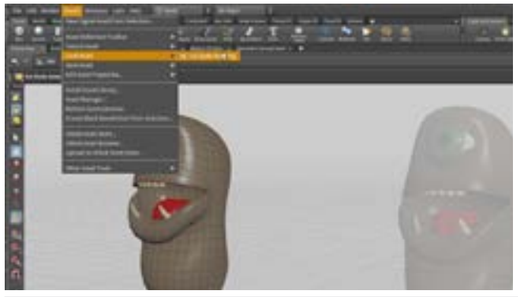


01 Select the three nodes then from the **Assets** menu select **New Digital Asset from Selection**. Set the following:

- **Operator Name** to *fur_dude_anim_rig*
- **Operator Label** to *Fur Dude Anim Rig*

For **Save to Library**, click on the browse button then click on $HIP the double click into the HDA directory. Select the fur_dude.hda file then click **Accept**. It is now set to: *$HIP/hda/fur_dude.hda*

Click **Accept** then **Accept** again in the Type Properties panel. This adds the new asset definition to the same HDA file. **Rename** the subnet to *fur_dude_anim_rig*.



02 Go to the Object level. **Alt drag** on the *furdude_rig* node to create a new geometry node and name it *test_rig*. Press t and move the test rig over to the left.

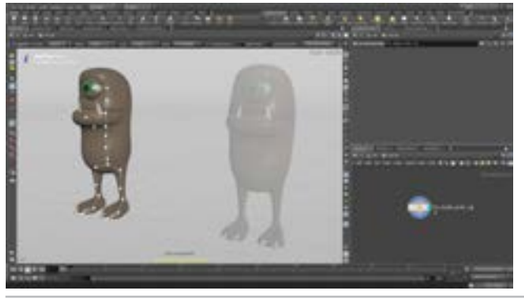
Double click to dive into it. Select the *fur_dude_anim_rig* node and from the **Asset** menu, select **Lock Asset > Fur Dude Anim Rig.**

Go back up one level. You now have two versions of the rig - the *test_rig* is locked and shows you what it is like to interact with the completed asset. Right now you have nothing to interact with. You will fix that soon.

Click in the Network editor, and press **Ctrl 1** to set a quickmark.

03 Navigate back to the *furdude_rig* object then into *fur_dude_anim_rig*. Click in the Network editor, and press **Ctrl 2** to set a quickmark. Now you will be able to quickly jump back and forth between these networks.

Select *rigpose* then click on the **Clear** button next to **Transformations**. Go to the **Asset** menu, select **Save Asset > Fur Dude Anim Rig.** From the Asset menu, choose **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Node** tab and set **Default State** to: `kinefx__rigpose`. Click **Accept**.



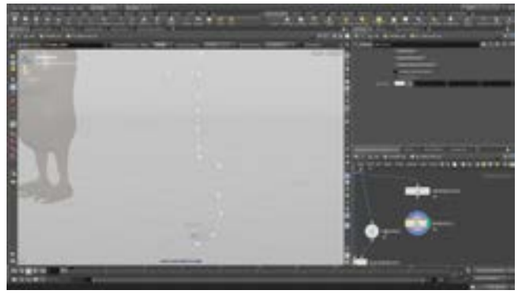
04 In the Network view, **press 1** to navigate back to the **test_rig**. In the Scene View on the left, expand the toolbar and click on the **Handle** tool.

You will now see joints displayed on the *test* rig. If you click on any of them, you will see that they can't be changed because their parameters have not been promoted to the asset. This is the beginning of building an animatable interface for this character.

PART TEN

Add More Control Joints

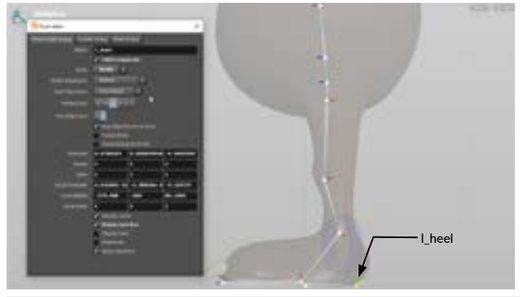
To provide more flexibility with the control rig, you can add joints such as a root joint for the whole skeleton, heel joints for a reverse foot setup or a look-at point for you to target with your eyeball. These joints will have the same names as the ones in the original rest skeleton and that will ensure that they are used to drive the motion on the character.



01 In the Network view, **press 2** to navigate back to the **fur_dude_anim_rig** and set its **Display flag**. Set the **Template flag** on the *fur_dude_capture_rig* node.

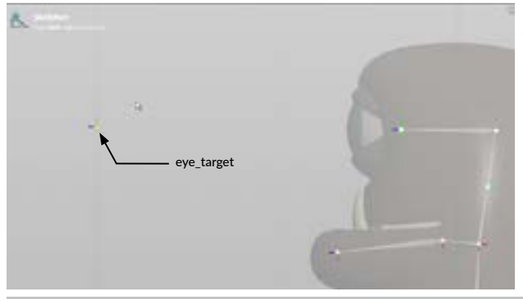
Branch off a **Delete Joints** node from the third output of the *fur_dude_capture_rig* node. Click on the arrow next to group then select fur dude's right leg joints. Press **Enter**.

Add a **Skeleton** node. Change to **Create Mode**. Turn on **Grid snapping** and add a point to the origin. Change back to **Modify** mode and rename the new joint to *furdude_main*.



02 Go to right view. Go back to **Create** model and **MMB-click** to break the connection to the main joint. Click on the *l_toe* joint then add a new joint where the heel would be.

Change back to **Modify** mode and rename the new joint to *l_heel*. Press **p** and with **Child Compensate** set to **On**, set the **Rotate** to **0, 0, 0**.



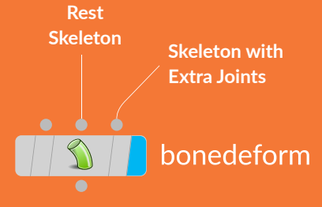
03 Pan to the area around the eye. Go back to **Create** model and **MMB-click** to any connections. Now add a new joint out in front of the eye.

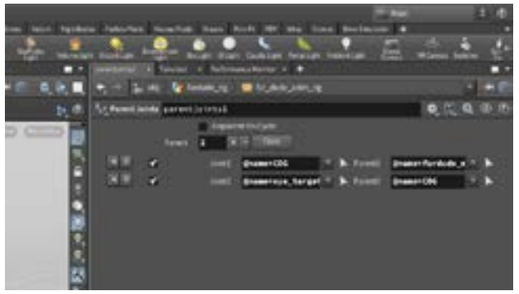
Change back to **Modify** mode and rename the new joint to *eye_target*. Press **p** and make sure that **Rotate** is set to **0, 0, 0**.

ADDING EXTRA JOINTS TO THE RIG

These extra joints were added to the stream of nodes on the far right while the middle rest skeleton will continue with the original joints. When fed into the bone deform, the extra "phantom" joints are ignored and only the original joints determine the final output of the rig.

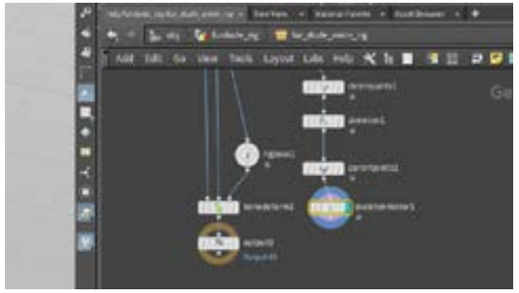
If you were to feed these into the middle rest skeleton input on bone deform you would get an error because these extra joints would not have corresponding capture weights in the incoming geometry.





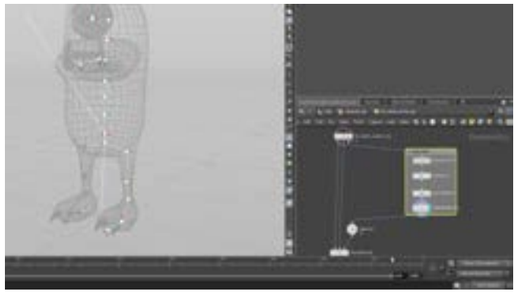
04 Add a **Parent Joints** node. Click the **plus sign** twice. Click on the arrow next to **Joint1** and in the scene view click on the *COG* joint. Press **Enter** (with your cursor over the Scene View) to accept. Now use the arrow next to **Parent1** and select the *furdude_main* joint.

For the second entry, click on the arrow next to **Joint2** and in the scene view click on the *eye_target* joint. Press **Enter** (with your cursor over the Scene View) to accept. Now use the arrow next to **Parent2** and select the *COG* joint.



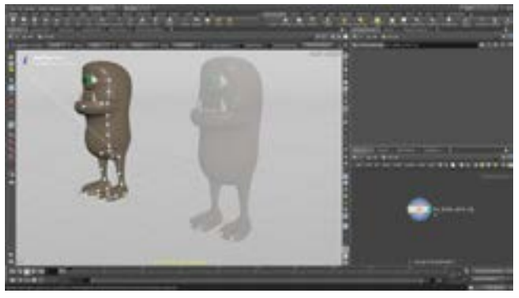
05 Add a **Skeleton Mirror** node to the chain. This creates a mirror copy of all the joints. Go to the Parameter pane and click on the arrow next to **Group**. Select only the left leg joints, including the new *l_heel* joint,and press **Enter**. Now only the leg is being mirrored.

Under naming, set **Find Tokens** to *l_* and **Replace Tokens** with *r_*. Now you have the right leg joints properly named.



06 Feed the skeleton mirror node into the *rigpose* node. Set the **Display flag** on the *bonedeform* node. The claws on the foot appear to have flipped. Go back to the skeleton joint, select the *l_toe* and **press p** to bring up the parameters. Set **Rotate** to **0, 0, -90**.

Select the four nodes you used to add joints and click on the **Add Network box** button. Position the box and center the nodes. Click on the box's title and enter *Add Joints*.



07 From the **Assets** menu, select **Save Asset > Fur Dude Anim Rig**. This saves the changes into the asset definition which will update on the *test_rig*. You still can't do anything with the *test_rig* because there are no parameters promoted to the top level.

You are now going to setup the main controls and promote parameters to start bringing the *test_rig* to life.

WHAT IS THE ROLE OF THE TEST RIG?

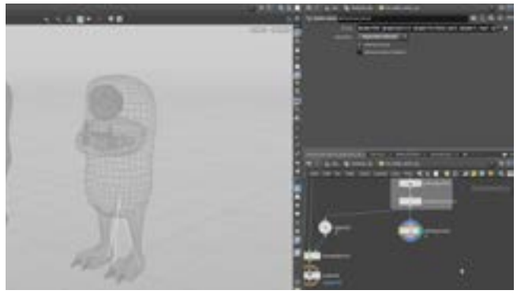
The rig you are working on is an unlocked asset which gives you the ability to work with all the joints inside the asset. But when the asset is published to an animator, they can only work with parameters that have been promoted to the top level of the character.

