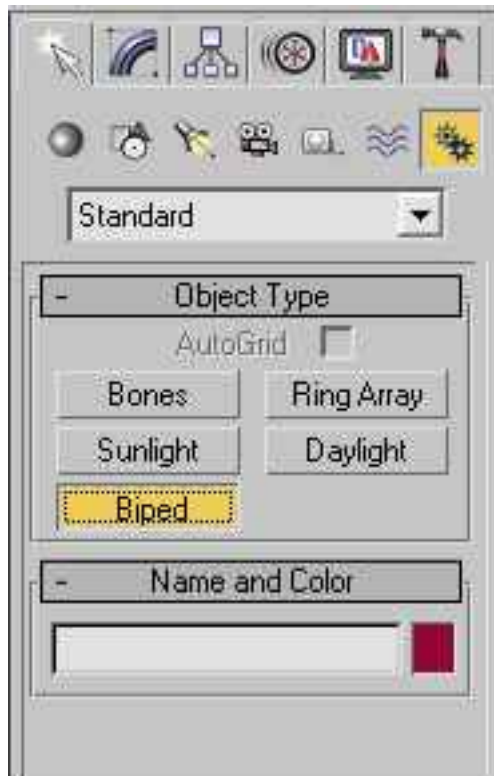


Step 2: Biped Rigging

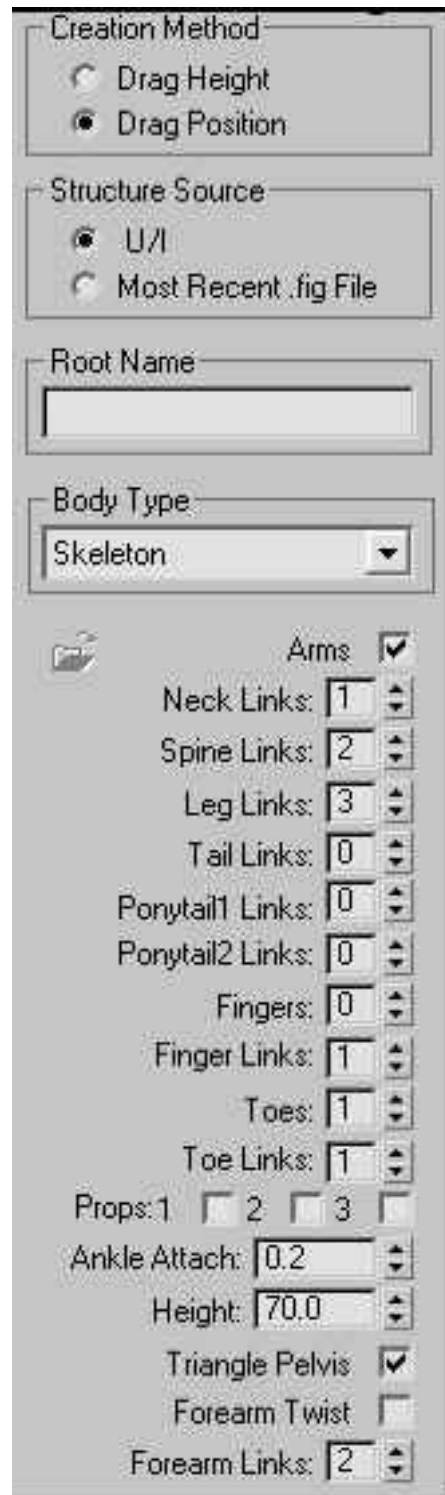
Ok now we are going to rig our model for ingame use in halo ce.

First thing we are gonna do is open up 3ds Max and creat a new scene. Also make sure that you have the Physique pluggin loaded up and running, that plugging should have been included with Character Studio for 3ds max, without it well im not sure what to do...

