The Bipedal Rig

by Tom Magill

This tutorial steps you through creating the entire skeletal rig with IK and SDK for an animatable bipedal character. This document is designed to compliment in-class tutorials and as such I am not including explanations and reasons to creating the rig in this particular way. Including these examples would more than double the volume of this document and is dealt with in detail during face-to-face time in class.

However, this document will be a good memory aid for all the steps involved in constructing the rig.

Intro

The polygon model featured in this document is one of the base meshes that come with Mudbox. It has been altered to have a mouth cavity and the arms have been transformed into a modelling T-stance.

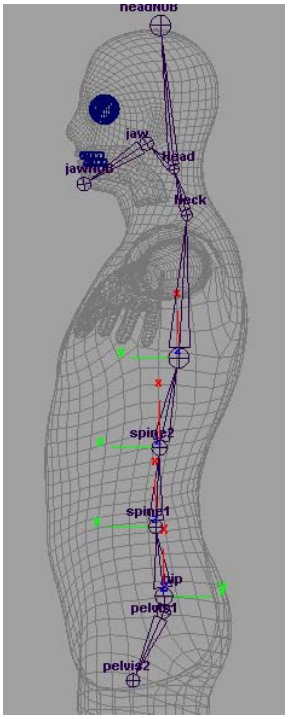
Setting up a rig is an involved process and this tutorial describes a minimalistic setup – a comparably simple rig that nonetheless allows for forearm twisting, foot roll and driven finger movement.

You own model will only be ready for rigging if you have completed mapping of texture coordinates.

This tutorial assumes that all tools and actions are at their default settings.

Duplicate your character model and hide the duplicate. You should keep this duplicate as a template for the creation of blend shapes.

1. The skeleton

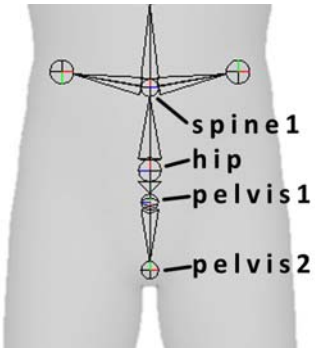


Create the spine and head in side view

Ideally the spine is a straight line. That way the x-axis of each joint points straight up. This is not anatomically correct, but it's good for torso animation. However, the spine joints should not run along the back of the model and should rather lie closer to the centre of the torso. So because the torso of this particular model is bent slightly backwards I made my joints slightly bend backwards too. Then I manually rotated the local rotation axis of each joint so that the x-axis points straight up (as opposed to pointing to its child joint).

To manually adjust the local rotation axis of a joint enter component mode and enable the filter for local rotation axis, which is the filter button that has a question mark: ?
You can also display local rotation axis when unselected (Display > Transform Display > Local Rotation Axis)

If after joint creation you need to modify the placement of joints try to stick to the Rotate and Scale Tools only. Afterwards you must not forget to Modify > Freeze Transforms (enter the options window and check that move, rotate and scale are all ticked). Don't use the Move Tool to adjust joints.



Note that the head pivots at the top of the neck and the jaw pivot should be at earlobe level.

To separate hip movement from the spine you should create a pelvic bone – that's two joints that point downward from the hip joint. The pic on the right is a close-up of a good pelvis set-up viewed from the front.

Depending on your character design you might want to have the spine1 closer to the hip bone.

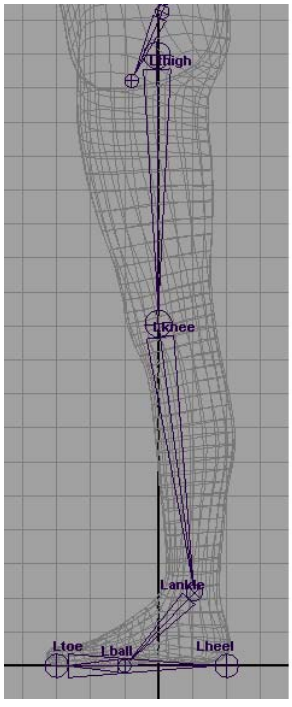
Create one leg in side view

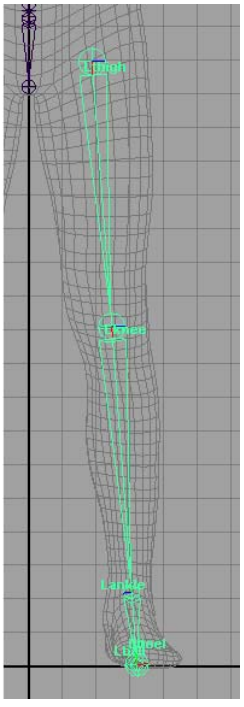
The leg gets created in side view.

Please note that the foot (ball, toe and heel) are snapped to the working grid. They do not lie in the middle of the foot but along the base of the foot.

Then go to front view and, through use of Rotate and Scale place the joints along the middle of the leg. At this stage the leg is still a separate joint chain to the spine.

Parent the leg to the backbone chain by selecting first the thigh joint, then the pelvis2 and Edit > Parent (p). If you want to check, there's a picture of the completed skeleton on page 5 where you can see how the thigh connects to the pelvis.