The *test_rig* is a second locked version of the asset and until you start promoting parameters and building controls, you will not be able to manipulate it. That is what makes it a great tool for verifying that the asset is ready for animation - if you can't work with the test rig then the animator will not be able to pose the character.

PART ELEVEN

The Main Controls

To add kinematics, you need to break the current hierarchy where the feet are under the COG. You can break off a few joints and reparent them to build the hierarchy you need. This reparenting happens off to the side then you will blend the results back into the original skeleton hierarchy which is important to make sure the bone deform functions properly.

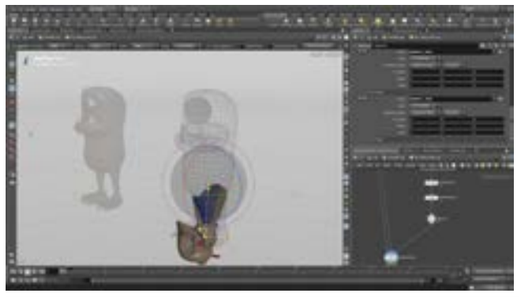


01 Branch off a **Delete joints** node from the *skeletonmirror* node. Click on the arrow beside Group and in the scene view select the *furdude_main*, the *COG*, the *pelvis*, the *l_heel* and *r_heel* joints. Press **Enter** then set **Operation** to **Delete Non-Selected**.

02 Add a **Parent Joints** node after the *deletejoints* node. Click on the **+ sign** to add a joint listing. Click on the arrow next to **Joint1** and in the scene view select the two *heel* joints. Press **Enter**.

Click on the arrow next to **Parent1** and in the scene view select the *furdude_main* joint. Press **Enter**.

Now the heel joints are free from the *COG* and can be used to drive the IK on the legs down the line. But all of the joints are parented to the *furdude_main* joint except for the *pelvis* joint which is parented to the *COG* joint.



03 Wire the *parentjoints* node into the *rigpose* node. Clear any existing joints from this node then set the **Display flag** on it. Now click on the *furdude_main* joint, the *COG* joint, the *pelvis* joint then the two *heel* joints.

If you now set the **Display flag** on the *bonedeform* node everything is mixed up because most of the joints are now missing. This can be fixed with a skeleton blend.

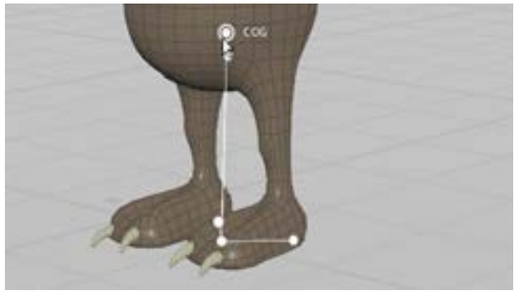
04 Add a **Skeleton Blend** node into the Network editor. Wire the *skeletonmirror* node into the left input and the *rigpose* into the right input. Then wire the *skeletonblend* node into the third input of the *bonedeform* node.

In the *skeletonblend* Parameter pane, set the **World Space** checkbox to **On** and *weight1* to **1**.

CONTROL GEOMETRY

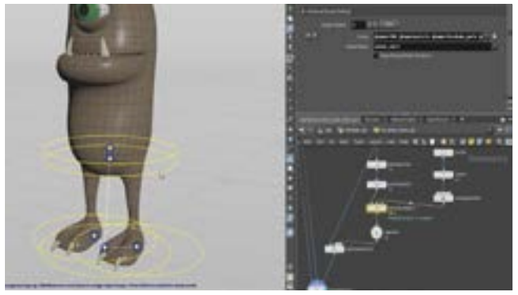
When you add a joint to a rigpose, it can be promoted to the top level of the asset to select and animate. The **Attach Control Geometry** node lets you assign geometry you build to different parts of the rig to make it easier to select joints for manipulation. You can create any shape you want for these controls.

A good example where this will help is the eyeball and the two lid joints which overlap. Control Geometry will make it easier to select these when you set up that part of the rig. For now you will use it for the main controls.



05 From the **Assets** menu, select **Save Asset > Fur Dude Anim Rig**. This saves the current setup. In the Network view, **Press 1** to navigate back to the **test_rig** where you can see that only the five joints listed in the rigpose are visible.

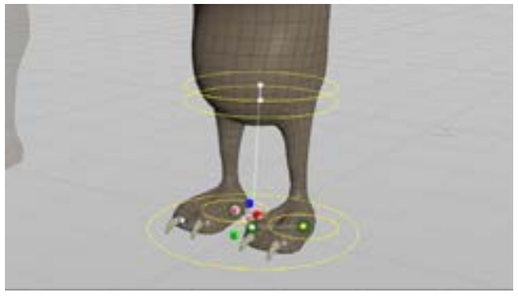
You cannot select and move them because the parameters haven't been promoted yet.



06 In the Network view, **press 2** to navigate back to the **fur_dude_anim_rig**. Add a **Circle** node to the network. Set **Orientation** to **XZ**. Set **Uniform Scale** to **0.2**. Set **Divisions** to **36** and **Arc Type** to **Open**. Add a **Color** node and set **Color** to **yellow**.

Follow with a **Merge Packed** node and set **Name 1** to *circle_ctrl*.

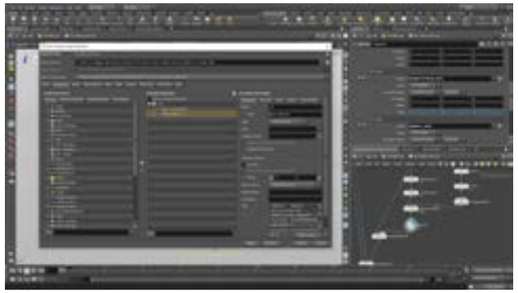
Place an **Attach Joint Geometry** node between the *parent* node and the *rigpose* node. Wire the *mergepacked* node into the second input. Set the **Display Flag** on this node and press **Enter** in the scene view. Select all the visible joints then **press G** and use your scroll wheel to find the *circle_ctrl* geometry. It gets assigned to all the joints.



07 In the Operation Control bar at the top, change the **Mode** to **Tweak Shapes**.

Select the *COG* and *Pelvis* joints then **press G** to bring up a transform handle. **Press E** to get the scale handle then click drag on the middle handle to scale in all three directions until these controls are a bit smaller (around 0.67 in the parameter pane).

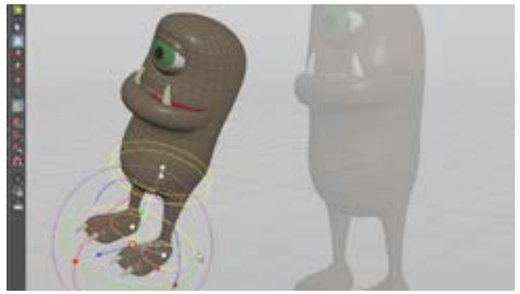
Select the two *heel* joints then **press G** to bring up a transform handle. **Press E** to get the scale handle then click drag on the middle handle to scale in all three directions until the heel controls are much smaller (around 0.3 in the parameter pane).



08 Select the *rigpose* node then select the four joints using the control geometry. To make this work on the test rig you need to promote parameters.

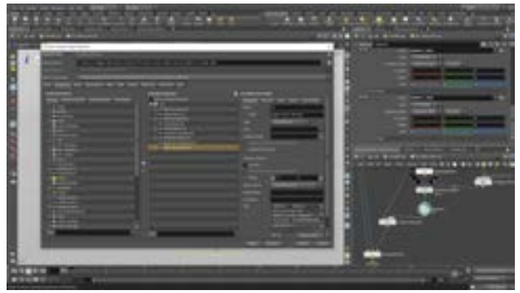
From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Parameters** tab.

On the *rigpose* node go to *furdude_main*, **RMB-click** on **Scale** and choose **Lock Parameter**. Now drag the on **Translate** and move it over to the Parameter list under *root*. Set it's **Label** to *Main Translate*. Repeat for the **Rotate** parameters and name them *Main Rotate*. Click **Accept** to finish and save the results to the asset.



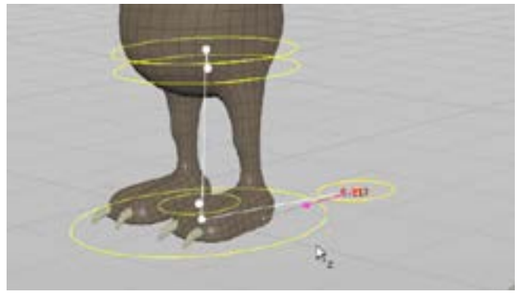
09 In the Network view, **press 1** to navigate back to the **test_rig**. The *test rig* now updates to show the new controls - as long as you have the **Handle** tool active. Select the *furdude_main* joint using its control geometry and now a transform handle appears that you can use to move the rig around.

Undo when you finish to put it back at the starting point..



10 In the Network view, **press 2** to navigate back to the **fur_dude_anim_rig**. From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Drag the **Translate** and **Rotate** for the *COG*, *l_heel* and *r_heel joints* and the **Rotate** for the *pelvis*. In each case, lock the unused **Scale** or **Translate** (for the pelvis) parameters for each joint.

Click **Accept** to finish and **Save** the results to the asset.



11 In the Network view, **press 1** to navigate back to the **test_rig** which has been updated to show the new controls. Select the *COG* and *heel* joints using the control geometry and transform the parts around.

Undo when you finish to put all the parts back to their original location.

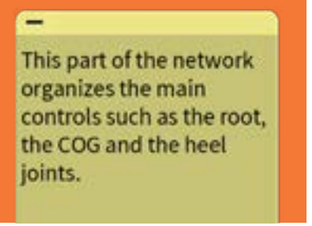
12 In the Network view, **press 2** to navigate back to the **fur_dude_anim_rig**. Select the nodes you used to set up the main controls and click on the **Add Network box** button. Position the box and center the nodes. Click on the box's title and enter *Main Controls*.

From the **Assets** menu, select **Save Asset > Fur Dude Anim Rig**. This has no effect on the rig but keeps the asset up-to-date. You may want to **Save** your Scene file too.

ORGANIZING YOUR NETWORK

Lining up nodes and adding **network boxes** are extra steps that are worth the added effort. The more organized your network is, the easier it is for you to work with later or for others to read what your intentions are.

