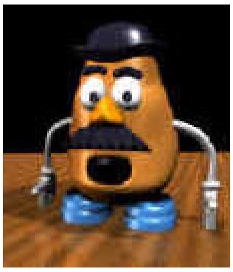
The Blended Planet

www.blendedplanet.com

Information and Resources for Blender Users

Products | Tutorials | Galleries | Links | HOME | EMail



Blender Character Animation Quickstart by David Weese

It's no big secret that one of Blender's historical weaknesses has been in the area of character animation. This shouldn't be viewed as as too big a tragedy since, quite frankly, the tools included in such luminary higher-end applications such as LightWave and 3DS Max have not been appreciably stronger.

In fact, until recently, the only real viable options for industrial strength character work were SoftImage, Maya, and Hash:Animation Master. Each of these choices are somewhat problematic - SoftImage and Maya are mighty expensive, especially if you want to run the app on several machines, and Hash, while extremely capable on paper, is - in practice - extremely (frustratingly) unstable, and really always has been.

But I digress. I bring you good news as someone who has worked with all the apps discussed above. Blender 2.25 has a plethora of tools that have finally come together to provide character animation capabilities that are not only strong, but elegantly designed with simplicity and ease of use that I have found satisfying to use...I think you will too.

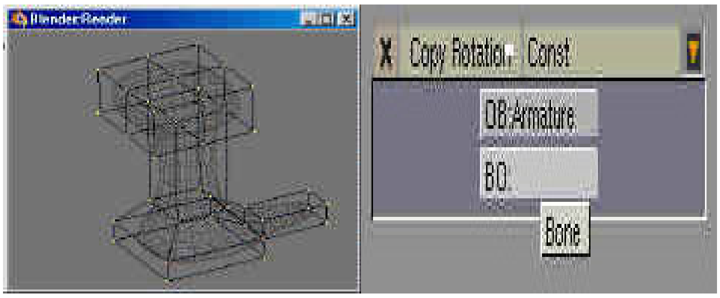
What constitutes what we might call "Strong character animation tools"? Glad you asked. The CG world seems to have arrived at a consensus of the features one should expect if you want to be able to quickly churn out solid character stuff :

1) Modeling with subdivision surfaces
2) A powerful bone deformation system
3) The Holy Grail - Non Linear Animation (NLA)
4) Weighted target morphing
5) Inverse Kinematics and related Constraints

In this article, I'm going to examine only the first three, and you will discover that they will keep us busy for quite awhile....

Lets first define these terms and break down how Blender addresses each of these requirements...

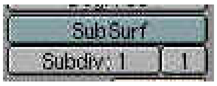
Subdivision Surfaces Modeling (SS)



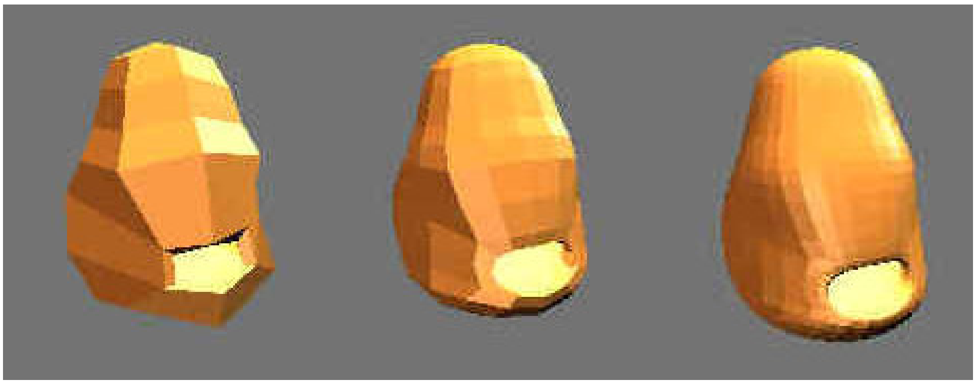
figures 1 & 2

SS means that you model and animate using a comparatively low resolution polygon mesh (fig 1), but when you render, the program dynamically subdivides and smoothes the model (fig 2) so you gain a much higher level of "smoothness" and detail at render time. Most model manipulation is done using the low res mesh, which acts as a lattice around the true, subdivided model (fig 1). Nearly every surface in the Mr. Potato Head picture at the top of this article used SS. The full size (640X480) image rendered in about 6 seconds on my Athlon 1.8 ghz machine.

Using Subdivision Surfaces in Blender is really simple. In the EditButtons Window (F9 Key) you will find the following buttons :



To turn a regular mesh into a subdivided mesh, just press the button called "SubSurf". This is a non-destructive operation - press the button again to go back to the un-subdivided mesh. Plus (it gets better!), you can use the two buttons below it to fine tune your SS experience. The button labelled "Subdiv: 1" lets you set the level of subdivisions, as demonstrated by Mr. P :



Subdivision levels of 0,1, and 2

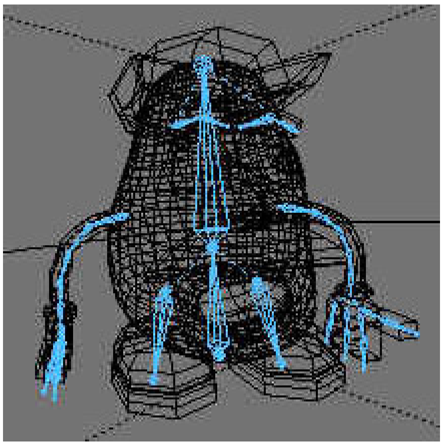
Furthermore, pressing the 2nd little numeric button allows you to set the render-time subdivision level. In ultra cool Blender fashion, we now have the ability to manipulate our model with little or no subdivision in the 3D viewport, yet tell Blender to go ahead and render at a higher subdivision level.

In case you wondered, SS was invented by and for the creative team at Pixar for use in films like Monster's Inc., Toy Story 2 and A Bug's Life. In addition to the benefits already illustrated, it should be obvious that SS ends itself well to flowing, organic forms.

Note : Any modeling technologies that compete with SS? There's really only one - NURBS - and the way things are panning out today, it appears that SS is the winner in the arena of character animation and entertainment, and NURBS have a much stronger foothold in the area of high precision mechanical design and product visualization.

You'll be glad to know Blender's SS implementation has a strong pedigree, being based on scientific papers by Catmull and Clark (the dudes at Pixar who created it).

Bone Deformation
The next requirement for character animation is bone deformation. You are almost certainly aware that this means using simple "bone" objects arranged in a "skeleton" which, when moved, rotated, or sized, transfer these motions to a polygon mesh which is "attached" to the skeleton, in essence providing an intuitive, "shorthand" way to indirectly manipulate the many, many points that comprise a model. By way of example, here is Mr. P's skeleton :



