Animator Friendly Rigging
Part 3b

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

Jason Schleifer

CONTENTS

Biped Arms

In Animator Friendly Rigging Part IIIa, started working on some techniques for rigging a character's arms for animation. We focused on forward and inverse kinematics, looked at stretching the arms, figured out how to lock the elbow, reduced counter-animation, and more.

This discussion and development of the Biped Arm Toolkit continues in Part IIIb, where we will explore twisting the arm, creating a powerful shoulder control, and finally applying all these techniques we explored to our actual JJ character! Exciting, isn't it?

First, we must review!

Why Are Arms Important?

Remember the following statement:

Extremely expressive, arms are integral to defining a character's personality

It's important to make sure that your character can express the most subtle bits of tension, to the most wild exclamations of explosive frustration, to the most loving caress.

Requirements for arms

- Requirement #1 – Ability to gesture. Most characters move their arms when they talk. They bring them up, wave them around, throw them this way and that.. they're incredibly expressive, and very helpful for getting a point across. The easiest way to animate that type of motion is with Forward Kinematics, because it allows for nice clean arcs and easy manipulation. Thus, we will need **Forward Kinematics** for general gesturing

- Requirement #2 - Other times characters will place their hands on the ground, or on the table, or on some other object. In these cases, we want to make sure the hand is locked off.. so we'll choose **Inverse Kinematics** for situations like this.
- Requirement #3 - If a character places their arm on a table, we want to ensure that their elbow is locked in that position. It can be extremely frustrating for an animator if they have to continually adjust the position of the elbow every time they move the torso. Therefore, we need some sort of **Elbow Locking** for ability to place elbows on table

- Requirement #4 – if a character has their elbow on the table and they're holding a glass of wine, you want them to be able to wave that forearm around very easily. The easiest way to do this is, again, with forward kinematics. Thus, we need some way to lock the elbow to the table but still use **Forward Kinematics** on the lower arm.
- Requirement #5 – In addition to waving their arm around, sometimes you want the character to lock their hand down. Take, for example, a character with his elbow on the table and his head in his hand. You will probably want to constrain the hand to the head, and the easiest way to do that is with **Inverse Kinematics** on the hand.
- Requirement #6 – You can't have an expressive or biomechanically correct arm without some sort of **Shoulder control**
- Requirement #7 – As part of making it easy to get any pose the animator wants, it is sometimes necessary to have the ability to move the whole arm where you need it.. dislocating the arm from the body. Thus, we need to allow for the **translation** of the **shoulder control**.
- Requirement #8 – A common problem I run into when animating is the fact that I'll have some great animation on my arm, but need to make minor adjustments to the torso. As soon as I move the torso, the arm will change orientation because it's a child of the torso's motion. This can be incredibly frustrating, and makes it difficult to get shots finished when you continually have to go back and forth. So, we're going to add the ability to have the arm's rotation **independent** from the shoulder.
- Requirement #9 – Finally, sometimes it's necessary to stretch the arm in order to get the exact pose you require. Thus, we're going to allow for **stretching** in both the **fk** and **ik** arms.

In Animator Friendly Rigging Part IIIa, we covered many of the requirements above. Now, let's finish the rest!

Arm Toolkit - Continued

Multiple Rig Inputs

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

To start where we left off, we're going to be looking at a way to handle twist most effectively. Remember the diagram below from the previous course:

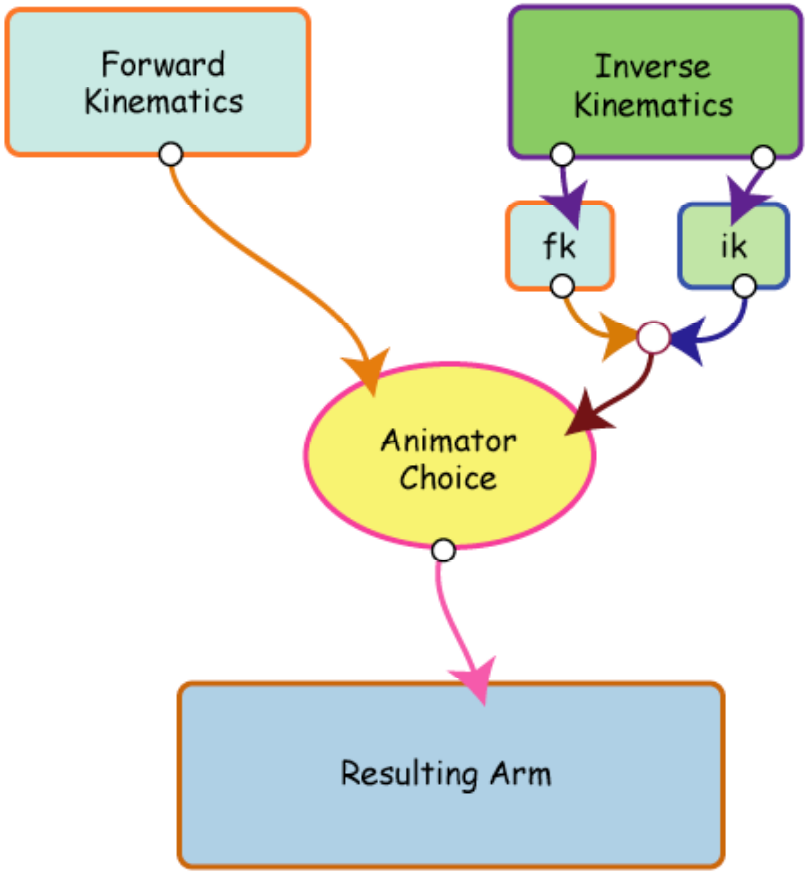

Figure 120 - Many rigs, animator chooses what rig works best for any situation

This diagram represents our goal to allow us to create as many different types of control structures as we need, in order to drive one resulting skinning rig, or

"resulting arm" in this case. What this means is we can start out with a single forward kinematics rig. Then, when we decide we need a fancy IK rig, we can simply plug that into our system and let the animator decide what type of *control* they want. If they want FK, they animate with that. If they want IK, they animate with that. It doesn't matter -- the *goal* is to make it so either rig can drive our resulting skeleton and therefore our deformations.

So what is the easiest way to connect the display arm to all these different types of rigs? The best thing to do is look at what common attributes all the rigs have, and then base our connections on those.

The common denominator for each rig is the joint structure of an **up_arm, low_arm, and hand.**

So if we set up an arm rig that will just attach to the up_arm, low_arm, and hand, we can ensure that the rig will do the right thing no matter what the animator decides should control it.

Handling Twist

We don't just want geometry that will attach to whatever joint we specify, we want geometry that will stretch with the scale changes from up_arm, to low_arm, to hand, and we want something that is going to *twist* correctly.

Remember the importance of looking at *reference* material. If we actually take a look at a person's upper arm and forearm as it moves around, you'll see that the skin doesn't just rotate 100% with the upper arm and 100% with the hand, it actually twists along the joint a bit. This is due to all the muscles and how the skin is attached.

For example, in a real person their forearm is actually made up of two bones, a *radius* and an *ulna.*