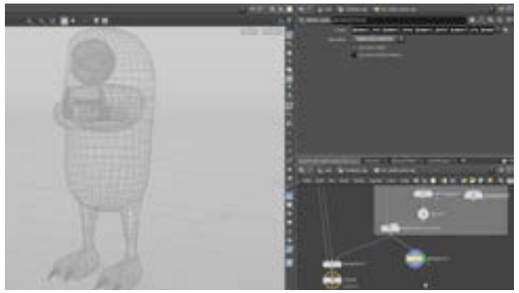
You can also add comments to each node and display them in the network or you can use **sticky notes** to explain larger blocks of nodes. Communication is always beneficial when creating networks in a team setting.



PART TWELVE

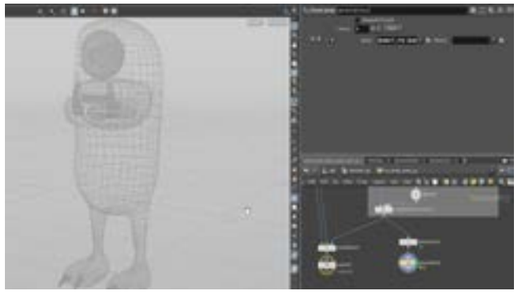
Inverse Kinematics for the Legs

To animate a character, Inverse Kinematics allow you to set up the legs so that moving either the feet or the hips causes the knee to bend appropriately. You are again going to pull out some joints from the main skeleton and set them up using KineFX. You will again blend the results back into the original hierarchy.

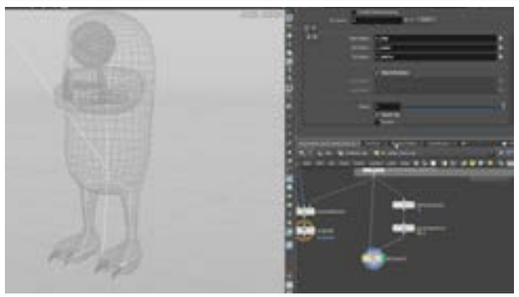


01 Rename the *skeletonblend* node to *skeletonblend_controls*. You are going to use these nodes a lot and it will be helpful to be able to identify them.

Branch off a **Delete joints** node from the *skeletonblend_controls* node and set its **Display flag**. Click on the arrow beside Group and in the scene view select the left and right *hip, knee and ankle* joints. Press **Enter** then set **Operation** to **Delete Non-Selected**.



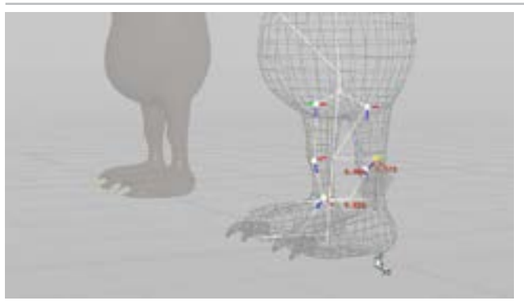
02 Branch a **Parent Joints** node and set its **Display flag**. Click the **+ sign** to add a Joint and set **Joint1** to *. Leave **Parent1** left blank. Now all of the joints are unconnected. This will leave you free to use them independently.



03 Place an **IK Chains** node into the Network editor. Feed the *skeletonblend_controls* into the first input and the *parentjoints* node into the second one. Set its **Display Flag**.

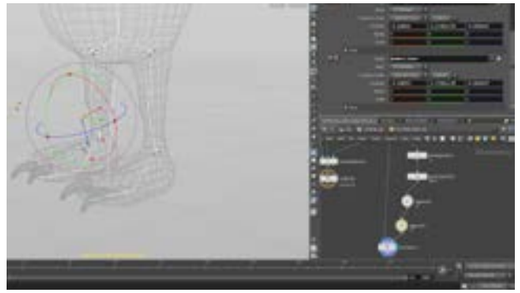
In the Parameter pane, click on the **+ sign** and in the folder click on the arrow next to **Root name**. Select the *l_hip* joint and press Enter (with your cursor over the Scene View). Set the **Mid Name** to *l_knee* and the **Tip Name** to *l_ankle*. You can use the arrow to select the joints or just type in the name. Set **Match by Name** to **On** and **Blend** to **1** then set **Orient Tip** to **On**.

Click the **+ sign** again and do the same for the right leg.



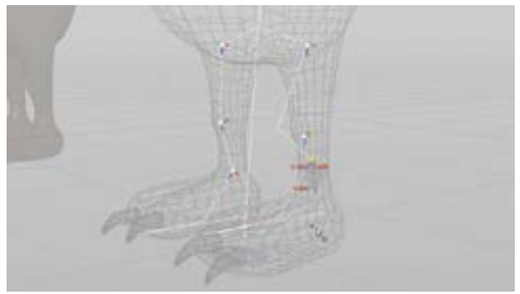
04 Add a **Rig Pose** node in-between the *parent* node and the *ikchains* node. Click on the *ankle* joints and move them around to see the Inverse Kinematics at work. Because you used **Match by Name** the ankle joints are acting as the end effectors for the IK chain.

You will see some flipping as you move the ankles. This is because the knee joints are being used as the twist effectors and they are not positioned very well - you need to move them in front of the legs.

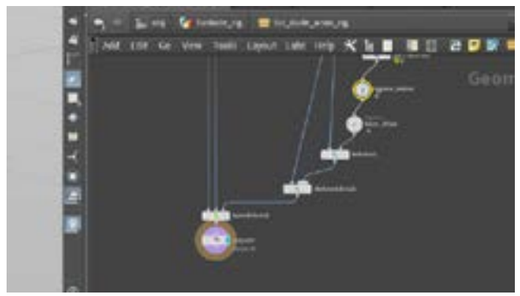


05 Add another **Rig Pose** node in-between the first *rigpose* node and the *ikchains* node. Press the **Shift** key and select both the *knee* joints and move them in front of the character. Now any flipping you had in your IK will have flipped back.

Name the first rigpose node to *rigpose_ankles* and the second one to *knee_offset*.

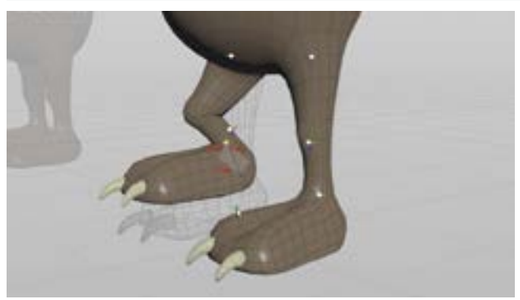


06 You can now go back to the *rig_ankles* node and test out the ankle joints. They don't flip now that you have the knees offset. You can also go to the *knee_offset* node and move the knees to act as a twist effector for the IK chain.



07 Add **Skeleton Blend** node and wire the *skeletonblend_controls* node into the first input and the *ikchains* into the second input. Then plug the output of the new **Skeleton Blend** node into the third input of the bone deform.

Rename the node to *skeletonblend_ik* and set the **World Space** checkbox to **On** and *weight1* to **1**.

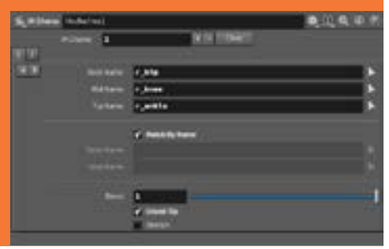


08 You can now go back to the *rigpose_ankles* node and test out the ankle joints to see how it affects the captured surface.

At this point, you are not going to save the digital asset to push these changes to the test rig. The *rigpose_ankles* node will not be how you control the ankles in the final rig. You are now going to build a reverse foot setup that will allow you to drive the whole foot setup and then the whole leg setup will be saved to the asset.

FK/IK BLENDING

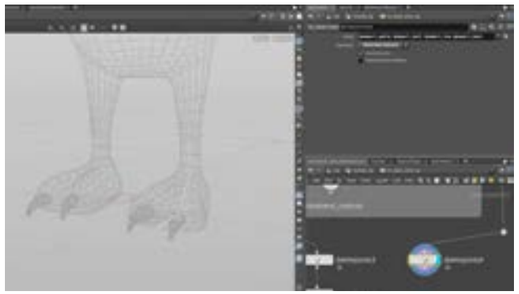
There are a couple of ways to blend IK and FK but you wont' be using them in this lesson because you will only be using IK for the legs. To get blending to work, you would need to promote parameters for the leg joints to the asset using a rigpose then you could either use the **Blend** attribute on the IK Chains node or use a **Skeleton Blend** between the IK solution and the rotating joints in the rigpose. If Fur Dude had arms then it would make more sense to set this up.



PART THIRTEEN

Reverse Foot Setup

To control the feet, you will create a classic reverse foot setup where the heel becomes the root then the toe, ball and ankle are parented to that. This can be easily accomplished in KineFX and the results blended back into the original skeleton. In this case you will completely rebuild the right foot but since the joint names align everything works properly.



01

Delete the *rigpose_ankles* node. You will control the ankles using a reverse foot setup.

Branch off a **Delete joints** node from the *skeletonmirror* node and set its **Display flag**. Click on the arrow beside Group and in the scene view select the *l_ankle*, *l_ball*, *l_toe* and *l_heel* joints. Press **Enter** then set **Operation** to **Delete Non-Selected**.

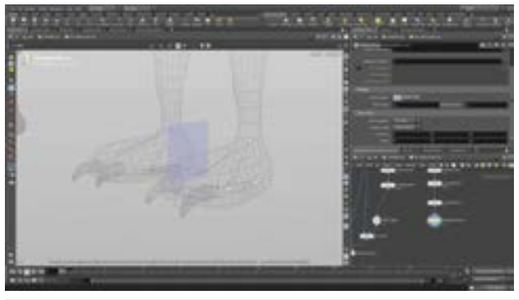
Note: If you want redirect the connecting wire to go around the network box then you can **Alt-click** on it to add dots.

02

Add a **Parent Joints** node and set its **Display flag**. Click the **+ sign** to add a Joint and set **Joint1** to *. Leave **Parent1** left blank.

Branch a second **Parent Joints** node and set its **Display flag**. In the Scene View click on the *l_heel* joint then the *l_toe* joint then the *l_ball* joint then the *l_ankle* joint. MMB-click to finish. In the Parameter pane they will be displayed in the following order:

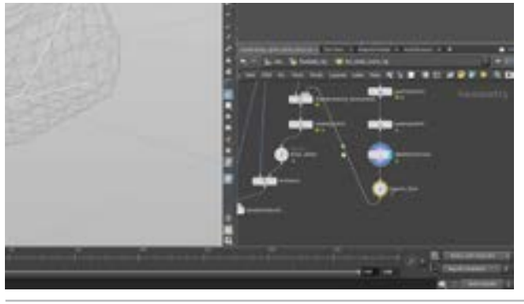
- **Joint1** to *@name=l_toe* | **Parent1** to *@name=l_heel*
- **Joint2** to *@name=l_ball* | **Parent2** to *@name=l_toe*
- **Joint3** to *@name=l_ankle* | **Parent3** to *@name=l_ball*



03

Add a **Skeleton Mirror** node to the chain. Go to the Parameter pane and under naming, set **Find Tokens** to *l_* and **Replace Tokens** with *r_*. This creates us the reverse foot for the right leg.

