

Character Rig 101 Doc version 1.0

This document is to help people just getting into 3D to get their characters setup quickly without too much hassle and still have a rig that is a pleasure to animate with..

There are many sections not written yet, however this version of the doc will get you a complete animatable rig to work with. The sections that are missing deal more with Skin deformations and alternate (more complex) methods for rigging a specific area. This part isn't written yet because there is much more information on that type of thing through the Alias|Wavefront and Gnomon Training videos. This is just a starter, but I believe it to be effective...

Before I get started...

I would like to give a huge thanks to Jason Schleifer for his kindness in allowing everyone access to his wonderful scripts! I spoke with him for about a month and he decided that many people can benefit from his scripts and that it is time for people to have them. This is a major contribution that we will all benefit greatly from! These scripts are just a start however, please take a serious look and Jason Schleifer's DVDs "Fast Animation Rigs" and "Maya Techniques | Integrating a Creature Animation Rig within a Production Pipeline". Both of these are very high quality DVDs and no Animator/Rigger should be without these in their library! (this means you!)

I would also like to thank Alias|Wavefront for making such a quality product and a special thanks to Carla Sharkey for helping me with the Alias|Wavefront permission to make this tutorial! This wouldn't be have possible without your help!

Ok guys... here it is... the Jake Character Rig 101 doc... Now, remember this is just my opinion on how to do things, so if you don't like this way simply don't use it... if you have a different/better way of doing something please email me at **jcallery@subvoicestudios.com** and let me know how/what you did. This way I can keep this doc updated and add in the other stuff... don't worry, your name will be on which part you donated... and please remember... information is to be shared... not kept for yourself.. Open Source for life!

NOTE:

A quick note on the scripts that accompany this tutorial. Jason Schleifer was amazingly kind to us all and has allowed us to download a set of scripts that compliment this tutorial. So please if you ever talk to him or write him, and you find this tutorial useful please send your thanks!

Download the scripts here: <http://homepage.mac.com/jschleifer/HomePage/melScripts.html>

Ok.. to install the scripts just copy all of the scripts to your "My Documents\maya\scripts" folder. This is assuming you are running Maya under NT.

Lets Get Started!

I usually do characters in the polygonal fashion... So that it what I am going to use through out this documentation. However as always this is basically a skeletal rig that can be used for any type of surface... As a side note, if you are using some other type of surface, I would build a low-poly version of your character and rig it as I am about to show... then use the rigged low-poly model to drive the high Resolution mesh for your actual character.. this will greatly increase your real-time response in the viewport... Ok enough talk... lets have some fun!

The Model:

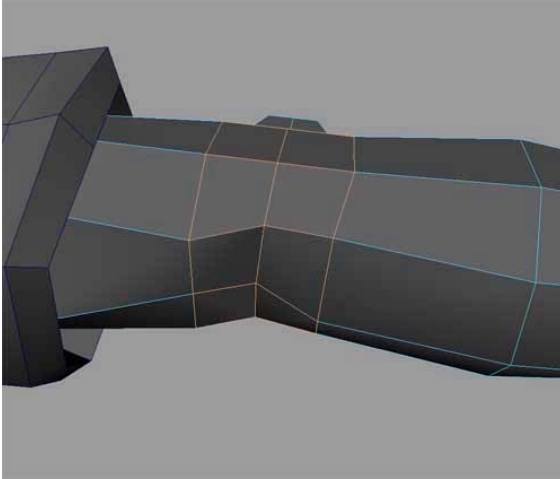
Ok.. As stated before.. I am using polys.. so, here is my poly character:



The first thing that I do is make sure that each piece of the model (assuming that your model is in pieces) is named with a good name so you can understand what you have selected. Also I put a “_low” after the name of each of the pieces to designate that it is the low poly version of that piece... For instance the arm is “r_arm_low”. (see note on naming conventions)

Once you have all of your pieces finished in low poly then what you want to do is put the arms, legs, body, etc.. each in their own layer (trust me this will save you so much time later). So put the arms in the “arms_low_layer” layer and the legs in the “legs_low_layer” and so forth. Now make each layer of type “reference” (change the middle box in the layer section to “R”) so that you can’t select them.

These are some things to note when using this particular rig and how your model should be built. One of the things that seems to work the best for me, is to have 3 flow lines going around the place with the skin is going to bend (elbows, knees, fingers). This gives good control when you paint weights... It will also be a big help when adding influence objects later on. The first and third flow line ring will “pin” the skin in place and the middle ring will be used to maintain volume when the joint bends. The fingers also require some consideration... The spaces of the skin between each joint in the finger should have one flow line ring around the finger to control the fat part between joints..



Quick note on naming conventions:

First off, in the scripting world the naming convention goes like this. If there are two or more words in a name then the words after the first are capitalized.. the first letter of the entire name is lower case.. for instance “lowPolyFoot” or “smoothHead_node”. This goes for every word that is not separated by an underscore “_”... One other thing to remember is that chances are good that you will be writing some sort of script to help you animate, and even the simplest of scripts can become hard to write if your names are tough for the script to generate... To solve this problem I label anything that is similar in the same way... for instance if I grouped all of the face extras together (teeth, ears, eyes, etc..) then the group would be called “faceExtras_group” same goes for nodes and things like “headSmoother_node”. This way if I want my script to go through the scene and select all of the groups I can tell it to find anything that ends in “_group” or another example would be, any extra iconic control that I make for the rig I end in “_ctrl”.. that way I can make a script that does things to every control like record their starting position so that the rig can be set back to the bind pose.. however all of this comes later... just try to keep in mind good naming conventions, so your life becomes so much easier! Also, don’t forget to name your joints!! (Did you like that segue?)

NOTE: These instructions were written assuming that positive “Y” is up and positive “Z” is coming towards you and that the front of your character is facing in the positive “Z” direction (the front of the character is facing you).

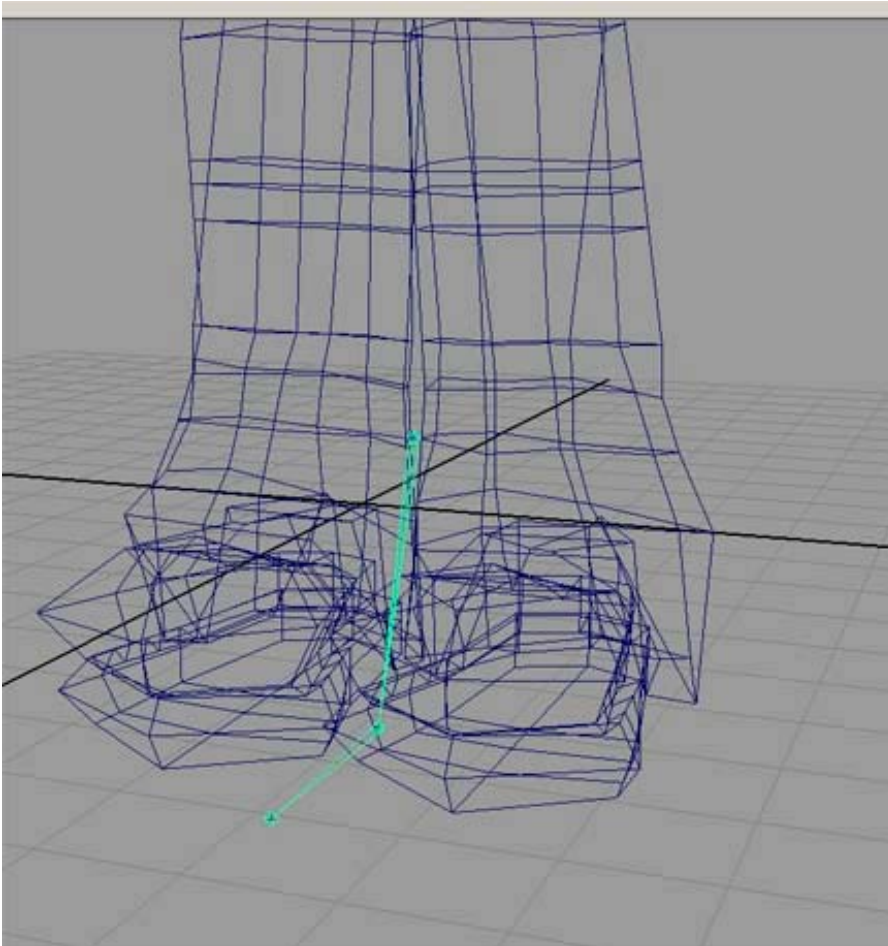
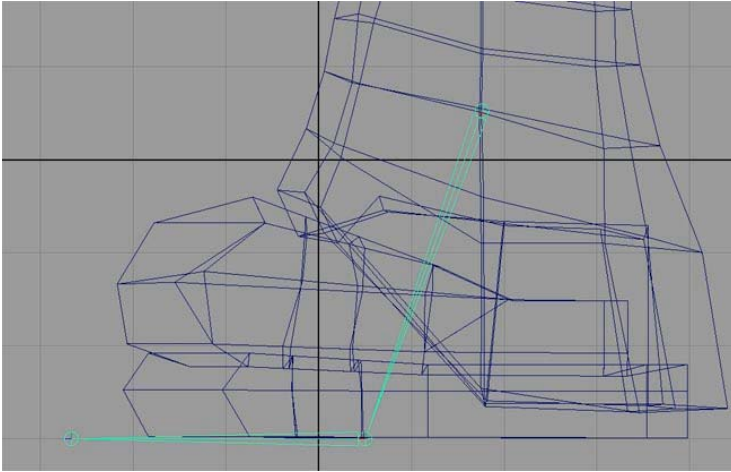
SKELETAL CONSTRUCTION

JOINTS:

This section will vary greatly depending on your character, however most of the concepts are the same.. so for this example I am using a biped.. I am going to start from the feet up through the neck..(simple to complex) As a note I usually built the legs, feet, spine, and skull in the side view and then move them into position before doing the arms and then finally mirroring them.. all this means is that until you move the joints they will appear in the middle of your character and not within the skin. Don’t worry, as Microsoft frequently states in their help docs.. this is not a bug.. this is a feature... ☺

A.The Feet

For my character the foot setup is going to take 3 joints.. the toe, the ball, and the ankle..(named from left to right).



As far as joint placement goes that really does depend on your character... I would do some quick smooth bind tests of each joint to see what placement is going to give you the best starting point for weight painting.. Assuming your character is symmetric (as far as the skeleton goes) then for now we are just going to build the left side then mirror the joints for the other side and then rename them. When placing the joints start at the ankle then ball then toe.

B. The Knee

Start at the hip, and place the hip joint some place in the middle of the hip, then place the knee in the middle of the three knee “rings”, making sure that the knee is forward of the hip to guarantee the proper preferred angle or bend for the upcoming ik. Place the hip, then the Knee, then parent the ankle to the knee joint. make sure to name your joints!

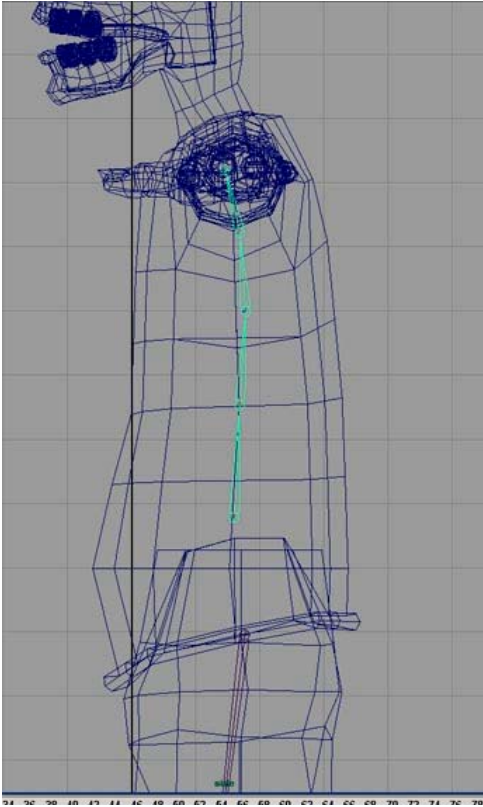
C. The Spine

This is always a tricky subject.. now there are many ways to do a spine and some are very complicated and some are very simple.. Your spine setup greatly depends on how you like to animate.. I am sure that some of you out there like to work the animation much the same way you would do stop-motion animation.. Move each joint and set key... For me though I never want to touch a single joint once the character is rigged. I find them cumbersome and tough to select from odd camera positions and they all look the same making it tough to grab the correct one.. (especially from the top) Not to mention that they are inside of the skin.. what a pain..

If your job is to make rigs for people then you must take into consideration how the people you are rigging for like to animate. Chances are that if you hold a rigging position, then your place of business is most likely fairly large, and therefore chances are good that many different people will be animating with your rig. Therefore you must make a rig that will behave like each animator wants it to... this is very difficult, but can be done... for the scope of this doc though I am going to rig for the way I like to animate... this cuts down on my options on how I can animate, but over the years I have realized how I like to animate... this is unfortunately different for all of us... So in the *Specific Rigs* section at the end I will show you some solutions to please everyone... if that is even possible... But for now this is the spine I like to use... It has its drawbacks, but it has plusses also...

From my research on this (yes, hours in front of the mirror is less than comfortable positions ☺) I have discovered that the spine works like this. There is the lower portion of the spine that basically mimics what the hips do. It is quite difficult to make the hips and the lower back twist in different directions.. So the lower back is just one joint connected to the root/middle hip. The upper back seems to be able to *roll* in a direction, however it is very difficult to make the bottom part of the lower back roll left while the upper part rolls right. So with that in mind the upper back is going to consist of three joints for smooth rolling. l_u_spine, m_u_spine, u_u_spine.

Start with the middle part of the lower back and place a joint. Place the next one at the base of the upper back. The next is just above that, and the last is just below the base of the neck.. Do not parent the lower back joint to the root yet. (you haven't made a root yet ☺)



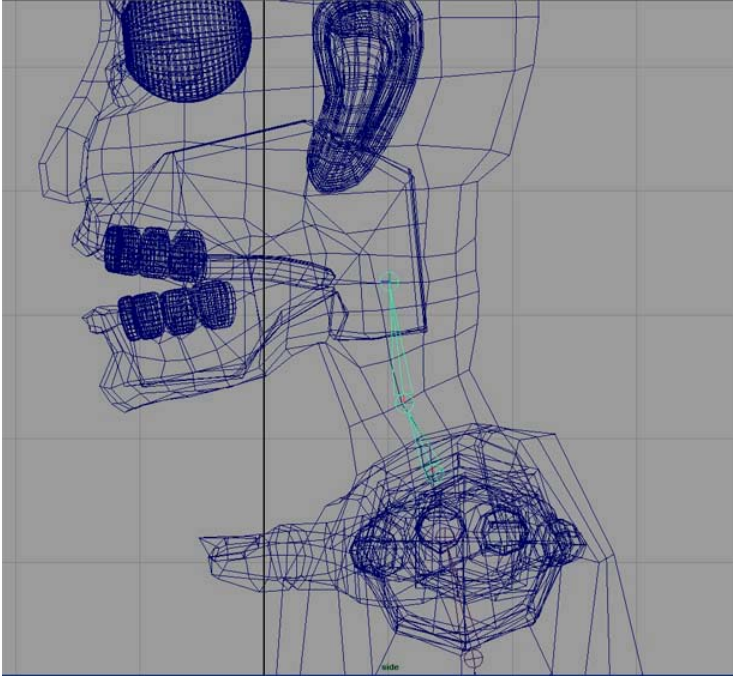
The spine is a fun one to mess with... so we will come back to rigging the rest of the spine in a but, but for now we are going to put down the rest of the joints.

D. The Neck

Again, from the many hours in uncomfortable positions in front of a mirror I have discovered that a decent neck rig can be created with as little as two joints (the less the better!) A better result can sometimes be achieved though the use of three joints, the choice is yours. It seems that the neck bends at the base (just behind the clavicles) to get a general position for the head, then the head bends at the base of the skull and the top of the neck to get direction. With this method you can bend the neck forward and look up as if looking up a ladder or up a ventilation shaft. Generally though I use one more joint in the center of the neck to achieve a slightly smoother bend in the neck.

Place a joint at the base of the neck behind the clavicles, then one in the middle of the neck and one at the top of the neck or the base of the skull. Do not parent the neck base joint to anything yet..

Have you saved yet? You should have saved at least 5 times already, and probably started a new version of your scene just incase something bad happens to the file! Also don't forget to name your joints! You will thank me later I promise!