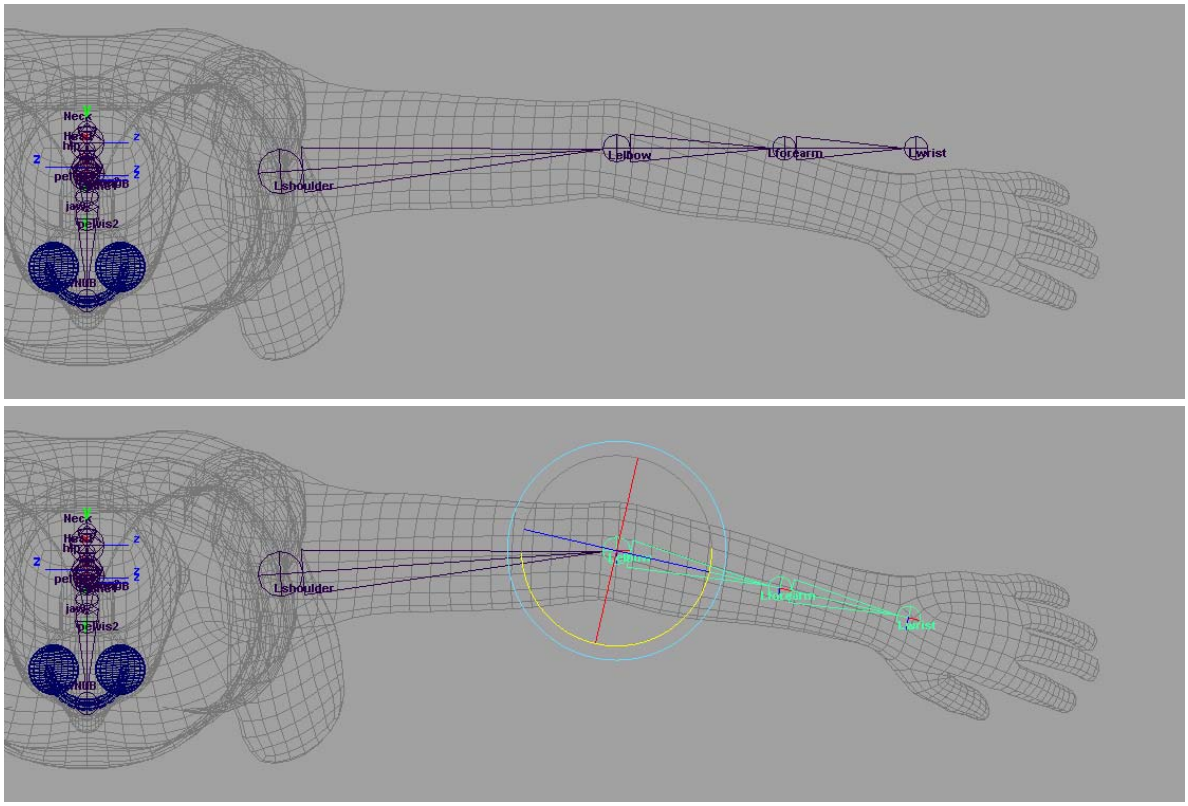




The heel joint is only temporary and will be deleted later.

Create one arm in top view

In top view create shoulder, elbow, forearm and wrist joint. Use the shift-key when creating forearm and wrist so that these are in a perfect straight line from the elbow. Then use the rotate tool to place them inside the arm model. It is important that the elbow-forearm-wrist is a straight line.

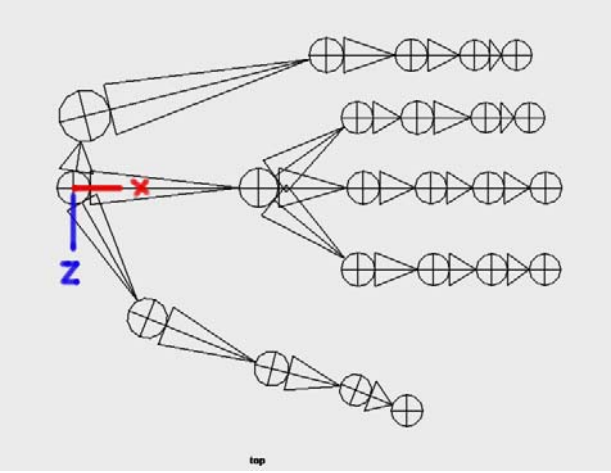


In front or perspective view you must then raise the joint chain from the ground and place it into the arm model. Rotate the chain on the shoulder if you need to point the wrist down.

Scale joints on their x-axis if you need to adjust joint placement. When finished, select the shoulder and freeze transformations.

Create the hand in top view

The hand skeleton is an independent hierarchy from the rest of the body and gets created separately. We'll attach it later using constraints. The structure of the hand should look like the pic on right.



The wrist joint's x-axis must point down the line of the hand. The easiest way to achieve this during joint placement is by starting at the wrist and placing the necessary joints for the *middle* finger before you create any other fingers. Alternatively

you can manually rotate the local rotation axis after joint creation. (See yellow box on page 2)

Pinky and Thumb have a joint at their base to allow cusping of the hand. The three fingers index, middle and ring do not need this so they can all branch off from one common joint.

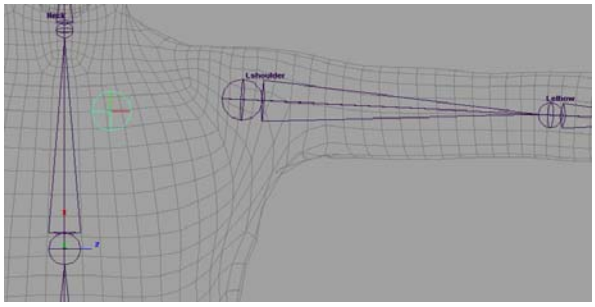
Point constrain the hand to the arm

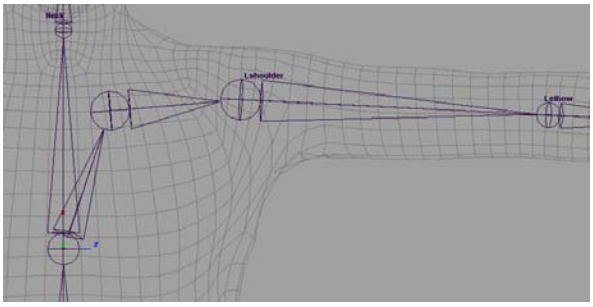
The hand skeleton is *separate* from the rest of the body skeleton. The hand's wrist joint must be in the same location as the armSkeleton's wrist joint. You get both wrist joints in the same spot using a point constraint: select first the armWrist and then shift-select the handWrist and then Constrain > Point. Make sure that the action is reset to its default before you use it, because 'maintain offset' must not be ticked.