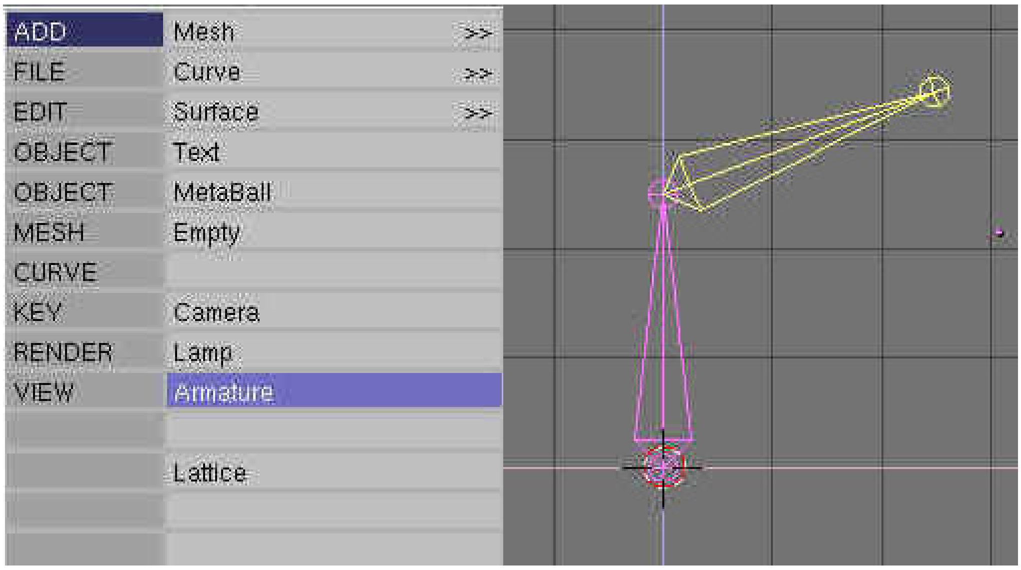
Now, if you are new to all this, the following may not seem particularly profound to you, but it's really quite cool from a workflow standpoint : Blender allows you to create a complex hierarchy of bones into a skeleton (it's called an "armature" in Blender), then have any number of separate models attached to the skeleton. In my example, Mr. P's head, eyes, arms, hat, etc. are all separate, unrelated meshes - in other words the model components do not all need to be part of the same mesh in order to be controlled by a single armature. We'll see shortly how to link the meshes and the bones together, so stay with me!

Good News and a Bug :
It's worth mentioning that an armature can actually be modified slightly to provide a bone system for multiple models, provided the models use

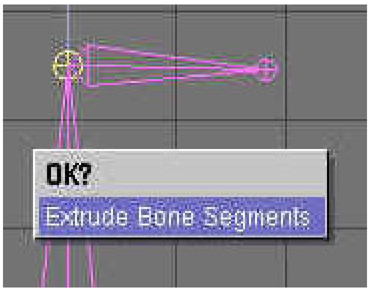
the same naming conventions for the bones and the parts of the body the bones deform. This is accomplished by setting up an armature with sensible bone names like "left bicep", "neck", etc., then saving the blend file. Later, you bring the armature into any other blend file by pressing SHIFT+F1 key (Load Library) and browse to find the armature and load it in. You can then adjust the positions of the joint so that they align well with your new model. If you have set up the model correctly (which we're about to see how to do), the armature should work fine. That means you can create a basic biped rig which could be reused as a deformation system for any number of different characters. In fact, any actions (such as a punch, kick, or walk cycle) that are defined for the armature ALSO are automagically imported. HOWEVER, when you browse the blend file, do not try to do this by pulling the armatures in via the 'armatures' folder - you have to go into the 'objects' folder - the armature will be there.

OK, Let's get rockin' :

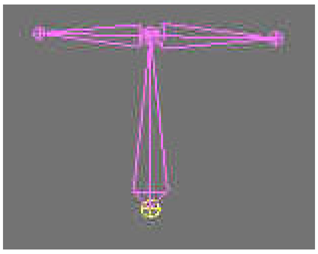
1) Creating Armatures - press the spacebar and do the obvious - click "Armature" :



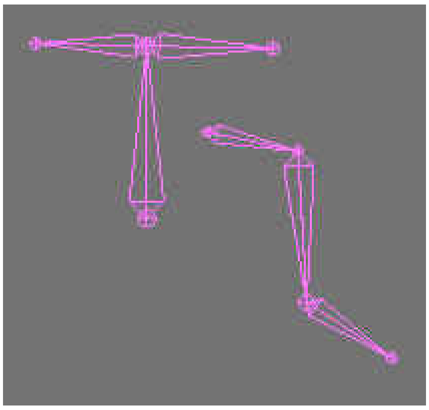
As long as you keep clicking with your left mouse, new bones are added to the armature. Hit ESC key to stop adding bones to the initial armature you've created. After you create an armature, you can select a joint with your right mouse button and press the E key to extrude additional bone segments, building up the armature to your requirements :



...but keep in mind you may NOT extrude a new bone from the root node of the skeleton as shown here :



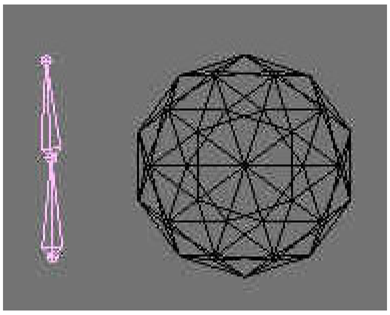
You may also simply add more armatures via the menu the same way you did initially, in order to add more bones to the structure. If you need to delete a bone, you MUST select both ends of the bone, so the entire bone is highlighted, not just a joint.



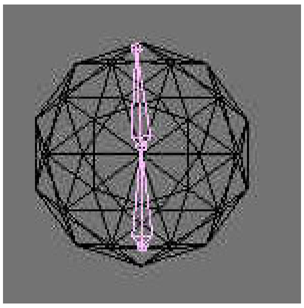
Bones added this way are not hierarchically linked to the other bones in the armature, but they all are considered part of the same armature object. This has a bearing on the kinematic characteristics of the armature. Right now, if I were to rotate the first bone I created, the second and third one would be moved as well in forward kinematic fashion. In other words, bone #1 is the *parent* of bones #2 and #3. Fortunately, it is possible to completely change the hierarchy later, by simply changing the parent-child relationships, which you'll see how to do soon...

Like most Blender Objects, you access the armature sub-object (edit) mode with the TAB key. Use edit mode to move the joints around to position them correctly within your model. If you want to grab ALL the bones in a single armature chain, grab any joint, then press the Lkey (select linked).

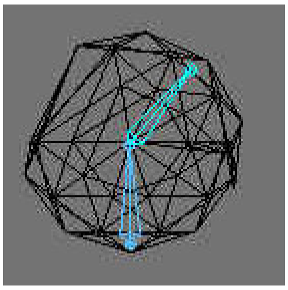
I should point out that bones SHOULD be placed *within* your model. I had speculated that perhaps you could place the bones away from the model as illustrated here :



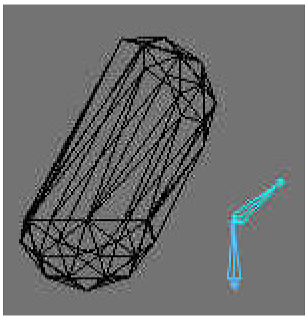
Instead of the more intuitive arrangement :



But this does not work, because the armature uses the bones' joint axis as a global center of rotation, so instead of getting the proper deformation like this :



An offset armature gives you this, entirely useless deformation :



So, if you think you might want to be able to manipulate your armature with the model entirely out of the way (and, yes, you very well might), the solution is to simply place the armature in a different layer, and turn off the layer containing the model in order to hide it when desired.

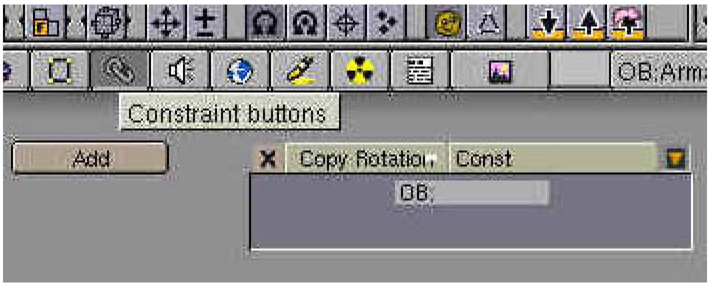
- OR, an equally groovy solution -

Use a rotational constraint! It's worth taking a moment here to talk about constraints within a skeleton, because they can be a major headache saver. The idea behind a constraint is that you are forcing one object to be "constrained" to another object in some way. For example, a "Copy Rotation" constraint forces the constrained objects rotation to perfectly match the rotation of another object.