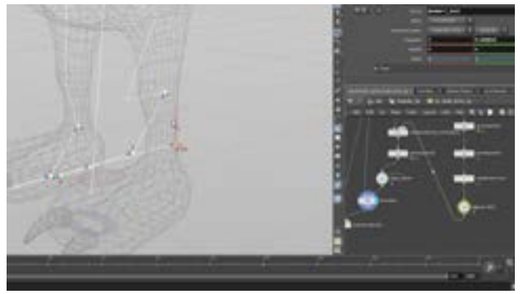
Because all the joints in this hierarchy have the same names as the original skeleton it will transfer the information properly when this rig is posed.



04

Now add a **Skeleton Blend** between the *deletejoints* and *parentjoints* that you set up earlier for the leg. Rename the node to *skeletonblend_reversefoot* and set the **World Space** checkbox to **On** and *weight1* to **1**. Click on the arrow next to **Group** and select the two *ankle* joints. Press **Enter**.

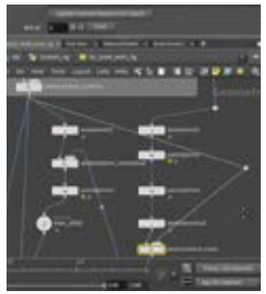
Add a **Rig Pose** node after the *skeletonmirror*. Rename it *rigpose_foot*. Feed it into the second input of the *skeletonblend_reversefoot* node.

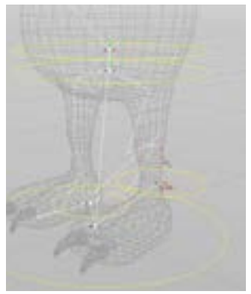


05

Set the **Display flag** on the *ikchains* node. Now go to the *rigpose_foot* node and select the *l_heel* joint. Now when you move it you are moving the whole foot and the leg chain. Select the *l_toe* joint and rotate it. Again the reverse foot works and the IK chain is activated.

When you finish press **Clear** to remove all the joints. You will add some of them back later. For the heel joints, you want to use the heel joints you set up earlier as part of the main controls.

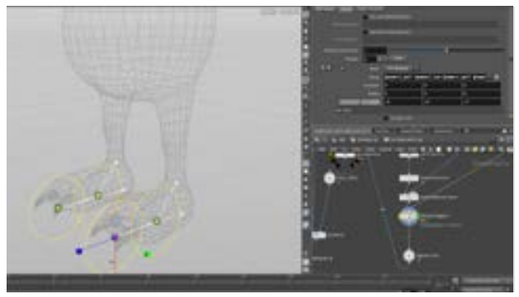




06

Now add a **Skeleton Blend** between the reverse foot *skeletonmirror* and *rigpose* node you were just using. Rename the node to *skeletonblend_heels* and set the **World Space** checkbox to **On** and *weight1* to **1**. Now feed the output of the *skeletonblend_controls* node into the second input.

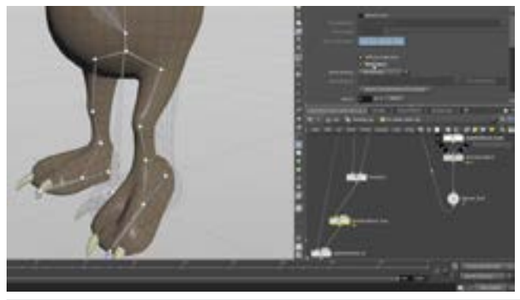
Click on the arrow next to **Group** and select the two *heel* joints. Press **Enter**. Now you can select the *rigpose* for the controls and move the left or right heel to control the whole setup. You can also grab the COG joint and if you move it up and down the IK chain works properly.



07

Put an **Attach Joint Geometry** node between the *skeletonblend_heels* node and the *rigpose_foot* node. Wire the *mergepacked* node from the **Main Controls** network box into the second input. Set the **Display Flag** on this node and press **Enter** in the scene view.

Select the *toe* and *ball* joints then **press G** and use your scroll wheel to find the *circle_ctrl* geometry. In the Operation Control bar at the top, change the **Mode** to **Tweak Shapes**. Select the *toe* and *ball* joints then **press G**. **Press E** to get the **Scale** handle then click drag on the middle handle to scale the controls to around **0.3**.



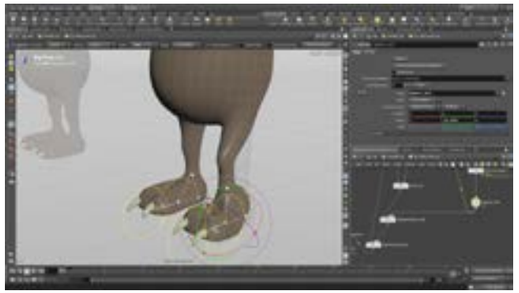
08

Set the **Display flag** on the *bonedeform* node. Pose the *l_ball* using the *rigpose_foot* node. You can see that rolling the ball causes the toe to point down instead of bending.

REVERSE FOOT SOP

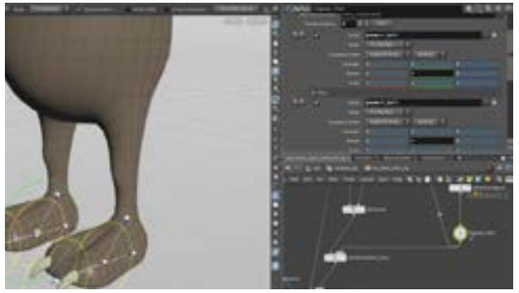
There is a **Reverse Foot SOP** that you could use to control the feet and drive the leg kinematics. The **Reverse Foot SOP** has sliders for controlling the roll of the foot an individual controls for all the parts. But you are not using it in this lesson.

Instead you will create a hand-built reverse foot solution to learn how joints can be manipulated to give you the control you need.



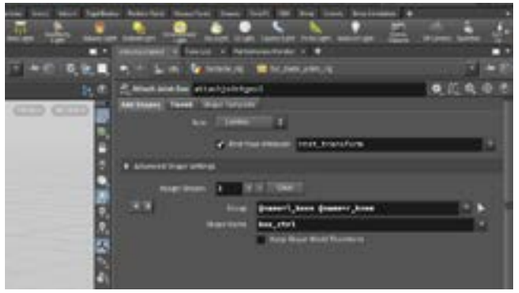
09 Now add a **Skeleton Blend** between the *ikchains* and *skeletonblend_ik* node you were just using. Rename the node to *skeletonblend_toes* and set the **World Space** checkbox to **On** and *weight1* to **1**. Now feed the output of the *rigpose_foot* node into the second input. Click on the arrow next to **Group** and select the *toe* and *ball* joints. Press **Enter**.

Now the toe will point in the right direction if you roll the *ball* joint.



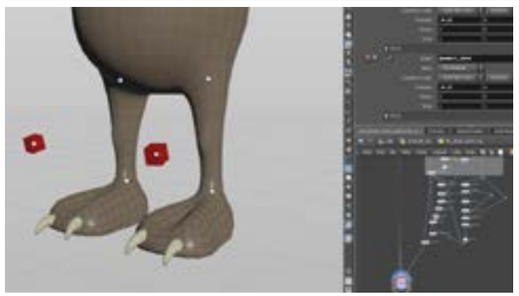
10 Select the *rigpose_feet* node. Make sure only the *left and right ball and toe* joints are listed on this node. For all of them, **RMB-click > Lock Parameter** on the **Translate, Rotate** and **Scale** parameters. Then go to the **Rotate Y** for each of them and **RMB-click > Unlock Parameter**.

Now if you select any of these four joints you will only get a Rotate Y handle.



11 Go back to the Main Controls Network box. Add a **Box** node to the network. Set its **Uniform Scale** to **0.02**. Follow this with a **Color** node and set it to a **red** color. Feed this node into the *mergepacked* node and set **Name 2** to *box_ctrl*.

Put an **Attach Joint Geometry** node between the *parentjoints* node and the *knee_offset* node. Wire the *mergepacked* node into the second input. Set **Mode** to **Tweak Shapes** then select the *knee* joints then **press G** and use your scroll wheel to find the *box* geometry.



12 Go to the *knee_offset* rigpose node. Make sure that only the two knee joints are listed. For both of them, **RMB-click > Lock Parameter** on the **Rotate** and **Scale** parameters. You will control these joints by moving them around.

Set the **Translate** values to **-0.15, 0, -0.05** for both knees.

Add a **Network Box** around all of these nodes used to define the spine and head joints and name it *Leg Controls*.

THE ROLE OF RIG POSE

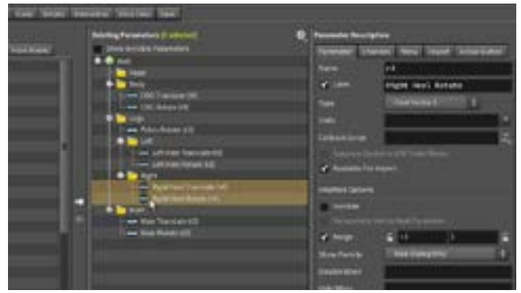
Up to this point the Rig Pose node has been used to test out the rig. When building an animation control rig, this node is also used to set up the parameters that will be promoted to the top level and define which joints will be visible when working with the digital asset.

As you set up these nodes it is easy to accidentally add parameters that you don't need which will add extra joints at the top level. You can also make the mistake of deleting parameters that you do need by clicking on the x button.

PART FOURTEEN

Promote the Leg and Spine Controls

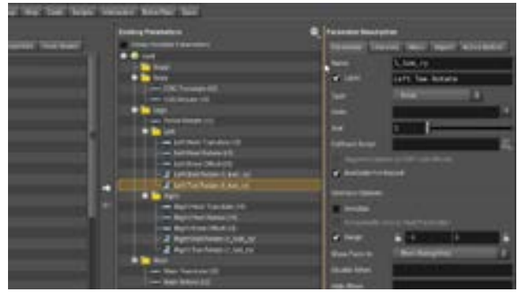
To make all of the leg and spine controls available to the animator, the parameters need to be promoted to the top level of the asset. This is an important step that is always needed to give the animator the control they need. This also means that you can keep certain parameters hidden that you don't want animators to work with.



01 From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Parameters** tab.

Create four folders. Name them *Head*, *Body*, *Legs* and *Main*. Drag the *COG parameters* onto the **Body** folder and the *Main parameters* onto the **Main** folder.

