

Character Setup in Softimage

BY DAVID GALLAGHER

● Before you begin breathing life into a 3D character, you'll need to make a skeleton to bend it and controls to move it. The "character setup" is the process between modeling and animation. Your character

setup should allow for a complete range of motions—and emotions—since a lack of foresight during the setup phase will mean a limited performance during animation.

In part one of this two-part article, I'll describe the process of placing a skeleton inside a Softimage 3D character; next month's part two will cover the controls and constraints. In addition, a web-only supplement covers the enveloping and vertex weighting process in Softimage.

Softimage excels in the area of character setup. It allows a freeform approach to skeleton creation and doesn't limit the workflow to a specific sequence of steps or an automation paradigm. Designing the setup to allow for any and all movement can definitely be challenging. It's a bit like a logic puzzle—a virtual machine designed to give you the control you want. But even simple machines can go wrong, so attention to detail is critical.

Let's put together a simple character skeleton and see how it's done. To demonstrate, I'll be using my "Stuart" character, a heavy-set, working-type fantasy character designed for an in-house project at my company, Cineframe.

Making The Skeleton The basic goal of a skeleton is clear: The beginning of each



The open workflow and configurability of Softimage make it a logical choice for high-end character animation projects. Well-planned skeletal and animation controls can help you bring your Softimage characters to life.

ANIMATORSANONYMOUS

joint functions as a point of rotation and deformation in the character's mesh. You need to place each joint carefully to get predictable results.

Series of joints are called chains, and all the chains for a character make up the skeleton. Chains are made up of a root, one or more joints, and an end effector. The two ends of the chain, roots, and effectors look just like nulls—three short intersecting lines representing each axis. As you move the end effector of an inverse kinematic (IK) chain, the joints change rotation to reach the end effector. Although they're called IK chains, you're also free to animate the rotation of the joints directly without using the end effector.

You can choose to use 3D or 2D chains. 3D chains allow each joint to rotate freely on all axes, while 2D chains keep all the joints in the single 2D plane of the first joint, allowing every other joint to twist on the X axis and turn on the Z axis but not in Y. In most cases, you'll want to use 2D chains. For instance, you normally wouldn't want to be able to bend an IK arm or leg sideways on the local Y axis—it's impossible for our own human bones.

To start, position the character in the workspace so it faces forward in the Front window. The Z axis points forward, and the X axis points side to side (see Figure 1). At this point, you should be basically done with the modeling. You can make modeling changes later, but it's easier if you don't have to.

All the body parts you want to bend with the skeleton should be parented beneath a single null, but not necessarily in a descending hierarchy. For instance, a hand doesn't have to be the child of the arm, but can be directly under the main null just like the arm. This arrangement makes it easier should you ever need to detach the arm to make changes, since you won't also have to detach the hand, thus losing the vertex weighting of all the fingers. (Detaching body parts later from the skeleton erases any work you've done to assign the control points to each joint.)

