Animator Friendly Rigging

Creating animation rigs that solve problems, are fun to use, and don't cause nervous breakdowns.

Jason Schleifer

CONTENTS

Biped Legs 5

Why Are Legs Important? 6

Requirements for legs 8

Leg Toolkit 9

Foot Roll 9

Create a better footroll 30

Knee Controls 44

Creating the Leg Rig 55

FK Joint Structure 60

IK Joint Structure 63

Result Leg 64

Create the fk anim controls for the fk leg 69

Create the foot ik 79

Foot Control 83

More Leg Controls 107

Create noFlipIK and stretchyIK legs 108

Set up the poleVector leg 134

Animation Checklist 139

Simple controls 140

Rig Requirements – ALL 156

Connect the leg geometry to the scale 163

Author Biography 172

Included Mel Scripts 173

js_addAllAttrsToStretchTSL.mel 173

js_addHalfJoint.mel 173

js_addNormalizeScale.mel 173

js_addSuffixWin.mel..... 173
js_addWorldScaleToDistance.mel..... 173
js_attrDraggerSingle.mel..... 174
js_autoRotate.mel..... 174
js_connectBlend.mel..... 175
js_connectBlendUI.mel..... 175
js_copyPivot.mel..... 175
js_copySetDrivenKeyUI.mel..... 175
js_createCurveControl.mel..... 175
js_createIkStretch.mel..... 175
js_createIkStretchUI.mel..... 175
js_createMeasureTool.mel..... 176
js_createSkelGeo.mel..... 176
js_createStretchSpline.mel..... 176
js_createStretchSplineUI.mel..... 176
js_createTwistySegment.mel..... 177
js_createTwistySegmentUI.mel..... 177
js_cutPlane.mel..... 177
js_error.mel..... 178
js_findOrError.mel..... 178
js_findRootOrError.mel..... 178
js_getOptionVar.mel..... 178
js_getOptionVarString.mel..... 178
js_getStretchAxis.mel..... 178
js_grepRename.mel..... 178
js_grepRenameUI.mel..... 178
js_hashRename.mel..... 179
js_hashRenameUI.mel..... 179
js_loadSelectedIntoButtonGrp.mel..... 179

js_multiConstraintSnapUI.mel....179
js_matchObj.mel....180
js_matchObjUI.mel....180
js_multiConstraintSnapUI.mel....181
js_pivot.mel....181
js_quickAddAttr.mel....181
js_renameGeoAsParent.mel....182
js_replaceHash.mel....182
js_rotationOrderWin.mel....182
js_selectFromRoot.mel....182
js_setDrivenKeyAutoDriveUI.mel....182
js_setDrivenKeyLength.mel....182
js_setUpMultiConstraint.mel....183
js_setUpMultiConstraintUI.mel....183
js_snapObjectToConst.mel....183
js_splitSelJoint.mel....183
js_splitSelJointUI.mel....183
js_toggleRef.mel....184

Biped Legs

In **Animator Friendly Rigging Part III**, we began extended JJ's rig to handle arms and fingers. While doing this we explored forward and inverse kinematics, setDrivenKeyframing, expressions, utility nodes, and creating complicated hierarchies in order to get interaction that's easy for the animator to use.

We're going to continue this exercise in Part IV by rigging the legs of JJ. We will then take a look at some of the other important elements of creating animation rigs: Mirroring controls, Snapping (fk to ik and more), UI Creation, additional rigging ideas, and rigs for props.

But first, we'll extend our JJ rig so he has legs and feet that are a joy to work with.

Why Are Legs Important?

Providing more than just locomotion, legs convey force and drive

In the movies folder you'll find some example movies of legs. Take a look at these examples and note the types of movements that are occurring.

In the first example you can see Bryce running up to a wall, leaping on to it, and running off. His leg here are doing a number of things that are seemingly basic, but actually contain some complicated motion
.
On the run up, it's just a basic foot plant, push, swing forward, nothing too complicated. Then he plants his leg right in front of the wall and pushes up, extending it totally straight. While doing this, he uses his momentum to swing his left leg up and onto the wall, planting with his *knee*. His right leg then swings up totally straight, while he pivots his body on his left knee.

Then, sitting on his bottom, he swings his legs over further, planting his heels. Finally, he spins around onto his toes again and pushes off to keep running.

As you can imagine, there are quite a few complicated actions going on here. The ability to use multiple pivot points on the feet would come in quite handy, as well as having a way to lock down the knee. In addition, we need to make sure the foot rig itself is solid enough to spin around 270 degrees without running into gimbal lock problems.

Figure 1 - Bryce running and jumping up a wall

In the second example, you can see Carlos doing pushups with his knees locked to the ground. Again, having a method of locking down the knees is extremely useful here.

Figure 2 - Carlos doing pushups with his knees locked to the ground

Take a look at the other movies and watch closely what the legs are doing. Go out yourself and try running around a bit and see what your legs do. Analyze here the force of the movement is coming from. Is it from pushing off the ground? Is your hip pulling your foot? Is it swinging from the thigh? There are a number of motivators for creating motion with your legs, and it's important to understand exactly what's driving everything.