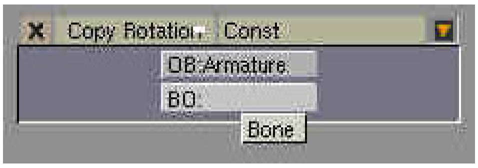
All you need to do is create a dummy bone in the skeleton that does not directly control any vertices. Place this bone somewhere out of the way, where it is not "mixed in" with a bunch of other bones and/or your model - somewhere you can reach it easily on other words. Then simply constrain the rotation of the deforming bone to the dummy.

Where is this useful? Well, say you have a dense model with a detailed skeleton, and you're going to require a lot of animation of the hand. Drilling down into a complex scene to grab and manipulate finger bones can be an exercise in frustration.

Instead, grab the finger bone just one time, and, after going to the constraint buttons window (see below), add a "copy Rotation constraint.



Where it says OB:, simply type the name of the armature this bone (and the dummy bone I just mentioned) are part of. Once you do this, the following appears :



Now type the name of the constraining bone (the dummy). Now, whenever you rotate the dummy bone, the bone in the hand rotates exactly the same way. Very useful...

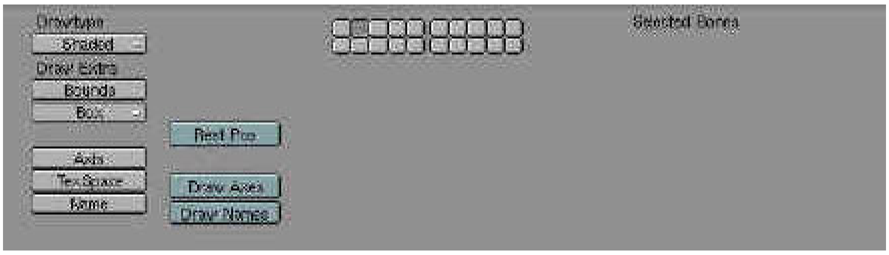
IMPORTANT VERSION 2.26 FEATURE, and it's a biggie :

If you grab any bone (or any other object, for that matter, and press the RKey to rotate it, then immediately press x,y, or z, the rotation will be constrained to the global x,y, or z axis. That's kind of cool, but here's the biggie : if you press Rkey, then x,y, or z TWICE (e.g. R-Y-Y), the rotation is constrained to the objects LOCAL x,y, or z axis. This means you don't have to rotate around the scene in order to find a position perpendicular to your object and get it to rotate the way you want it to! With bones, by default, the local x and z axes rotate the bone and the Y axis is the roll axis.

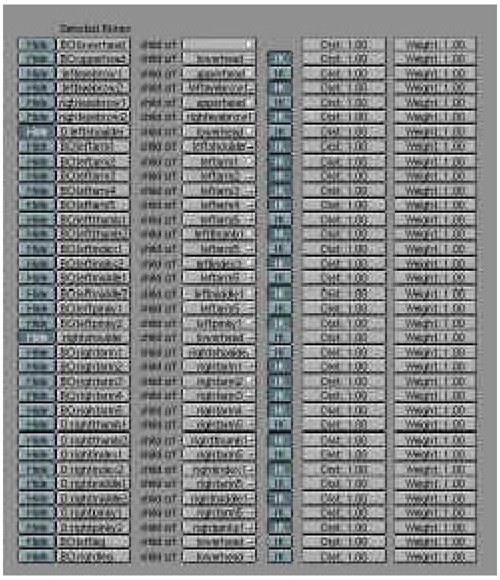
The next step to creating a useful skeletal deformation system is to somehow link the bones to the model. We'll break it all down in detail, but briefly, the way it works is that you 1) Name all your bones, then 2) assign clusters of vertices from your model/s to named vertex groups, and 3) make darn sure the names of the bones and the names of the vertex groups match.

Get it? We'll name a bone "right_forearm". and we'll also create a vertex group called "right_forearm". If the model is parented to the armature, the bone will automatically deform any vertex group that has the same name. Elegantly simple and intuitive, is it not?

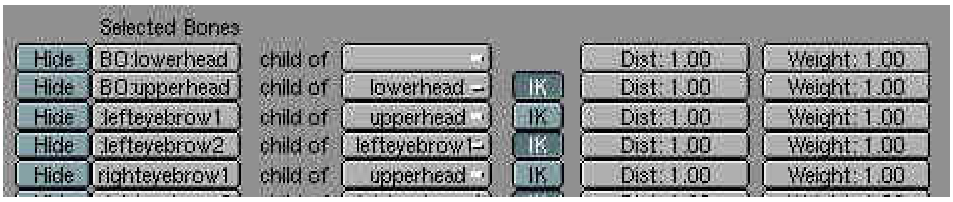
Step One - Name your bones. What you do is enter edit mode for the armature. In the editbuttons window (F9 Key) here is what you'll see :



Hmmm...Something appears to be amiss - where are all the bones? In fact, if you right mouse click on a bone to select it, nothing happens. This is a little weird, but the secret is that you have to hit the "A" key (with your mouse over the 3D window, obviously). so that all the bones in the armature are selected - all the bones will turn yellow. In the case of Mr. potato head, doing so revealed this mess :



Which, when viewed a little closer, looks like this :



	Selected Bones					
Hide	BO:lowerhead	child of			Dist: 1.00	Weight: 1.00
Hide	BO:upperhead	child of	lowerhead	IK	Dist: 1.00	Weight: 1.00
Hide	:lefteyebrow1	child of	upperhead	IK	Dist: 1.00	Weight: 1.00
Hide	:lefteyebrow2	child of	lefteyebrow1	IK	Dist: 1.00	Weight: 1.00
Hide	righteyebrow1	child of	upperhead	IK	Dist: 1.00	Weight: 1.00

Technically you do NOT have to select all the bones in an armature object to be able to edit their names and whatnot. In fact, it's actually a little easier if you use the Lkey (select linked) instead of the Akey (select all), because that way you can mess with a single armature at a time.

Now, before you get all discouraged and drop out right now, let me tell you this is probably the most boring and tedious part of the whole process. If you can get through this, you're halfway home. You have to go through and make sure all the bones have appropriate names that represent their function, and ensure that the hierarchal relationship between bones is correct.

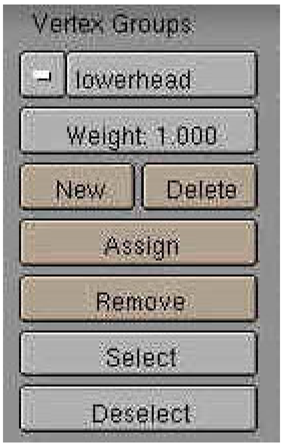
This is important : Blender has a "pose flipping" option. This makes it possible to create half an animation such as a walk cycle (e.g. - the right foot takes a step), then Blender can automatically create the opposing motion (the left foot step). However, if you want to take advantage of this option, you MUST name your bones using conventional suffixes so Blender knows which bones are symmetrically corresponding to each other. So, name the right thigh bone "thigh.r" OR "thigh.right" and name the left thigh bone "thigh.l" OR "thigh.left". If you like to do absolutely EVERYTHING by hand, you may safely ignore this convention.