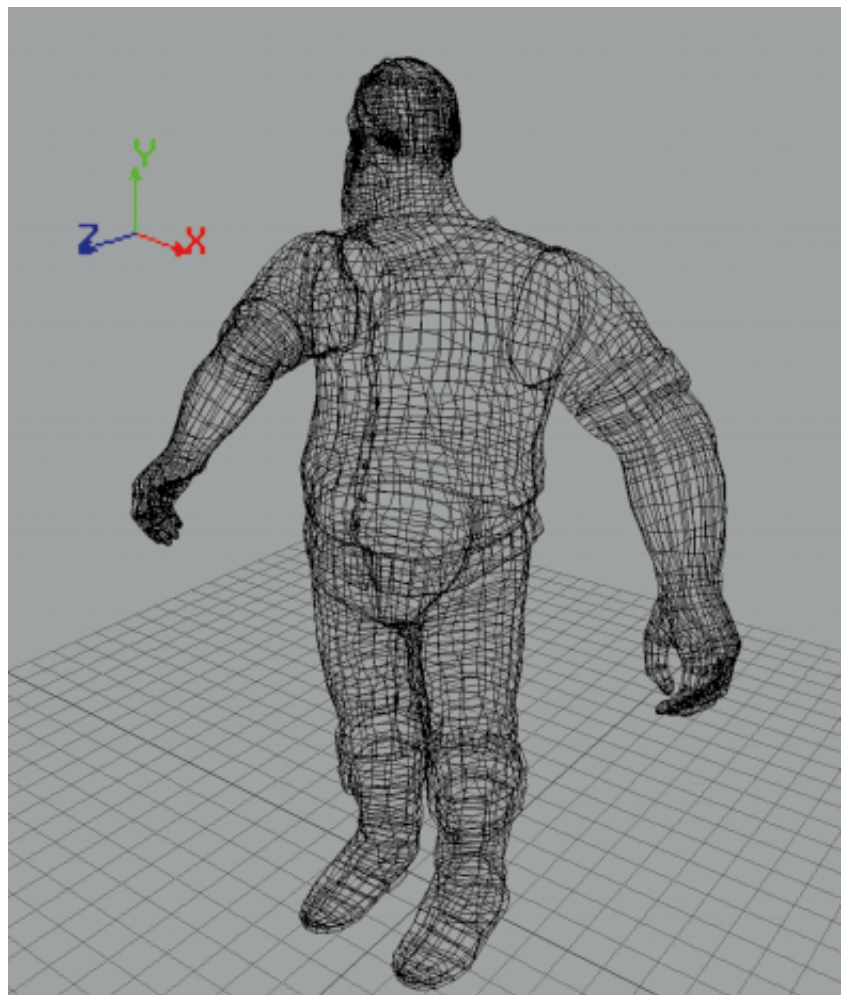


FIGURE 1. The character is ready for skeleton placement, facing down the Z axis.

Put Down Your Arms To begin drawing a skeletal chain, first decide which projection window (Front, Right, or Top) is closest to the plane of the needed 2D chain. (Don't draw it in the Ortho or Perspective windows; it makes unpredictable chain orientations.) Let's start with the arms. For instance, if you've modeled the arms sticking out perpendicular to the body with palms down in a neutral pose, draw the arm chain in the Top window. This correctly orients the chain for bending, since you always draw 2D chains along their XY axis (Z points toward you). Choose Skeleton→Draw 2D chain, and starting somewhere near the shoulder, click to place the chain root (the base of the chain) and first rotation point of the chain, click near the elbow, and then again at the wrist.

The exact placement of each of these rotation points is very important, but you don't have to get it perfect for the chain's initial creation. You must get the number of

joints right, though, since Softimage *does not* allow you to add or remove joints from a chain after it's made.

If you want to use IK at all, always draw a bit of a kink in the chain, never draw it in a straight line. This way, Softimage knows which direction you want the joint to bend. Otherwise, it may bend your character's arm backwards.

In our example, the arms are modeled at a 45° angle, so they are aligned with neither the Top nor Right views. In this case, choose either of those views and make the chain the best you can (see Figure 2, p. 52). Of course, since the geometry is not aligned with the view, there's no way to get the lengths of the joints and their positions right, but we can fix those in a minute.

Now rotate the whole chain into place to align it with the structure of the arm. Most of the time you'll want to rotate the whole chain on X if it is drawn in the Right window and Y if drawn in the Top. (This procedure is

ANIMATORSANONYMOUS

like rotating in Global Z, since chains take their orientation from which window they were drawn in and are automatically oriented with global coordinates only when they are drawn in the Front window.)

Depending on how the arm was modeled, you also may need to cock the joint in Y (or X if drawn in the Top) to refine the orientation of the chain root so it's perfectly aligned with the arm geometry. Orbit around the arm and rotate the whole chain until it's in place and mimics the orientation of the model arm structure (see Figure 3).

Fix Up The Joint Once the orientation is correct for the chain, refine the lengths and placement of the joint itself. Use the Skeleton→Move Joint and Skeleton→Move Branch commands to reposition the joints, root, and end effector. (Note: Softimage won't let you change the first joint, but this doesn't matter since it's attached to its root and you can change that.) Using the left or middle mouse button, drag each of the parts into place.

As you move the part you've chosen, the other parts scale or move accordingly. With Move Joint, all adjacent parts scale to accommodate the change; with Move Branch, all the children move with your selection. I used the word scale in the previous sentence, but note that it's *not* the same as manually scaling the joints with the Scale menu, something you should never do during the setup stage. Move Joint and Move Branch make the change *without* altering the Scale=1 property of all normalized objects.