Figure 3 - legs can do many things on a character, not just hold them up! They can be quite expressive.

Requirements for legs

- Requirement #1 – **Inverse Kinematics**. About 95% of the time your character's feet will be on the ground. Thus, inverse kinematics on the feet is very important.
- Requirement #2 – As you can see from the example movies, having the ability to **lock the knee** on the ground is also important.

- Requirement #3 – Sometimes characters will need to let the legs flow free, so **forward kinematics** is a biggie here.
- Requirement #4 – One thing that animators spend most of their time doing is trying to get the feeling of weight. A way to help out with this is to have a good foot pivot and foot rolling system that allows them to get the weight of the character on the ball of the foot. So we need to develop a really easy to use **foot roll** and **foot pivot system.**

Leg Toolkit

As before, we're going to go through a series of examples where we explore various controls on our foot and leg rigs. This gives us an opportunity to experiment, to practice, and to try options *before* committing to them.

Foot Roll

Back in the early days of 3d computer graphics animation there was no way to have an easy foot control. Quite often people would have to use forward kinematics to position their legs--something not too difficult if your character was running like a crazy person flailing their legs all over the place, but can you imagine trying to do something subtle like a weight shift? Every time you would move the body you would have to adjust the upper leg, then the lower leg, then the foot, then the toe.. and if it didn't line up *exactly*, even for *one frame* the believability would be lost.

Talk about frustration! Luckily we now have the ability to create all sorts of complicated networks of nodes to lock the foot down. But what's the *best* way to create a foot rig? There are a number of ways of creating a foot rig, all of which will work just fine. What we'll explore here is something that is incredibly powerful in that it allows the animator total control over what the foot is doing,

but also gives some automation for the complicated (but repetitive) task of performing a foot-roll.

Let's start off with a similar setup to what we had on the hand rig, where we created a hierarchy that allowed us to lift the palm and twist it side to side.

1. **Create a new scene**
   - Choose **File > new**
2. **Create a leg structure**
   - Go to a side view.
   - Choose **Skeleton > Joint tool**
   - Create a leg structure that looks like the following.

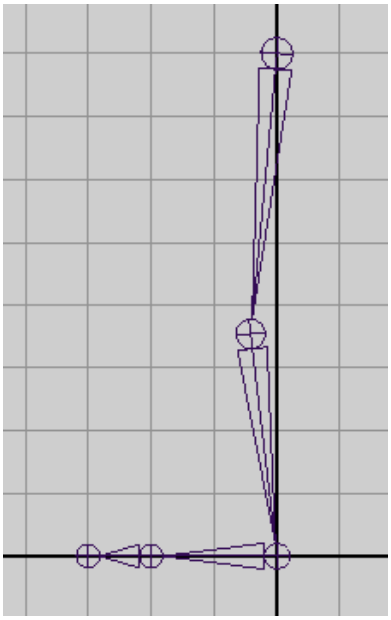

Figure 4 - joint structure for a leg

3. **Add Geometry to it**
   - Select all the joints

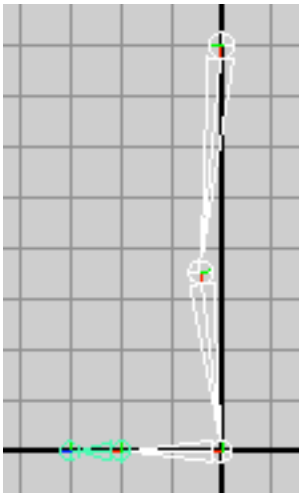

Figure 5 - all the joints selected

- Click the **Add Geo to Selected Joints** button on the **animator Friendly Rigging Shelf**

Figure 6 - geometry created

4. **Scale feet**
   - Select the two pieces of geometry for the feet
   - Scale them so they're wider and foot-shaped.

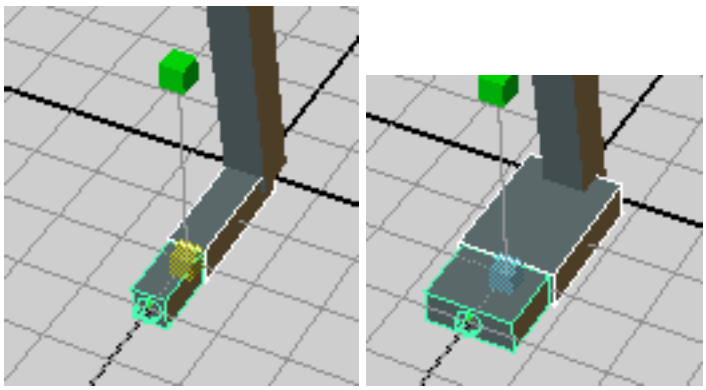

Figure 7 - scaling the feet

5. **Create ikSolver**
   - Choose **Skeleton > IK Handle Tool > Option Box**
   - Set the **current solver** to **ikSCsolver**

Figure 8 - use the ikSCsolver for the legs

   - Click on **joint1** and **joint3** to create an ikHandle for the leg.
   - Click on **joint3** and **joint4** to create an ikHandle for the foot.
   - Click on **joint4** and **joint5** to create an ikHandle for the toe