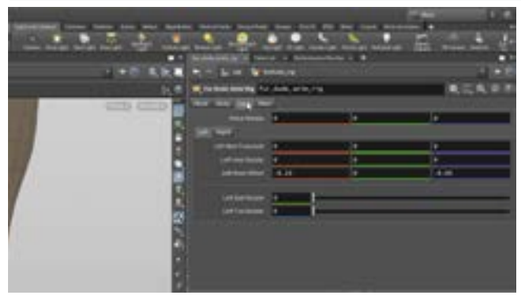
Put two folders inside the **Legs** folder and name the **Left** and **Right**. Add the *Left Heel* parameters into the **Left** folder and the *Right Heel* parameters into the **Right** Folder.



02 From the *knee_offset* rig pose node, drag the *l_knee* translate values into the **Left** folder in the **Leg** folder. Rename it **Left Knee Offset.** Click on the **Channel** tab and set the default to **-0.15, 0, -0.1.**

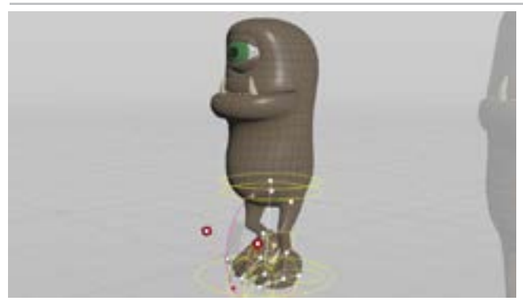
From the *rigpose_foot* node, drag the **Rotate Y** from the *l_ball* joint, name it **Left Ball Rotate** and set its **Range** to **0 to 30**. **Rotate Y** from the *l_toe* joint, name it **Left Toe Rotate** and set its **Range** to **0 to 20**. Drag a **Separator** in between the *Knee Offset* and *Ball Rotate* parameters.

Repeat for the right knee and the right foot.



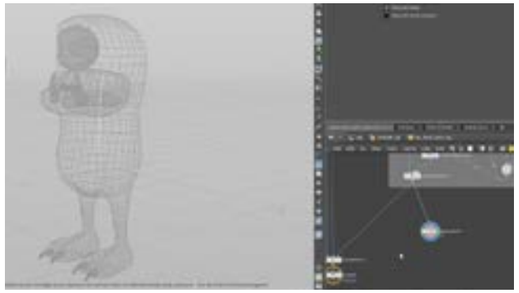
03 Click **Accept** and now the new parameters and controls are saved into the asset. Go up one level to see the asset parameters laid out on the *furdude_rig*.

These parameters are for the unlocked rig that you are working on. Rather than test out these parameters, it is a better idea to play around with the *test_rig*.



04 In the Network view, **press 1** to navigate back to the *test_rig* which has been updated to show the new controls.

Play with the test rig in the Scene view using the handles or using the parameters in the floating panel.

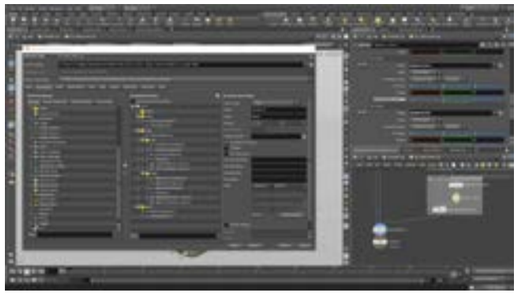


05 Branch off a **Delete joints** node from the *skeletonblend_ik* node and set its **Display flag**. Rename the node *deletejoints_spine*.

Click on the arrow beside **Group** and in the scene view select the *spine1*, *spine2*, *spine3*, *neck1*, *neck2* and *jaw* joints. Press **Enter** then set **Operation** to **Delete Non-Selected**.

06 Add a **Rig Pose** node and rename it *rigpose_spine*. Now add a **Skeleton Blend** between the *skeletonblend_ik* and *bonedeform* node. Rename the node to *skeletonblend_spine* and set the **World Space** checkbox to **On** and *weight1* to **1**. Now feed the output of the *rigpose_spine* node into the second input.

Add a **Network Box** around all of these nodes used to define the spine and head joints and name it *Spine Controls*.



07 Set its **Display Flag** then in the Scene view press and hold the **S key** then select all the joints. This will add them to the rigpose list.

Lock the **Translate** and **Scale** parameters for all of these joints. You will only be rotating them.

08 From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Parameters** tab.

Drag the *neck1*, *neck2* and *jaw* onto the **Head** folder and name them Neck 1 Rotate, Neck 2 Rotate and Jaw Rotate. *spine1*, *spine2* and *spine3* onto the **Body** folder and name them *Spine 1*, *Spine 2* and *Spine 3*. Add a separator between the COG parameters and the spine parameters.

Click **Accept**. This will save the new controls to the rig. You can now explore them using the *test_rig*.

SPINE CONTROLS

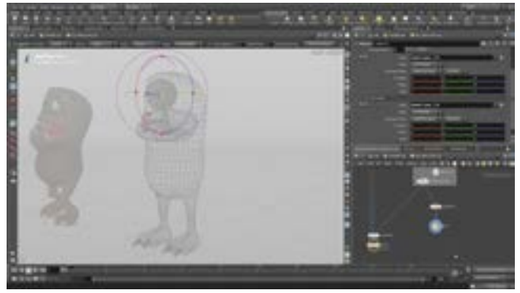
For this rig, you set up the spine using joint rotations. This is known as **Forward Kinematics** and just like other parts of the rig, you pulled these joints out from the main skeleton then set them up in a Rig Pose to be promoted to the top level.

You did not use control geometry for these parts because the joints are easy to select in the Scene view. You don't need to have control geo for all the joints.

PART FIFTEEN

Eye Controls

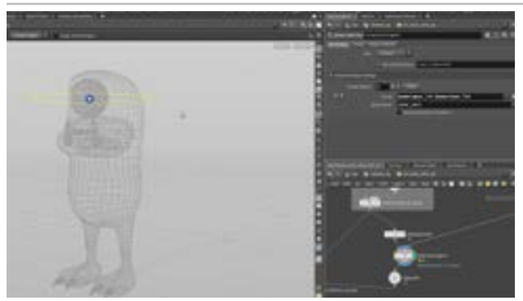
The next step is to set up the eyelids with control geometry to make it easier to select these overlapping joints. You will also set up the eye target joint as a look at for the eyeball using a different section of Houdini called VOPS. When these parts are rigged, you will again promote the appropriate parameters to the character's asset.



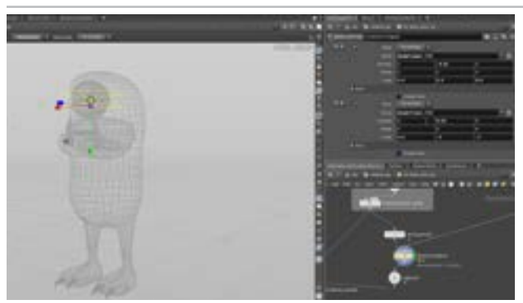
01 Branch off a **Delete joints** node from the *skeletonblend_spine* node, set its **Display flag** and name it *deletejoints_eyelids*. Click on the arrow beside **Group** and in the **Rig Tree** select *upper_lid and lower_lid*. Move your cursor over the Scene View and press **Enter** then set **Operation** to **Delete Non-Selected**.

Add a **Rig Pose** node and rename it *rigpose_eyelids*. Set its **Display Flag** then in the Scene view press and hold the **S** key then select all the joints. This will add them to the rigpose list.

02 Now add a **Skeleton Blend** between the *skeletonblend_spine* and *bonedeform* node. Rename the node to *skeletonblend_eyelids* and set the **World Space** checkbox to **On** and *weight1* to **1**. Now feed the output of the *rigpose_eyelids* node into the second input.



03 Place an **Attach Joint Geometry** node between the *deletejoints* node and the *rigpose* node. Wire the *mergepacked* node into the second input. Set the **Display Flag** on this node and press **Enter** in the scene view. Make sure the **Mode** is set to **Assign Shapes**. Select all the two eyelid joints then **press G** and use your scroll wheel to find the *circle_ctrl* geometry.



04 In the Operation Control bar at the top, change the **Mode** to **Tweak Shapes**.

Select the *eyelid* joints then **press G** to bring up a transform handle. **Press E** to get the scale handle then click drag on the middle handle to scale in all three directions until these controls are a bit smaller (around 0.5 in the parameter pane).

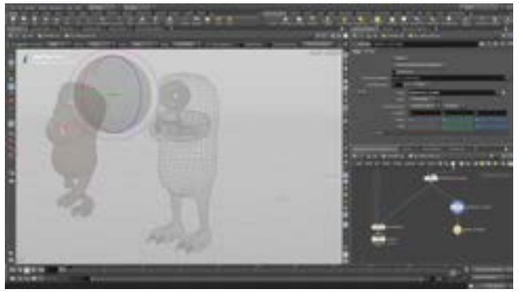
Now select the *upper_eyelid* joint and move it up by **0.02** then select the *lower_eyelid* joint and move it down by **0.02**.

05 From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Parameters** tab.

Lock the **Translate, Rotate** and **Scale** parameters for all of these joints. Unlock the **Rotate X** parameters. From the *rigpose_eyelids* node, drag the *Rotate X* from the *upper_lid* to the **Head** folder. Set the **Range** from **-10** to **30**. Next, drag the *Rotate X* from the *lower_lid* to the **Head** folder. Set the **Range** from **-20** to **20**.

Add a Separator to divide the eye controls from the head controls.

Click **Accept**. This will save the new controls to the rig. You can now explore them using the *test_rig*.



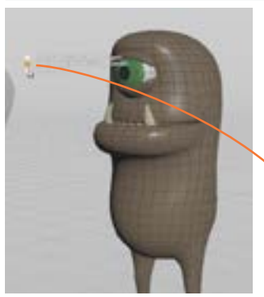
06 Branch off a **Delete joints** node from the *skeletonblend_eyelids* node, set its **Display flag** and name it *deletejoints_eyes*. Click on the arrow beside **Group** and in the scene view select the *eyeball* and *eye_target* joints. Press **Enter** then set **Operation** to **Delete Non-Selected**.

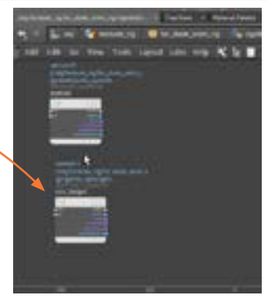
Add a **Rig Pose** node and rename it *rigpose_eyetarget*. Set its **Display Flag** then in the Scene view select the *eye_target* joint. This will add it to the rigpose list. You do not need the *eyeball* joint - it will be controlled by a look at constraint. Lock the **Rotate,** and **Scale** parameters for the *eye_target* joint.