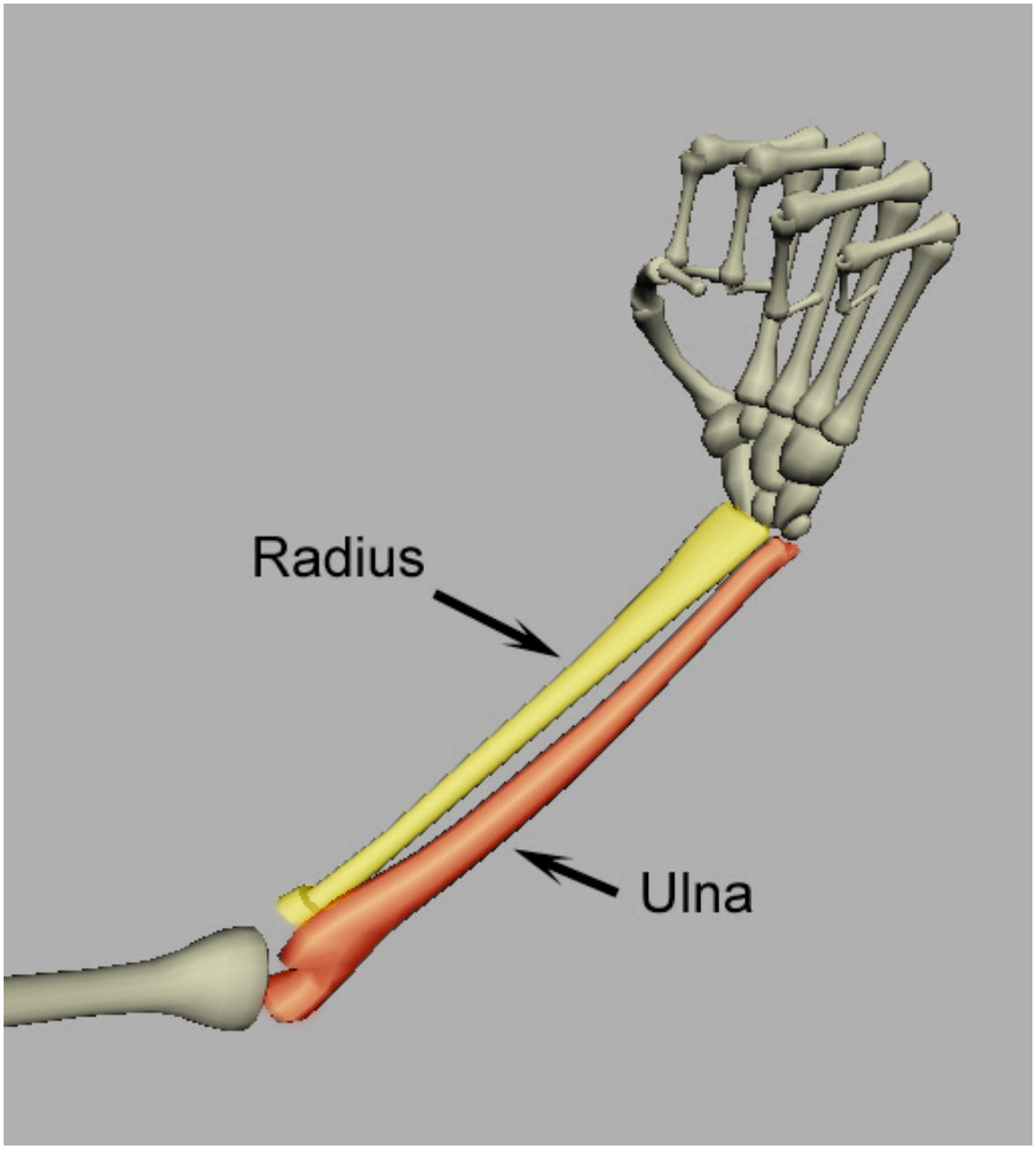

Figure 121 - radius and ulna bones in the forearm

As you twist your wrist, the radius wraps *around* the ulna, pulling the skin.