Now, can I give you a little advice before you go too far with this on your own? When you are creating armatures, select them all every time you add a few bones - like I just described - and *name them as you go!* It can be a hairy mess looking at a bunch of bones that have been given default names by Blender, and you have to try to figure out manually that "bone.034" is actually "right shin". It *can* be done - you simply hide all the bones except one by pressing the "hide" buttons on the left, then see which bone is left over and name it appropriately. Plus, by looking at the "child of" popups, you can sort of figure out which bone is which. And in all fairness, bones are named in sequential order according to the order you created them. But, that's really doing things the hard way. Save yourself some stress and name bones as you go.

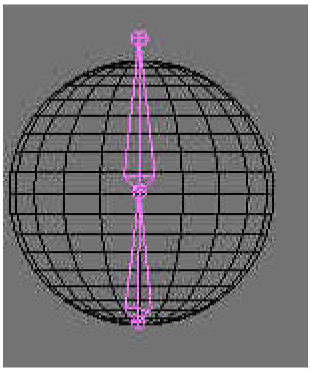
The buttons labeled "IK" indicate the root of a kinematic chain. By default chains are kinematically linked. This means that you cannot move the child bone (NOT labeled IK) away from it's parent (labeled IK). You can rotate the child all you want, but it is locked to the joint between the bones. If you disable the IK relationship, the child bone can be moved away from the parent bone. In most cases this is not a particularly useful thing to do.

Now, feel free to mess with this if you want to, but for most of us the entry buttons labeled "Dist" and "Weight" are of no use whatsoever. They are a holdover from Blender's old bone deformation system, where there was no way to specify which vertices were influenced by which bone. It only comes into play if you parent a model to an armature, but don't bother to set up any vertex groups. In this instance, Blender has to dynamically calculate the deformations based on the proximity of bones to the mesh vertices. This system was kind of workable, but believe me, the new vertex-group deformation system is 10,000% more reliable and easier to work with, plus it's a LOT faster. 'Nuff said.

OK, you've named all your bones. How to get the model to behave? It's really pretty simple. simply select your model and enter edit mode (TAB key). On the EditButtons page (F9 key), look for this group of buttons :

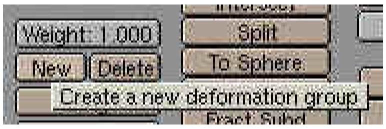


Let's go through the process step by step. I'll show you how I do it, using a very simple ball model and 2 bone armature as shown below. I already named the bones "hi" and "lo" for obvious reasons :

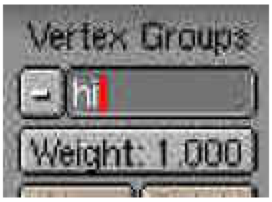


The more complex the model and bone system, the longer this'll take, but I was able to rig Mr. Potato Head perfectly in about a half an hour. I think you'll agree this is a pretty systematic and efficient approach.

First off, go ahead and create the vertex groups. You do this by pressing "New" :



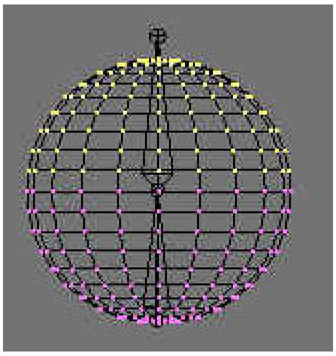
You will then see a new vertex group in the drop down listing. Type a name that is exactly the same as the name of the bone you intend to match up with this vertex group. Keep in mind these name ARE case sensitive. "Hi" and "hi" are two different names :



You can go ahead and create a vertex group for all your bones. This might seem counterintuitive if you are not familiar with Blender's object-oriented data structure. When you create the vertex groups, you are really just creating a container object that can have any number of vertices added to it later. I found this confusing at first...I thought if I selected a bunch of vertices, then created a Vertex Group, the selected vertices would automatically BE that Vertex Group. Not so! As I said, all you are doing now is creating empty Vertex Groups. We will assign specific vertices to these groups next.

I assume you created both a "hi" and a "lo" Vertex Group if you are following along. The order of operations is as follows to avoid confusion :

1) Select the appropriate vertices on your model :



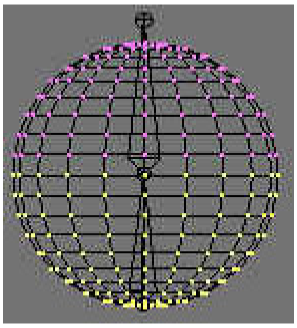
2) Make sure you have the appropriate Vertex Group selected in the Vertex Group dropdown selector - "hi" in this case :

3) Assign the selected vertices to this Vertex Group by pressing the "Assign" button.

4) Now to avoid confusion, press the Button labeled "Deselect". The fact that all the selected vertices are now unselected assures you that Blender has correctly assigned these vertices to this particular Vertex Group. If there is any doubt in your mind, simply press "Select" to see

that all the vertices in this group are now once again selected.

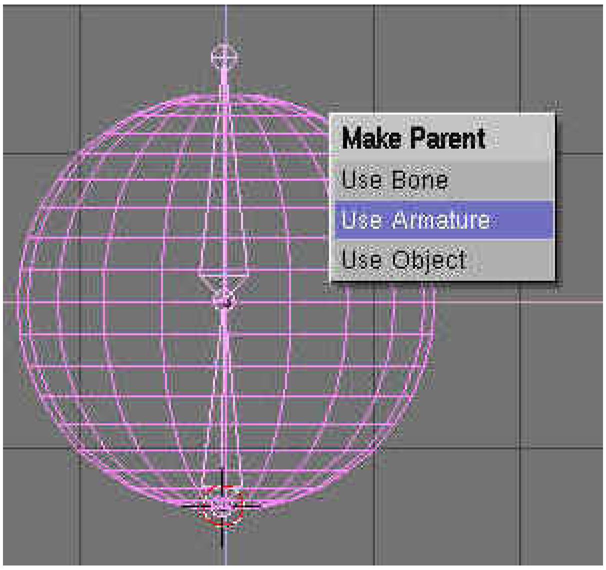
5) Do the same thing for other groups of vertices. In this first example, I will add all the remaining vertices to the "lo" group :



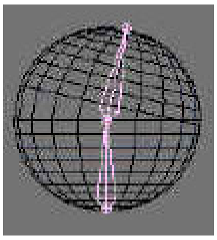
A couple notes regarding the other buttons. "Delete" does just what you'd think - it deletes a vertex group. If you delete a vertex group, you are NOT going to delete the vertices in your model. All you're doing is deleting the container that holds the vertex group information. The "Remove" button is a simple way to take vertices out of a group.

The "Weight" button, as discussed earlier in regard to Armatures, is kind of a holdover from the old days. We'll discuss weighting a bit more later, but for now take it on faith that in most cases, there is little benefit to messing with it.

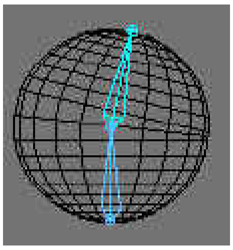
Now we have a 2 bone armature, and a 2 vertex group model. Let's just parent the model to the armature :



We'll getting into creating poses with the bones shortly, but for now, take a look at what happens if we rotate the hi bone :



As you can see, all the vertices assigned to "hi" move, and NONE of the vertices assigned to "lo". This leads us to the matter of weighting. In most cases, if you're dealing with organic models, you don't really want this kind of rigid deformation. In other words you want a smooth transition from the influence of one bone to another, right? More like this :



The secret is that, when you create your vertex groups, simply assign the row of vertices near the joint to *both* Vertex Groups! Blender will intelligently share the deformation between the bones. You can get a lot more complex with it, by adding intermediate bones and giving them a weight of less than 1.000, but so far I have found this to be more trouble than it's worth.