As usual, try to stick to rotate and scale when placing the joints into the hand. And reset transforms afterwards. If, however, you need to use the move tool, then ensure that the local rotation axis points correctly afterwards. The correct alignment of the local rotation axis of the fingers is that x should point to the child joint and the z axis usually is the one that the fingers can rotate on when bending.

In single joints (those that are not branching) you can use the Skeleton > Orient Joint command to automatically re-align the x-axis to point down the hierarchy. If you don't do this the X-axis will no longer point down the length of the bone and that means the bone can't rotate nicely around its own axis.

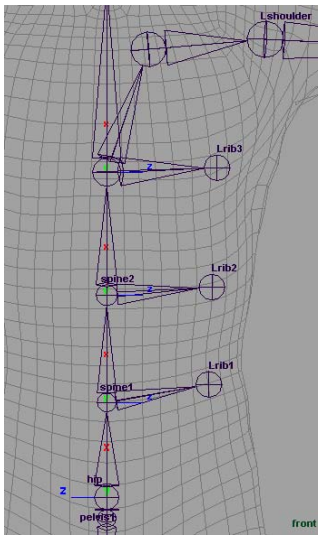
Create a joint for the clavicle in front view





After placing a single joint for the clavicle you then parent the shoulder to it and then parent the clavicle to the top spine segment (not the neck!).

With the clavicle selected: Skeleton > Orient Joint.

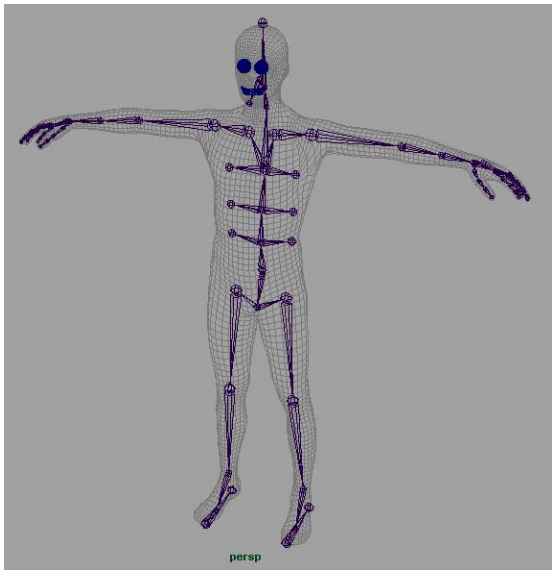


Ribs

It's a good idea to create some ribs. Create a floating joint and then parent it to the spine. The image to the right shows finished ribs for the left body half.

Name joints and mirror the skeleton

As you create joints you should also name them. Symmetric joints (those that exist on the left and right side) should have a prefix like L or R.



Once the left half of the skeleton is completed you can use Skeleton > Mirror Joint to create the right half. Make use of the 'Replacement names for duplicated joints' option in the mirror action to automatically rename all L into R.

Note: after mirroring the hand you need to point-constrain the RhandWrist to the RarmWrist since the constraint won't get mirrored.

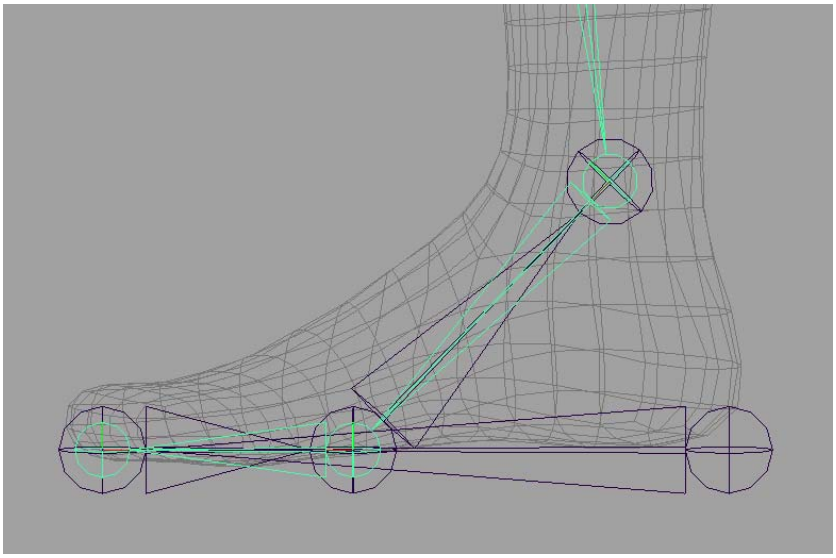
2. Foot Roll

Before applying the first skin we need to create the reverse foot. Remember that earlier I said that the current heel joint will eventually be deleted. We want to create the reverse foot and delete the current heel before applying a skin.