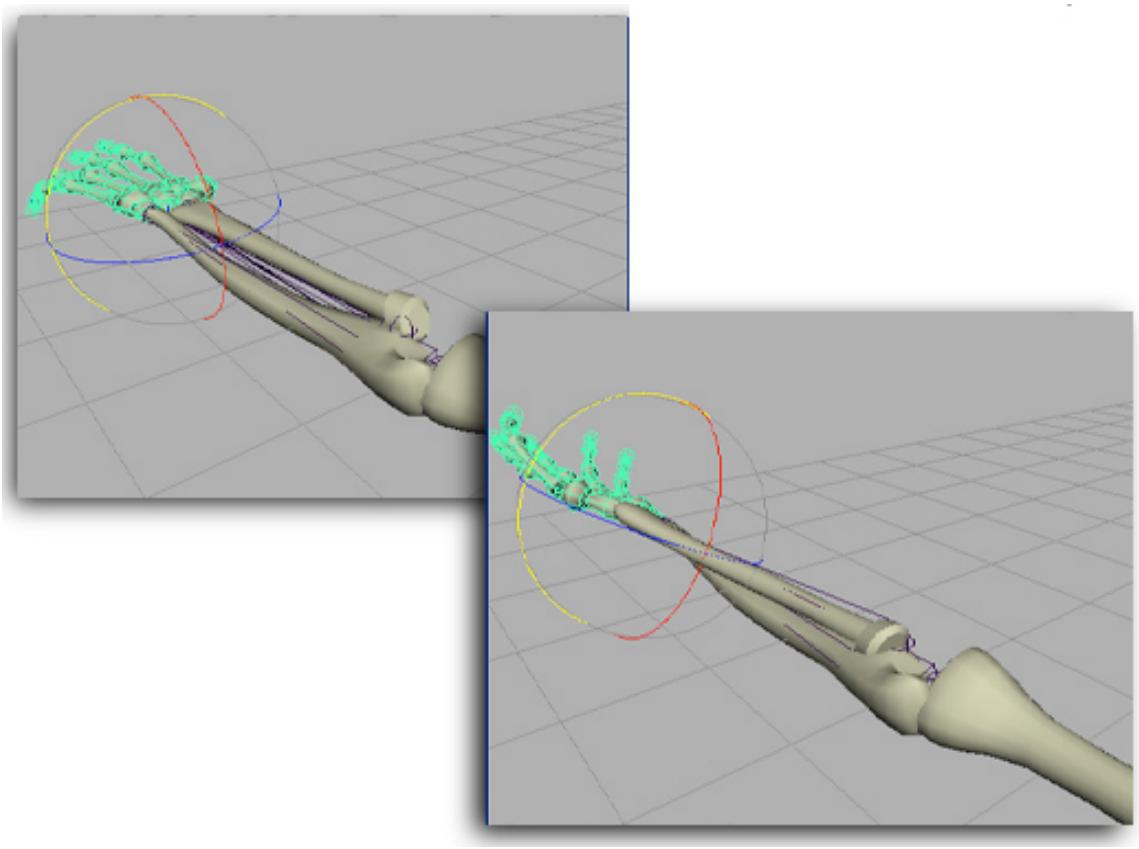

Figure 122- forearm twisting

Look at your upper arm when you rotate our elbow in and out and notice that it works in a similar way. In this case, it's not two bones rotating around each other, it's the muscles pulling on the skin, but in effect the top doesn't rotate as much as the base.

We can simulate this interaction by creating a series of joints that travel along the length of the forearm (or upper arm), and twist based on a percentage of how much the wrist (or elbow) is twisting.

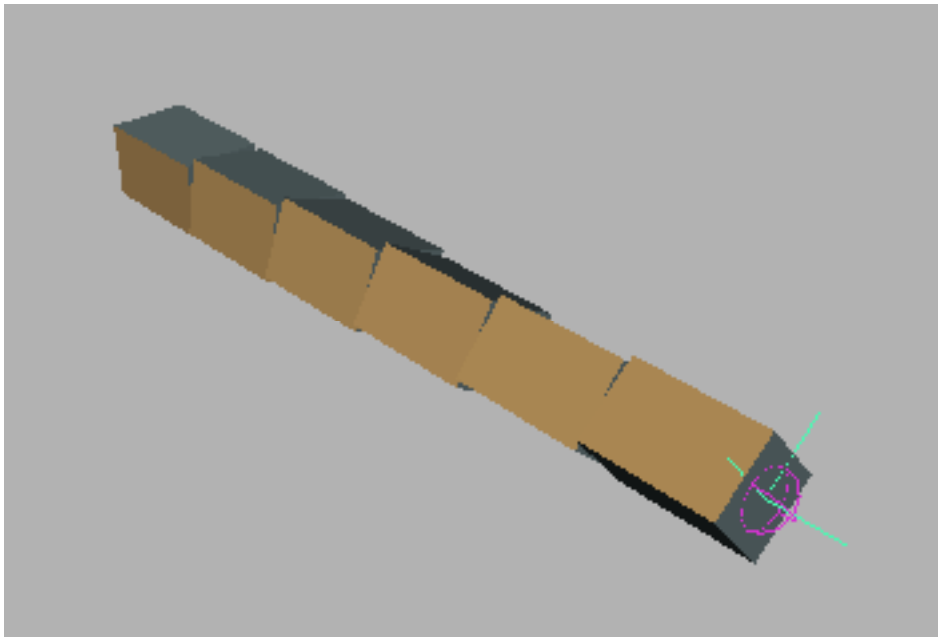

Figure 123 - joints twisting to simulate radius and ulna

What else does this series of joints look like?

The torso rig! Of course! We can adapt the splineIK rig that we're using for the torso to make it work on the arms of our characters. Using this, we just hook up the top and bottom to the various parts of our arms, and they should twist correctly along the joints.

Create a twistable arm segment

1. **Create a new Scene**
   - Choose **File > New**
2. **Create a joint segment**
   - In a **front** view, choose **Skeleton > Joint Tool**
   - Create a joint segment like the following:

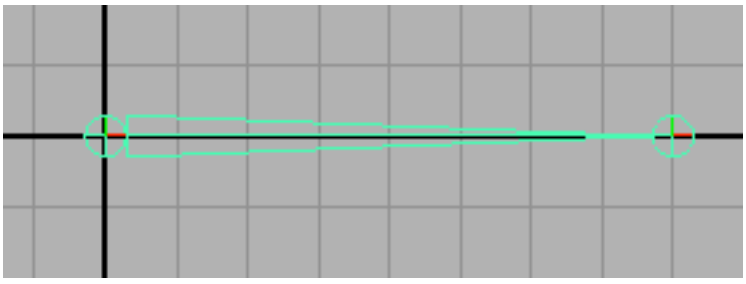

Figure 124 - joint segment created

3. **Segment the joints**
   - Select **joint1**
   - Click on the **Split Selected Joint** button in the **Animation Mentor Friendly Rigging** Shelf
   - Choose **5** segments