I suppose if you are trying to realistically rig a model of an anatomically correct human, and you want everything to move perfectly naturally, you'll want a little more control. If so, you can use vertex weight painting, where you actually paint the model to strengthen or weaken the deformation effect of any given bone. This tool is similar to that found in Maya, and quite frankly, I find Maya's implementation of vertex weight painting is also more trouble than it's worth in the vast majority of situations. The Mr. P model, which animated very well in Pixar style, was rigged completely using the technique I just described.

Creating Actions and Poses for your Model

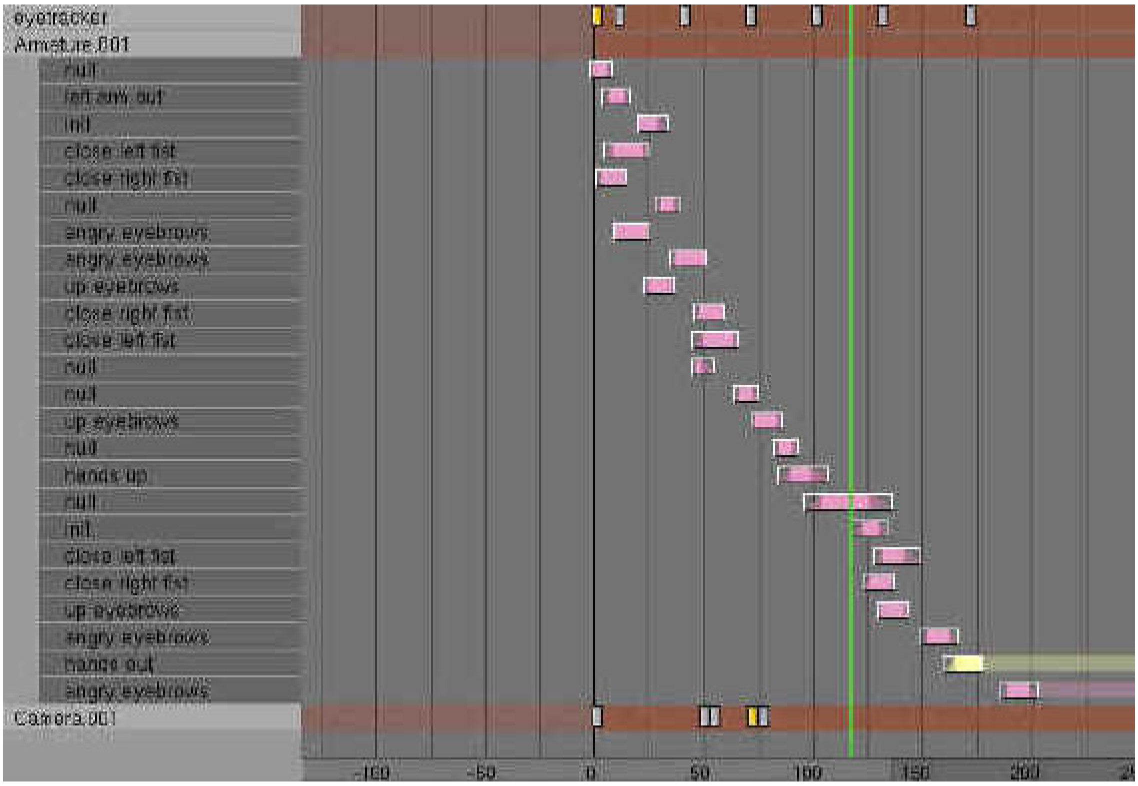
Once your character is rigged, it's time to animate, right? There are two valid approaches - linear and non-linear. The linear approach is to simply start keyframing the skeleton, and keep on setting keyframes all the way through to the end of the scene. This is how you used to have to do it with EVERYBODY'S software - 3DSMax, Lightwave, SoftImage, etc. And in fact, really only SoftImage and Animation Master have done a good job of combining powerful non-linear systems that are also pretty easy to understand and use.

Until now, that is, and you can quote me...**Blender's Non-Linear**

Animation (NLA) System is one of the best in existence. It's simple, elegant, intuitive and stable.

What does Non-Linear (NLA) mean? The idea is that you want to be able to create different key poses or very short key animations of your character, and then layer the actions (or reuse them) by placing animation clips in a timeline, as if you were editing video clips. The best implementations also allow you to blend motions, so that you could, for example, create an action that is a simple walk cycle of two steps. Then you create an action where the character turns his head. In your non-linear animation "mixer", you can layer the clips, so your character walks multiple steps (you loop the walk cycle clip) *and* turns his head. Since the two motions use different bones, they do not interfere with each other - the motions are perfectly blended.

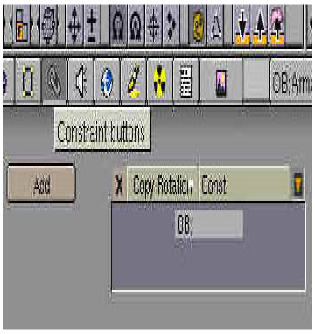
It would be pure fantasy, I suppose, to expect to be able to size the clips, and as a result have the action clips play a little faster or slower, right? Well, in fact, Blender does all this, and more, and it's easy! here's a screenshot of Mr. P's animation timeline :



As you can see from the screenshot, there are multiple instances of many actions in this timeline, and you can even set blend-in and blend-out's for each clip, so different actions smoothly transition from one to another. Now let's look at how it's all done...

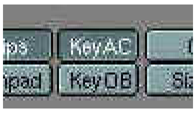
Non Linear Animation (NLA) In Blender
Creating Actions

The best way to teach you this is s-l-o-w-l-y, step by step. The first thing you need to know is that Armatures have a interesting characteristic called "pose-mode". Most Blender objects, by contrast, have 2 'modes', object mode and edit mode, and you access edit mode by pressing the Tab key. In the case of armatures, edit mode is NOT used for creating keyframes or poses of the skeleton. Rather, you use edit mode to construct your skeleton or adjust things like bone lengths or joint positions. To create animatable poses of your armature (skeleton), you enter pose mode by pressing CTL + TAB key. When you do this, the bones turn blue to indicate they are in pose mode :



Now, before we start going too crazy with animation keyframes, let me point out a neat new feature that may save you some stress. It's called "autokey" and it mirrors a feature common in some competing 3D apps. With autokey turned on, you do not have to press the I-key and insert keyframes manually. Instead, Blender automatically sets keys whenever you rotate, scale, or translate a bone.

Autokey can be set separately for objects or armatures. To do so, open up the information window at the top of the Blender screen. Find these toggle-buttons :