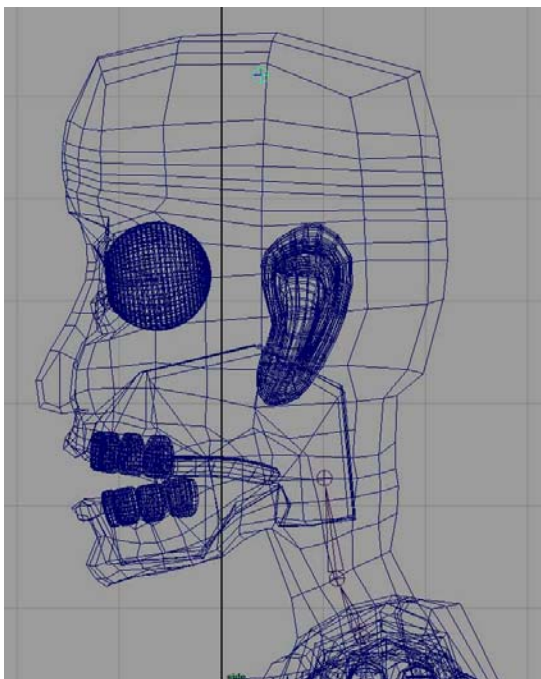
The neck will be rigged in a similar manner to the upper spine.. Sounds like fun doesn't it!

E. The Skull (ooohh scary ☺)

The skull is actually very simple as far as the skeleton goes.. its just one joint. However it brings up a great point.. We like to use the metaphor of human bones to get our heads around the concept of using joints to move meshes. The metaphor actually works quite well, but it sort of blinds us to what is actually going on in the software. There are many tools in many different packages that use metaphors to help us understand what the tool does in the digital world... It just compares the digital tools to a physical tool, so we can quickly grasp what the tool is supposed to do. This is fine but the real interesting things the tool can do are hidden by this metaphor. So whether or not you care to know (you should care you know ☺) I am going to try my best to explain joints to you. Ok here goes... A *joint* is simply a position in space that is going to be used as a center point. A digital joint usually includes the joint (the ball end) and a bone (the pyramid "bone"). It turns out that the *bone* part does nothing... it has no bearing at all on the skin or anything else for that matter.. it is simply a visual representation of a distance and a direction from one joint to the next. Therefore the only part that really matters is the ball joint part. Like I said before the joint part is just a pivot point for whatever is attached to it. (Skin, geometry, etc..). Smooth binding, simply calculates for each vertex what percent of the rotation of the joint each vertex should rotate around the joint. Weights go from 0 meaning 0% of a rotation, to 1 which is 100% or all of the rotation. For instance if a vertex is *weighted* to a particular joint 0.1 or 10% that would mean that if the joint rotates 90 degrees the vertex only rotates around the joint 9 degrees. Since each vertex rotates in a perfect circle around the joint (pivot) geometry inside the bend will seem to collapse.. this is where influence objects help tremendously.

It is also worth noting that edges don't really exist on poly geometry either. The vertex is all the data the computer has to work with.. the edges are generated from these vertices. Therefore all poly geometry is simply a collection of points in space creating what is called a point cloud. Now that you know this try and think of cool things you can do with these tools and information that the tool was not designed to do. A quick example of my work that uses vertices as position information is a piece I did that makes a face out of planes, that is driven by black and white video... What I did was shoot some video of my face lit from above (to create highlights and shadow areas). I made the footage black and white and imported it into Maya as a displacement map on a plane. The vertices of that plane were raised and lowered based on the gray value of

the pixel that vertex represented in the footage. I then wrote a script that made a bunch of squares and for every frame moved those squares to the same position in space as each of the vertices of the displaced plane... pretty cool huh?

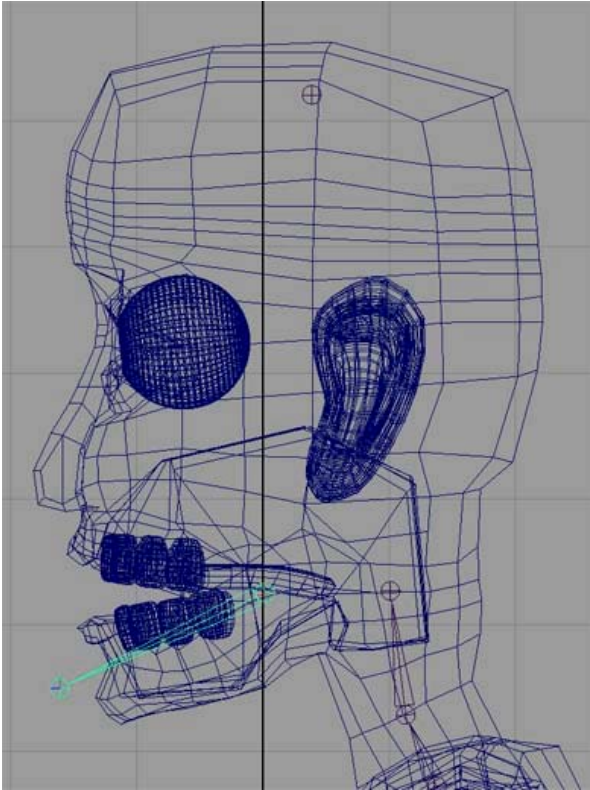


Ok.. so what all of this babbling means for the skull is this.. all you need in the skull is one joint.. This joint is basically a geometry collector. That means that most of the head will be weighted 100% to this joint so it doesn't bend or move at all.. So place a single joint at the top of the head.. call it "skull"

F. The Jaw

The jaw is a fairly simple setup, the trick is in the weighting of the skin at the corners of the mouth... the other trick is in the blendshapes for the cheeks based on the jaw open distance. We will get to that later... (This whole rig thing is pretty in-depth huh?)

First put down a joint to be the base of the jaw, then place one more joint at the tip of the jaw, just past the chin... Now remember that the only joint that matters is the "jawBase" joint. This will serve as the pivot point for all of the vertices in the jaw area. What this means is that the jaw tip isn't even needed... however it does make the jaw setup easier to look at.. ☺ When it comes right down to it.. you will be looking at your rig for very long periods of time and these times will most likely be very late at night, so it better make absolute sense to you, because your brain won't be in good shape at times when animating... which is fine, because animating is about feeling and not thinking, but if your rig makes you think then you are in some serious trouble ☺



NOTE: The jawBase is used as the pivot for the vertices, so put it in a spot that makes sense to rotate the jaw with. Also don't forget to name your joints and save lots and lots!

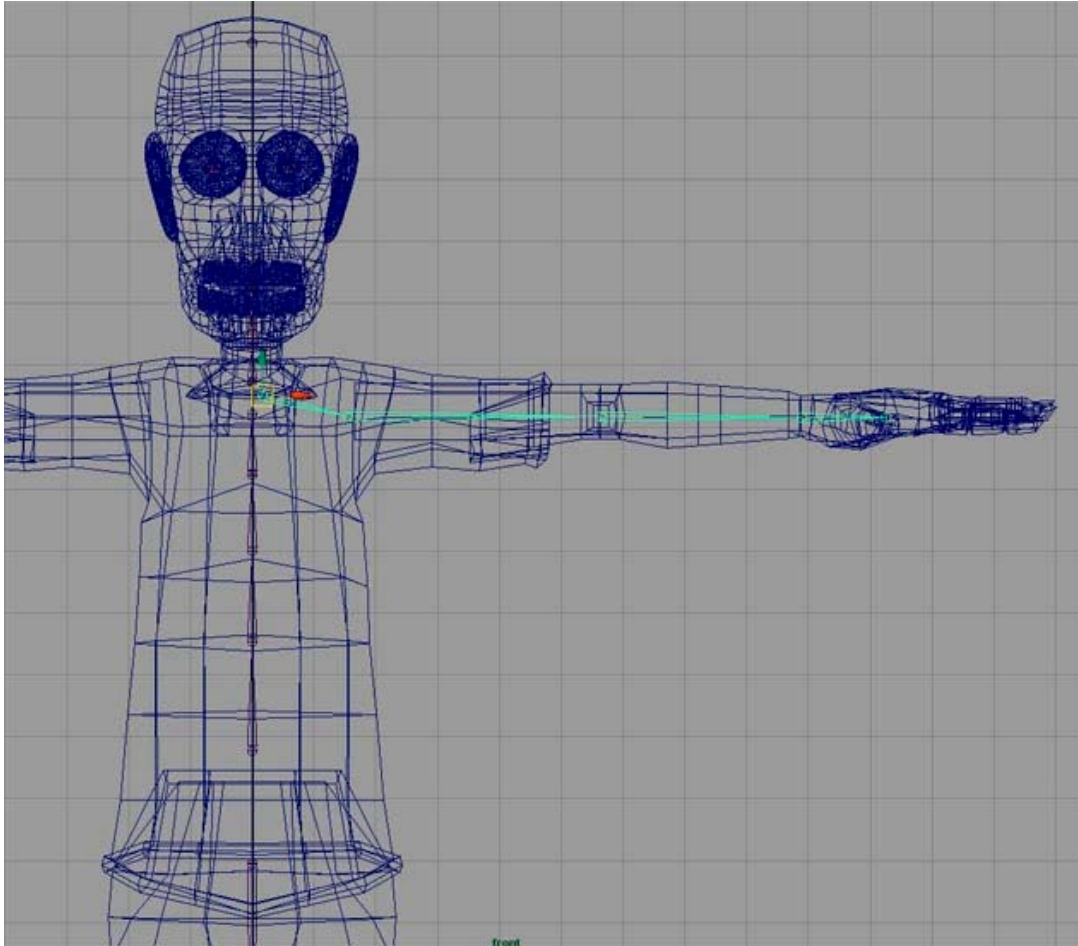
G. The Arms

The arms are pretty much like the legs turned on its side.. however instead of a hip we have a shoulder and a clavicle and a hand to deal with (ok.. so not much like a leg). So lets start putting down our joints. For now we are going to be doing just the left arm and we will mirror the right arm later.

Switch to front view.

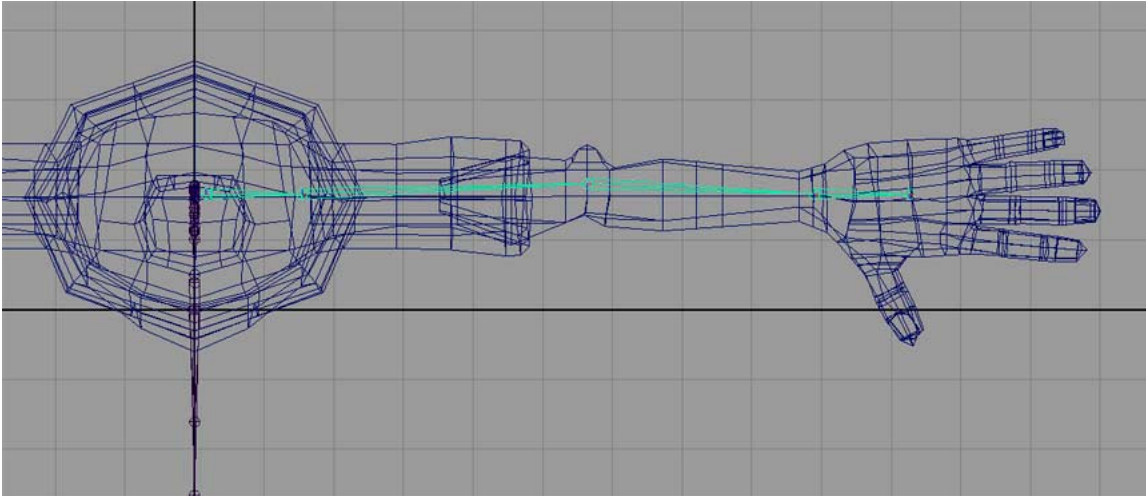
Place a clavicle joint just to the right of the neckBase joint, then place the next joint, the shoulder joint, exactly level with the elbow, then put down the elbow joint perfectly level with the shoulder joint and then the wrist joint then a palm joint.. Most people don't use a palm joint but I do and I'll show you why later.. (it has to deal with correct orientation of the iconic controllers ☺)

Label your joints! and also save it if you haven't.. please remember to save many times and make many versions.. this way you won't loose much time if your file gets fudged up..



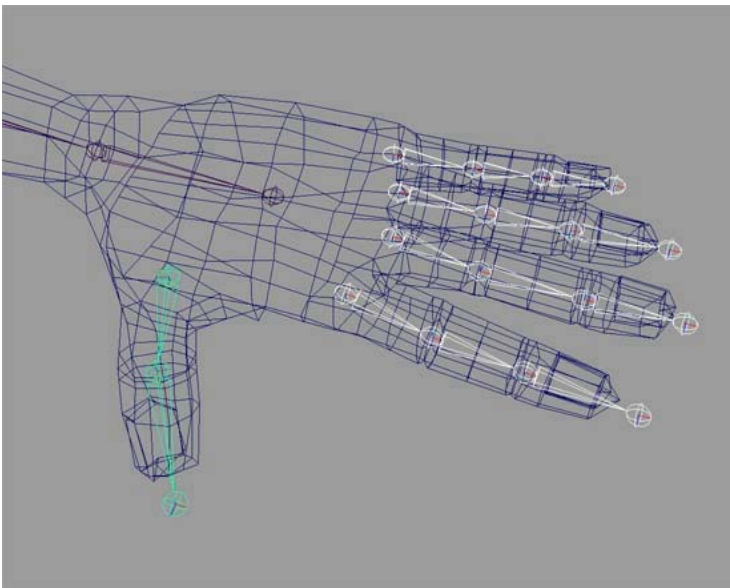
Once the front view placement is complete switch to top view. Now you want to slide the entire arm joint system back so that the clav, and shoulder joints are where you want them. Now you must move the elbow back to ensure a proper preferred angle for the IK system. To do this, select the elbow, and switch to the move tool (w)... Now press insert. This will allow you to move one joint at a time. Now move the elbow joint back slightly so it is behind the shoulder joint...

Also move the clavicle joint toward the front of the character to where it would really be if the character existed in the physical world. Again think of the joint as a pivot for vertex rotations.



H. The fingers

The fingers are straight forward. Place a joint at the knuckle, the second finger joint and the third joint at the end of the finger and one just past the tip of the finger... Make sure to put the joints in the middle of the *three rings* for each joint of the finger.



Parenting The Joints

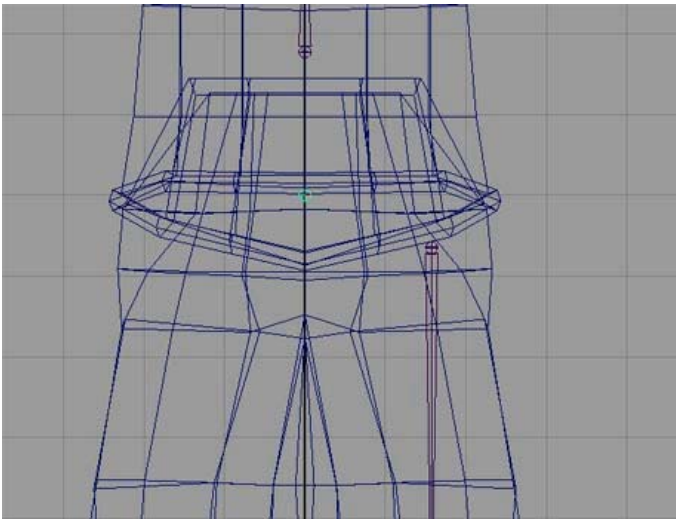
Ok.. cool.. the joints are all placed nicely in the side view.. now lets place them correctly in 3D space and parent them all to each other..

Switch to front view and move the leg joints to their correct places inside the leg. It would be best

move the leg joints by the upper most joint (hip) to insure that the joints are all in a vertical line to get the best results from IK.. if your character's legs are bowed then add an extra joint ("innerHip") between the root and the top of the leg (hip joint). Then move the outer hip joint outside of the mesh to get the knee where it should be.. Now when weighting just weight the outer hip joint to nothing and weight the new inner hip joint as you would have weighted the now outer hip joint.

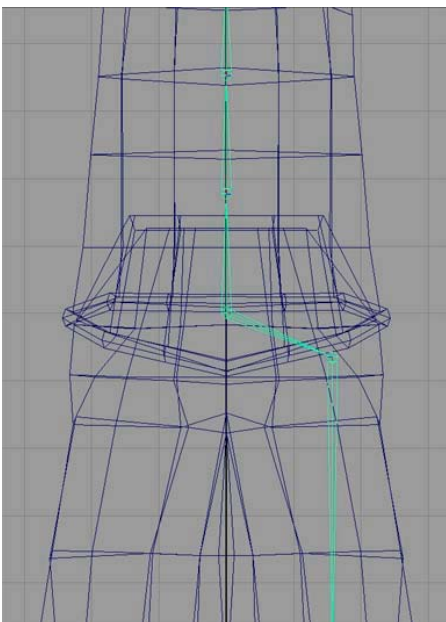
Now is a good time to make sure that all of your joints are in the right places... don't rotate any of them... just select them, select move and press insert and move them to where they need to be.. Shortly we will be taking care of orientation... get your joints to where you want them to be because soon it will be hard to move them without breaking something... This rig is designed in such a way that going back to fix things is not difficult, however there are still some things that are better done right the first time... this is one of them...