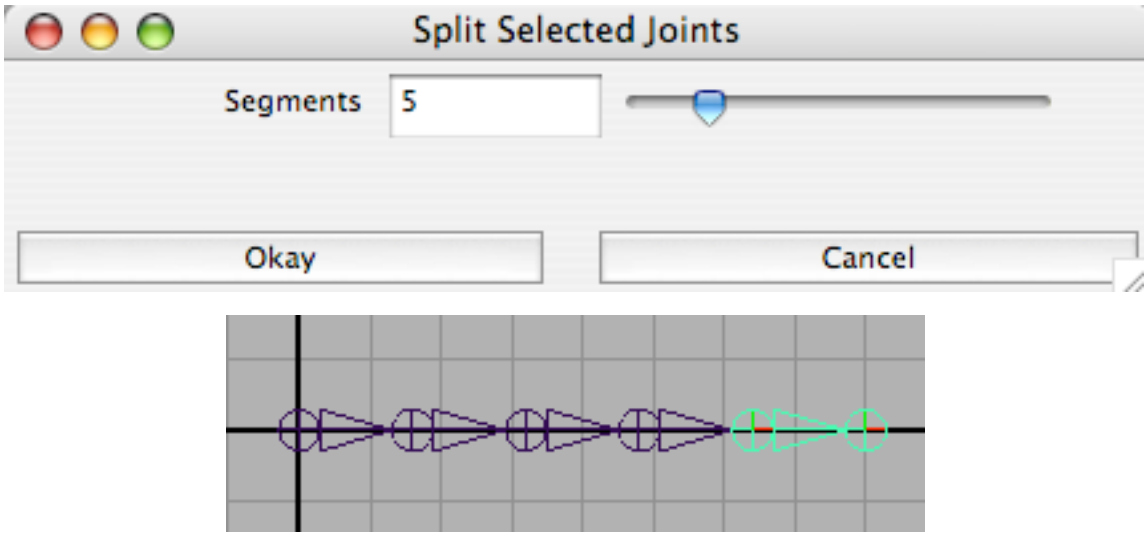

Figure 125 - Segments set to 5

4. **Rename all the joints**
   - Select all the joints

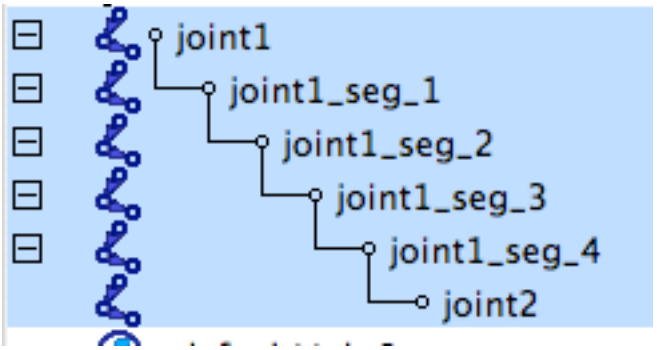

Figure 126 – Selecting all the joints to rename

   - Click on the **Hash Rename** button in the Animator Friendly Rigging Shelf
   - Enter **seg_#_joint**

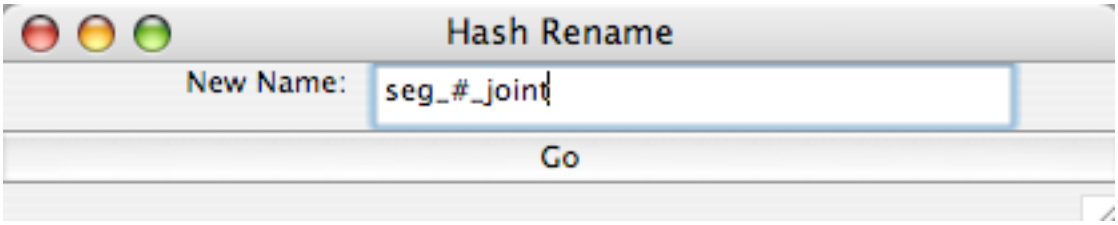

Figure 127 – Entering a new name. If you enter a # mark in the name, it will replace it with a number. You can use the #'s as padding. For example, if you enter: seg_####_joint it will rename the joints seg_0001_joint, seg_0002_joint, … seg_0100_joint…