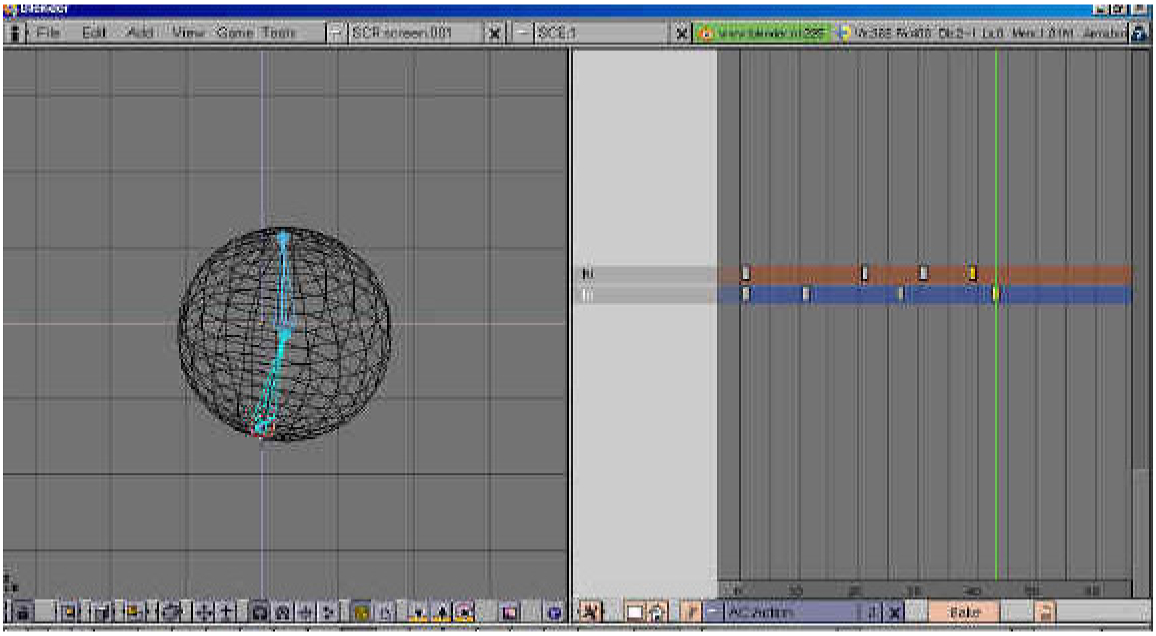
KeyAC refers to automatic keyframing for actions, and KeyOB refers to automatic keyframing for objects. In my case, I've been using Blender for about 3 years, so I'm used to manually keyframing everything. Plus I always found the automatic keyframing in 3DSMax a little annoying, because it always seemed like you'd accidentally keyframe something you didn't want to, and wind up with a tangle mess of competing keys. So, I leave KeyOB toggled off.

However, I have found that for actions, where you are animating a single armature in isolation (and generally dealing with a fairly short animation cycle), it is more convenient (and fun) to let Blender automatically insert keyframes whenever I rotate a bone. So I leave KeyAC toggled on.

The rationale behind autokey is that you will be really bummed out if you adjust 43 different bones so that on frame 126, your model is striking *the absolute perfect pose, then when you play it back to review your animation, all your work is lost because you forget to set keys for every single bone, and so, of course, Blender simply interpolates between the keys that occurred before and after frame 126. If this were to happen, you are, quite simply, screwed.*

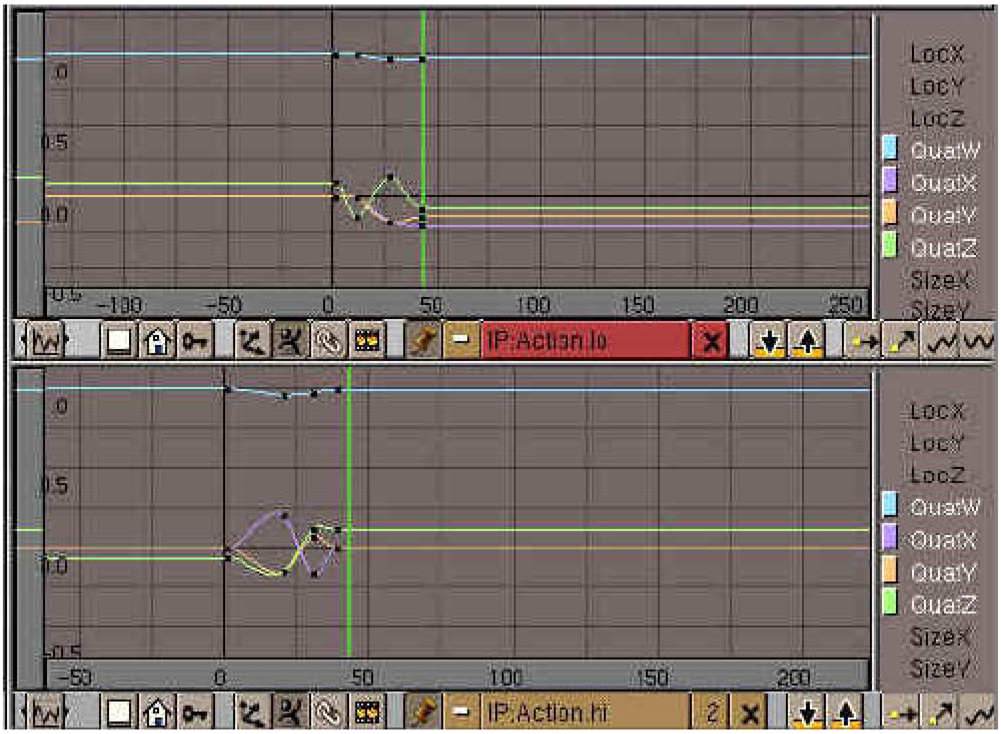
Now, for the sake of simplicity, lets assume you do NOT want to create any actions just yet. Your animation is going to be really simple, so you'd just as soon create keys in linear fashion. Assuming you did toggle KeyAC ON, you could now just move bones, jump ahead a few frames, move some more bones, etc., and when you jump out of pose mode (by pressing TAB twice), you simply press ALT+A key to play the animation and enjoy your creation.

HOWEVER, with any other type of animation, this would be the end of the story, but in all reality, there's a little more that took place under the hood. Without telling you, Blender went ahead and created an Action (named, conveniently enough, "Action"). You can see this by splitting your 3D view into two screens, then holding your mouse over the right hand viewport and pressing SHIFT+F12 to reveal the Actions Window:



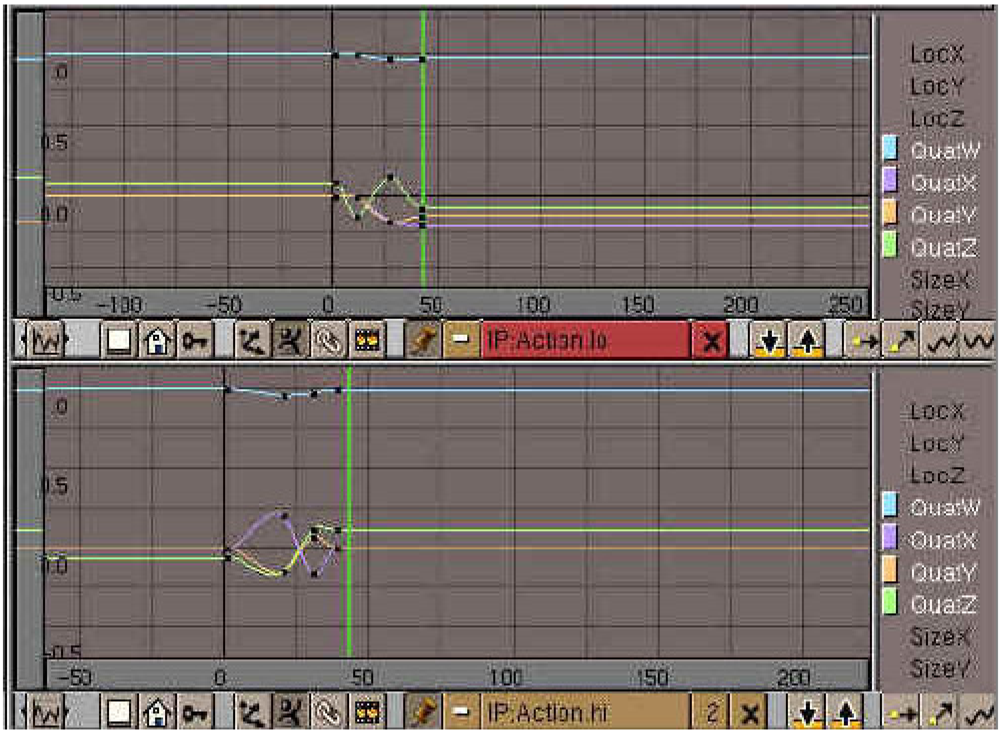
This screenshot tells us a lot. First of all, we can see that Blender has automatically created an Action object, and inserted keyframes for both the "hi" and "lo" bones. It's probably pretty obvious that you can select any of the little dots that represent keyframes (with your RMB) and move the keys around using the standard Gkey method.