Ok, now that you are sure of the placement you want for the joints it is time to parent them all together.. the first thing we need is a root joint for the whole structure... Place a single joint at the center of the hips and name it "root"

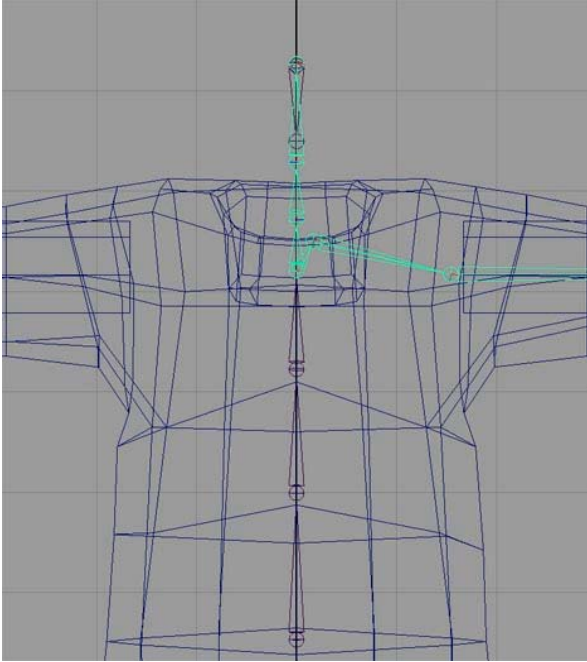


Now select your Hip joint and then select the root joint and press "p" or go to Edit->Parent... This will connect the two joints making the root joint the root of the structure..

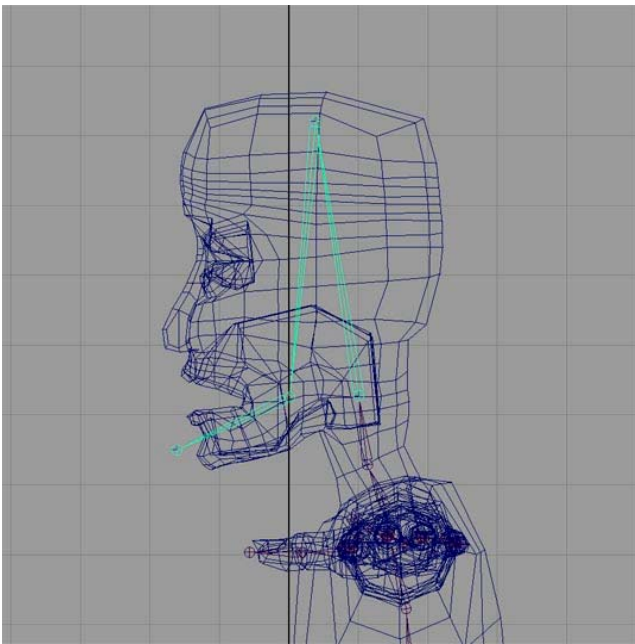
Now select the lower spine joint and then the root joint and press "p"



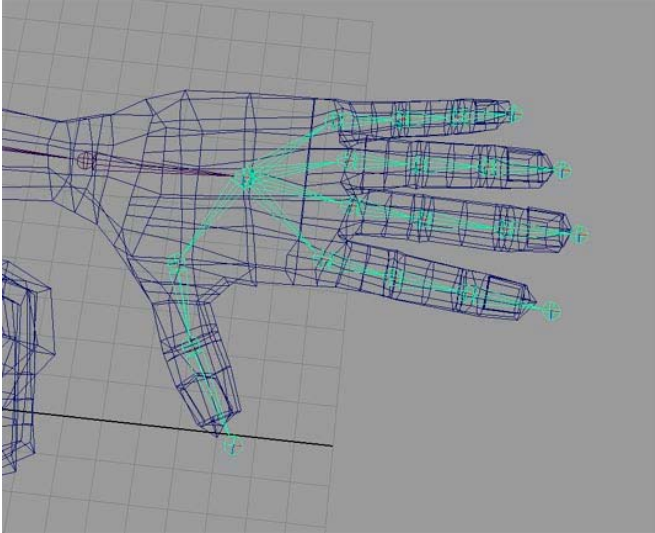
now parent the clav joint and the "neckBase" joint to the tip of the "t_u_Spine" joint..



now parent the "jawBase" and the "neckTop" joints to the skull joint



now take the base joints of each finger and parent them to the palm joint

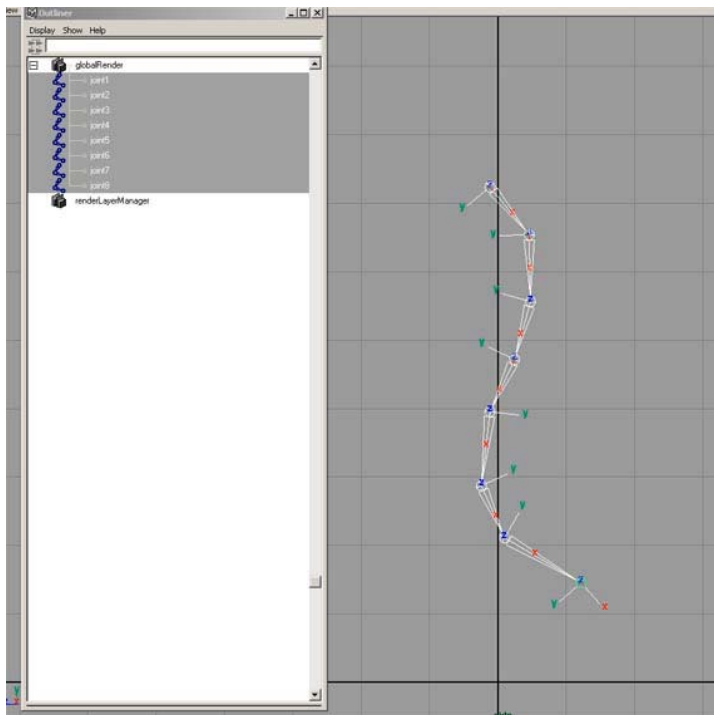


There you go! Half a skeleton! Two more steps and you will have a whole skeleton!

The Joys of Joint Orientation

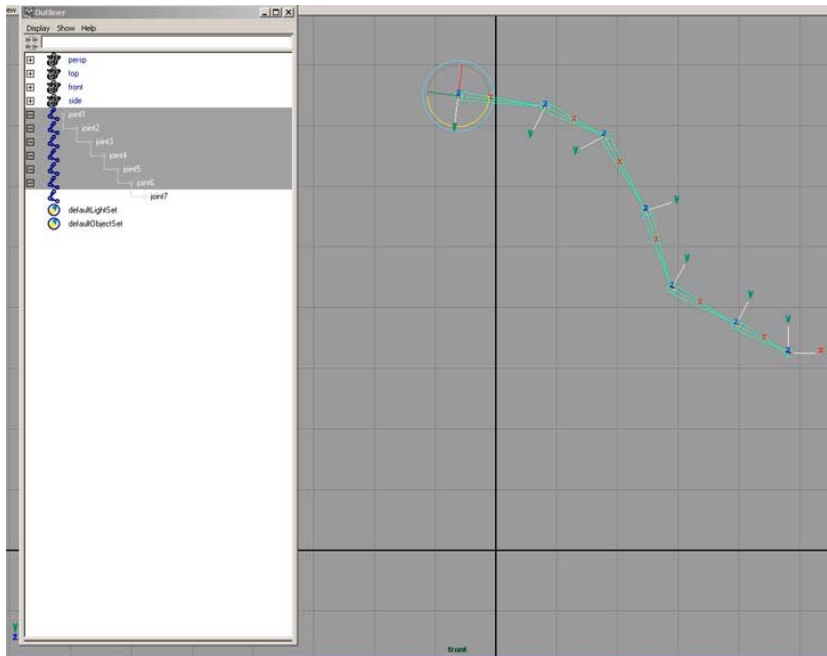
This is probably the most important part of building a skeleton. What this essentially means is that for any given joints the surrounding joints have a matching local rotation axis or orientation. To help explain my point... an example...

Lets say this is a spine or tentacle or rope or whatever... what is important is the local rotation axis... to view these select your joints (one at a time) and Display -> Component Display -> Local Rotation Axis



As you can see the rotation axis in the curve go opposite directions depending where in the curve you are... this is bad because if you want to curl the chain it curls wrong...

Select the first joint and then shift->select each joint one at time after that... now rotate..



To fix this the easy way is to use a script by Jason Schleifer
“jsOrientJointUI.mel”

This can now for the first time be obtained from Jason Schleifer's Website at
<http://homepage.mac.com/jschleifer/HomePage/melScripts.html>

If you don't have the script you can wait until the end of May 2003 and I will have the instructions on the hard way to do it... or just search the net for a tutorial on joint orientation... there are a few good ones out there... I can't send you to those links because I don't have permission from the owners, however I assure you that they are out there... There are essentially two ways of doing it... one involves copying the rotation values and putting them into the joint orientation boxes in the attribute editor... The other way involves unparenting joints and aiming the joint down to the next one with a specified up-vector... Sorry, I don't have it yet... But soon ☺

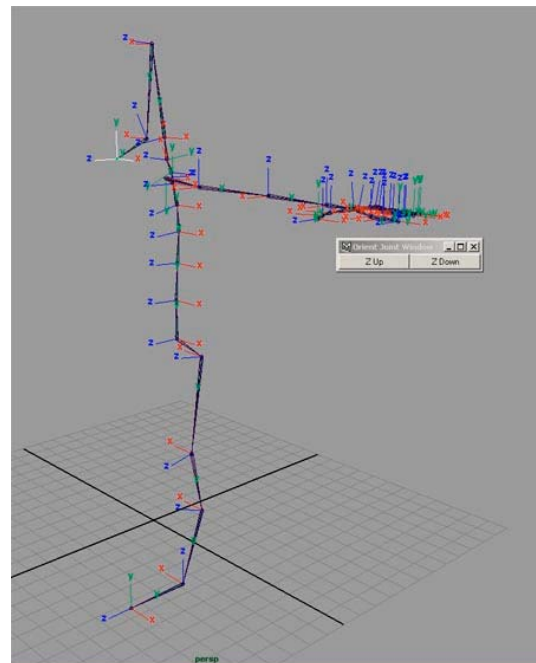
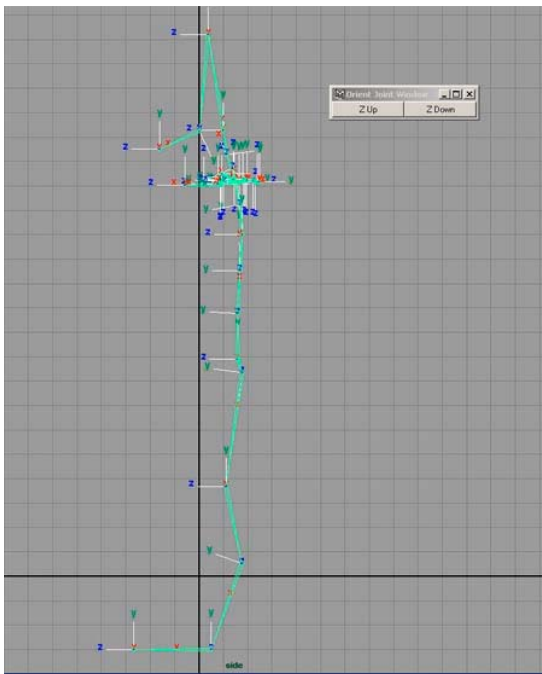
If you have jsOrientJointUI.mel then just follow these directions...
type the following command into the command line window at the bottom left of maya's UI:

```
jsOrientJointUI.
```




Now simply select your joints one at a time and press either Z-up or Z-down to make all of your local rotation axis point in the same direction.. Note this script points the Y-Axis toward the next joint.

Now that has been explained, I will just do the same thing for each of the joints in my character..
Note: Don't orient the root joint.. and don't worry about the tip joints...



And there you have it.. properly oriented joints... ☺ Mmmm... joint orientation...

Mirroring The Joints

Cool... there is half a skeleton...

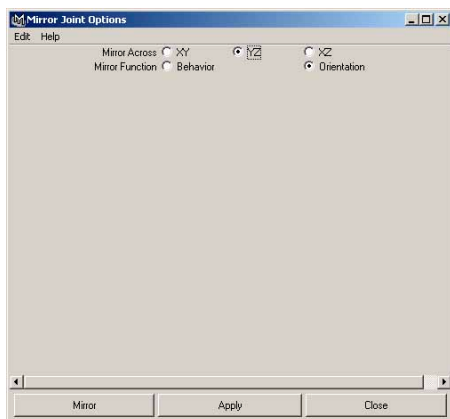
Have you been saving? how many versions are you up to now? Make another one.. just for kicks..

Oh. don't wine.. it saves quickly... (Note: don't save Maya files across any kind of network!

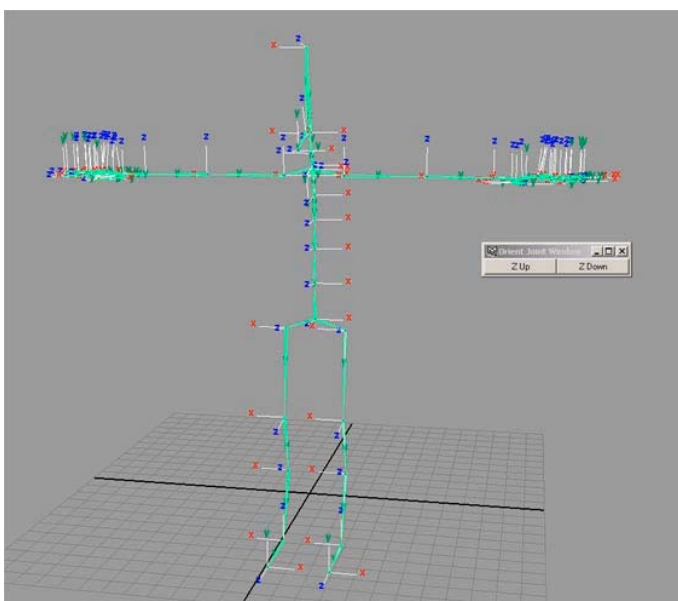
Local saving only.. then when you are done.. copy what you have onto the network!)

ok... now... to make the right side with very little work...

ok.. select the clav joint... Now go to the option box for skeleton->mirror Joint->options
 Set mirror Across YZ and Mirror Function to Orientation... depending on which direction
 you built your skeleton you might have to change the Mirror Across setting to do what you want..
 A tip for Mirror Across: The direction that is not in the pair is what it is mirroring... for instance
 YZ means it is going along the X axis to mirror...



Ok.. now do the same for the hip joint...



Ok... one whole skeleton... schweet! Now for the really icky parts ☺

Note: to turn off the local rotation axis select the joints and select display->component display -> Local Rotation Axis again... and they will turn off.

One last thing... use the outliner and select all of your joints (or just select them in the viewport) and go to Modify -> freeze transformations -> options and select only rotation and press apply. Now if you absolutely must return to the bind pose and there is no other way of doing it.. you can turn off your IK and manually rotate each joint back to 0,0,0 and you will be guaranteed that you are at your bind pose... (when I say select all of the joints I don't mean just grab the root, I mean shift select each and every joint... just in case I was unclear!)

NOTE: This might be a good place to save... then save a test version and do a smooth bind to your geometry... just to see if you joint placement was correct... if you have to go back and move

your joints (which you should if they are not right.. this will not be correctable later) then just select the joint, select the move tool then press (insert) and move it where you want.. don't forget to re-orient your joints after you move them! good luck!

SKELETAL RIGGING

IK Setup

Ok the “controllers” part of the lesson is derived from Jason Schleifer’s Maya Master Class from siggraph. This is a technique that he calls Iconic Representation of Character Controls... and I must say that it is a very cool idea... since I was on a mission to never have to touch a joint again this was a great stepping boulder to that goal. The concept is simple and I am ashamed of not thinking of it my self... What we are going to do is this... we are going to model some nurbs curves in pretty shapes to represent our controls for our character. This way when we are looking at our character from all kinds of weird angles we can easily grab the control we want. Many people use locators, but since they all look the same it is very hard to tell which control you are selecting... So.. lets start with the feet!