07 Add a **Rig Attribute VOP** node. Wire the *deletejoints_eyes* node into the first input and the *rigpose_eyetarget* into the second one.

Now add a **Skeleton Blend** between the *skeletonblend_eyelids* and *bonedeform* node. Rename the node to *skeletonblend_eyeball* and set the **World Space** checkbox to **On** and *weight1* to **1**. Now feed the output of the *rigattributevop* node into the second input.

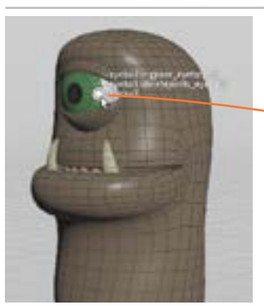
Set the **Display Flag** on the *bonedeform* node.

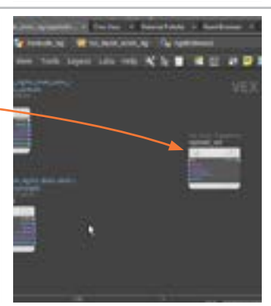




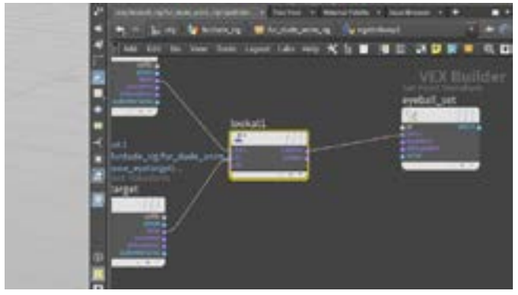
08 Double click on the *rigattributevop* node to dive into it. In the Scene view, click on the *eyeball* joint then drag the *eyeball (deletejoints)* version into the Network editor. This gives you a **Get Point Transform** node that focuses on the *eyeball* joint coming from the **First Input**.

Click on the *eye_target* joint then drag the *eyetarget (ripose_eyetarget)* version into the Network editor. This gives you a **Get Point Transform** node that focuses on the *eye-target* joint coming from the **Second Input.**





09 Click on the *eyeball* joint then drag the *eyeball* version into the Network editor. This gives you a **Set Point Transform** node that focuses on the *eyeball* joint.

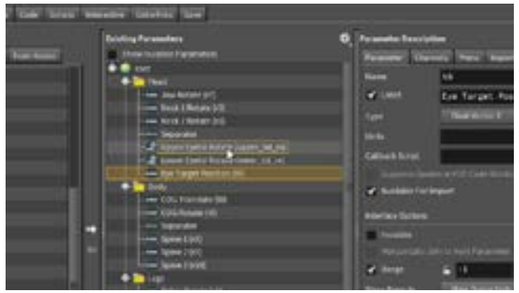


10 Now press **tab > Look At (KineFX)** and place the node in the middle. Wire the **xform** output on the eyeball *getpointtransform* node into the **from** input on the *lookat* node. Wire the **xform** output on the eye_target *getpointtransform* node into the **to** input on the *lookat* node. Wire the **outxform** output on the *lookat* node into the **xform** input on the *eyeball_set* node.

The eyeball geometry flips. Select the *lookat* node and set the **Look At Axis** to **Z** to match the orientation you set on the rig when it was first set up.

11 Put an **Attach Control Geometry** node between the *deletejoints_eyes* node and the *rigpose_eyetarget* node. Wire the *mergepacked* node from earlier in the network into the second input. Set the Display flag on this node and go to the Handle tool. Make sure the **Mode** is set to **Assign Shapes**. Select the *eye_target* joint in the 3D view then **press G** and use your scroll wheel to select the *square_ctrl*.

Set the **Display Flag** on the *bonedeform* node. Click on *rigpose_target*. Select and move the *eye_target* joint. This orients the eyeball. **Undo** to set it back to it's original position.

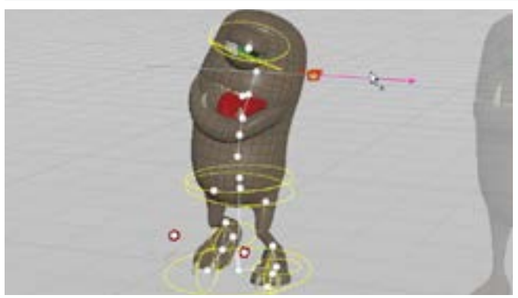


12 Add a **Network Box** around all of the nodes used to define the eyes and name it *Eye Controls*.

From the **Assets** menu, select **Edit Asset Properties > Fur Dude Anim Rig**. Click on the **Parameters** tab.

From the *rigpose_eyetarget* node, drag the *Translate* parameters from the *eye_target* to the **Head** folder in the *eye* section. Name them **Eye Target Position**.

Click **Accept**. This will save the new controls to the rig.



13 You can now explore them using the *test_rig*.

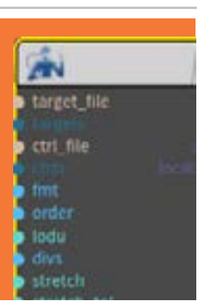
You now have all of the parts finished for this control rig. It is now ready for you to animate a walk cycle. To do this you will create a second copy of the test rig and animate using that network.

This digital asset can create multiple instances in multiple scene files and if you later need to come back and make a change, all of the assets will update. This is the pipeline advantage of working with a digital asset rig.

RIGGING IN VOPS

The Rig Attribute VOP offers a range of different solutions beyond what you have seen in this lesson. IK chains can be built using this approach and the IK chain SOP you used earlier has one of these inside it.

You can also use VOPS to set up a curve solver, realistic shoulder, reverse foot and more. The ability to drag joints from the Scene view to the VOP network is a unique workflow that helps speed up workflow.



PART SIXTEEN

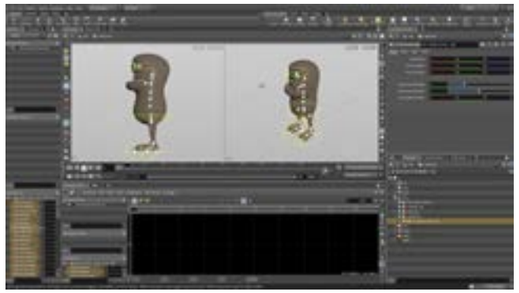
Animate the Rig

It is time to keyframe a walk cycle for the fur dude character. This will involve new tools such as the Channel List to pin down channels for blocking out the motion. The results will be a quick and dirty walk cycle designed to see fur dude in action. The goal is to map out a basic keyframe workflow to learn a bit about how to animate KineFX rigs.

01 Go to the object level. **Press tab > Geometry** and place down the node. Rename it *walkcycle*. Turn off the **Display Flags** on all of the objects.

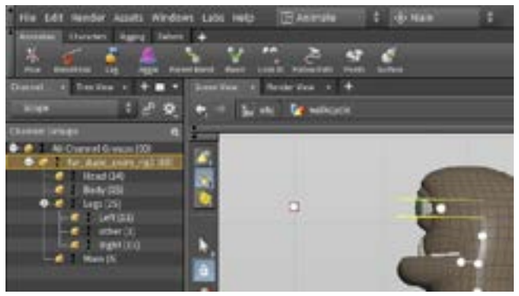
Double-click to dive into *walkcycle* and in the Network view, press **tab > Fur Dude Anim Rig**. Press **Enter** to place it at the origin. This is a new locked version of the fur dude rig that you will animate from scratch.

This puts another version of the character asset into the scene. You can have multiple versions in this scene file or in other scene files and they will all reference the same asset definition on disk.



02 From the **Desktop** menu (the one that currently says Build), choose **Animate**. This gives you panes that are designed to work with a keyframe workflow. You may need to go back into the *walkcycle* object.

The **Channel List** on the left will play a key role in blocking out the animation of the character. The **Animation Editor** lets you display and edit animation curves. In this lesson you will block out the motion and won't be doing any curve editing.



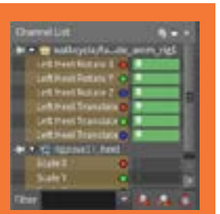
03 In the Network view, select the *fur_dude_anim_rig* inside the *walkcycle* object. Go to the Parameter pane and click on the box icon in the top right. Choose **Parameters and Channels > Create Nested Channel Groups**. Click **Close** in the pop up window. The parameters from your asset are now listed and organized based on their folders.

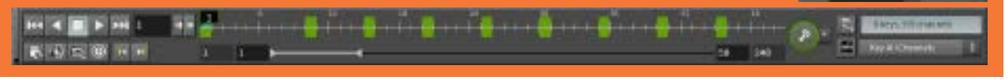
Click on the **Pin** icon next to the *fur_dude_anim_rig* channel group to pin these channels. Make sure your timeline is set to **frame 1** then **press k** to keyframe all the channels.

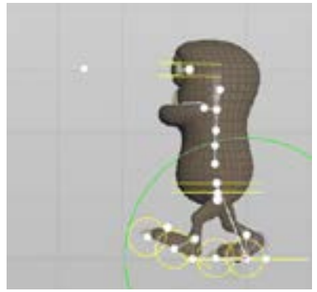
HOW CHANNELS WORK

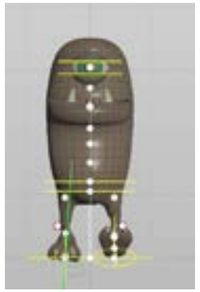
When you select a joint, the channels load into the channel list. Press **k** to keyframe them.

If you want to keep them loaded in the list for blocking purposes, you can pin them or add them to a channel groups to pin them all together. You can build the Channel groups directly from the asset. They are organized based on how you built the UI for your character. You can also build your own groups to pin down specific channels.



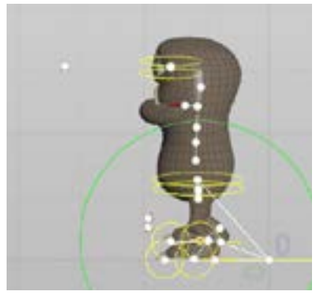






04 Set the **Timeline** to start at **10** and end at **50**. Go to **frame 10**. With all the channels pinned, **press k** to set another keyframe. You want to **keyframe first** then pose. Any posing will update the value for that keyed frame.

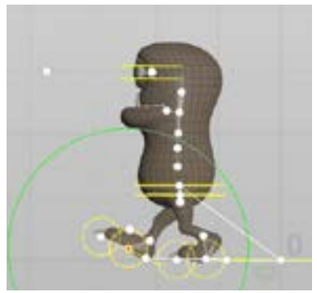
You want to create a pose where the *left heel* moves forward and rotates up. The *COG* will go down a bit and the *right ball* rolls forward.