However, if you are an obsessive-complusive control freak type like me, you also immediately wonder how you can edit and fine tune the IPO's associated with this animation. Never fear! Split the right hand window a couple more times (by middle mouse clicking the center splitter bar - duh), then turn both of the new windows into IPO windows (SHIFT+F6).



Now, traditionally, Blender only allows you to visualize 1 IPO at a time. If I right mouse click the name "hi" or "lo" on the left hand side of the Actions window, BOTH IPO windows change to show the IPO curves for the currently selected object. The same thing happens if you click the popup IPO selector to change which IPO you're viewing - BOTH IPO windows will reflect the change.

Fortunately the Blender programming crew had the presence of mind to realize that editing Action IPOs can be a hairy mess, and editing one IPO may require viewing another, related IPO for a different bone at the same time. Hence, a new feature called "pinning" the IPO window. See the little stickpin icon in the IPO window header? If you toggle it on, that IPO window is pinned, and will continue to show you that object's IPO even if another object is selected. In this way you can edit multiple IPO's at the same time. In this illustration, both IPO windows are pinned so I am able to edit the IPOs for "hi" and "lo". Very handy...



So much for you control freaks. for the moment let's assume you do NOT need to tweak the IPO's. One other thing you do need to keep in mind at this point is that if you create a new action, Blender will create it as a duplicate of the current action (the same way Blender does with textures and materials). Simply select all the keys in the new action with the Akey, and hit "x" or "Del" to delete the keys and start with a clean slate.

Blender seems to blur the line a bit between an action, which is an isolated bit of animation, and the "live" animation in your project. I still don't have a good grip on this, or its benefits. (In fact, you probably are wondering what the heck I'm talking about.) If you dive in and create multiple actions, you will begin to understand. You can jump out of pose mode and jump out of edit mode, you will see that the changes to the bones are retained, when what you really wanted was for the armature to return to its neutral "rest" position. To avoid any frustration related to this phenomenon, it's a really good idea to create a neutral action that you use as the starting point for any other actions you create. The way to do this is to :

1) jump into pose mode
2) select all the bones with the Akey
3) press the Ikey
4) Insert a key for Avail - this keys everything for every bone
5) press Akey in the Action window to select all keys
6) press the Dkey to duplicate all the keys
7) press the Gkey to grab the duplicate keys and move them ahead 10 or 20 frames

Creating this neutral pose provides you with a rock-solid way to return to your model's original rest position.

All of which brings us to an interesting point you will want to take advantage of : Actions do NOT have to contain any action! You can, for example, on frame 1, move your armature into a specific pose, then simply select all the keys and duplicate them and move them to a few frames later to create a static pose. Why would you want to do this? Actually you will want to do it quite often, because by doing so, you are essential creating morph targets for your entire skeleton. You can then use the non-linear animation (NLA) editor to blend these static poses into a more complex animation. There are two gotchas to be aware of :

1) Don't key *all* your bones like you did in your neutral pose above! If you do, then when you use your static pose, *all* the bones will try to move to their static positions. This is *NOT* what you want. You want ONLY the bones that are in a *DIFFERENT* position from the neutral pose to have keys. (eg, if the pose is an extended hand, and the rest of the model is in a contorted position because of other actions, you wouldn't want the extended hand pose to force all the other bones back to their neutral positions, right? You ONLY want the hand to be affected by the pose).

2) It's not sufficient to create keys on the action's first frame only. You must create duplicate keys a few frames later. Why? Because if you don't, you will go to use the pose in the NLA editor, and (being only 1 frame wide), the animation clip in the timeline is infinitely small and this makes editing it's relationship to the greater animation really, really difficult.

These two points are probably pretty hard to visualize - don't feel bad. After we cover NLA in the next section, it'll all start to make a little more sense.

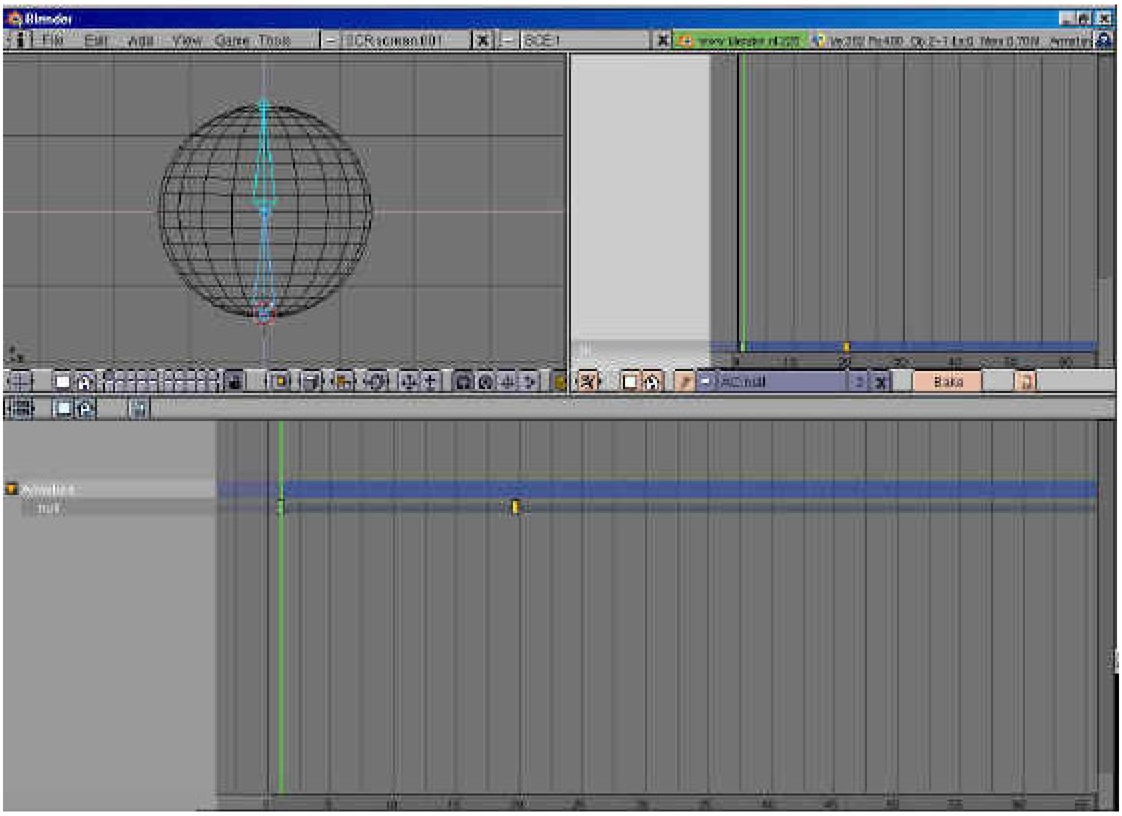
***TIP** : Thanks to Carl for this one...if you want to permanently delete actions that aren't being used from a Blender file - press shift + f4 in any window (data browser), go to the actions - you will probably have to go up one level, then you can just click actions and there they are...*

Actions always have a fake user (F) this is because otherwise you would use all the actions you weren't using (naming in an actuator for example)

Use your right mouse button to select all the actions you wish to delete, then press F, now you see that all the F's with the actions you selected are gone, the fake user is gone.... the number after the actions should now be 0, indicating no users for that action. Save... quit blender and restart the file, now they should be gone....