- Click **Go**

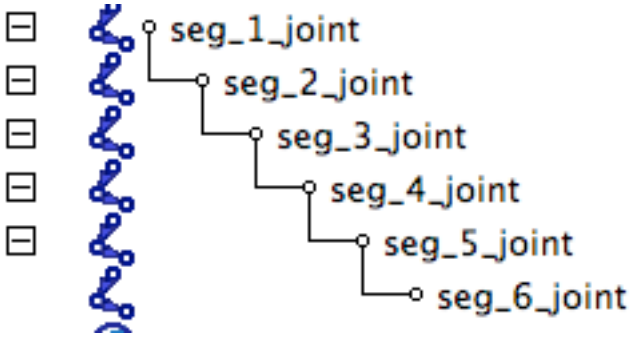

Figure 128 - renaming the joints

5. **Add Curve**
   - Choose **Create > EP Curve > option Box**
   - Set **Curve Degree** to **1 Linear**

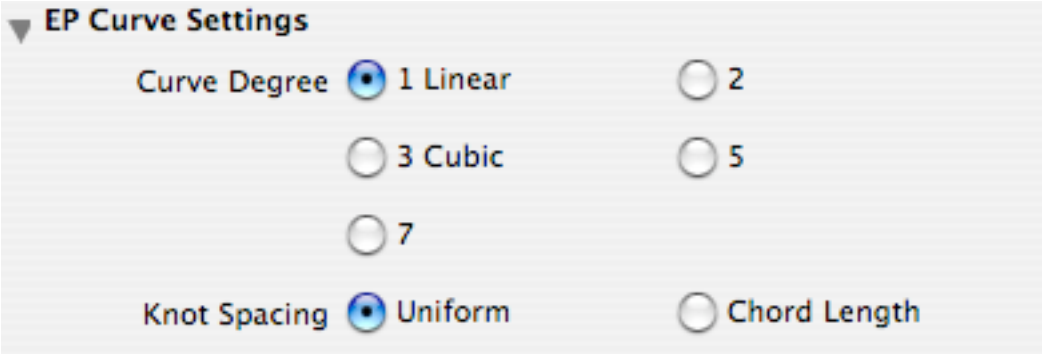

Figure 129 - Curve Settings

   - Create a curve that goes from **seg_1_joint** to **seg_6_ joint**

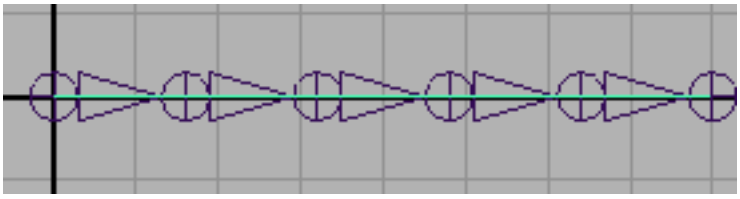

Figure 130 - Creating a curve

6. **Rename the curve seg_curve**
   - Rename **curve1** to **seg_curve**
7. **Create an IK Handle**
   - Choose **Skeleton > Ik Spline Handle Tool > Option Box**
   - Turn off **Auto Create Curve**