05 Go to **frame 15** and **press k**.

Roll the *left heel* flat but don't move it. Move the COG to align with the left foot. Move the *right heel* up in line with the other foot. Rotate the *right ball* back to flatten the foot.

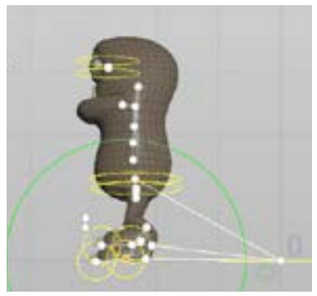
Rotate the *COG* a bit towards the left foot. You can also add some rotation to the three *spine* joints to emphasize this tilt.

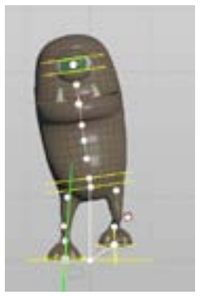


06 Go to **frame 20** and **press k**.

Now you want to create a pose where the *right heel* moves forward and rotates up. The *COG* will go down a bit and the *left ball* rolls forward. This is the opposite of the pose at frame 10.

Rotate the *COG* and *spine* joints back to center them.

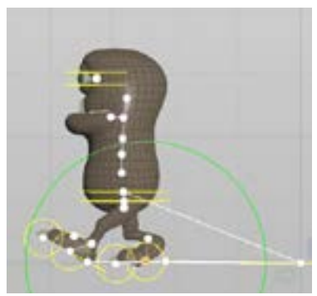




07 Go to **frame 25** and **press k**.

Roll the *right heel* flat but don't move it. Move the *COG* to align with the right foot. Move the *left heel* up in line with the other foot. Rotate the *left ball* back to flatten the foot.

Rotate the *COG* a bit towards the right foot. You can also add some rotation to the three *spine* joints to emphasize this tilt.



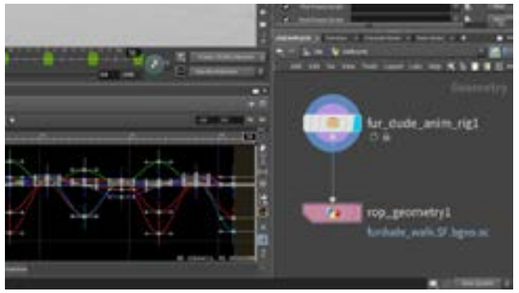
08 Go to **frame 30** and **press k**.

You want to create a pose where the *left heel* moves forward and rotates up. The *COG* will go down a bit and the *right ball* rolls forward.

Rotate the COG and spine joints back to center them.

09 Continue this pattern up until frame **50**. You can repeat the same poses to keep the walk cycle moving forward.

At this point you can go back and tweak any pose to refine the motion. You can also explore creating some overlapping action using extra keyframes. You can animate some eye movement and maybe the eyelids blinking. You can also go beyond 50 frames if you would like a longer animation.

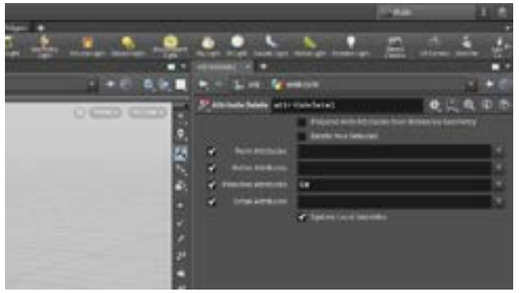


10 Connect the output of the *fur_dude_anim_rig* node into a ROP Geometry node. This will allow you to export out a cache of the fur dude geometry. Set the **Output File** to

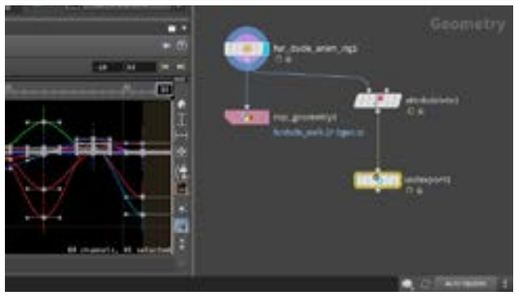
```
$HIP/geo/furdude_walk.$F.bgeo.sc
```

Next **Set Valid Frame Range** to **Render Frame Range**. **RMB-click** on the **Start/End/Inc** parameter and choose **Delete Channels**. Set the **Start** to **1** and the **End** to **50**.

Click the **Save to Disk** button to store the cache to disk. You will use this to add fur.



11 Branch an **Attribute Delete** node off of the *fur_dude_anim_rig* node. Under **Primitive Attributes** choose **Cd**. This removes the color from all the body parts.



12 Connect the output of the *attributedelete* node into a **USD Export** node. This will allow you to export out a the fur dude geometry to the USD format. Set the **Output File** to

```
$HIP/usd/furdude_walk.usd
```

Next **Set Valid Frame Range** to **Render Frame Range**. **RMB-click** on the **Start/End/Inc** parameter and choose **Delete Channels**. Set the **Start** to **1** and the **End** to **50**.

Click the **Save to Disk** button to store the cache out the USD file. You will use this later in the rendering process.

CACHING OUT ANIMATION

Because of Houdini's procedural nature it is not really necessary to cache out the animation. You could reference it from this network to another network for grooming or to Solaris for conversion to USD. The advantage of caching is that you lock down your animation and work with a flattened file on disk. This is a very production-friendly approach.

In Solaris a USD file referenced from disk is also more efficient. Since Houdini references files from disk, you are always free to change your animation and output the new sequence and it will be picked up automatically.

PART SEVENTEEN

Add & Groom the Fur

Fur dude gets his name for a reason and you are going to work with a variety of grooming tools to add and shape the hair. Using a desktop designed for grooming, you will add frizz, clumping and hair dynamics which will simulate as fur dude walks. The end result will be ready to be exported for rendering.

01 Got to the **Grooming** desktop. Put the four existing objects into a network box and call it box *Rig & Animate*.

In the Network editor, press **tab > File**. Place the node then double click to go into it. Click on the **browse** button next to **Geometry File** and go to *$HIP/geo*. Select *furdude_walk.$F.bgeo.sc*. Press **Accept**.

Add a **Blast** node, set it's **Group** to *fur_dude_body* and turn **On** the **Delete Non-Selected** checkbox to focus on the body. Set the **Display Flag** then go to the Object level and rename this *fd_anim*.

Alt drag to create a copy it and name it *fd_rest*. Dive in and change **Geometry File** to *$HIP/geo/ furdude_walk.1.bgeo.sc.*

02 Now move the time slide forward a bit. One object has a static version of fur dude and the other is animated. Click the **Add Fur** button . Select the *fd_rest* object and press **Enter**.

Now select the *fd_anim* object and press **Enter**. Turn off the **Display Flag** on *fd_rest_anim*, *fd_rest_deform* and *fd_rest_hairgen*.

Turn on the **Display flag** for *fd_rest* and *fd_rest_groom*.

03 Select the *fd_rest_groom* node and from the **Hair Tools** shelf, click on **Set Guide Length**. Turn the Randomize Button to On. Set M**in Length** to **0.03** and from the menu on the right select **Texture**.

Use the file browser button to select *$HIP* then go into the *tex* directory and choose *fur_length.jpg*.

Now set **Max Length** to **0.15** and again choose **Texture**. You can use the **arrow** on the right to select the *fur_length.jpg* image.

You now have the eye, lip and the bottom of the feet masked out and the rest of the fur with random lengths.

04 From the **Hair Tools** shelf, click on **Bend Guides**. Set **Angle** to **45** to add some bending to the guides.

Next, click on the **Frizz Guides** tool. Set the following:

- **Frequency** to **15**
- **Amplitude** to **0.005**
- **Random Amplitude** to **0.02**

This keeps the hairs from looking too straight when rendered. You can add more frizz if you want a more tangled look.

05 From the **Hair Tools** shelf, click on **Clump Guides**. Set **Clump Size** to **0.02** to **Tightness** to **0.5**.

Next change the **Clump Profile** so that it stays high for a bit longer then tapers off near the end.

Now go to the Object level and turn **off** the display on the *fd_rest* and the *fd_rest_groom* and turn **on** the display on the *fd_anim* and *fd_rest_sim* and *fd_rest_hairgen*.

Select the *fd_rest_groom* node and set **Density** to **20000**.

06 Select the *fd_rest deform* node and from the **Hair Tools** shelf, click on **Simulate Guides**. On the *fd_rest_sim* node, go to the **Vellum Constraints** tab and under **Bend** set **Stiffness** to **5**.

Alt-drag on the *fd_anim* node to make a copy and rename it *fd_collision*. Dive into this node and on the *blast* node set **Delete Non Selected** to **Off**. Add the tongue, upper teeth and gums to the **Group** selection then add a Null node and name it *COLLISION_OUT*.

At the object level, select *fd_rest_sim* then under **Vellum Collisions, set External Collisons to On then** set **Collider SOP** to *../fd_collision/COLLISION_OUT*.

07 Click on the Caching tab and set **Set Valid Frame Range** to **Save Frame Range**. **RMB-click** on the **Start/End/Inc** parameter and choose **Delete Channels**. Set the **Start** to **1** and the **End** to **50**. Click **Save to Disk** to run the simulation.

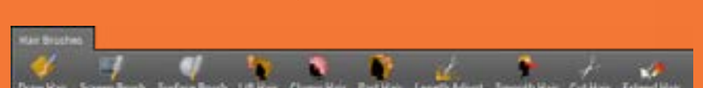
Now set the **Load from Disk** checkbox to **On**. The cache will now be used to define the fur instead of the hair calculating for each frame.

08 Select the *fd_rest_hairgen* node. Under **Distribution**, set **Density** to **1000000**. Scroll down to Guide Interpolation and set **Clump Crossover** to **0.25** to create a it of overlap between clumps. This gives you an idea of what furdude looks like with a full head of hair.

These are not the hairs that you will render in Houdini's lighting context called Solaris. You will instead bring in the guide hairs and render using a hair procedural at render time.

HAIR BRUSHES

The grooming desktop also has hair brush tools that let you work interactively on the character's surface. You can lengthen, smooth, cut and extend hair. These tools were not needed to set up fur dude's grooming but later you may want to explore them to further style the final look.



PART EIGHTEEN

Set up an Render the Shot

To render the shot, you will reference the USD files into the Solaris Stage then add a backdrop. Solaris is a Houdini context that uses LOP nodes to set up a USD Scene Graph. Next, you will import the fur then add and position a camera and a light. The Karma renderer will then be invoked to create a preview render of the shot then render out the animated sequence.

01 Change the desktop to **Solaris**. Choose **Stage** from the path bar. In the Network View press **tab > Reference** then click to add a **Reference** node.