Using the left mouse button (the prompt says X Translate) you can move your selection only up and down the X, or length, of the previous joint. Using the middle mouse button, Free Translate, you can move the selection in all axes. But if you move parts of a 2D chain at all from side to side in Z, against the normal XY plane that it's supposed to maintain, you break the IK. As the chain next recomputes when the end effector is moved, the chain realigns itself to

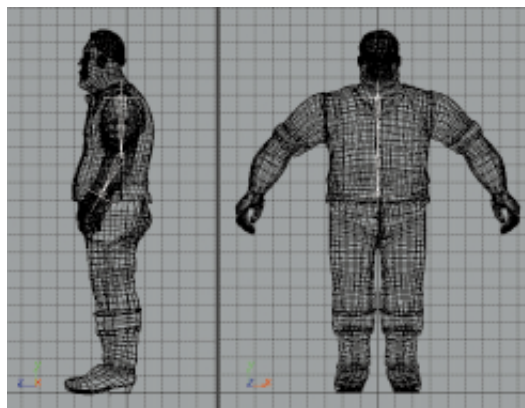


FIGURE 2. The arm chain is properly drawn in the Right window, but in the Front window, you can see it's too short and not rotated like the arm.

stay in 2D. It will simply pop back into place, but it's best not to have any surprises in this process.

In our example with the rotated arm chain, you may want to use the Ortho window to ensure that all your Free Translate changes happen along the axis of the 2D chain. Select the Joint, then in an Ortho window, click on Z. The Ortho window aligns with the joint and the Free Translate changes happen in the proper plane—and we get no surprises later (see Figure 4, p. 54).

We've been talking about using two joints for the arm, but you can also add a third joint to the arm chain to accommodate forearm twisting. Just make a very small third joint at the end of the chain so we can use it later (see "Twisting a Wrist," p. 56).

Out on a Limb To get these joints placed correctly, study real-world bone relationships by looking at your own limbs changing rotation. It all seems obvious until you do it. You'll probably want to place the elbow joint near the middle of the flesh, but slightly closer to the outer part of the elbow. If you place the joint too near the edge of the elbow, the elbow will retain shape nicely, but the crook of the arm will self-intersect badly. If you make an extra wrist joint, draw it straight off from the main forearm joint, so it will have a near zero rotation in Z. Make the end of the arm chain (and the beginning of the hand chain) just outside the hand as it joins to the arm.

You have to do some reinventing each time you set up a character, since each

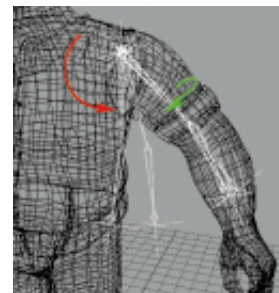


FIGURE 3. Rotate the whole chain sideways in X, then twist in Y as needed so your joints match the arm geometry.

character body type will react differently to the placement of joints. For instance, making a thin body work well is much easier than making a fat or muscular one work. The fatter parts are more likely to self-intersect when rotating, a nagging problem in 3D. Thinner limbs make vertex weighting easier and are more forgiving with joint placement.

Once you place the shoulder joint or chest joints, you can test the envelope to see how that arrangement bends the geometry. Use Skin→Global Envelope to get the body to move temporarily with the skeleton. (Visit the *3D Design* web site at www.3d-design.com for a web-only supplement on envelope weighting, also known as vertex assignment.) Ignore the fact that many points will be wrongly assigned—focus on the chain in question to determine if you've chosen the right placement for each point of rotation. You can worry about mismatched points later. When you're done testing, simply use Skeleton→Reset Actor then Cut the chains from the character geometry.

You can mirror the chains onto the other side with Effect→Symmetry. Continue this process for each of the needed skeletal chains.

For fingers and hands, I like to use one joint for the palm area, then four three-joint chains for the fingers. Start the finger chains in the middle of the palm flesh, directly beneath the knuckles. Orient the finger chains cocked in Y a bit so that as the fingers close, they bunch together. The thumb has three joints as well, the first

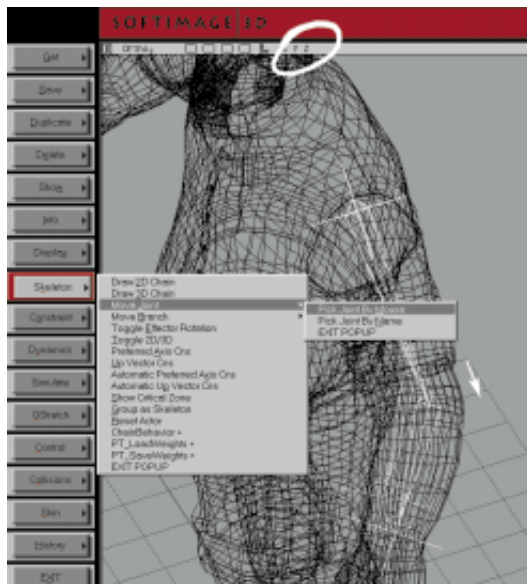


FIGURE 4. Never resize the joints using the Scale command. Get them into place with Skeleton→Move Joint or Move Branch commands.