Create the reverse foot rig

- Select the left ankle
- Edit > Duplicate
- Edit > Unparent
- Use the down-arrow keys to travel down the skeletal hierarchy until you reach the heel
- Skeleton > Reroot Skeleton
- Skeleton > Orient Joint
- Rename these four new joints: "LrevHeel", "LrevToe", "LrevBall" and "LrevAnkle"
- Click on the knee joint and use the down-arrow keys to travel down the skeletal hierarchy until you reach the Lheel.
- Delete Lheel

Because there are two toe, ball and ankle joints sitting on top of one another it is a good idea to make them appear at different sizes. I tend to increase the radius of the reverse joints and decrease the radius of the foot joints. Each joint has a radius attribute which you should use the change the joint size.



In this picture the character skeleton is highlighted green and the reverse foot skeleton is dark blue.

Inverse Kinematic chains

- Select the reverse heel and press Ctrl+h. This hides the reverse foot for now.
- Skeleton > IK Handle Tool > Options
- Reset the tool. This turns on the ikRPsolver which is the one we need here (as opposed to the ikSCsolver which we don't want).
- Create 3 IK chains:
    - one from the thigh to the ankle
    - one from the ankle to the ball
    - one from the ball to the toe
- Unhide the reverse foot
- Select LrevAnkle then shift-select the ankle's IK Chain
- Constrain > Point
- Select LrevBall then shift-select the ball's IK Chain
- Constrain > Point
- Select LrevToe then shift-select the toe's IK Chain
- Constrain > Point

Pole Vector Constraints

Pole vectors control the twist of a joint chain around the IK axis. Using pole vectors you can for example control the orientation of the knee.

- Create a locator
- Name it LkneeAim
- Use v-snap to move the locator to the knee joint.
- Now move the locator in a straight axis (probably the z-axis) so that it is well in front of the knee
- With the locator selected Edit > Freeze Transforms

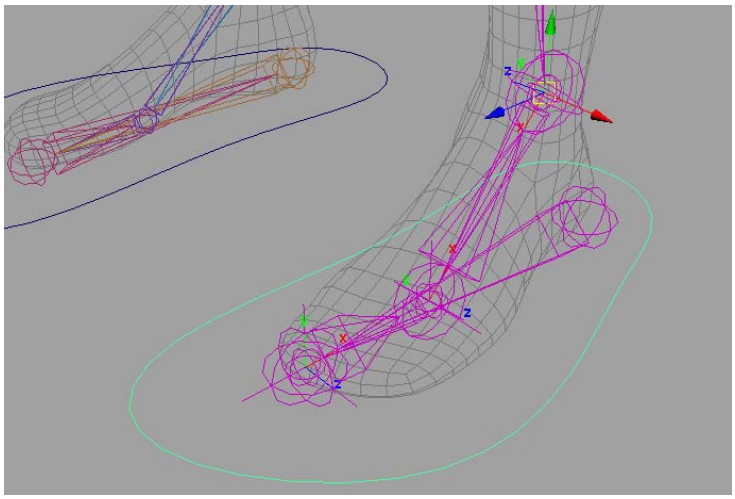
- With the locator still selected Shift-select the ankle's IK handle and then Constrain > Pole Vector.

Now the knee will always point towards the locator.

Foot Control

The foot (and hand) eventually gets animated through a NURBS shape. That one object will hold many of the animatable attributes, like foot roll, or finger movement in case of the hand.

- Create a NURBS circle
- Name it LfootCTRL
- Shape it so it is bigger than the modelled foot – you want to be able to easily select this shape during animation.
- Move the shape's pivot point to an akle joint. Use vertex snapping (V key) but do turn off the viewport's polygon display while you do it – otherwise the pivot point might snap to mesh vertices instead of the ankle joint)
- then **freeze transformations**
- Parent the LrevHeel to the LfootCTRL



Skin

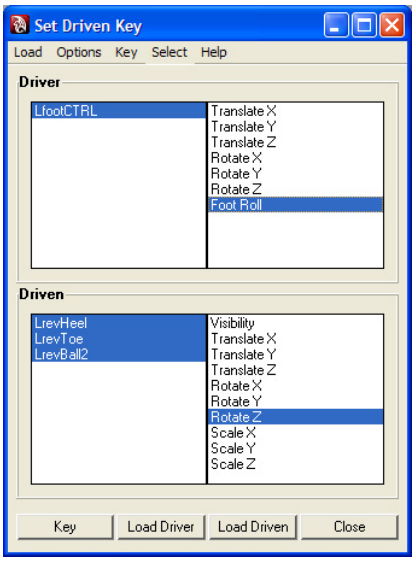
At this stage you should add the first skin. It is very likely that during the process of generating the rig you might have to delete skins and re-skin, so there is no point in painting weights at this stage. We do want to put a quick skin onto the character to see how the mesh will deform, but ignore weighting problems for now. The final skin gets applied only after the entire rig is completed satisfactorily.

- Select the character mesh
- Shift-select the hip joint and the L & R hand wrist joints
- When adding the skin (Skin > Bind Skin > Smooth Bind) set the Max. Influence to **3**

Foot Roll attribute

The LfootCTRL animates the left foot. To get the foot to roll properly during walking we will now add an attribute to this shape called 'Foot Roll'.