NLA

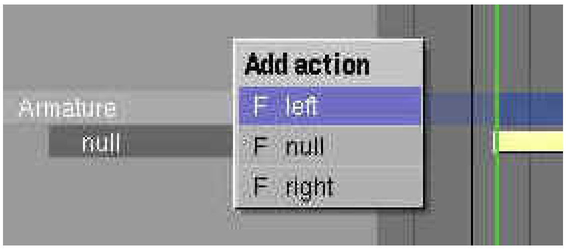
You've already seen a screen shot of the NLA editor. You get to it by pressing CTL+SHIFT+F12 key. I generally open it at the bottom of the screen, so the layout is something like this :



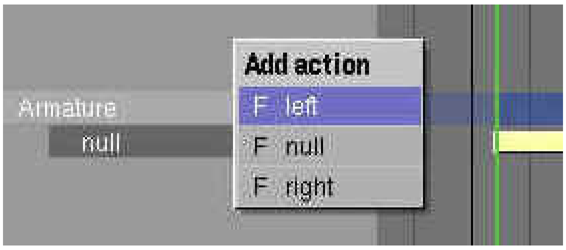
The simple scene here has three actions, "left", "right", and "null", and all of these are actually static poses as I described earlier.

My goal is to have the object bend left, then right, then straighten up, and you will see that although these are static poses, the NLA will smoothly animate between them.

First, Note that there are keyframe "dots" in the NLA window. These represent the keys for whatever action is currently active. This is a feature of dubious value, once you see how the NLA editor works. Really we want to get rid of these keys, but if you select and delte them, you are actually deleting all the keys for that action! the thing to do is place your mouse over these keys in the NLA window and press the Ckey. Confirm that you want to convert these keys to an NLA strip.

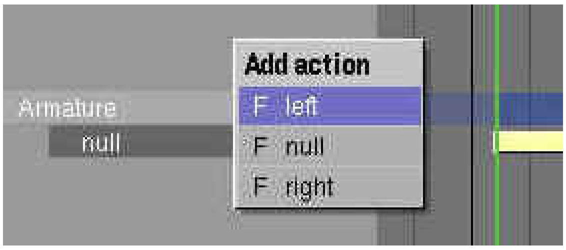


Now you have a strip that "points" to the corresponding action, but if you delete the strip you are not damaging the action it points to :

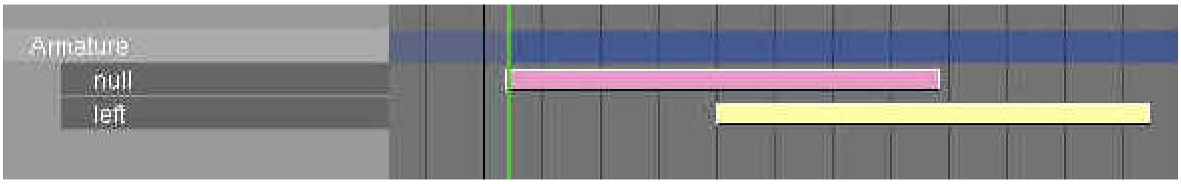


In this case, the active action happened to be "null" and I happend to want to start with this pose, so I will leave it alone. Notice however than when you made this conversion, the Action window changed to reflect the fact that there is no action currently active.

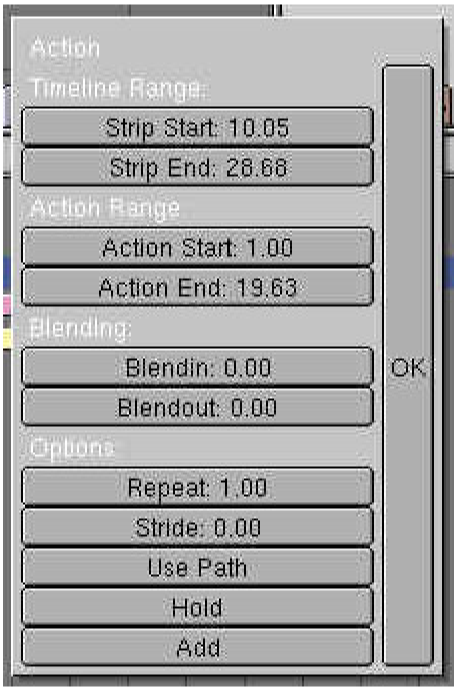
Moving right along, in order to insert an action to the timeline, place your mouse over the name of the armature you're interested in (there's obvously only one in this case), and press SHIFT+Akey, and choose which action you want to insert. We want to add "left" :



Now the new action is mixed into the timeline. I used the Gkey to move this new clip to the right so it starts around frame 10 :



If you play the timeline now (ALT+Akey), you will see that the armature changes instantly from the "null" pose to the "left" pose. That's bad, because we want to smoothly transition into this pose. In order to do this, select the "left" clip, and press the Nkey. this is what you'll see :



Action

Timeline Range:

Strip Start: 10.05

Strip End: 28.68

Action Range

Action Start: 1.00

Action End: 19.63

Blending:

Blendin: 0.00

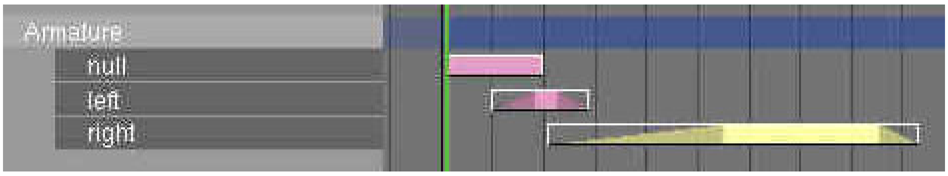
Blendout: 0.00

Options

Repeat: 1.00

Stride: 0.00

What we want to do is change the Blendin to 8. This means this action will blend into the animation over 8 frames. As you will see, this has the desired effect. Next, I added the "right" action and blended it in as well as blending it out, so that after it played, the model returned to its neutral postion.



Note here the DISADVANTAGE to using static poses. You will also see from the screenshot that I wanted the "right"action to go more slowly. I simply selected it and scaled the action strip (Skey). This effectively "spreads" the animation over a larger number of frames. Then, by making the Blendin much longer, the action transitioned in much more slowly. I also gave action #2 a blendout, so that when it's strip ended it did not suddenly jump to the pose in the third action. Play with this a bit and it will all make sense.

By the way, the "Repeat" option allows you to make the action "loop" a certain number of times. Use path allows you to link a motion to a path (such as a walk cycle). "Stride" is the distance (in Blender units) that the character moves in a single cycle of the action, and is only useful if "USE PATH" is specified.

"Hold" can be quite useful. If enabled, the action will freeze in it's final position even after the clip has stopped playing, instead of returning the armature to it's "rest" position. This option pretty much make Blendout's unneccessary, but you'll want to have a neutral pose as discussed above if you will need to get the model back to it's neutral position sometime late rin the timeline.

Finally, the "Add" button causes keys for an action to be added to all other concurrent action clips. This is generally desirable, especially if you are mixing actions which affect common armatures. For example, an action of an opening hand added to an action of a swinging arm to make a complete baseball pitch animation. You want Blender to intelligently blend the motions of the wrist from the two actions, as opposed to having one sequentially replace the other. Make sense?

Well, that's it for now! If you see Mr. P's final animation, feel free to download it.

Also, feel free to let me know how you liked this tutorial!

Products | Tutorials | Galleries | Links | HOME | EMail