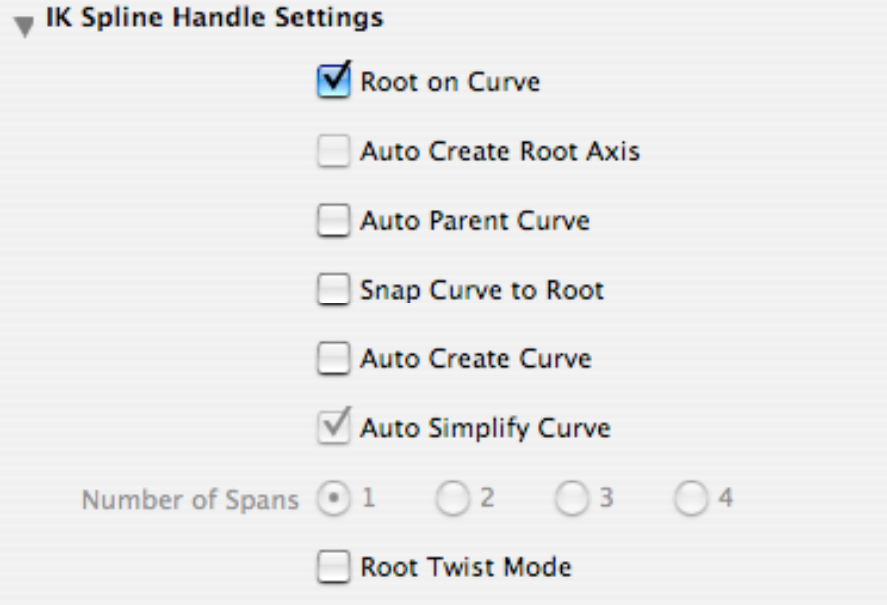

*Figure 131 - Auto Create Curve turned **off***

   - Select **seg_1_joint**
   - Select **seg_6_joint**
   - Select **seg_curve**

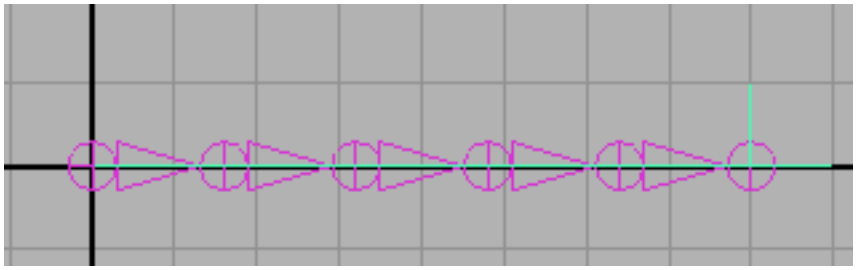

Figure 132 - Creating the spline IK Handle

8. **Rename ikHandle**
   - Rename **ikHandle1** to **seg_ikHandle**

Create Controls to drive the Spline IK

Now that we have a chain, we have to have some way of controlling the segment. Since we're going to be hooking this chain up to a joint structure, we might as well use joints to control it!

9. **Create the controls**
   - Create two joints that look like the following:

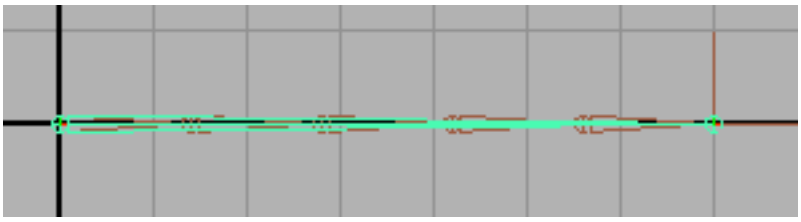

Figure 133 - Creating the control joints

10. **Rename the joints**
    - Rename **joint1 seg_start_ctrl**
    - Rename **joint2 seg_end_ctrl**

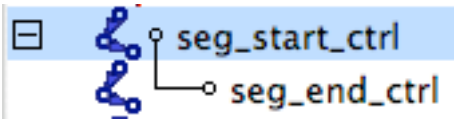

Figure 134- renaming the control segments

11. **Turn on the local rotation axis for the joints**
    - Select **seg_1_joint, seg_2_joint, seg_3_joint, seg_4_joint, seg_5_joint**
    - Choose **Display > Transform Display > Local Rotation Axis**

Set up the controls to manipulate the twist

We're going to try another method for handling twist other than **object rotation**. In the case of the arm, it's much more convenient to tell the twist to happen based on an up-vector like an object. For example, on the wrist we know that we always want the upper part of the arm to line up with the upper part of the wrist, so why not just place a locator above the wrist and aim at that?

12. Create two locators

- Choose **Create > Locator**
- Name the locator **seg_start_loc**
- Duplicate the locator
- Name the duplicate **seg_end_loc**.