- In the channel box highlight the scale and the visibility attributes
- RMB and choose Hide Selected
- Modify > Add Attribute
    - Long name: footRoll (no spaces)
    - Data Type: Float
    - Minimum: -5
    - Maximum: 10
    - Default: 0
- Select these three reverse foot joints: LrevHeel, LrevToe and LrevBall
- Animate > Set Driven Key > Set…
- Now select the LfootCTRL
- In the Set Driven Key window click ‘Load Driver’
- From the driver attributes, choose Foot Roll
- From the driven attributes choose Rotate Z
- Press the ‘Key’ button



- On the LfootCTRL set the foot roll attribute to –5
- Rotate the LrevHeel so that only the ball and toe lift off the ground (as if the foot was just planting itself on the ground during walking)
- In the SDK window press the ’Key’ button

- On the LfootCTRL set the foot roll attribute to 5
- Rotate the LrevBall so that the heel lifts off the ground as if in the process of walking (note: this is the polygon mesh’s heel I’m referring to – not the heel joint)
- Press the ‘Key’ button

- On the LfootCTRL set the foot roll attribute to 10
- Rotate the LrevBall back to 0
- Rotate the LrevToe so that the heel and ball lift off the ground as if the foot was preparing to release itself from the ground
- Press the ‘Key’ button

The foot roll is now finished. You can close the SDK window. Try out the foot roll attribute. If everything worked properly the foot should plant, carry and lift smoothly.

Sideways tilt

A problem occurs when you tilt the footCTRL on its z-axis. If your character is expected to perform this animation then you will need additional pole vector constraints. What happens is that the leg joints do not follow the tilt of the reverse foot’s joints. You can set up pole vector constraints like you did with the knee, seeing as you have ikRPsolvers running through these very bones. Since you already learned how to do this, I want to

mention here and alternative method: you can orient constrain the foot joints to orient themselves with the reverseFoot joints.

- Select the LrevHeel
- Shift-select the Ltoe
- Constrain > Orient > Options → turn on 'Maintain Offset'
- Add
- Select LrevToe
- Shift-select Lball
- Constrain > Orient

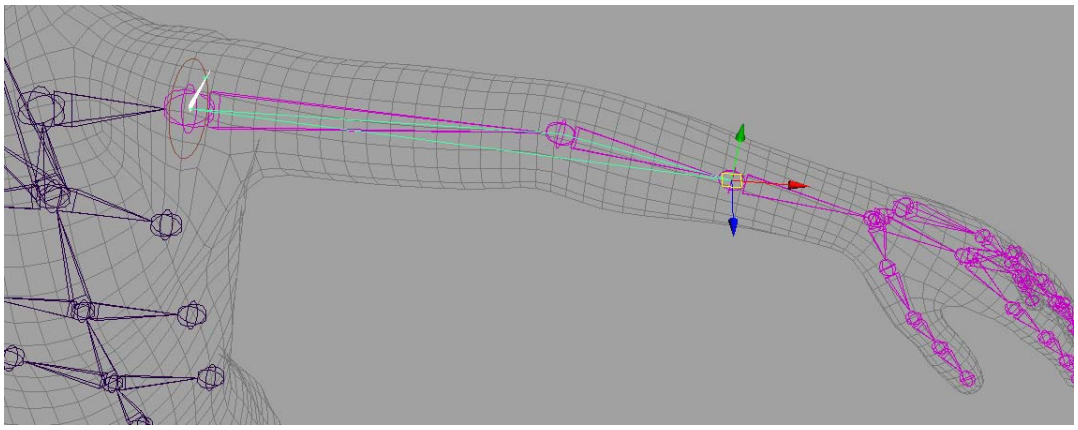
Test it. Sideways tilt should work now.

Und jetzt: repeat all this for the right foot before you move on to the next section.

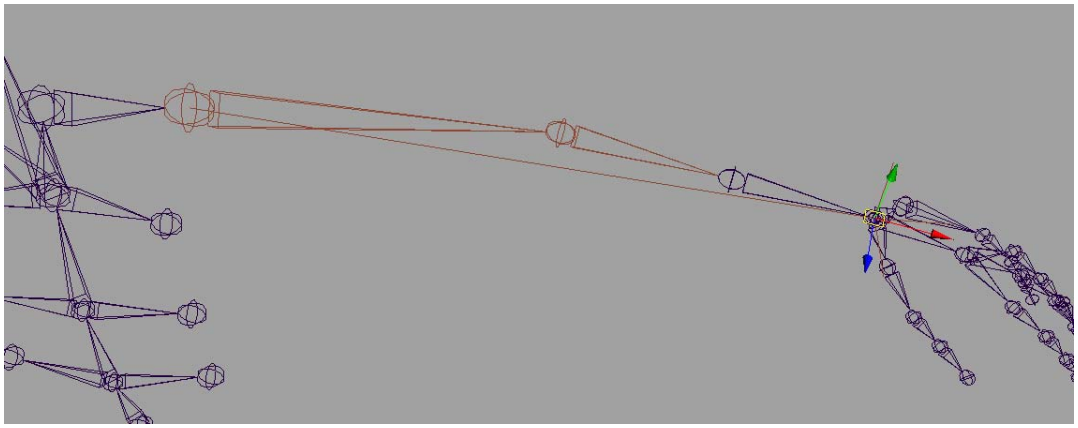
3. Arm rig

IK chain for the arm

- Using the ikRPsolver create an IK chain from the shoulder to the Lforearm



- Select the Lforearm joint
- In the Outliner RMB and choose 'Reveal Selected'. This will unfold the hierarchy until the Lforearm node is displayed in the Outliner
- Parented under the forearm joint you will see the IK effector.
- Select the effector in the Outliner
- Press the Insert key
- Show > Polygons to hide the body mesh (we don't want vertices to get in the way of the next action)
- Holding down the v-key (snap) drag the effector over to the wrist joint. Make sure you drag on the yellow gizmo-centre and not on just one axis.
- Hit the Insert key again to exit 'move-pivot' mode



If you now select the IK handle you can test the arm bending.