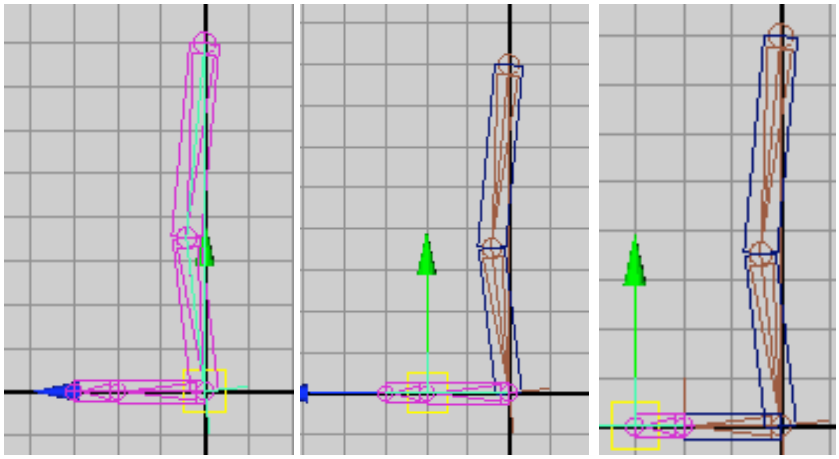

Figure 9 - creating ikHandles for the leg

6. **Rename joints and ikHandles**
   - Rename the joints and ikHandles to match the following images:

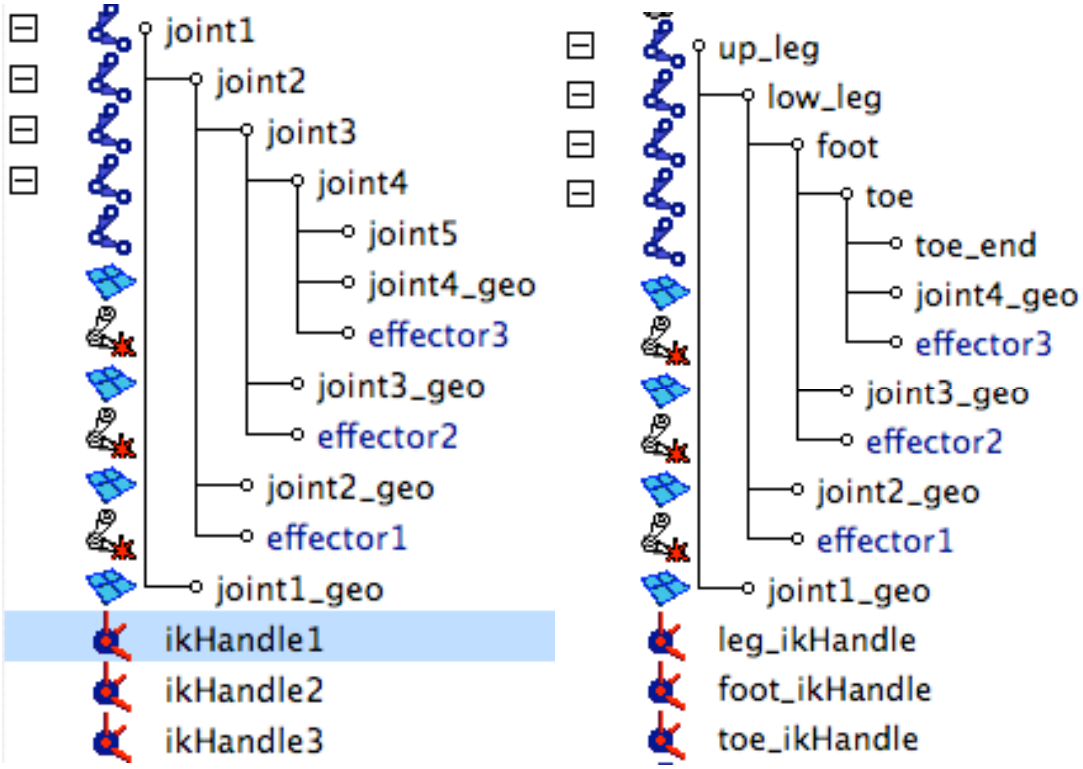

Figure 10 - renaming the joints and ikHandles

7. **Create a locator for the whole foot**
   - Choose **Create > Locator**
   - Rename the locator **foot_loc**
   - Parent the three ikHandles under **foot_loc**

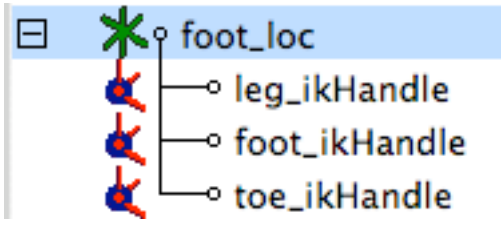

Figure 11 - foot_loc is now the parent of all the ikHandles

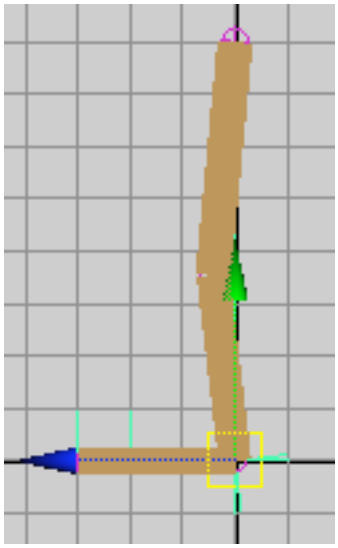

Figure 12 - foot_loc is located at the ankle and is the parent of the ikHandles for the foot