13. Position the locators above the ends of the segment

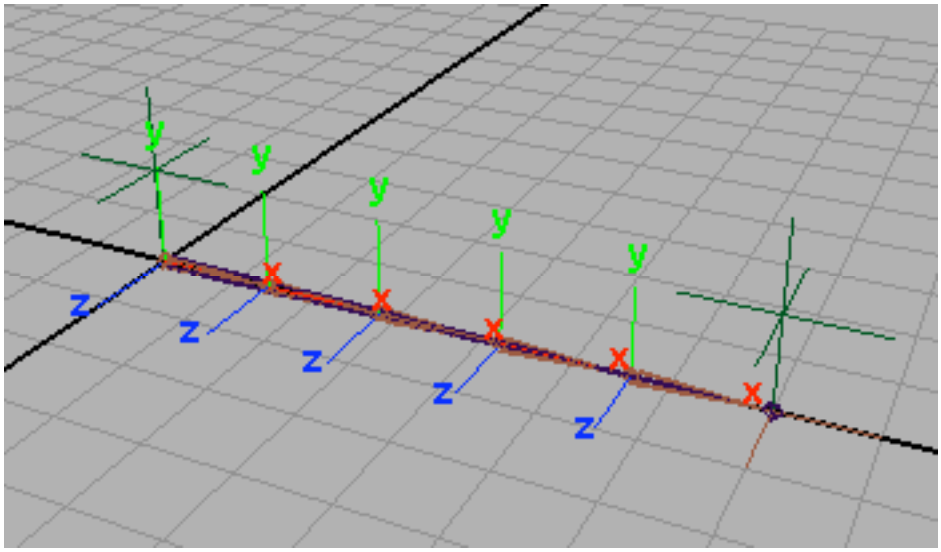

Figure 135 – Positioning locators above the ends of the segment

14. Set the Advanced Twist Controls

- Select **seg_ikHandle**
- Load the **Attribute Editor**
- Set the **Advanced twist Controls** to look like the following:

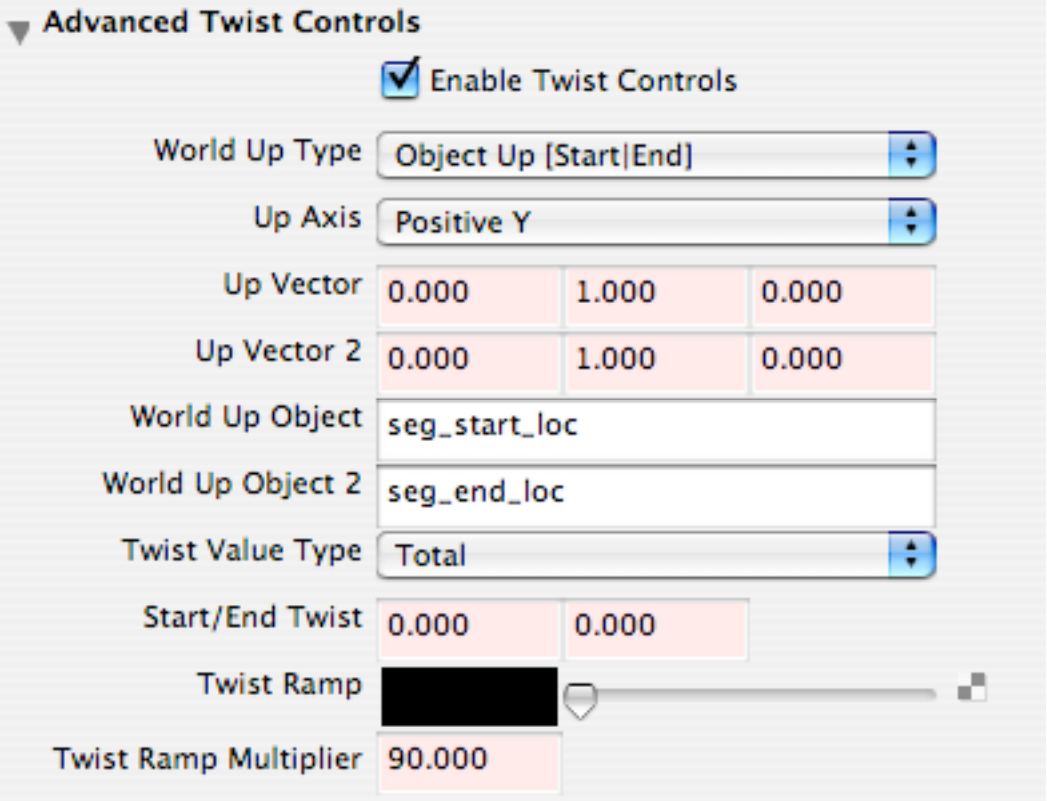

Figure 136 - Setting the advanced twist controls

15. Parent the locators under the controls

- Parent **seg_start_loc** under **seg_start_ctrl**
- Parent **seg_end_loc** under **seg_end_ctrl**

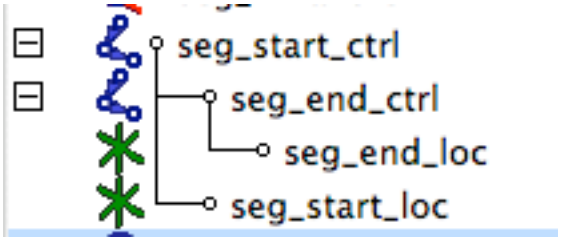

Figure 137 – Parenting the locators

Want to learn more? Purchase the entire DVD at:

http://jasonschleifer.com