Pole Vector Constraints

Just like the knee the elbow needs a control object. Repeat the pole vector setup that you did for the knee using a locator and placing it precisely behind the elbow. Don't forget to freeze transforms on the locator.

Hand Control

Just like the LfootCTRL the hand will be controlled through a NURBS shape. The setup is slightly more complicated here because we need to consider the forearm twist.

- Create a NURBS circle
- Name it LhandCTRL
- Edit > Group

You now have a circle with its pivot at world 0,0,0 and it is grouped with itself, the group and the circle both have their pivot in exactly the same spot. This is very important.

- Select the LhandWrist
- Now add to the selection the hand group (not the LhandCTRL!) – best to use the outliner so you can be sure you selected the group.
- Constrain > Point
- Constrain > Orient
- In the Outliner you must now delete those two constraints again – the group should not remain point nor orient constrained to anything
- Now you can select the LhandCTRL shape and remodel it so that it is considerably bigger than the character's hand. This makes it easy to select during animation. But be sure not to move the pivot point of this shape in the process!

Setting up controlling connections

- Select the LhandCTRL
- Now shift-select the LhandWrist joint
- Constrain > Point

- Constrain > Orient (the hand must not flip now! – if it does check that the group that LhandCTRL is grouped under is in fact properly oriented – see previous step)

The handCTRL now drives the wrist movement and rotation

Controlling the arm movement

- Select LhandCTRL
- Shift-select the ikHandle of the left arm
- Constrain > Point

The handCTRL now moves the arm as well as the hand

Controls for forearm twist

- Open the Expression Editor (Window > Animation Editors > Expression Ed.
- Write this for the expression:
  - Expression Name: characterRigControls
  - Expression: `Lforearm.rx=LhandCTRL.rx/2;`

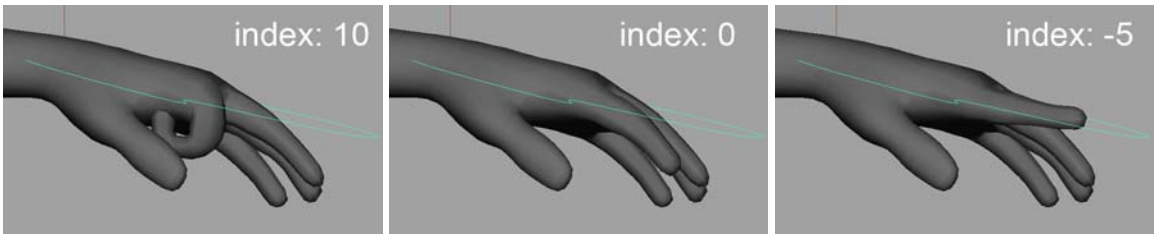
  Careful, expressions are case and space sensitive, so if you use different names make sure you spell the node names correctly
- Click the 'Create' button

If all went well the forearm joint will now spin half the amount of the handCTRL on the x-axis. This will create the proper twisting of the lower arm.

Hand Attributes

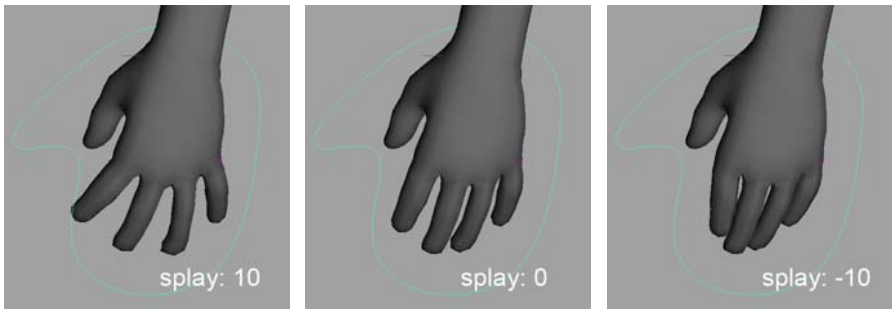
The handCTRL should hold all animatable attributes for finger movement. It does not need to keep scale and visibility attributes.

- On the LhandCTRL highlight the scale and visibility attributes
- RMB and choose Hide Selected
- Create 6 new attributes: thumb, index, middle, ring, pinky and splay. If you want to you can give these attributes limits (I know I would). I recommend –5 to 10 for the fingers and thumb and –10 to 10 for the splay
- Using SDK get the finger attributes to drive the fingers to curl up and in the negative direction the fingers should over straighten.



The fingers need only one axis to bend (the z-axis if everything is set up correctly). The base of the thumb on the other hand might need all three axis controlled in order to rotate properly.

When the thumb and finger attributes are finished you can move to setting up the splay attribute. The splay is meant to rotate only the first finger joint on its y-axis so that the fingers can spread out (+10) or get close together (–10).



4. Master control shape

It's good to create two shapes that control the major movement of the character.

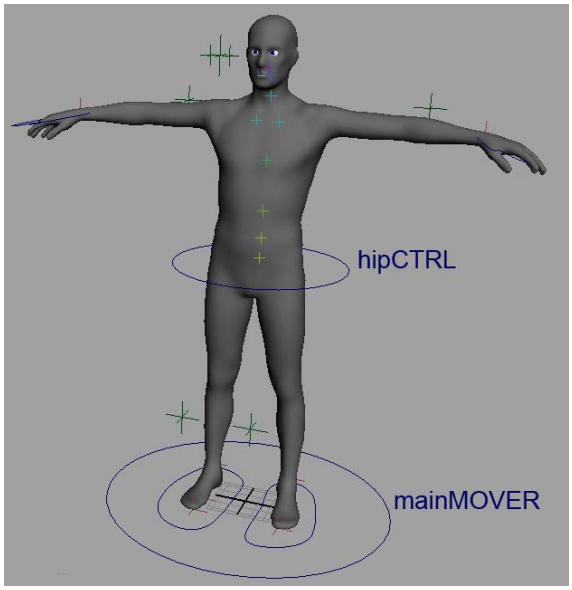
The one I like to call hipCTRL is a NURBS circle to which the hip joint gets point-constrained.