8. **Create a locator to lift the ball of the foot**
   - Choose **Create > Locator**
   - Move the locator the location of the ball of the foot

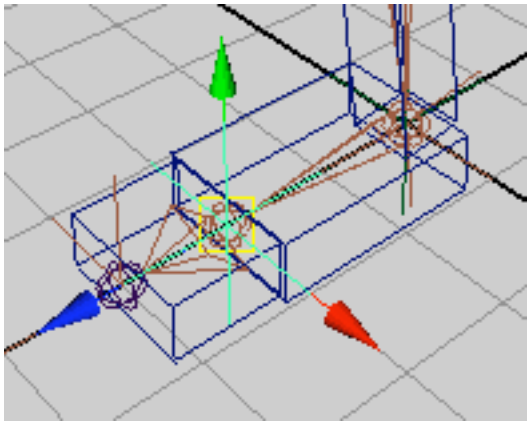

Figure 13 - locator moved to the location of the ball of the foot.

   - Rename **locator1** to **ball_loc**

Figure 14 - renaming locator1 to ball_loc

9. **Parent ball_loc under foot_loc**
   - Parent **ball_loc** under **foot_loc**

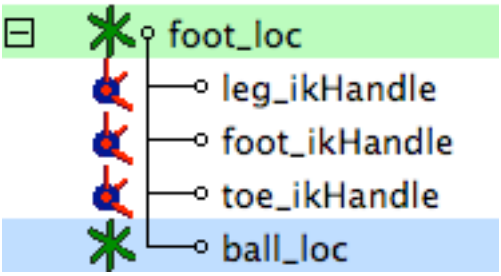

Figure 15 - parenting ball_loc to foot_loc

10. Parent leg_ikHandle under ball_loc

- Parent **leg_ikHandle** to **ball_loc**

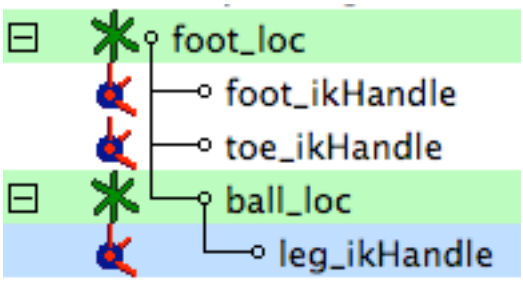

Figure 16 - parenting leg_ikHandle to ball_loc

Now when you rotate **ball_loc**, you can see that it lifts the heel of the foot as you would expect.

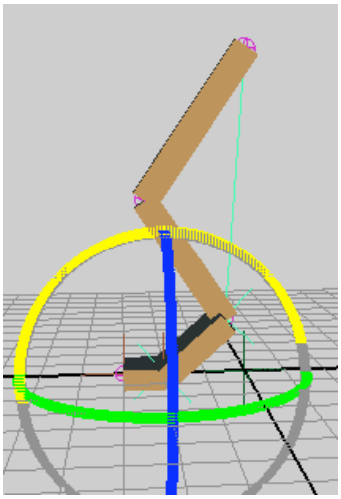

Figure 17 - foot now rotates correctly

Our next step is to create a similar lift for the toe of the foot.