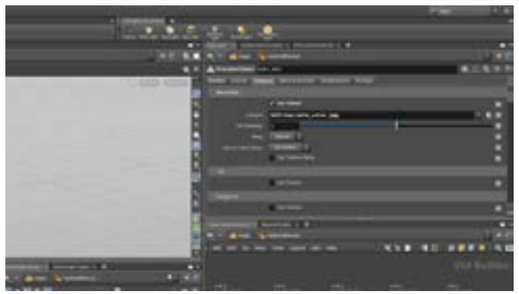
Next to **Reference File**, click on the **File Pattern** and find the *furdude_walk.usd* file. Rename the node to *furdude*. Set the **Primitive Path** to */char/`@sourcename`* - this will use the node name and place it into a group called *char*. In the **Scene Graph Tree** expand *char* then *furdude* to see all of the named primitives.

In the Scene View, use your view tools such as **spacebar-h** for homing the view to get a better look at the walk cycle.

02 Press **tab > Material Library. Wire** it into the output of the *reference* node then set its **Display Flag.**

Go to the **Material Palette** pane. Click on the arrow next to */stage/materiallibrary* to open up this area. Scroll through the material gallery on the left of the palette and drag a **Principled Shader** materials into the *materiallibrary* working area.

Go to the Network view and **Alt-drag** this material to create four more. Rename the five materials *body_mat*, *eyeball_mat*, *eyelid_mat*, *teeth_mat*, and *tongue_mat*. You can also see the materials in the **Scene Graph Tree**.



03 For *furdude_body_mat*, under the **Surface** tab, set **Base Color** to **1, 1, 1**. Click on the **Textures** tab and under **Base Color** click on **Use Texture** then use the button next to **Texture** to call up the file window. Click on **$HIP** in the side list then click on the *tex* folder to open it and then click once on *skin_color.jpg* to select it. Click **Accept** to assign the texture to the material. Next set the **Roughness** to **0.5** and **Reflectivity** to **0**.

Assign *eye_color.jpg* and *eye_lid.jpg* to their materials using the same method. Set *tongue_mat* to a **redish pink** and *teeth_mat* to a **yellowish-white**.

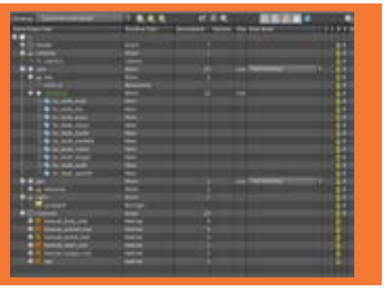
04 Go to the **Stage** level. After the *Material Library* node, add an **Assign Materials** node. From the Scene Graph, drag the *fur_dude_body* to the **Primitives** field then click on the arrow next to **Material Path** and choose *body_mat*. Now click the **Plus Sign** next to add four new entries. Assign them as follows:

- *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 SCENE GRAPH

When working in Solaris, the geometry and materials you add using LOP nodes are added into the **Scene Graph** and converted to USD. When you add a light and a camera, they will also become part of the USD Scene Graph.

It is not necessary to completely understand USD to light and render in Houdini as an artist but once you start thinking about the pipeline of your project, USD will become a useful tool in managing your shots.



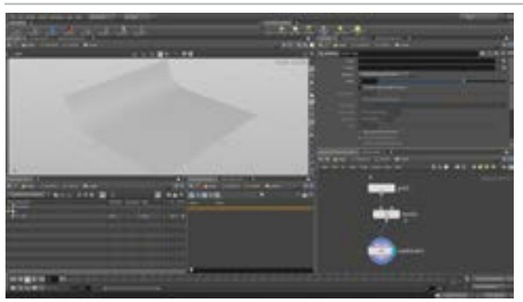
05 In the Network view, press **tab** and type out **SOP Import**. Click to place the node. Rename it *hair*. Set **Import Path Prefix** to */char/$OS*. Click on the node icon net to SOP Path and navigate to the *fd_rest_hairgen* node.

Add a **Merge** node in-between the *furdude* and *materiallibrary* nodes. Wire the *hair* node into it.

06 Go back to the **Material Palette** pane. Open up */stage/materiallibrary*. Drag the **Hair** material into the *materiallibrary* working area.

Leave the **Root Color** and **Tip Color** set to the defaults for the hair. Next click on the **Secondary Reflection** tab and set **Root Color** to **dark grey** and **Tip Color** to **medium grey**.

Go back to the Assign Material node, From the **Scene Graph**, click on the arrow next to **Primitives** and select the fur curves. Next, click on the arrow next to **Material Path** and choose *hair*.

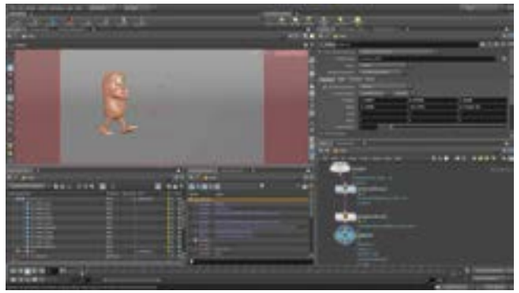


07 In the Network view, press **tab** and type out **Grid**. Click to place the node. Rename it *backdrop* and wire it into *merge*. Set **Import Path Prefix** to */geo/$OS*. **Double-click** on the *backdrop* node to dive down to the geometry level.

Select the *Grid* node and set the **Size** to **50, 50** and **Rows** and **Columns** to **10**. **RMB-click** on the *grid* node's output and type **Bend**. Place the bend node and set its **Display Flag** then set: **Bend** to **75, Capture Origin** to **0, 0, -10, Capture Direction** to **0, 0, -1,** and **Capture Length** to **10. RMB-click** on the *bend* node's output and type **Subdivide.** Set its **Display Flag** then set **Depth** to **2.**

08 Go to object level and set **Rotate Y** to **-45** degrees. Add material and assign it. You can leave the default grey or add your own base color.

Set time range from **10-50**. You will be rendering this sequence after the hair has had 10 frames to settle down.

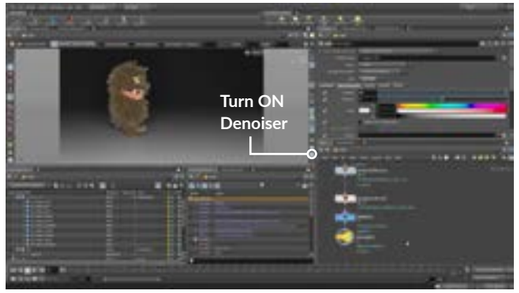


09 Use your view tools to look at *furdude* from the front. From the **LOP Lights and Camera** shelf, **Ctrl-click** on the **Camera** tool. This adds a camera node into the network and you are now looking through the camera in the Scene View.

Press the **Lock Camera/Light to View** button so that view changes can be used to reposition the camera. Now **Tumble**, **Pan** and **Dolly** in the Scene View to reposition the camera so furdude starts on the left and the moves to the right. Scrub the timeline to make sure the camera works for the whole sequence.

10 Now turn **Off** the **Lock Camera/Light to View** button then tumble around until you are looking down on Fur Dude. From the **LOP Lights and Camera** shelf, **Ctrl-click** on the **Area Light** tool. This adds an *arealight* node to the end of the chain.

Select the *arealight* node and from the **Base Properties** tab, set the **Intensity** to 2.



11 From the *Persp* menu, choose **Karma** to render with Karma in the Scene View. You can move to different frames in the timeline and the Scene View will update quickly.

Karma is designed to work with USD which is why everything in the LOP context is converted to the USD scene graph. You can only use the Karma renderer from this part of Houdini.

To get a cleaner image when you render, you can turn on the Denoiser if you have an Nvidia graphics card. Be sure to install the **Denoiser** from the **Render** menu and then turn it on in the **Display Options** bar.

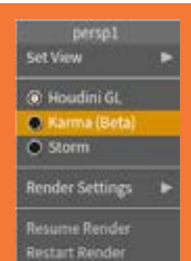
12 Press **tab > Karma** to add a **Karma Render Settings** and **USD Render ROP** node. Wire them into the end of the chain. Select *karmarendersettings* and on the **Image Output > Filters** tab set **Denoiser** to **nvidia Optix Denoiser**. Set the **Output Picture** to *$HIP/render/walk/furdude_walk_$F2.exr*. The *$F* in the name is needed to add frame numbers to the renderings and the *2* is the padding of the frame number.

Select *usdrender_rop*. **RMB-click** on the **Start/End/Inc** parameters and choose **Delete Channels**. Set the **Start** to **10** and the **End** to **50** Select the *usdrender_rop* node. Click on **Render to Disk**.

KARMA RENDERER

You are now going to render the sequence using Houdini's renderer Karma. Karma is designed to render USD and is known as a Render Delegate. At first you will render in the Scene View. Press **d** in the Scene View to bring up display options to control your rendering. You can turn on a denoiser, set Pixel Samples and define the Image Resolution.

Later when you set up a **Karma LOP**, there will be render settings on that node that you will use to create the final output being saved to disk.



13 When you finish, choose **Render > Mplay > Load Disk Files** and open up the rendered images to review the final sequence.

Later you can branch off another Karma node to up the resolution and render settings for your final rendering. It is always good to complete test renderings at a lower resolution first to make sure that everything is working the way you expect it to.

14 If you want to tweak the hair and fur settings, you can pin the Scene View to the LOP network then go back to the object level and from the **Simulate Guides** node, set the **Load from Disk** checkbox to **Off**. This will allow any changes to flow through to the final rendering.

Here you can see that the hair was shortened and made much frizzier. You can do anything you want. When you are finished be sure to re-cache and then turn **Load from Disk** back to **On**.

15 You can then go back to the Solaris network and maybe change the colors on the fur and re-render.

For the final render you may want to up the **resolution** to **1920x1080** and change some of the quality settings. For instance you can set **Pixel Samples** to **128** and **Light Sampling Quality** to **16**.

CONCLUSION

You have now rigged, animated and rendered the Fur Dude character using the **KineFX** tools in Houdini. Along the way you have touched on various important steps including the creation of a **capture rig** then the layering of a **animation control rig** on top of that. You have packed up the character into a **Houdini Digital Asset** and then keyframed a traditional **walk cycle**.

You then added fur with various grooming tools then rendered using **Karma**. This gives you a complete workflow for creating a shot using this character. You can then go back and refine any aspect of the procedure to create multiple iterations as you seek out the perfect result.

As mentioned before, the KineFX toolset is currently designed for retargeting and motion editing which are not covered in this lesson. These rigging and animation tools will continue to evolve and this lesson provides a taste for what is coming in Houdini's procedural rigging workflow in future versions.