- Create a NURBS circle
- Name it hipCTRL
- Use v-snap to center it on the hip joint
- Scale it large enough to be bigger than the torso
- Freeze transformations
- Select the circle, then the hip joint and Constrain > Point

Now when you move the circle the hip should follow

The one I like to call mainMover is also a NURBS circle

- Create a NURBS circle
- Name it mainMOVER
- Scale it bigger than the stance of the character – so that the footCTRL shapes lie within the circle
- Freeze transforms
- Parent to this shape the masterCTRL, the two footCTRL, the two handCTRL groups (*!! Careful: parent the groups!! not the control shapes*), the eye aim locator and the knee & elbow pole vector locators



When you move the mainMOVER everything should move with it. This is your handle to place the character in any place and orientation in the scene. This gets done only once and this shape does not need to be part of the Character Set (for Character Set see last chapter in this tute).

Please observe that in the image to the right some joints along the torso have their selection handles displayed.

(Display > Transform Display > Selection Handles)

This is a very helpful feature for selecting

objects that might otherwise be difficult to grab.

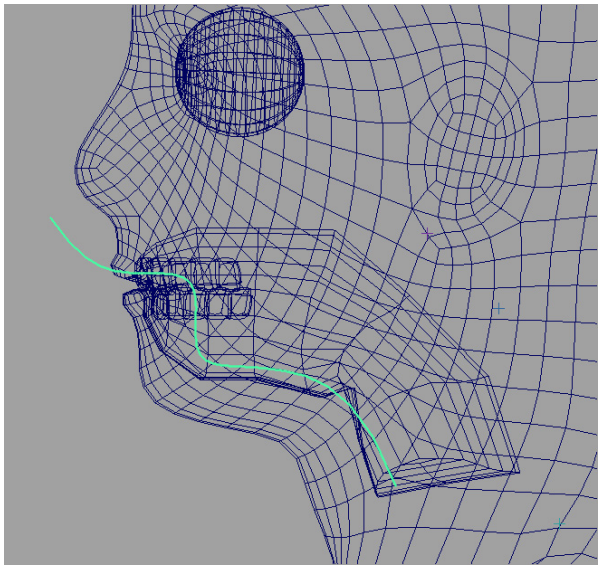
You can also utilise Display > Animation > Joint Labels to get joint names displayed in the viewport. You can see this in action in the pictures on page 2 of this tutorial. You access the label of a selected joint through the Joint Labelling section of the Attribute Editor.

5. Teeth and Tongue

Teeth and gums are modelled as separate objects to the body mesh. The top teeth'n'gums get parented under the head joint and the bottom teeth'n'gums get parented under the jaw joint.

The tongue is trickier:

- Start by creating a poly or NURBS sphere
- Scale it long along the y-axis and wide along the x-axis. So this means that the y-axis is the length of the tongue and the x-axis is the width of the tongue
- Rename the sphere tongue
- Create a NURBS curve from the back of the throat along the bottom of the mouth cavity and moving over the lower teeth out of the mouth
- Name it tonguePath



- Select the tongue, then Shift-select the tonguePath
- Animate > Motion Paths > Attach to Motion Path > Options
- In the options window set Y as the Front Axis and Z as the Up Axis
- Click the Attach button
- Still with tongue and path selected Animate > Motion Paths > Flow Path Object > Options
- In the options window set the Divisions Front to 15
- Tick 'local effect'
- Click the Flow button
- With the tongue selected you can see the list of inputs in the channel box. Click on the input named motionPath1

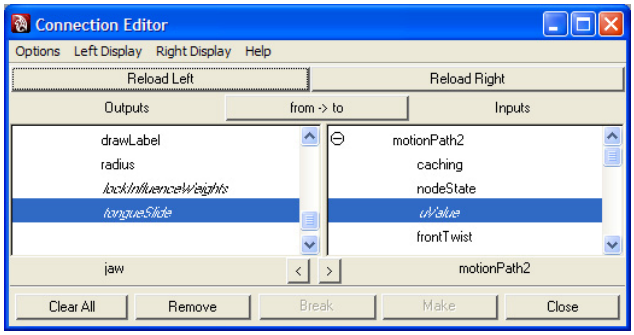
- The field of the U Value should have an orange background colour indicating that it is currently animated. And indeed, if you move the time slider you should see the tongue sliding along the path. At value 0 the tongue should be at the base of the throat and at value 1 the tongue should be well outside the mouth. If your tongue slides in the different direction then you should reverse the path direction: Surfaces menu > Edit Curves > Reverse Curve Direction
- Please break connections of the U Value to remove the current animation on the tongue.