11. Create toe lift locator

- Choose **Create > Locator**
- Move the locator to the position of the toe

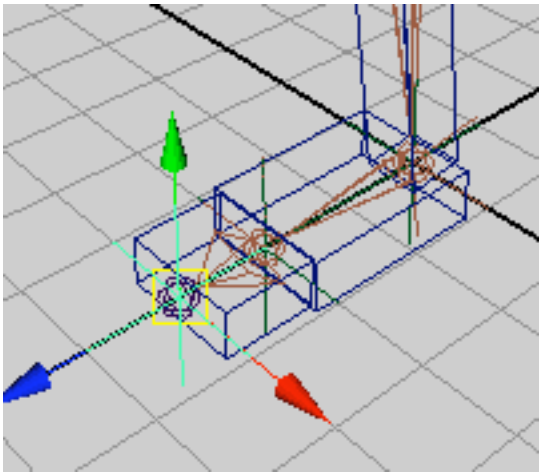

Figure 18 - locator moved to the position of the toe

- Rename **locator1** to **toeLift_loc**

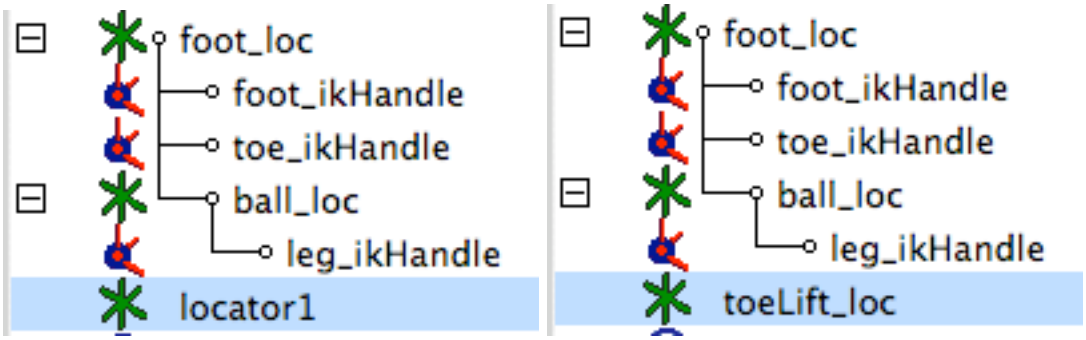

Figure 19 - locator1 renamed as toeLift_loc

12. Parent foot_ikHandle and toe_ikHandle and ball_loc under toeLift_loc

- Parent **foot_ikHandle, toe_ikHandle,** and **ball_loc** under **toeLift_loc**

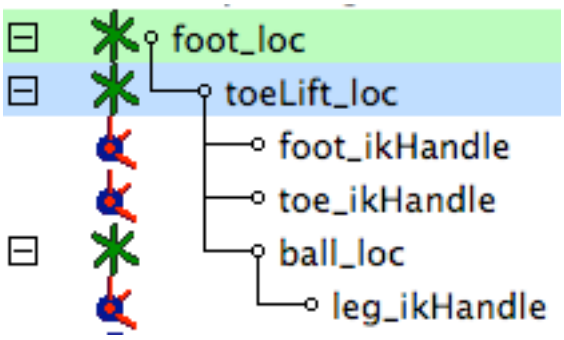

Figure 20 - foot items parented under toeLift_loc

Now you can use toeLift_loc to lift the entire foot by the toe. This is the first part of our foot-rolling system. But currently it only rolls forward, so we need to add the ability for the foot to rest back on its heel.

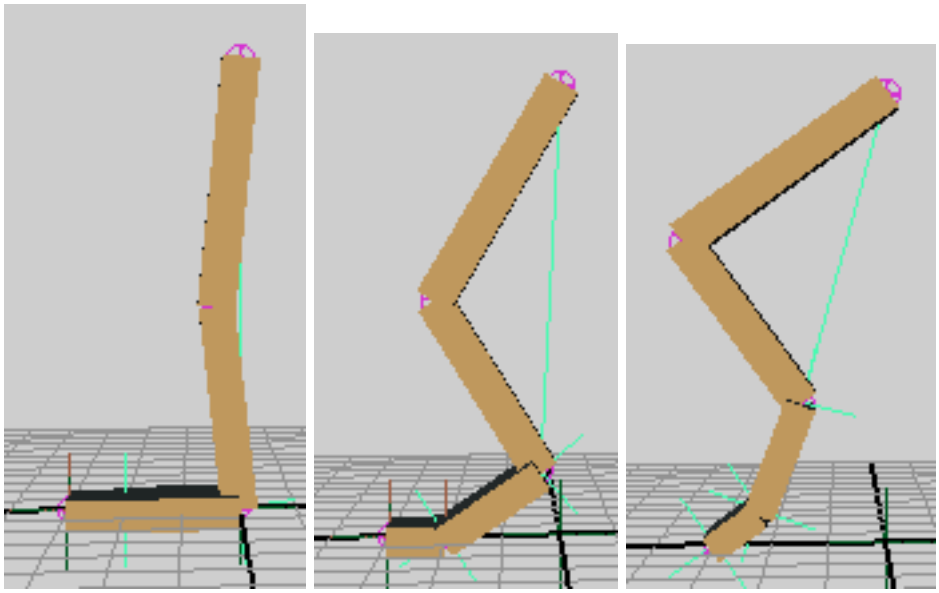

Figure 21 - foot rolling forward using ball_loc and toeLift_loc

13. Create a heel locator

- Choose **Create > Locator**
- Move the locator to the location of the heel
- Rename the locator to **heel_loc**