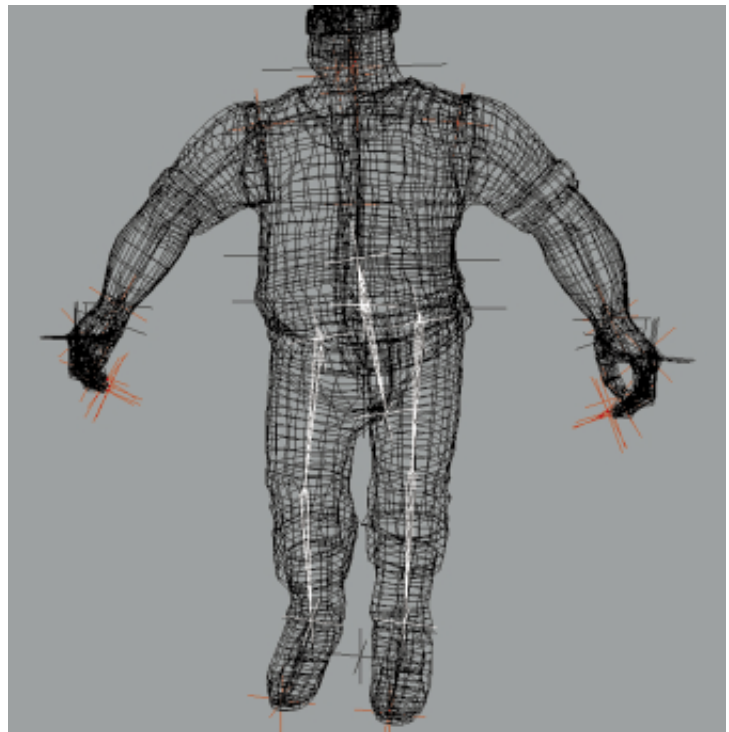


FIGURE 5. Independent hip rotation isolates the upper body rotation from lower body rotation. This means Stuart can do the hula or swivel his hips when he walks.

plunging into the flesh of the palm.

You can draw the shoulders as single joint chains starting to the left and right of the neck, well below the shoulder skin. Lift your shoulder to see where the point of rotation is. Some animators don't include shoulder joints. You can get away with this for certain character types simply by allowing the points to mush in between the arm chain and the spine/chest chains.

Hips and Whips The hips can be a single joint chain from the belt area to below the crotch. Note that when you rotate *only* your hips side to side, the point of rotation is just below your belly. This means you would draw the joint top to bottom, not from the crotch up to the waist. Proper arrangement of this joint will allow you to sway your character's hips from side to side as it changes balance, independent from the rotation of the upper body (see Figure 5). This is desirable because it allows the animator to tweak the hips without having to compensate each motion in the upper body. Isolating the animated elements means having more control over them.

Making hip joints from bottom to top is also possible, but it defeats this independent counter-rotation potential. That arrangement forces the upper body to come along, or the action would tear the waist. Some Softimage users make a hip joint coming out the back of the hips, perhaps to make it easier to select, but also to be able to animate the end effector to move the hips and legs (which are parented to the hip).

It may seem easier to whip the hips around using an end effector, but keep in mind that IK, which uses the end effector instead of joint rotation, is mostly useful for a chain if it moves in only two axes; that is, no twisting. Hips need to be rotated in all three axes for best results, so direct rotation is the best choice. Otherwise, attempting to control three axes of rotation with both translation function curves and rotation function curves may be too confusing.

Based on that principle, a shoulder joint, for instance, could be controlled easily using IK instead of rotations. The shoulder never needs to twist in X, but only to move about in Z and Y. You could use rotations, of course, but you may find it simpler to control the joint with the translation of the

end effector, and you lose no needed axes of rotation to do so.