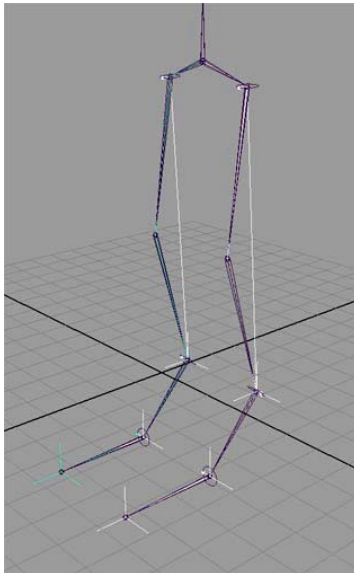
Leg IK

The first thing we need to do to the feet is set up the IK...

Select the IK tool and make an IK handle from the hip to the ankle... then make another from the Ankle to the Ball... and one last one from the ball to the toe... Now name them all ☺

l_ankle_IK, l_ball_IK, l_toe_IK (for the left side)

Now do the same for the right..



Now we are going to do some special grouping... why we do this will all become clear once we install the controller... worry not young jedi.. just trust me for now ☺

There is two different ways to do this...

I will explain the other way later... you can choose which way you like to work.. try them both!

Way One ☺

Ok.. first select the “l_ankle_IK” and go to edit->group.. this will group the IK handle to itself... thus making a clean transform node for us to mess with...Name that group “l_ballLift_group”

Deselect everything...

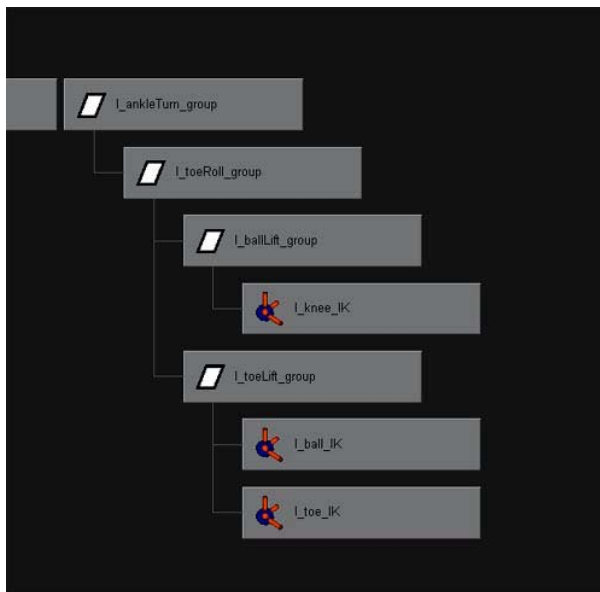
Select "l_ball_IK" and shift select "l_toe_IK".. Edit->group... Name that group "l_toeLift_group"

Deselect everything...

select the "l_ballLift_group" and the "l_toeLift_group" (the groups not the parts inside the groups) and Edit -> group... Name that group "l_toeRoll_group"...

Deselect everything...

select the "l_toeRoll_group" and Edit -> group.. this will group that group to itself.. I know...sounds silly.. but I promise it will all come clear to you.. ☺ name that group "l_ankleTurn_group"...



Now we must move the pivot points of the groups...

select the "l_toeLift_group"... select the move tool... now press insert and snap the pivot to the "l_ball" joint... (to snap hold down "v" while moving the pivot)

Now select the "l_ballLift_group" and snap its pivot to the "l_ball" joint...

Now select the "l_toeRoll_group" and snap its pivot to the "l_toe" joint(the tip)...

Now select the "l_ankleTurn_group" and snap its pivot to the "l_ankle" joint...

Cool.. now do the same for the right foot...
(read carefully... this can get a bit confusing...)

Way Two

(needs to be written)

"Reverse Foot Method"

Arms IK

Select the IK tool and make an IK handle from the "l_clav" joint to the "l_shoulder" joint.

Now make another from the "l_shoulder" joint to the "l_wrist" joint..

Name the first one "l_clav_IK" and the second "l_wrist_IK"

Have you been saving?

Now do the same to the right side...

Arms IK with mid-forearm joint
(needs to be written)

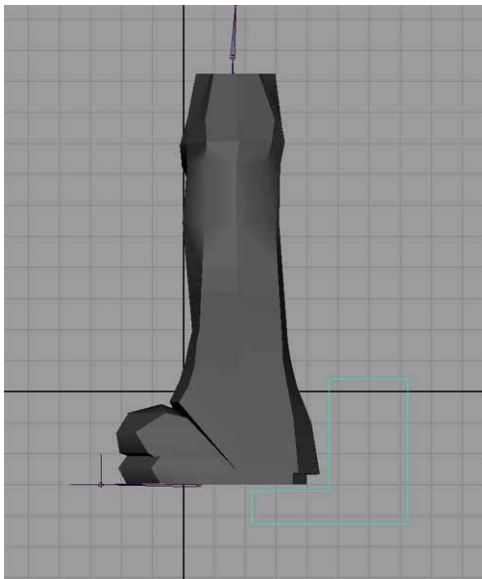
CONTROLLERS

Foot Controls

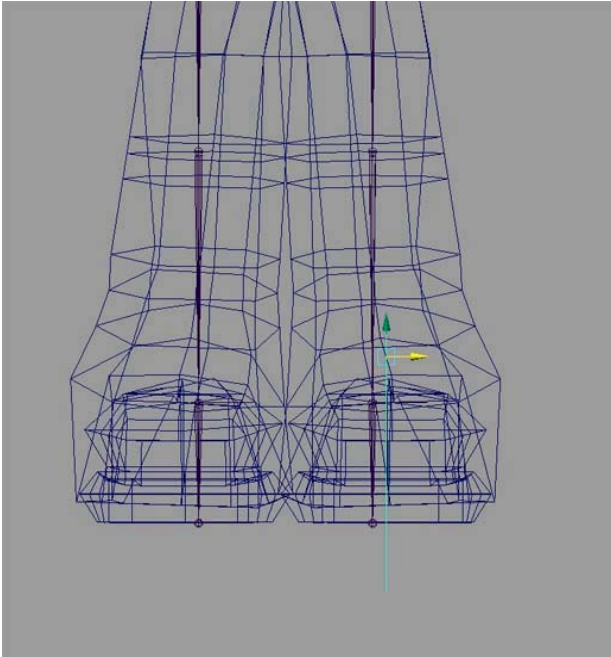
Ok.. now that the IK is finished it is time to make some controllers to animate our character with... We will as always start with the feet...

Switch into the side view and select the EP curve tool..

With the EP curve tool draw a foot controller.. any shape you like.. I would however suggest that the shape extends up the back of the calf a bit so you don't have to look under the floor to grab your foot controller.. Name it "l_foot_CTRL"



Now switch to front view and align the controller to the center of the foot



Now snap the pivot of the new controller to the "l_ankle" joint...

With the new controller selected Modify -> Freeze Transformations -> options
select scale, rotate, and translate...

Press Apply.. now when you want to get back to the bind pose you can just set your controller
to 0,0,0 in transform, scale, and rotate...

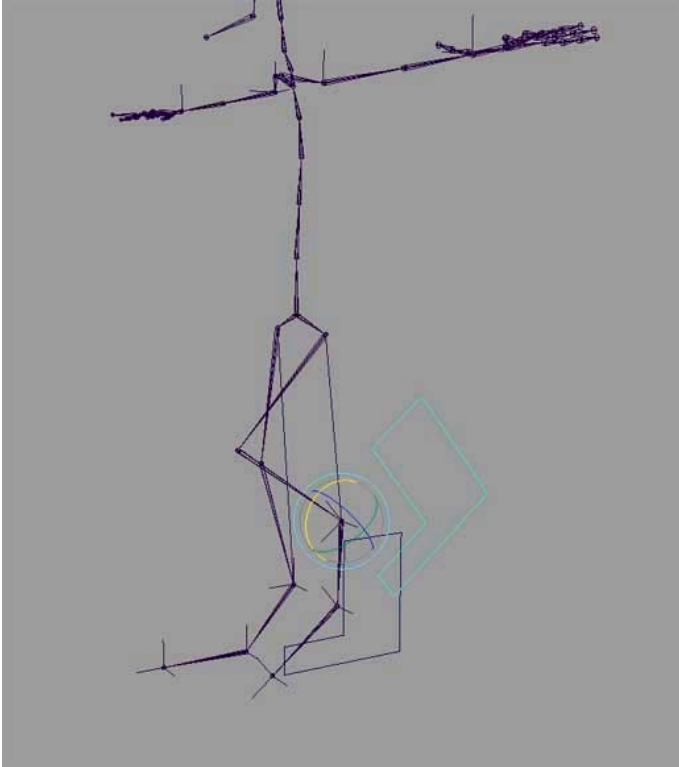
Do the same for the right side..

Now select the "l_foot_CTRL" and then shift select the "l_ankleTurn_group" (make sure you
select the group! Use the outliner to make your life easier)

Now Constrain -> point

And then Constrain -> orient

This *attaches* the Ankle turn group to the controller.. now when you move the controller the whole
foot moves...



do the same for the right side.

If you do move your stuff around to test it be sure to undo it or zero it out before you continue...

Toe Attributes

Now we are going to take care of the toe lift, and the toe roll, and ball lift... Use these to make a nice foot roll...

Select the "l_foot_CTRL" and press **ctrl-a** to get the attribute editor...

In the attribute editor go to the attributes menu and select add Attribute

Fill in the Attribute name : Ball_Lift

Set the data type to float (numbers with decimals e.g. 3.141592)

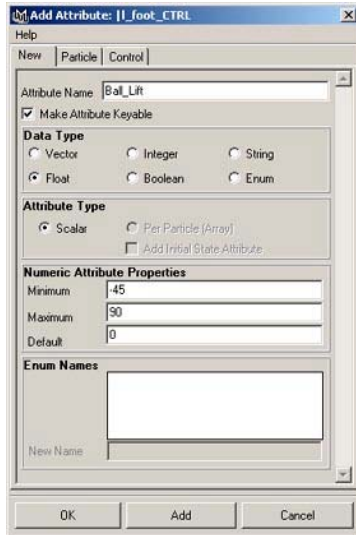
Set Minimum to -45

Set Maximum to 90

Set default to 0

Note: These numbers can be anything you want... I just like this many values to work with.. Play around with these until you find numbers you like to use...

Now Press **Add**



If you'll notice, in the channel box for the "l_foot_CTRL" a new attribute has been added, yeah!

Make 3 more attributes: Toe_Lift, Toe_Roll_X, Toe_Roll_Y

Do the same for the right side

Driven Keys

Now its time to *link* the new attributes to the foot actions... we are going to do this through driven keys... This same method will be used later on the fingers...

Animate -> Set Driven Key -> set -> option..

this is where the key frames are set for the different foot attributes...

Select the "l_foot_CTRL" and select **"load driver"**...

Highlight "Ball_Lift" in the right pane in the Driver Section...

Now select the "l_ballLift_group"

Press **"Load Driven"**

Highlight "RotateX" and press **"Key"**

This sets a key for when the "Ball_Lift" attribute is at 0...

Now select the "l_foot_CTRL" and set "ball_lift" to 90...

Select the **"Selection"** menu from the Set Driven Keys window and select **"Driven Items"**

Now rotate the "l_ballLift_group" as far forward as you want it to go and press the **"Key"** button in the set driven keys window

now select the "l_foot_CTRL" and set "Ball_Lift" to -45

Select the "l_ballLift_group" again and this time rotate it as far back as you can and press the **"Key"** button... now you should have a working "Ball_Lift" attribute! ☺

Repeat these same steps for each of the other attributes...

It seems that this Driven Keys technique is used most with finger setups...

Simply make an attribute in your finger control for each finger curl and key the joint rotations with those attributes for the fingers... now you can curl each finger one at a time or select all of the attributes and curl them all at once... its usually good to have a *spread* attribute also to open then fingers up...

(See Driven Keys for Feet)

Also as a quick note if you select an attribute in the channel box and then drag left and right with the middle mouse button in a viewport it will increase and decrease the values for that attribute.

PoleVector Constraints

Ok.. one last thing for the foot setup... Pole Vector Constraints...All this means is basically point the joint at this object in space... very useful for knees and elbows...

Create -> polygon Primitives -> sphere

set the axis and height to six sections or so... some low count..

Select the new sphere and press **ctrl-a** (attribute editor)

drop down the display section.. then drop down the Drawing Overrides

Check **“Enable Overrides”**

Uncheck **“Shading”** and **“Texturing”**

Now select the **"Shape"** Tab (second one from the left)

Drop down **"Render Stats"** and uncheck everything in that section..

Now you have a sphere that will not render and will not shade in the viewport... makes for a good control...

Select the sphere and shift select the "l_ankle_IK" and constrain -> poleVector Constraint

Now the knee will point to the sphere... very cool... name the sphere "l_knee_CTRL"

Now do the same for the right side...

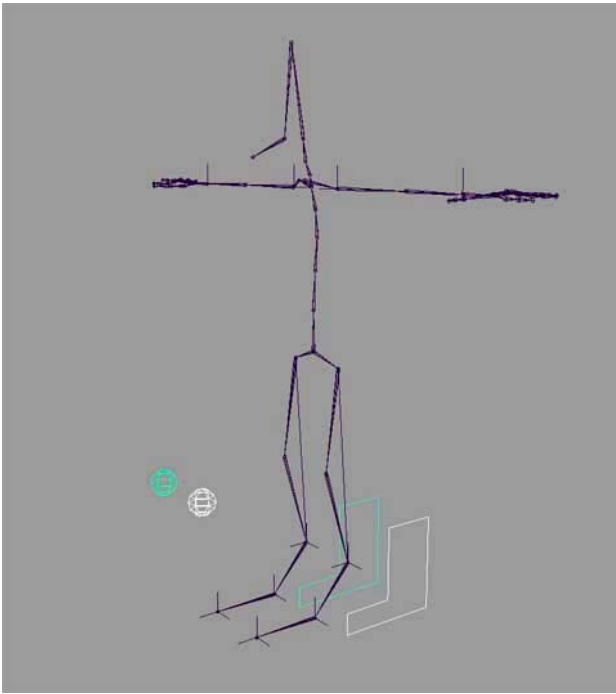
Now this is another part that is up to how you want to animate... What I do next is parent the "l_knee_CTRL" to the "l_foot_CTRL".. and the "r_knee_CTRL" to the "r_foot_CTRL".

To do this group the "l_knee_CTRL" to itself and name the group "l_kneeCTRL_group" now select the group and shift-select the "l_foot_CTRL" and press "p".

Now do the same for the right side.

The reason for the group is to make a clean transform node on the knee control.

now after you have parented them or not parented them select the two spheres and modify -> freeze transforms (make sure all of the boxes are checked in the options) again now to get back to the bind pose just set all of these attributes to 0,0,0

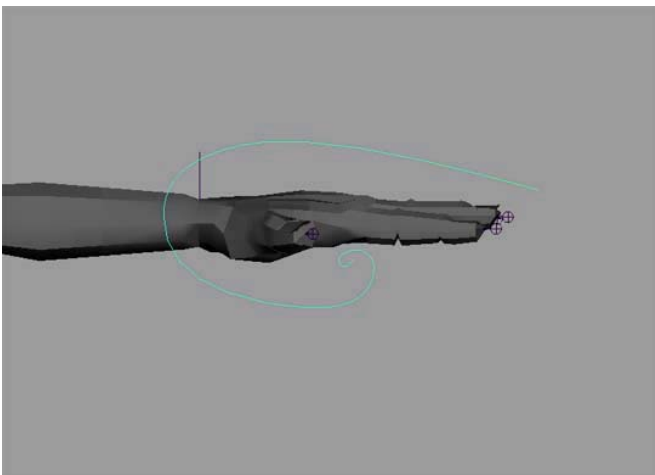


There you go.. the feet are done for the skeleton.. ☺

Arm Controls

Ok.. Switch to front view and show your hands and draw a neat curve to be your hand controller.. then name it "l_hand_CTRL"

(have you been saving??)



Switch to side view and line the hand control up with the ik handle...(doesn't have to be exact) Switch to perspective view and select the hand controller... Select the move tool and press **insert**.

snap the pivot to the wrist joint.

If you are lucky enough to have the jsConstObj script from Alias|Wavefront's Creature Rig DVD then you can simply select the hand controller and shift select the wrist joint and type:

"jsConstObj" in the command line window.

Now go to the outliner and select the wrist joint and delete the "point Constraint" created by the script.