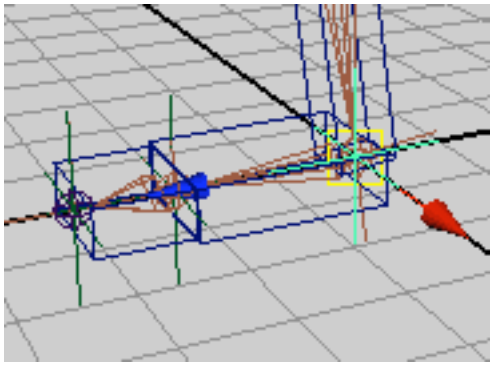

Figure 22 - heel locator positioned correctly

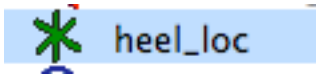

Figure 23 - locator1 renamed to heel_loc

14. Parent heel_loc

- Parent **heel_loc** to **foot_loc**
- Parent **toeLift_loc** to **heel_loc**

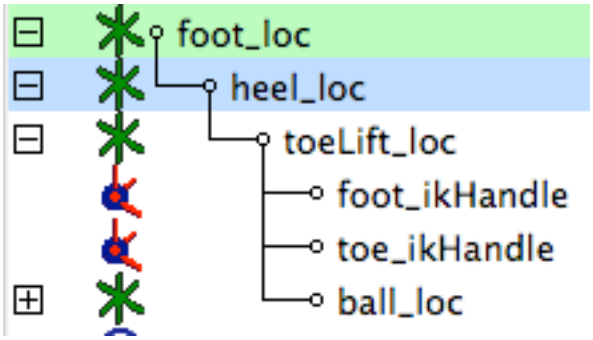

Figure 24 - heel_loc is now the parent of toeLift_loc an the child of foot_loc

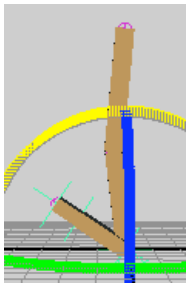

Figure 25 - foot rotating up on the heel

We now have a system of locators where we can lift the foot on it's heel, lift the heel up onto the ball of the foot, and lift the whole foot on it's toe.

But imagine trying to animate with this system, it would drive you crazy! There are too many things to try and select, so just as we simplified things for the hand rig where we added attributes to the hand control for bend and side, we'll do the same thing for the foot.

15. Create attributes on foot_loc for heel lift, ball_lift, toe_lift

- Select **foot_loc**
- Click on the **Quick-Add Attributes** button in the **Animator Friendly Rigging** shelf

- Enter **heel_lift ball_lift toe_lift**

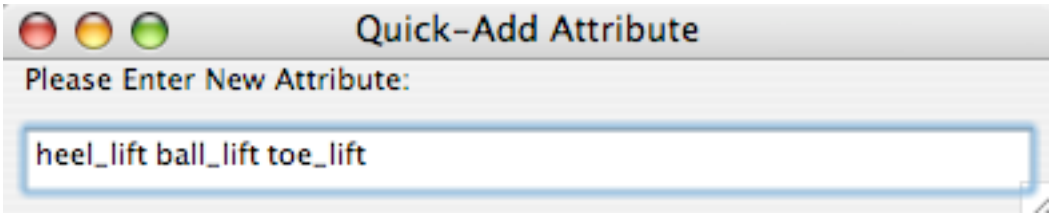

Figure 26 - attributes to add to foot_loc

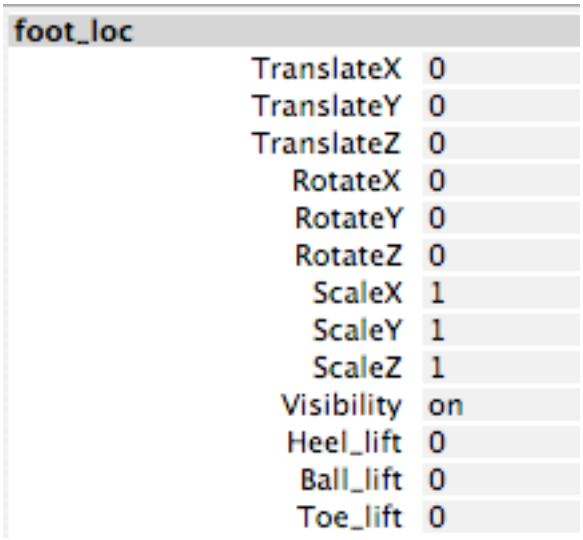

Figure 27 - attributes added to foot_loc

16. Connect heel_lift, ball_lift, and toe_lift

- Select **foot_loc**
- Choose **Window > General Editors > Connection Editor**
- Click **Reload Left**
- Select **heel_loc**
- Click **Reload Right**
- Connect **foot_loc.heel_lift** to **heel_loc.rotateX**