Make the legs children of the hip joint or its end effector so the legs move with the hips. And when the legs move independently, the hip joint provides an anchor for all the vertices in the groin/rear that won't be completely assigned to the legs.

Draw the legs with two joints—hips to knees and knees to ankles. Then start another chain for the foot, with one or two joints for toe bending. Version 3.8 of Softimage comes with ready-made skeletons, using Get→Skeleton. The high-resolution skeleton even has separate chains for each of the toes! Most of the time, though, you won't need that kind of foot articulation.

Use Your Head The head can be a single joint chain with a second chain for the jaw. Make the neck a one or two-joint chain ending where the head chain begins. You can make the head joint a child of the neck end effector, so that when the neck moves, the head follows along, or you can reverse that order and bring the control back to the head. Use Constraint→Position to attach the neck's end effector to the head chain

ANIMATORS ANONYMOUS

root. You may find this easier than using the rotation of one object to drive the placement of another, since you can focus on the head's position and rotation and let the neck just follow along (see Figure 6).

A common approach for the spine is to use a single two- or three-joint 2D or 3D chain. (A 2D chain will override rotations in Y unless no IK is used.) A spine has to be able to rotate on all axes up several points of articulation for full mobility. As with the example of the neck, you can use a bottom-up or top-down approach. When a single chain is used the standard way, the position of the upper body is controlled by rotating several joints in succession from the bottom of the spine to the base of the neck. Each joint up the chain will inherit the rotations from lower joints. During animation, each rotation function curve for each joint

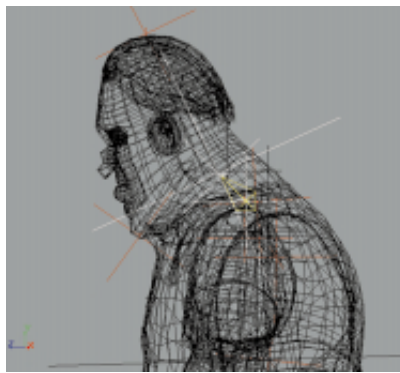


FIGURE 6. Constraining the neck to the head reverses the usual order of things. Now you can place the head and let the neck just follow along.

must be closely watched to create the desired motion. You may find that an easier approach is to control the motion from the top down, as with the neck. (I'll demonstrate a method for doing this in part two.)

A relatively recent development in the software is the ability to add any object to a skeleton to make it deform the body just

like the joints (bones). For instance, you can add scores of nulls to the skeleton hierarchy in the face area to drive facial movements. You can pull around or rotate the skeletal nulls to deform the face, either replacing or augmenting shape animation proper. This feature is called general skeletons, and I refer to them as skeletal nulls, since most of the time nulls are used. The skeletal nulls can also be arranged to translate based on the movements of the skeleton and other nulls, making smooth transition zones, solving problems in such tricky areas as the underarms and pectorals.

Tune in next month for part two of our look at character setup in Softimage, which will discuss constraints and expressions and how to use them to efficiently manipulate the skeletons we've placed in our character. ●

Twisting a Wrist

Without adding a third joint to the arm chain (or another such solution), you can't convincingly twist a 3D wrist and forearm. If you attempt to rotate an entire forearm joint to simulate this action, the elbow will become mangled. If you attempt to treat the problem with only the hand joint rotating, the wrist will crunch. The twisting action along your forearm can be correctly represented only with the help of another joint.

The deformation takes place throughout the length of the forearm as the radius and the ulna twist about each other, which makes the most logical 3D joint placement, which is two joints split halfway down the forearm. But an extra joint drawn like this would need to be stopped from free rotation on the Y axis or the IK will make the forearm bend like spaghetti. You could apply a rotation limits constraint to prevent any rotation in Y and still allow twisting in X, but when you do this, it creates problems later on—it introduces an offset because the rotation limits con-

straint isn't taken into account in the IK algorithms. The result is that the effector can't reach an IK constraint target at most positions.

To get around this, you can make the "wrist" or extra forearm joint very small at the end of the chain—so small that it's barely considered in IK solutions. To test this, draw three test arm chains: one with two equally sized forearm joints, one with a smaller joint at the end, and one with a tiny joint at the end. When you drag the end effectors, observe how the last joint bends. The smaller the joint, the less it bends to reach the end effector.

Even though the small joint will affect only a tiny area around the wrist, later on in the enveloping process you can force the small joint to grab more geometry up the forearm for a smooth, gradual twist.

—David Gallagher