Now select the hand control and shift select the "l_wrist_IK" and constraints -> point constraint.

If you don't have the jsConstObj script do the following:

Now select the hand controller and shift select the "l_wrist_IK" and constraints -> point constraint

Now select the hand controller and shift select the wrist joint and constraints -> orient constraint

if all is well and your wrist joint is flat like it should be if you oriented your joints correctly the wrist joint shouldn't have moved at all... (if it did move see below)

now select the controller and freeze transformations on it... again to make it easy to get back to the bind pose.

My Wrist Joint Moved!

If your wrist joint moves, undo the orient constraint.

select the wrist joint then shift select the controller and constraints -> orient

now go to the outliner and delete the newly made constraint.

now go to vertex mode or CP mode on the curve and select all of the points

and rotate them until they are vertical again... once this is done exit component mode

and follow the instructions above...

do the same for the right side...

Now to setup the clav controllers..

create a new poly cube..

go into the attributes and turn off shading and texturing and all of the render settings...

name the cube "l_clav_CTRL"

snap the Cube to the shoulder joint

scale the cube to a nice size (easy to grab but not too big)

Select the cube, shift select the "l_clav_IK" and constrain -> point.

select the "l_clav_CTRL" and group it to itself.

name the group "l_clavCTRL_group"

Select the "l_clavCTRL_group" and shift select the "u_u_spine" joint and parent them together...

freeze transformations on the cube

Cool... we are almost done with the clav_CTRL... All that is left is to give a polevector constraint to the clav_IK... If you don't, when you turn your character around the clav_Iks will flip over and twist your shoulder all up... and that is just no good... sooo.... do this:

We are going to make a locator and move it to the shoulder and parent it to the clav_CTRL and then use that as the pole vector constraint for the clav_IK

create -> locator

name the locator "l_clav_LOC"

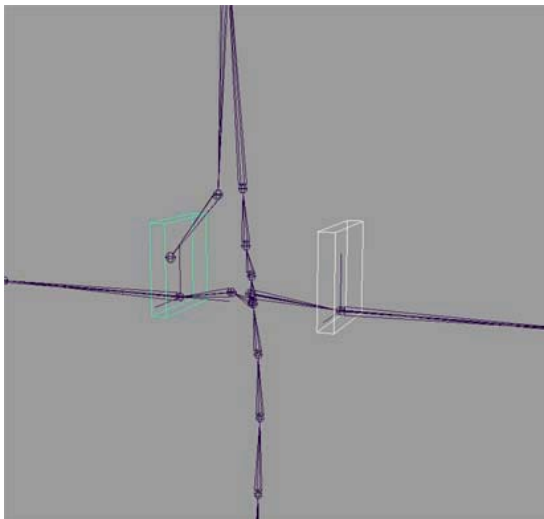
hold down "v" and point snap the locator to the shoulder joint...

select "l_clav_LOC" and shift select "l_clav_IK"

constrain -> pole vector

now select "l_clav_LOC" and shift select "l_clav_CTRL" and press "p"

do the same for the right side.



Now the elbows... These are exactly the same as the knees...

Make a new poly cube.. turn off the display stuff and the render stuff...

move it up to the elbow joint.. select the cube shift select the "l_wrist_IK" constraints -> pole vector

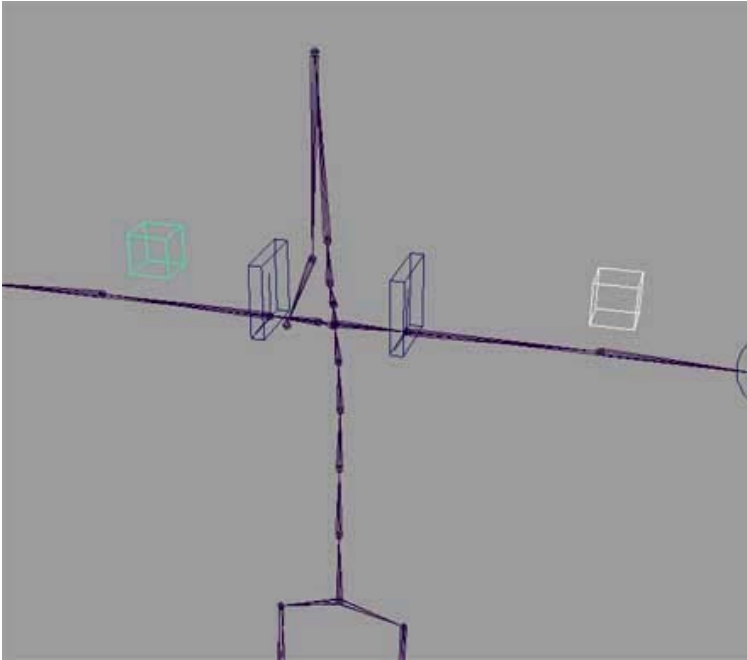
name the cube "l_elbow_CTRL"

group the "l_elbow_CTRL" to itself and name the group "l_elbowCTRL_group"

parent the "l_elbowCTRL_group" to the "u_u_spine" joint (if you want to)

Freeze transform the cube...

do the same for the right side..

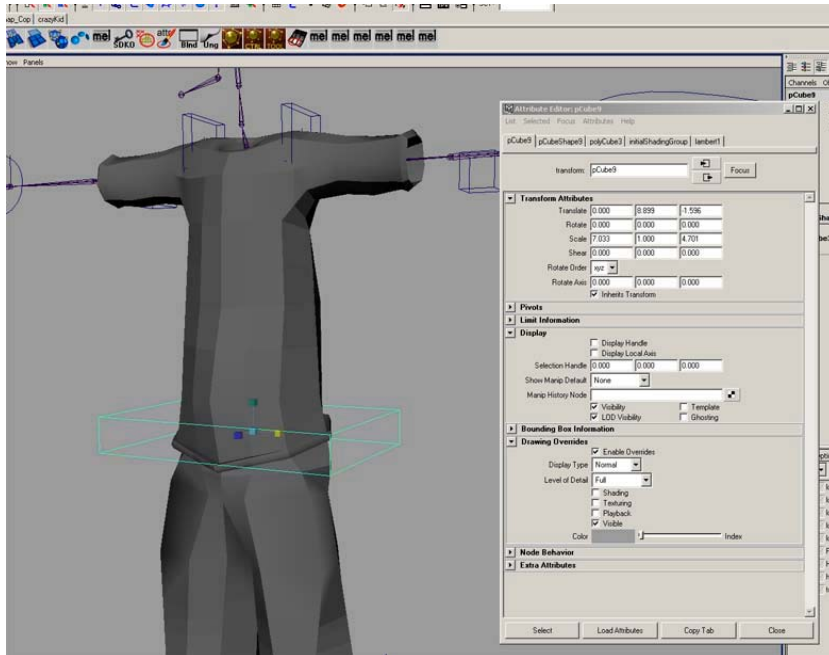


HIP Controls

OK.. Half way there...

The first thing we want to do is create a control for the hips (root)

Its going to work just like the Clav Controls... Make a poly cube and scale it to a decent shape to go around the hips...



Don't forget to go into the attribute editor and turn off all of the renderable features and to enable drawing overrides and turn of shading and texturing...

Rename the ctrl to "hip_CTRL"

Now freeze transforms on it so that everything is set to 0 and scale is set to 1

If you have jsConstObj:

select the "hip_CTRL" and shift select the root joint and type:
"jsConstObj"

If you don't have jsConstObj:

Select the "hip_CTRL" and shift select the root joint...

Now constrain -> point constraint

Now constrain -> orient constraint

Deselect all and then select your new "hip_CTRL"... move it around a bit and rotate it to make sure that all is well... when you are done playing (take your time.. this is fun remember?) just zero out the translations and the rotations... this should get you back to its original positions...

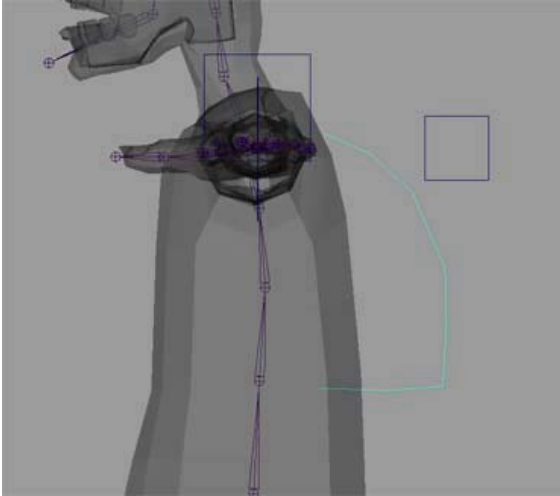
In case you were wondering why you don't want to just parent the root to the "hip_CTRL" is because that puts the entire skeleton under the "hip_CTRL"... now your layers can get all screwed up and things will simply not behave in a peachy fashion...

BTW: Did you save? a new version even? If you did cool.. if you didn't... for shame on you! ☺

Cool.. you got some hips.... Now for a spine ☺

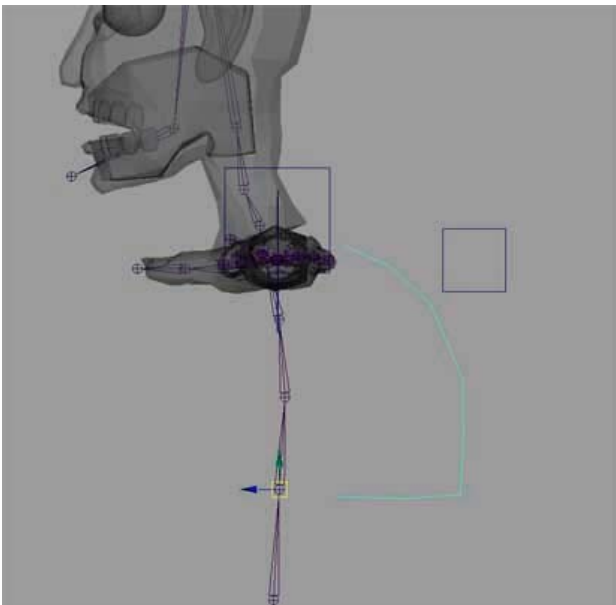
SPINE Controls

The first step is to create a curve to be used as the control itself.. In side view draw a curve to be your upper back control... I prefer simple shapes that can be recognized quickly... however it is your rig... so make it whatever you want... ☺



Now you want to move the curve in line with the spine from the front of the character and then snap the pivot of the curve to the "low_upperSpine" joint.

This is going to be the base rotation spot for all of the spine joints... Also rename your curve to "uSpine_CTRL"...



Now select the "uSpine_CTRL" and group it to itself. Name the group "uSpineCTRL_group"

select the group and shift select the root joint.. now parent them together...

Deselect all and reselect the "uSpine_CTRL" and freeze transform on it...

Nice... now we have a control to work with...

Spine Rig Method 1: FK

Open you new best friend the **connection editor** (window -> general editors -> connection editor)

Incase you don't know this is how the connection editor works... stuff on the left side when connected sends its data to stuff on the right side... so for instance say we load a cube into the left side... The list consists of attributes for that object... to load things into each side... select the object and press **Reload left** or **Reload Right** this loads the object... once there are objects loaded into each side (this can be the same object if you like) you can start connecting things...

Ok.. so say for example we loaded a second cube into the right side (assuming the button at the top is set to "from -> to") and say for some reason we want to make the cube2 rotate the same as cube1... Granted this is not the only and probably not the best way to achieve this but it gets the point across... on each side in the connection editor drop down the rotate section's triangle... this will reveal Rotate X, Y, and Z... Select Rotate X on the left side and then select Rotate X on the right side... now as if by magic, cube2 will rotate on its X-Axis at the same time that Cube1 rotates on its X-Axis... kinda cool huh? So this long winded explanation for something you probably all ready knew will stop now and we will just set up the spine ☺

If you have jsConstObj:

Select the "uSpine_CTRL" and shift select the "low_UpperSpine" and type: "jsConstObj" in the command line window.

now select the "low_UpperSpine" joint and in the connection editor press the **Reload Left** button.

now select the next joint up (mid_upperSpine) and press the **Reload Right** button.

now on the left side select "rotate" and then on the right side select "rotate"

now select the "mid_upperSpine" joint and press **Reload Left**.

select the next joint up "top_upperSpine" and press **Reload Right**.

select "rotate" in the left pane and "rotate" in the right pane.

Now the three joints will all turn the same amount creating a curling action.

If you don't have jsConstObj:

Open the connection editor if you have not... select "uSpine_CTRL" and press the **Reload Left** button to load the controller into the left pane...

Select the "low_UpperSpine" joint and press **Reload Right** to load the joint into the right pane...

So we now want the rotation data from the controller to adjust the rotation data of the joint itself...

so select the Rotate attribute of the controller and the Rotate attribute of the joint (in the connection editor)...

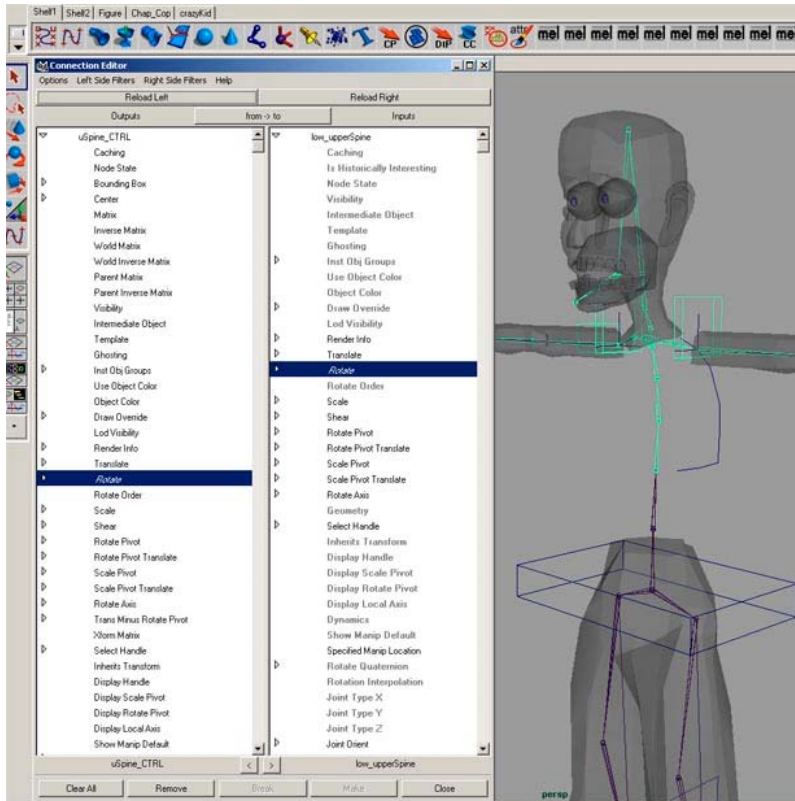
Now the spine joint will rotate with the controller...

Cool.. so now lets connect the other two upper spine joints...

Select the next joint up (mid_upperSpine) and press the **Reload Right**

Now make sure that Rotate is selected in the left pane and then select Rotate in the right pane..

Do the same for the "top_upperSpine"...



Now rotate the Controller... the "upperSpine" will now curl in the direction that you rotate the controller... yeah! ☺ the spine is rigged... simple enough huh?

This particular spine setup will do a good job for most “normal” bipeds... It does some what limit your exact control over the spine, but I think it works well anyway... Like I said before there are lots of ways to rig a spine.. ranging from simple to very complex so play around and see what you like!

Spine Rig Method 2: Spline IK

(Insert images for Spline IK setup)

The main advantage to using Spline Ik over FK for the spine is that you can rotate the hips and the shoulders independently of each other. This means that when animating your character you don't have to counter animate the shoulders every time the characters rotates his hips. The only down fall that I can see is that it is a little bit less precise control over the exact shape of the spine. However it seems to work great for me. Try this and let me know what you think about it!

Don't forget to save as much as you can!

Select splineIK handle tool options...

Set curve to autoSimplify=2

Turn off “AutoParent”

Leave the rest of the options to the default options

Select the one joint above the root in the spine and then select the “clavCenter”

This will make the splineIK handle from the lower back joint to the “clavCenter” Joint

name the curve that is generated from the splineIK tool “backCurve”

name the ikhandle “back_ikHandle”

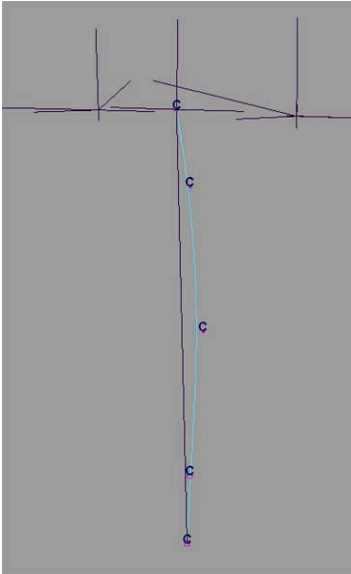
Now make relative clusters out of each CP on the curve.