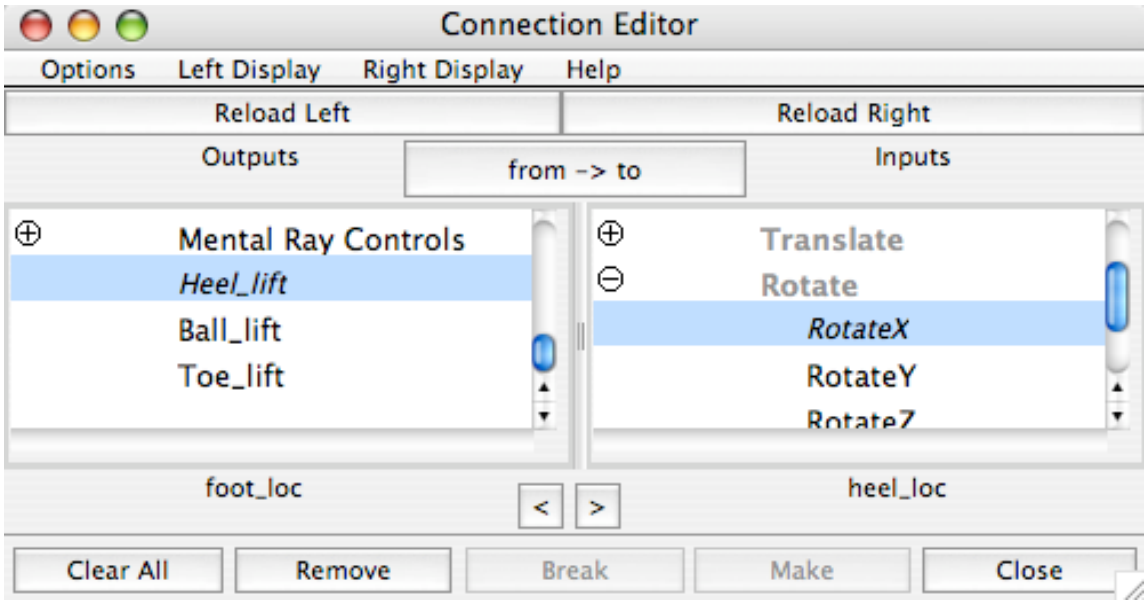

Figure 28 - connecting heel_lift to heel_loc.rx

- Select **ball_loc**
- Click **Reload Right**
- Connect **foot_loc.ball_lift** to **ball_loc.rotateX**

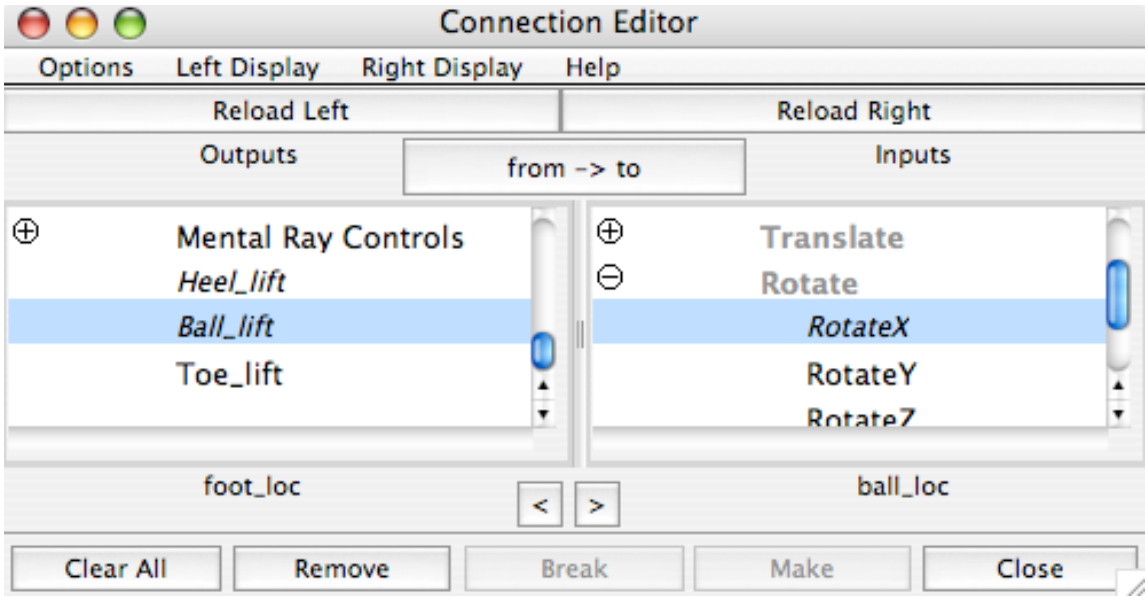

Figure 29 - foot_loc.ball_lift connected to ball_loc.rx

- Select **toeLift_loc**
- Click **Reload Right**
- Connect **foot_loc.toe_lift** to **toeLift_loc.rotateX**

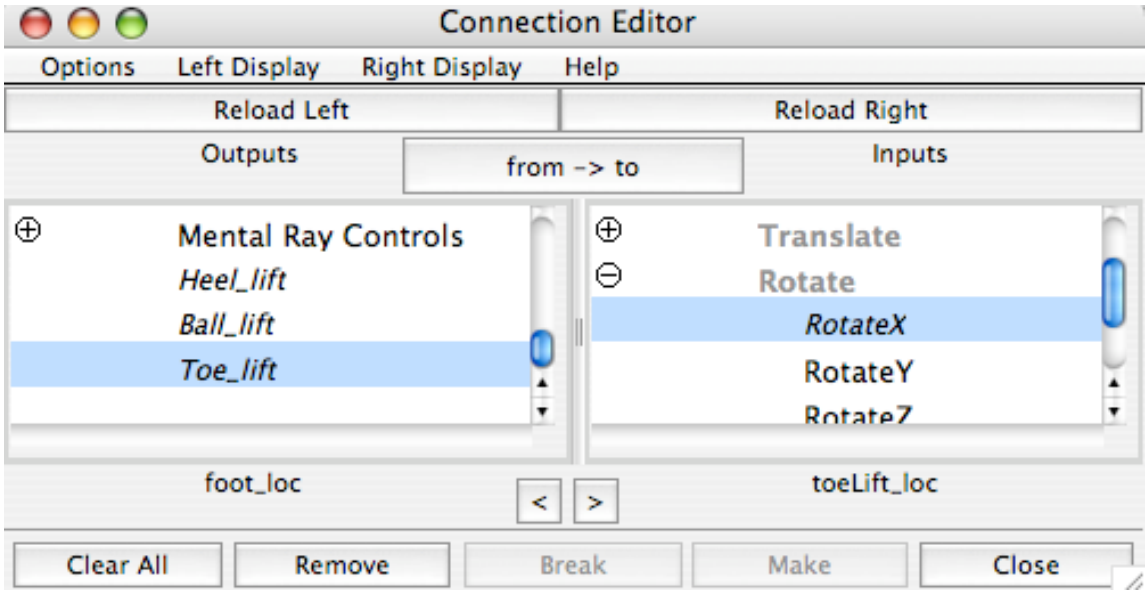

Figure 30 - connecting toe_lift to toeLift_loc.rx

Now you can use foot_loc to animate the heel lift, the ball_lift, and toe_lift to lift the foot.