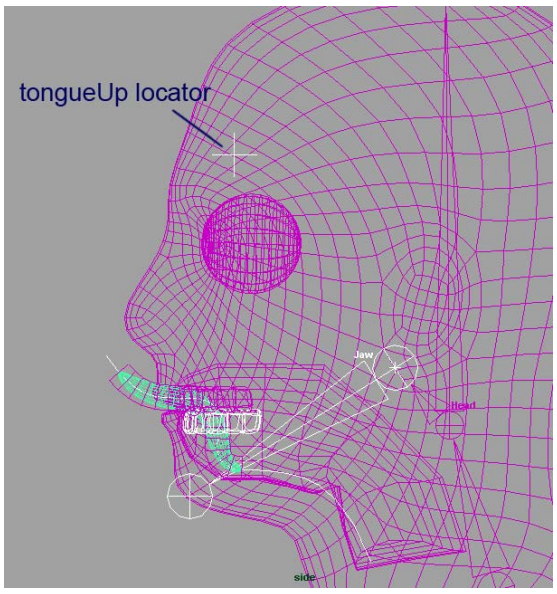
Now you can manually animate the tongue whenever needed. Tongue and jaw animation go hand in hand, so as a last step we will create an attribute on the jaw joint so that you can animate the tongue while the jaw is selected.

If the tongue ever leaves the FFD during animation it is likely you forgot to tick 'local effect'. In this case you can still turn the 'local' attribute on using the FFDs attribute editor.

Tongue slider attribute

- Select the jaw joint
- Create a new attribute of data type float named tongueSlide with a minimum value of 0, a max. of 1 and a default of 0
- Use the Connection Editor to connect the new attribute to the U Value of the motion path. TIP: loading the jaw into the connection editor is easy, but in order to get the motionPath loaded you need to select the tongue, open its Attribute Editor (Ctrl+A), browse the tabs at the top until you have motionPath1 and then click the 'Select' button in the bottom left corner of the Attribute Editor. Now that the motionPath node is selected you can 'Reload Right' in the Connection Editor.





Optimising tongue twist

When the head tilts sideways the tongue will keep being oriented horizontally. We'll now create a look-at target that becomes the new up vector for the tongue.

- Create a locator
- Name it tongueUp
- Place it inside the head above the tonguePath
- Select the tongue
- Ctrl+A to open the attribute editor

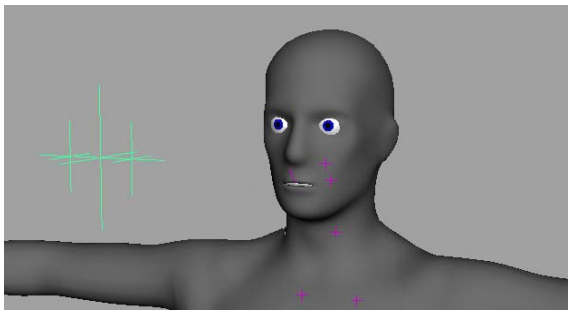
- Browse the tabs at the top until you find motionPath1
- Under World Up Type choose Object Up
- Under World Up Object type in the name of the locator: tongueUp

The tongue will now orient its twist according to where the locator is placed. You need to play a bit with the position of the locator until you get a good place that does not twist the tongue as it slides along its path. Once you're confident that the locator is in a good spot, freeze its transforms and parent it under the jaw bone.

Remember this tutorial creates only a minimalistic rig. We will teach a much better head rig in the "Realistic Head" assignment.

6. Eyes

Eyeballs should be NURBS or polygon spheres. These get parented under the head joint. You can use some locators as look-at targets for the eyes. The eyes then receive aim constrains to make them always point in the locators' direction. Turn on 'maintain offset' when applying the aim constraints! I like to have one central locator that two aimed-at locators are parented to. That way I can animate both eyes with the parent locator or I can choose to animate each eye separately using the two child locators.



7. Blend Shapes

Blend shapes need to be applied before the skin gets applied. Because you just have a quick skin at the moment and you have not yet gone through the trouble of painting good weights, the skin is easily removed (Skin > Detach Skin). If you already have a functioning skin, make sure you apply blend shapes with Deformation Order: Front of Chain or change the input order manually in the 'List of Input Operations' after applying the blend shapes. To get to the 'List of Input Operations' RMB the mesh and choose inputs > all inputs. Here you can MMB-drag nodes to change their order.

8. Character Set

O.K. Final step before this thing is ready for animation. You need to group all animatable attributes under one character node. It's sometimes preferable to have multiple separate character nodes, like one for the lower body, another for the upper body and a third for facial expressions, but in this example we want to create a minimalistic rig. It is up to your personal preference to take this a step further and create multiple character nodes.

Clean up your animatable objects

The animatable objects are any objects in the rig that get directly selected for animation. This includes the hand and foot control shapes, the pole vector locators for knee and elbow, the clavicle joint, all the spine joints, the neck, the hip, the head and jaw joints, the master control shape, the pelvic joint and perhaps even the blend shapes (the last is personal preferences – I'd leave them out for now). All these objects by default have attributes for translation, rotation, scale and visibility. Some have extra attributes that we created in the rig setup, such as the tongueSlide or the footRoll. The cleanup process that you should now go through involves hiding any attributes that are not going to be animated. This includes all the scale and visibility attributes. The pole vector locators don't need their rotation attributes (rotating the locator does nothing for the animation after all). Most joints don't need their transform attributes. You don't need to hide attributes for joints that are driven by other nodes because these joints never get directly animated by you. The arm and leg joints for example can be ignored because they are themselves being animated through an IK chain. The fingers can be ignored because they are driven through the SDK setup you created earlier. The IK chains can be ignored because they are constrained to the animation control shapes. Get it?

So go through every animation handle and every joint that is a likely candidate for direct animation and hide the unnecessary attributes. Do that now. It's worth it.

…

Finished?

Good.

Now select all these objects one last time – every control shape, every joint that is going to be directly animated, etc. and once they are all selected go to Character > Create Character Set > Options

- Reset the settings
- Give the character a name
- Click 'Create Character Set'

NOW you're ready to start animating. Begin by creating a simple animation that tests all deformations and use this to paint good skin weights.