Select a CP then Deform -> Create Cluster -> options

Check the “Relative” box

Press the “Create” button

Do this with each of the 5 CPs...



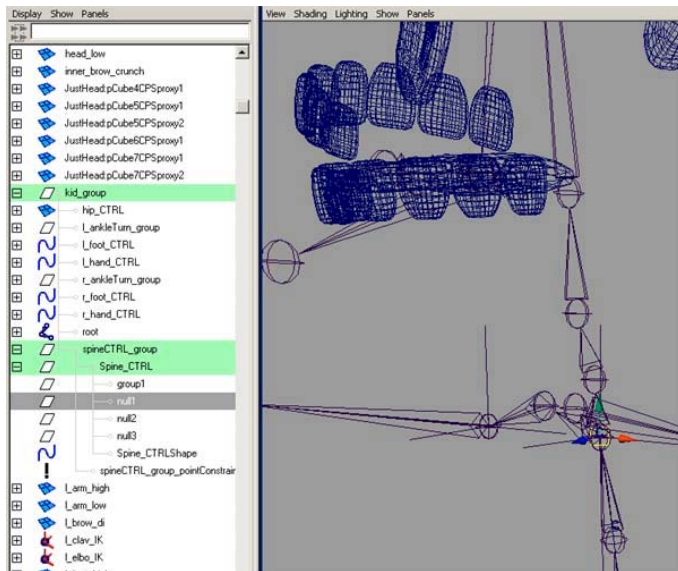
Now deselect everything and Edit->group...

I know, I know... it sounds stupid making a group with nothing in it... however it helps us out quite a bit as you will see...

(if you accidentally deselect the empty group, simply re-select it in the outliner or press undo)

Take the empty group and point snap the group to the cluster...

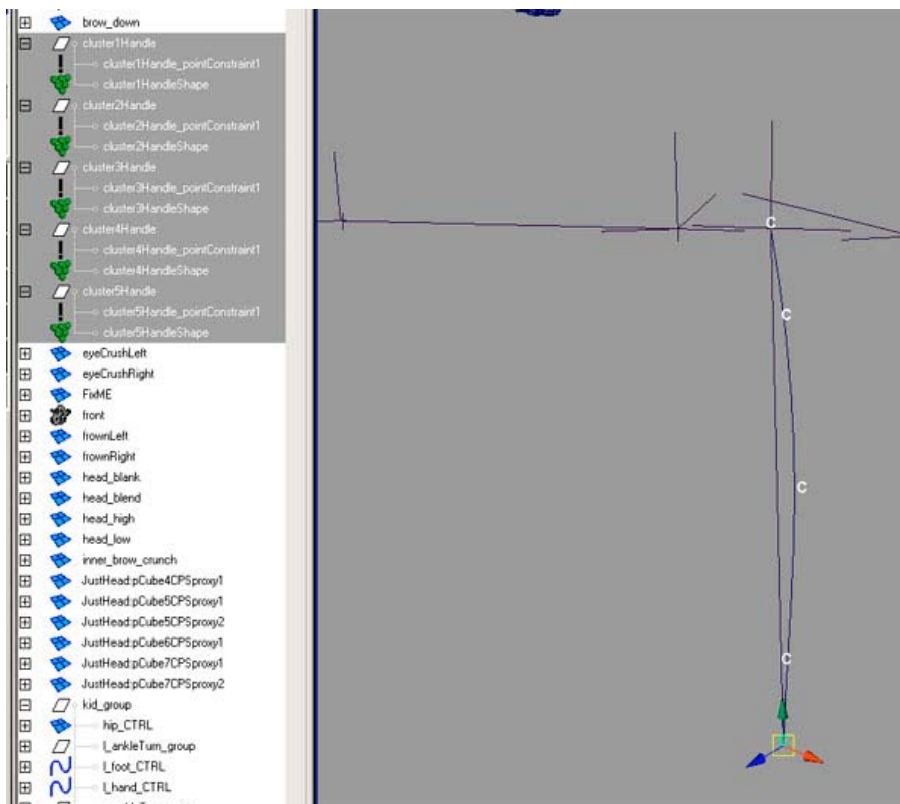
do this for each cluster... name each empty group NULL# the # should correspond with the number of the cluster in the splineIK for the back...



select the “NULL1” in the outliner and CTRL-Select “cluster1Handle”

now Constrain -> point

This will constrain each of the clusters to the empty group... this will allow us to keep the clusters with the character without getting double deformations... The clusters themselves are now outside of the main hierarchy and are relatively moved by the NULL groups...



Now parent the proper NULL groups to the proper controller.

parent the top 3 null groups to the spine control

parent the bottom 2 null groups to the hip control..

Now we need to make it so that the spineCTRL will move along with the hip CTRL, but not rotate with the hip controller.

group the spine controller to itself and name the group spineCTRL_group

snap the pivot of the spineCTRL_group to the root joint...

point constrain the spineCTRL_group to the hipCTRL...

Now we need to handle the twist of the back... to do this we are going to calculate the difference in Y rotation between the hips and the shoulders and then change the twist attribute to evenly twist the spine from the shoulders to the hips.

select the back_ikHandle and make this expression on it:

```
back_ikHandle.twist = -hip_CTRL.rotateY + uSpine_CTRL.rotateY
```

Congratulations! You now have a very flexible spine setup. Have fun with this one!

NECK Controls

Setting up the neck is just like setting up the spine, but instead of using 1 controller for 3 joints we are going to use 2... One for the bottom half of the neck and one for the top of the neck...

Go ahead and draw a curve to represent the bottom of the neck...

Now move the pivot to the "neckBase" joint and snap it there...

Load the controller into the left pane of the connection editor and then load the "neckBase" joint into the right side....

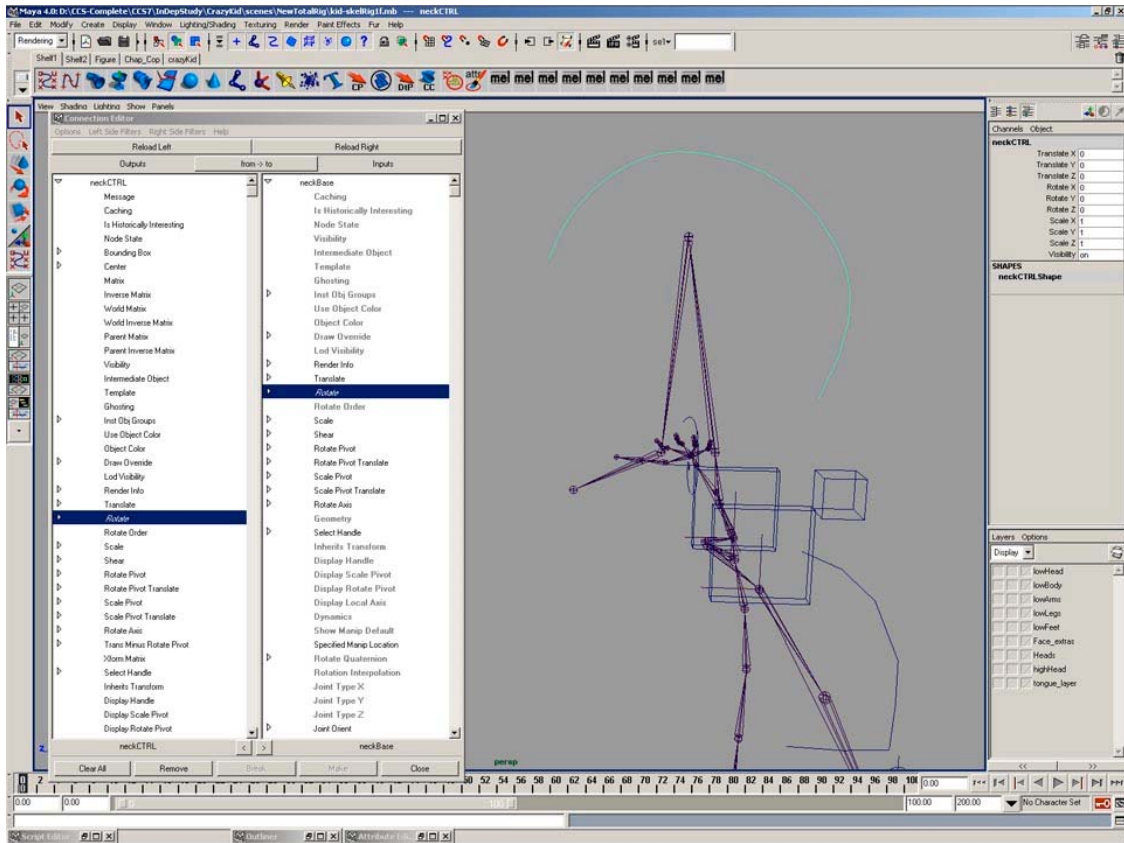
Now select **Rotate** on both sides...

Load the "neckMid" joint into the right side and select **Rotate**...

Now the bottom of the neck should be rigged...

group "neckCTRL" to itself and parent the group to the "clavCenter" joint.

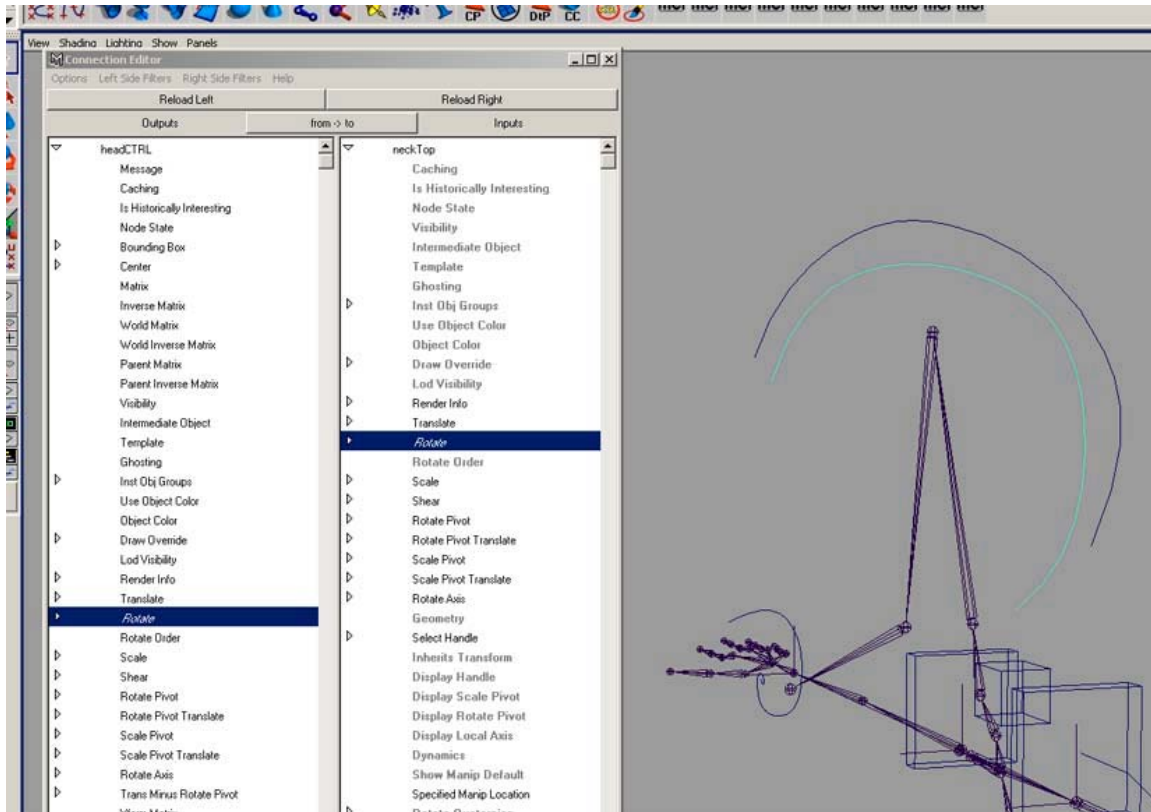
Now freeze transform on the "neckCTRL"...



Now to do the top of the neck (the head swivel section) draw another curve...

Move the pivot to the "neckTop" joint...

Now in the connection editor connect the rotate for the controller to the "neckTop" rotation...



Group the controller to itself and parent the group to the “neckMid” joint.
freeze transform on the controller...

Now the neck should be set up... play with it... and don't forget to save!! ☺

Head Setup

(WRITE EYE MODELLING SETUP AND PUPIL SETUP)

SMOOTH BIND SETUP

Connect Poly Shape

For this particular rig... the heart and soul of the efficiency is the ability to switch between high poly and low poly and the ability to add blend shapes at any time... Again to achieve most of this we will be using the connection editor to use the mesh data from one object, send it through a smooth node and then feed the newly smoothed mesh data into a new mesh... this way you can bind the low poly mesh which is much easier to weight and have the low mesh drive the detailed high poly mesh... I would like to thank Dirk Bialluch for bringing the core of this technique to light through his CPS script and his great tutorial that goes with it...

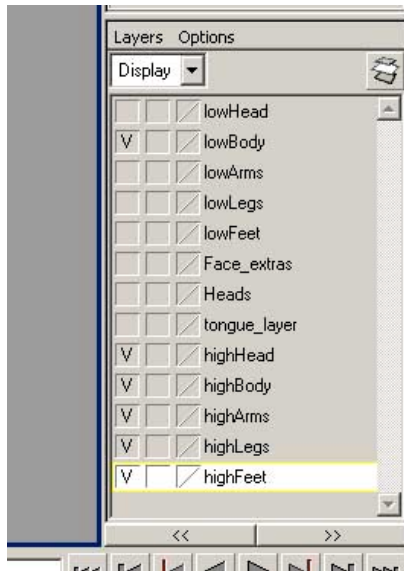
Note:

This is a good time to bring up texturing and UV mapping... I am about to explain two slightly different methods to handle the smooth binding of a character. The problem here is if you do the body like I am about to show you then you need to make sure that you have your UV unwrapping finished before you bind your character. The UV mapping needs to happen before the mesh is deformed or you will get very strange texture problems. However if you do the body (and arms and all other pieces) with the same method as the head you can do your UV mapping at any time during the

rigging process. All you have to do is unwrap the Uvs for the blendBody or blendHead (or whatever piece you are working on) and not the lowBody or lowHead objects. What happens is when the mesh data is piped from the blend object to the low object the UV coordinates will also get transferred. So I would suggest doing all of the pieces of your character just like the head is done, that way you have much more flexibility to make changes after things are bound.

The first thing that I think everyone should do is download the CPS script from Highend3d.com or wherever you can find it and run through the very first part of the tutorial... (the T1 section) This will help you get a better handle on what this whole ConnectPolyShape thing is about...

To get started with this we are going to need to make a few new layers for all of the high-res versions of the objects...



Now Open up the outliner... right Click somewhere in the right and turn off **show DAG objects** only and make sure that **show Shapes** is checked

create a poly cube and name it "torso_high"...

This next part is a bit of MEL... (oh quit whining... its good for you ☺)
Type the following into the Command Line (the white box under the time slider... also as a note MEL is case sensitive, so watch your caps..)

createNode polySmoothFace -n "torsoSmoother"

after you have that typed in press **enter**... the command will be executed and a smooth node will be created... (feel proud yet?)

In the outliner your newly created smooth node entitled **"torsoSmoother"** will be at the bottom of the list...

Ok here is what happened.... scripting languages are languages that are basically a bunch of commands that can usually be used at the command line one at a time... this allows for a bunch

of commands to be executed in a row just by running the script... Granted the scripting languages allow for runtime decisions to be made and to add logic to commands also but for this example the script is simple. This little bit of MEL works like this:

the “createNode” part is a command... this command takes certain options called arguments... if you look in Maya’s Docs you can find all of the argument options for every MEL command that exists... The options used here are the node type and the new node name... createNode is run and it is given the node type which is in this case the “polySmoothFace” node.. the “-n” says name the node what is in quotes after the ‘-n’... so the resulting node in this case will be called “torsoSmoother”

Congrats you have just made a new node ☺

This new node needs some data to be fed into it so it can smooth it and pass the data back out through its outputs...

Open the connection editor...

Open the outliner...

In the outliner find "torso_low"...

hit the plus next to it and select the “torso_lowShape”...

In the connection editor press **reload left**...