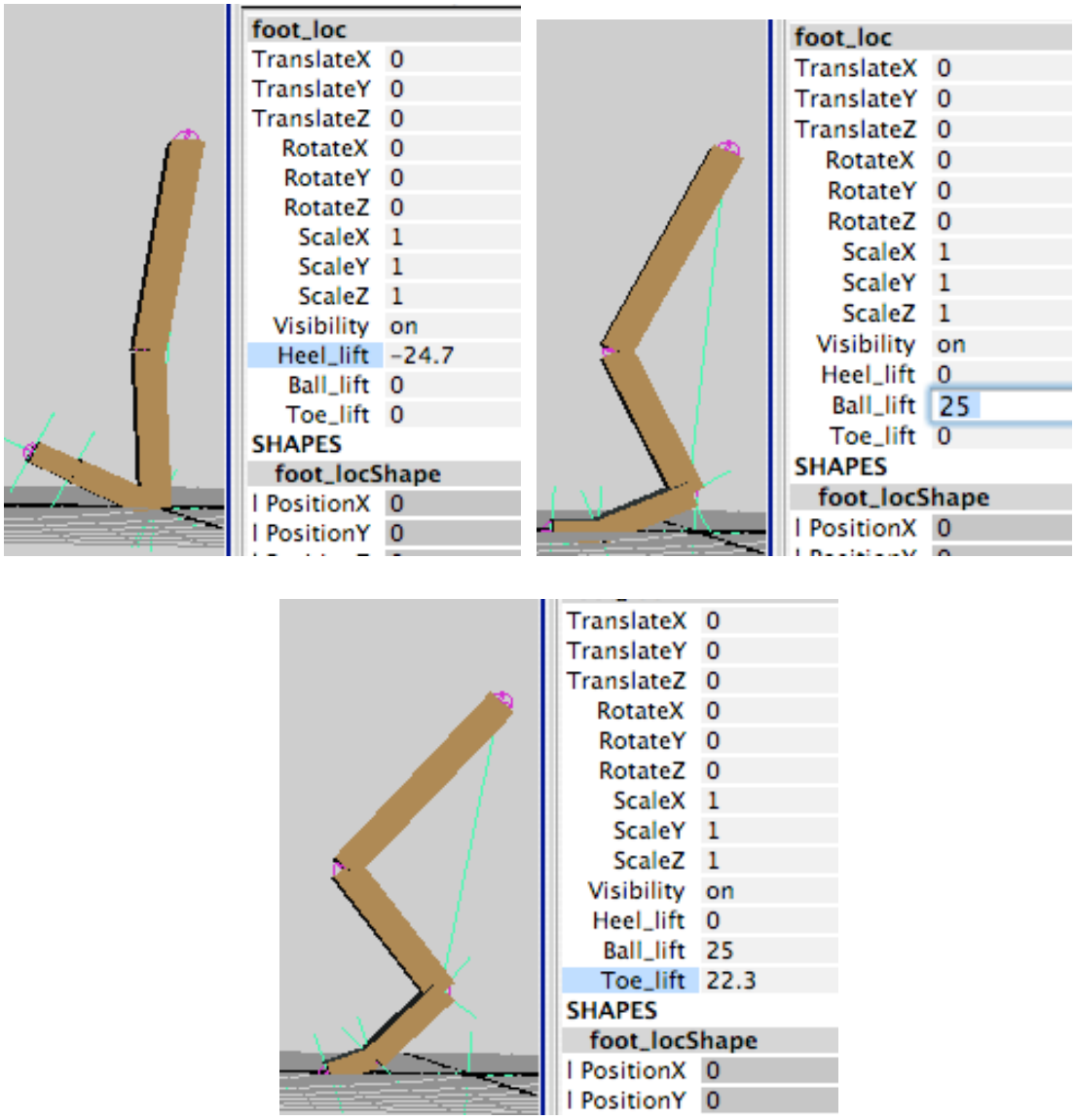

Figure 31 - using the attributes on foot_loc to animate the foot rolling.

Save file as: **foot_toolkit_v1.ma**

While powerful, this is still a bit of a pain to work with for the animator, especially with the **heel_lift.** Think about it. It's not possible to have the foot up on its heel at the same time that it's up on the ball or on its toes. Why not make a slider to handle it, just like we did with the side-to-side control on the palm of the hand?

Want to learn more? Purchase the entire DVD at:

http://jasonschleifer.com