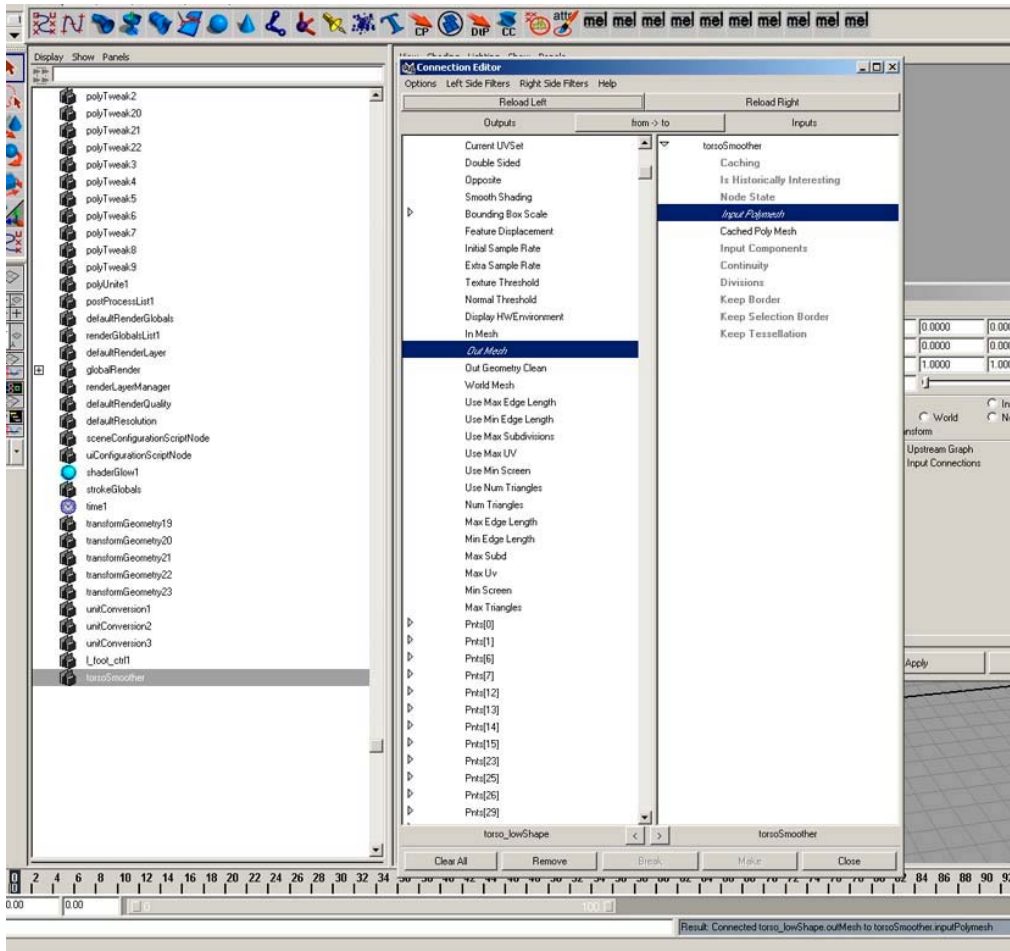
Now the shape node for the "torso_low" object is loaded...

Now in the outliner find the "torsoSmoother" node...

Press **reload right**...

In the left pane select “**outMesh**” and in the right pane select “**inputPolyMesh**”...

(if you don’t see these attributes make sure that “show hidden” is checked in the "rightSideFilters" menu of the connection editor... and also make sure that you have the shape node selected for the torso and not the whole torso object...



now load the "torsoSmoother" into the left pane and the "torso_highShape" into the right pane..

select "Output" in the left pane and "In Mesh" in the right pane....

this takes the output geometry from the smoothNode and pipes it into the mesh of the duplicate...

Now all we have to do is tell the smooth node which component to smooth...

In the command line type the following:

```
setAttr torsoSmoother.inputComponents -type componentList 1 "f[*]"
```

this sets the "inputComponents" attribute of the smooth node to faces and says to smooth all (the*) of the faces...

now you should have a high and low version sitting on top of each other... put the high poly version on the "torso_High" layer...

Do the same for the other parts of the character except for the head... Actually you may want to hold off on doing the rest of the body, because you may like the way the head is done and use that method for the entire character... (see previous "note")

NOTE: As a note there is a method for fixing geometry when the joints bend by using blend

shapes driven by joint rotation on the geometry... so for an elbow a blend shape can be used to pull the parts of the arm out that collapse from the weighting... simply drive the blend shape slider with the rotation of the elbow joint... (use driven keys)... if you are going to use this method instead of influence objects then read the section about the head before doing the rest of the character....

Pretty sweet huh? Now whatever you do to the lowPoly version will happen to the highPoly version... so that means that whatever happens to the vertices on the lowRes version will move the vertices on the highRes version. So what we are going to do is bind the lowRes version to the skeleton. This will allow a very quick response when animating and will severely cut down on the number of vertices that need to be weighted...

Next we are going to do the same thing for the head... but we are going to do a slightly modified version of what we did for the body... However the principal is still exactly the same...

The difference is that we want to be able to add blend shape targets (morph targets) to the head at any time during production and also do the UV unwrapping at any time during production... If you have ever tried to add a blend shape to a piece of geometry after it has been bound you will find that Maya will not be able to do it... It complains that a blend shape node cannot be found... this is because that the blendNode is below the smoothBind nodes in the history... therefore the target cannot be added because Maya can't find the node to add the target to...

Soooooooo.... what this means is that we are going to need 3 different heads...

1 head to add the blend shapes to... 1 head to bind to the Skeleton and 1 head to be the final high poly version...

Basically what this means is that we do the process twice... however the link from the blend shape head to the lowRes head will not have a polySmoothFace node between them...

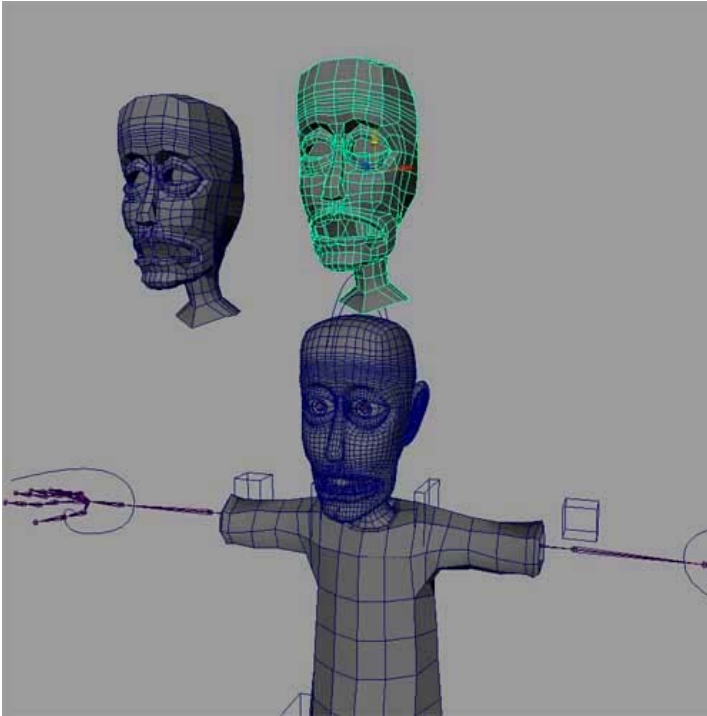
Ok enough blabber (seriously though... it is good to really understand what you are doing... so when you run into problems you can fix them... or come up with your own methods) lets do it!

NOTE: A quick note about making morph targets... always make a blank one... that is an exact untouched copy of the original head... and just sit it up by the rest of the targets... This blank head is good if something bad happens to your blendHead and it shifts on you or other bad things happen... just trust me.. it will come in handy later.. ☺

Ok.. Make a duplicate of the "head_low" object... move it to the left, off of the original head and add it to the same layer that your morph targets are on... (you do have them on a separate layer don't you?)

Name this head "blend_head"... This is going to be used as the head that has the blendShape node on it... It is going to drive the "low_head" object... The "blend_head" will also be the object that you would want to do your UV unwrapping on.

Now select the "head_low" object again and one more duplicate it... this time don't move the duplicate at all... Name this new head "high_head". This is going to be the final head or the result of all of the deformations (blending and binding)....



I moved the head just to show you the three heads... remember the only head you are going to move is the blend head... leave the head that is going to be the highPoly Head where it is...

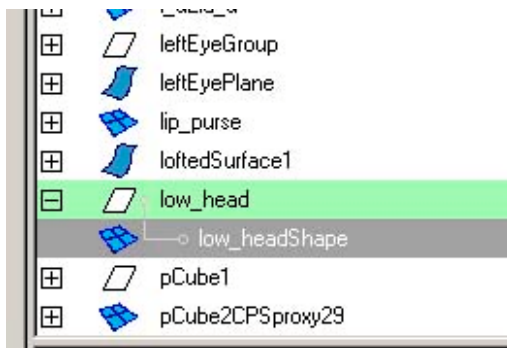
Now open the outliner and the connection editor...

In the outliner make sure that “**show DAG objects only**” is not checked and that “**Show Shapes**” is checked...

Now find the “**blendHead**” object and click the plus next to it.. this will allow you to select the “**blendHeadShape**” node...

Select it and press **Reload left** in the connection editor...

Now find the “head_low” object and press the plus sign and select the “head_lowShape” and press **Reload Right** in the connection editor...



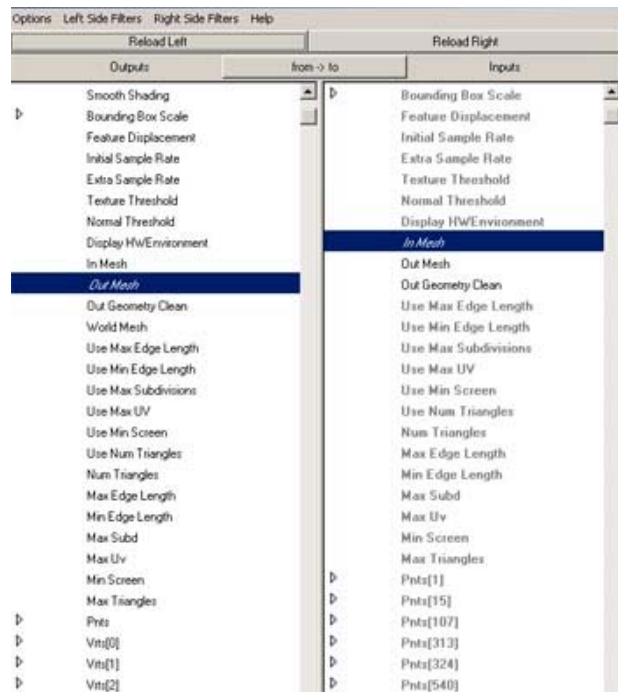
In the left side find “**Out Mesh**”... there will be a lot of things to look through... mine usually end up being between the long lists of stuff Like edges and

vertices...Once you find it select it...

In the right side of the connection editor find **"In Mesh"** and select that...

Remember if you can't see them... make sure that **"Show Hidden"** is checked in the Left and Right side menus...

Now the two meshes are connected.. whatever happens the vertices of the "blendHead" will now change the vertices of the "low_head"



Now we need to drive the "high_head" with the "low_head".
This part works just like the body does.

First make a new smooth node:
createNode polySmoothFace -n "headSmoother"

Open the connection editor...

Open the outliner...

In the outliner find "low_head"...

hit the plus next to it and select the "low_headShape"...

In the connection editor press **reload left**...

Now the shape node for the "low_head" object is loaded...

Now in the outliner find the "headSmoother" node...

Press **reload right**...

In the left pane select **“outMesh”** and in the right pane select **“inputPolyMesh”**...

(if you don't see these attributes make sure that “show hidden” is checked in the "rightSideFilters" menu of the connection editor... and also make sure that you have the shape node selected for the torso and not the whole torso object...

now load the "headSmoother" into the left pane and the "high_headShape" into the right pane..

select **“Output”** in the left pane and **“In Mesh”** in the right pane....

this takes the output geometry from the smoothNode and pipes it into the mesh of the duplicate...

Now all we have to do is tell the smooth node which component to smooth...

In the command line type the following:

```
setAttr headSmoother.inputComponents -type componentList 1 "f[*]"
```

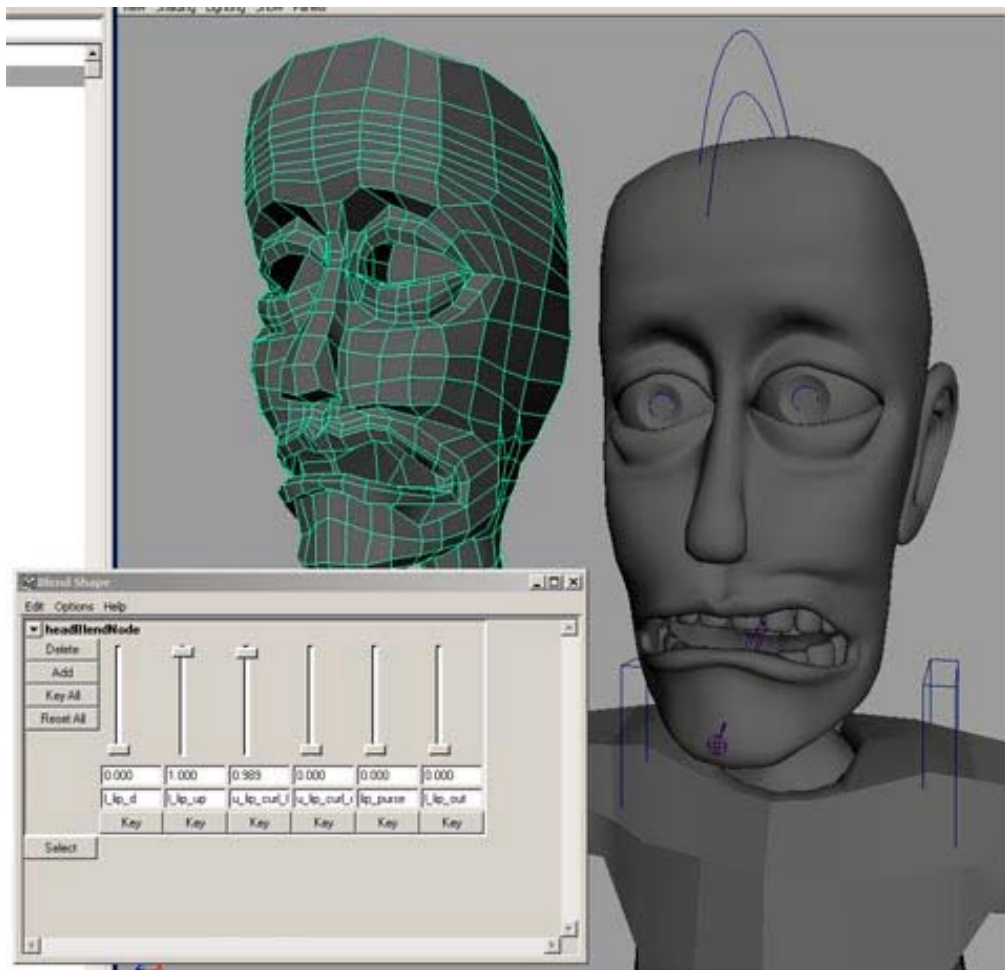
Cool... so lets try what we have been working towards...

Make some morph targets if you haven't already (duplicate the “blank_head” to make them)

Now select a morphTarget and shift select the “blend_head” and

Deform -> Create BlendShape

Cool.. now test out the blending... turn on the “high_head” layer and turn off the “low_head” layer... Now move the blendShape sliders around... the high head should deform properly... If not, undo then blendShape deform and retrace your steps... good luck!



Ok.. now you know the two methods that I use the most... so why don't you plan out what you want to do and then decide which method is right for your character, and finish up making the highPoly versions of the model...



Smooth Bind

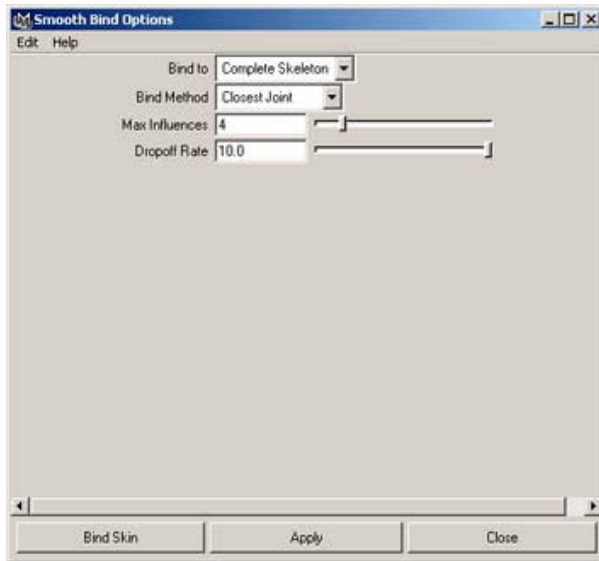
Sweet... we are almost done... We still have to bind the lowPoly versions of the mesh, weight that, and possibly add influence objects.. so lets continue
(Have you been saving? Different versions?)

Ok... Turn off all of the high version layers and turn on all of the lowRes layers..

Select the root joint and shift select each of the different parts of the mesh...

Now Skin->Bind Skin -> Smooth Bind -> options

Set the bind options as in the following picture...



The "Max influences" and "Drop Off Rate" are very important...

I have mine set to 4 and 10 respectively...

Max influences is the number of joints that can influence any particular vertex at one time... I like to keep that at 3 or 4... this makes the weighting process easier because that are not so many joints effecting each vertex...

Drop off rate is how fast the effect of each joint drops off... I like to set that at 10 because each joint will again effect less vertices... This really helps in the fingers and the shoulders...

Remember these options only effect the initial bind... what you do to the weights afterward has no bearing on these numbers...

Ok.. now press "Bind Skin"...

Cool.. the next thing is kinda fun... move your controls around and see what works and what breaks... Move the arms around paying close attention to the shoulders and the armpit... also check all of the major joints like the knees and the elbows... What you will probably notice is that "volume" is lost when bending most of the joints... we will weight each joint so that it is as good as weighting can get it... then we can add influence objects that will correct for volume loss... all very cool stuff ☺

As you can see... I have some immediate problems... the pants are sticking together... What I would suggest doing is start at the extremities and work your way in to the torso... this way you can lock the weights on each joint as you work and the weighting won't get screwed up...

Ok.. I'm going to start at the feet... I do all of the joints with the same method as I do for the legs and the feet... so I am only going to explain how to weight the feet and legs.. but I am going to give hints/tips for the rest of the body...

Select the foot geometry and Skin->Bind Skin -> Edit Smooth Skin -> Paint Skin Weights Tool -> options...

This will turn the geometry black and white... how white an area is, is how much effect the selected joint has on that vertex... So 100% white means that the vertex will rotate 100% of its joint's rotational arc... that means that that vertex will rotate the exact same amount as the joint...

So find the "l_ball" joint in the list of joints in the Paint Weights window and select it... (aren't you glad you named your joints?)

This is a good time to bring up a few tips...

1) USE YOUR SHELVES!!! ☺

Shelves are the greatest thing ever... you can make your own or use the default ones... in Maya 4.0 the shelves are at the top of the screen.. these are basically places to hold Icons for commands that you use most often... for instance I put the "Split Polygon Tool" and "Center Pivot" tools up there so I can just click that button instead of looking through menus...

To add a button for a tool that already exists simply use the menus at the top and hold down CTRL+SHIFT when you release the mouse button on the tool you want... this will add that tool to the shelf you have selected... To edit the shelves LMB click the down triangle all the way to the left of the shelves... Play with these... once you get to use them you will never want to be without them...

2)Meshes can be weighted in deformed positions... what that means is that you can bend your leg mesh and paint on it while it is bent... this is a great time saver and takes out much of the guess work when painting...

3)Hold RMB over a joint while painting and select "Paint Weights" and that joint will automatically be selected for you...

4)Turn on "Wireframe on Shaded" in the Shading menu for the view port... it seems easier to edit when you can see the vertices and edges...

Ok with that said... lets paint some weights... the two options you really need to worry about while painting weights are the Opacity option and the Value option.. These work just like the airbrush in PhotoShop... Play with these settings and try to get a feel for how you like to paint... also play with the Profile..(what brush you are painting with) This will allow for softer edges while painting or nice sharp edges... Also remember that when painting you are only really effecting vertices... so don't waste your time trying to paint a face... just paint over the vertices...

Two more tips:

While painting hold down the "b" key and MMB drag to change the size of you brush... also hold down "n" and MMB drag to change the value..

I was told once by a guest speaker at my school to only use Replace and Smooth... This does seem to work well but if you lock the joints you have finished working with, you should have not trouble using the other two...

To lock weights that you have finished... select the joints that you are done with in the Paint Skin Weights window and press "Toggle Hold Weights On Selected" Once you do that, that joints weights will not be effected...

I'm sorry this is so general but you really do just have to play with it and see how you like to work... Try all of the options in the window! Good luck!

Try not to worry too much about the weighting... just get it as good as you can and fix the rest with blend shapes (just like the head) or influence objects...

Just to let you know... I normally reference my characters and sometimes props into the scene instead of importing them directly... this allows for much more flexibility later on down the line... if you reference your characters in then you can edit the weights on the source file and the new weights will be updated in all of the files that reference that character... You can also change textures and things in the source file... there are limitations to referencing... when it works right it's great... when it doesn't, it really really doesn't... so test any different type of change before you commit it to the real source file... Grouping and parenting things after you have already referenced a file may break things... just think about what you are doing... think about how it will affect files that have already been animated... for instance if you delete a control that was animated in another scene those keys would be lost and that part of the character will not longer move like you had it... Trust me... test things first... it takes longer up front, but it will save you much time in the future... Just a suggestion...

If someone out there wants to write a really good and really detailed tutorial about skin weighting or UV unwrapping please email me and I'll include it in the next revision of this doc... Thanks!

Face Extras

Ok.. there are two eye setups that I use...one setup allows the eyes to follow an object and be directly controlled if you want... For me it is easier to just control the eyes directly because most of the time it is how the eyes look on camera not if they are really looking at something.... So for now I am just going to show you the simple one where you control the eyes directly...

Select your eyeball and group it to itself (just select the eye and press 'g')

Name the group "leftEye_group"

Now with the new group selected group the group to itself...

I know this sounds crazy but it works well... essentially when you group something to itself you are basically making a new transform node for it... so you rotate the eye ball, while moving the group of the eye according to the head... this way you have two clean ways of controlling the eyes...

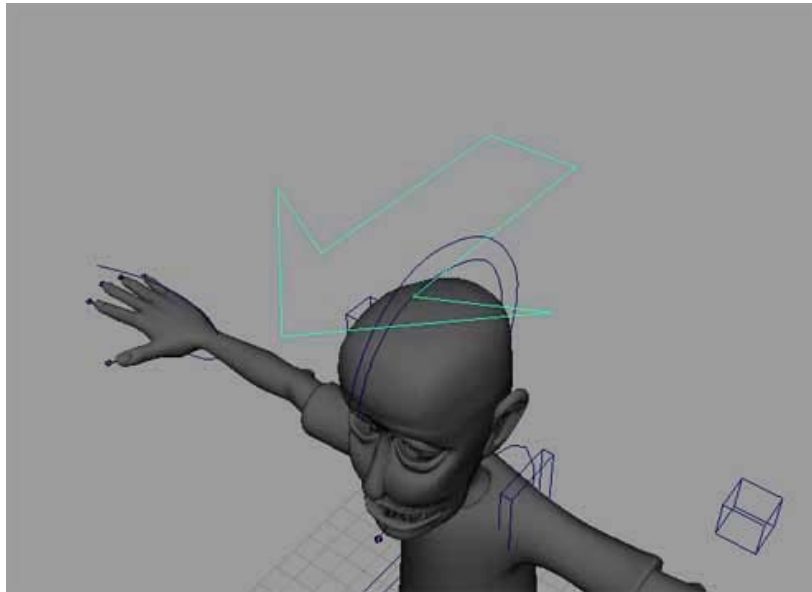
Now name the new group "leftEye_parent_group"

with this second group selected shift select the skull joint and parent them together...

Now the eye will move with the head while leaving a clean transform node at the first group...

Do the same for the right eye... (change the names of course)

Now above the head use the curve tools to draw a controller for the eyes.. I like to use the arrow... this way the eyes will point to where the arrow is pointing...



With the controller built, center the pivot (Modify -> Center Pivot)

Select the Controller and Parent it to the skull joint...

Select the Controller and Edit -> Freeze Transforms

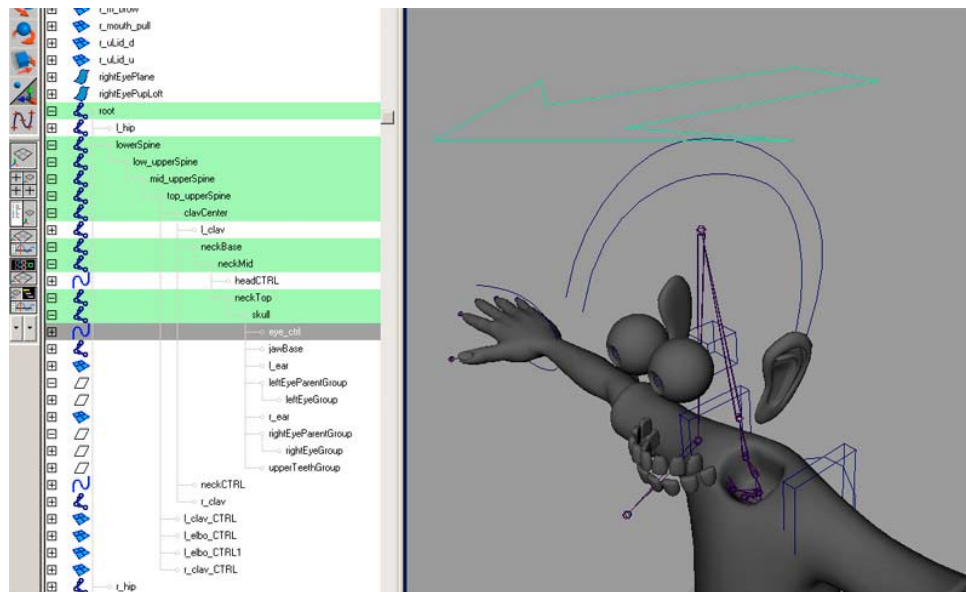
Now the arrow should move with the head...

Select the eye Controller

Now CTRL select the "leftEye_group" in the outliner...

Now Constrain -> Orient

Now your left eye should point with the arrow...



Now you have some working eyes... There are lots of cool thing you can do with eyes but I am going to save that for a different tutorial that will be added to this one...But this should get you started...

CleanUp

After this section you could very well be done with rigging your character... the only thing to do after this is influence objects if you want to...But this is basically a complete rig that should be a joy to animate... for me it is... and for me that is the point. I want to just have fun breathing life into small energized bits of phosphor... Yeah! ☺

Ok.. this section is really important and once it is done you will be able to animate much faster and have fewer things that are keyed... so less little slashes on the time line and less little dots in the graph editor ☺

Basically it works like this... you want to hide and make non-Keyable everything that you don't need to touch... you also should make all attributes of the controllers that do not need to be keyed...(such as scale or visibility)

We are also going to color code the controllers so that it is easier to see which is left, center, and right...

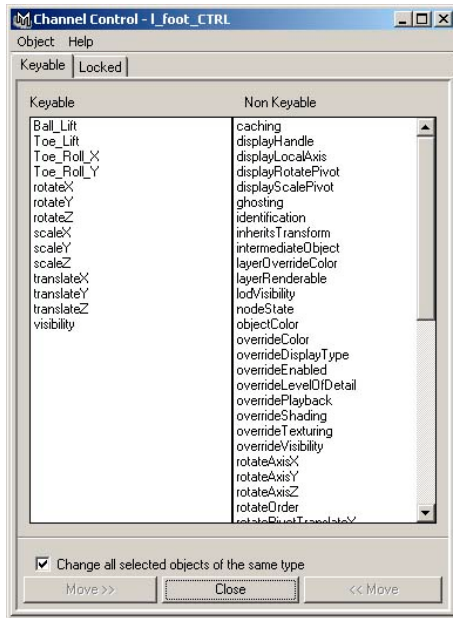
First thing... in the viewPort turn off IK handles and Joints... you shouldn't need them from now on...

NOTE:Before I didn't mention that you should make an attribute on the Main Head control that controls the opening and closing of the Jaw.. just use Driven Keys to rotate the "jawBase" joint...

Now time to turn off the attributes we don't need

Select one of the foot controllers...

now Window -> General Editors -> Channel Control..



This will pop up the channel control window... this is where you can hide or make attributes “non-Keyable”... this takes those attributes out of the channel box when you select the object..

Select scaleX, scaleY, and scaleZ in the left pane and press the move button...

now those attributes are gone from the channelBox... cool...

Basically do that for all of the attributes you don’t want...

For instance on the Clavical controllers you don’t need rotate or scale and for the eye control you don’t need translate or scale... Remove as much as you can... this will help you speed up your workflow and give you less to keep track of...

You could also probably get rid of the visibility attribute also... if you are trying to make your character pick up something... please don’t try to use the Visibility Switch technique... This is just messy... use constraints it will make your life much easier... and constraints are cool anyway ☺

Ok... See it didn’t take as long as you thought it would....

Now a dash of color and we can be done!

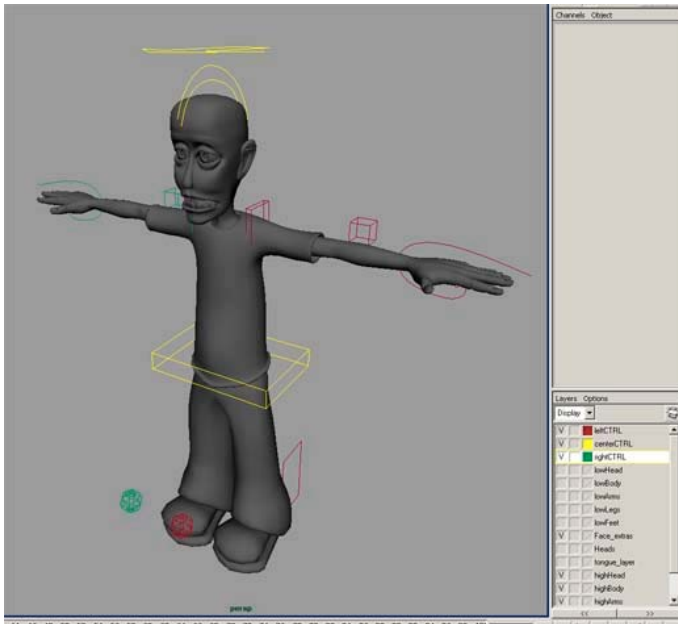
Make three more layers... named “leftCTRL”, “rightCTRL”, and “centerCTRL”

Select all of the left side controllers and select the leftCTRL layer, RMB -> Add Selected Objects

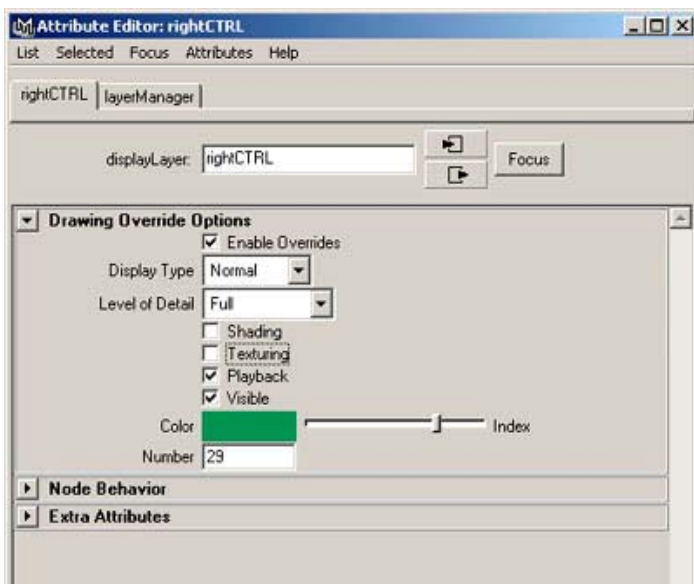
Do this for the right and center controls...

Now double click the “leftCTRL” layer and pick a new color for it...

Do that for each of the three layers... now you controls are easy to see from any angle and easy to pick too...



Sometimes though when you add some of the controls to the new layers they become shaded again... to fix this RMB the layer and select attributes... turn of shading and texturing... Also in the layers, set all of the highRes Layers to "R" or reference so you don't accidentally select the highRes geometry..



THINGS TO ADD/FIX

Use driven keys to rotate the spine if the constrain doesn't work properly...

write driven keys for fingers...(not just feet!)

More elaborate Eye Setup (pupil and aim)

UV Unwrapping

Detailed paint weights section