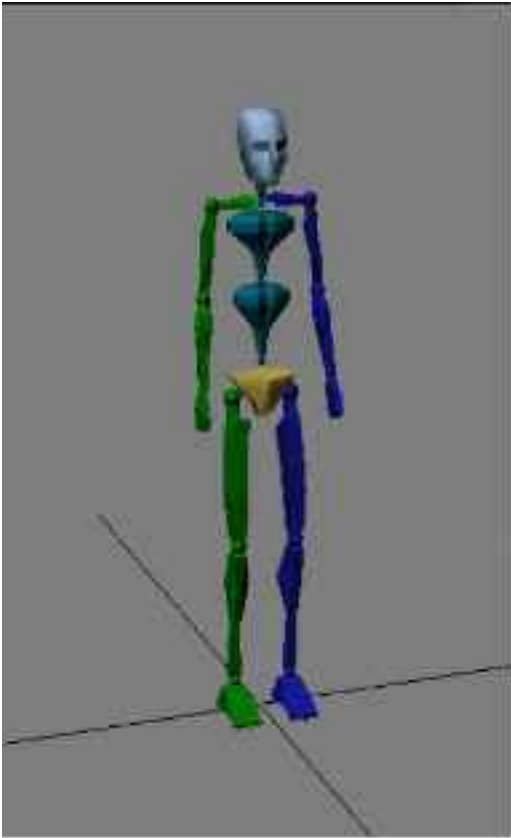
ok now first we are gonna creat a biped



next enter the following specs for your biped
Make sure all of it is exactly the same



Now click and drag the biped to the center of the plane. It should look like this



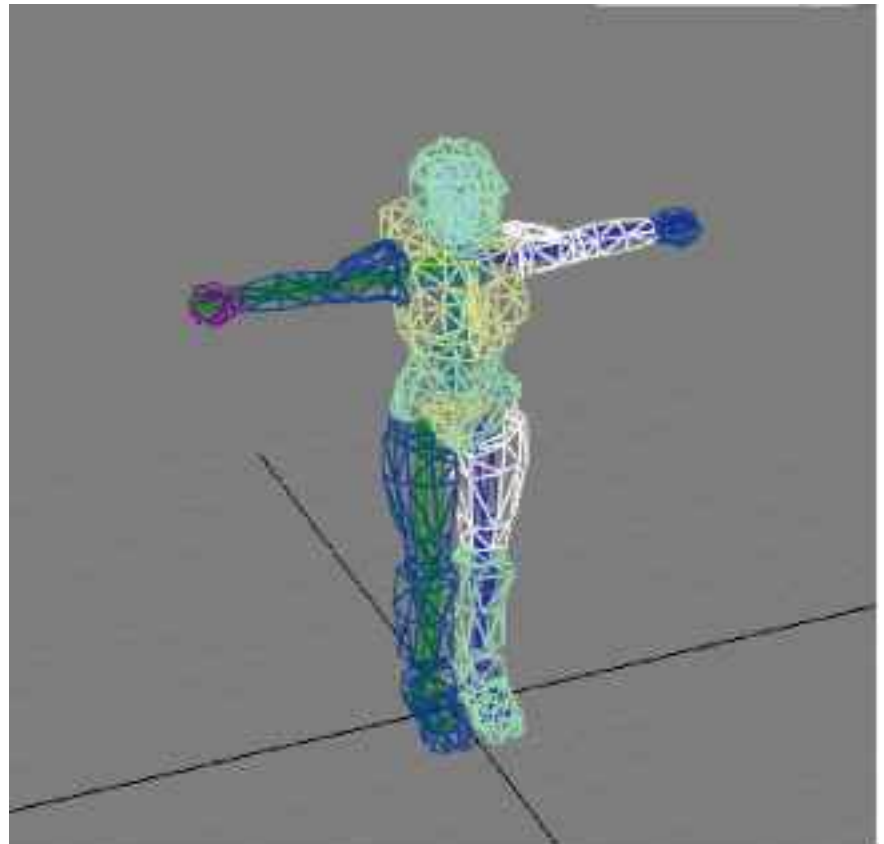
next you must unlink the toes by selecting them and clicking the unlink button at the top toolbar.



this is because the halo biped has no toes and if you want to use all of his old animations u will have to do this to match the node data of the old one. next right click them and convert them to editable meshes twice so that they are not exported.

Now we can import our model and adjust the biped to fit it.

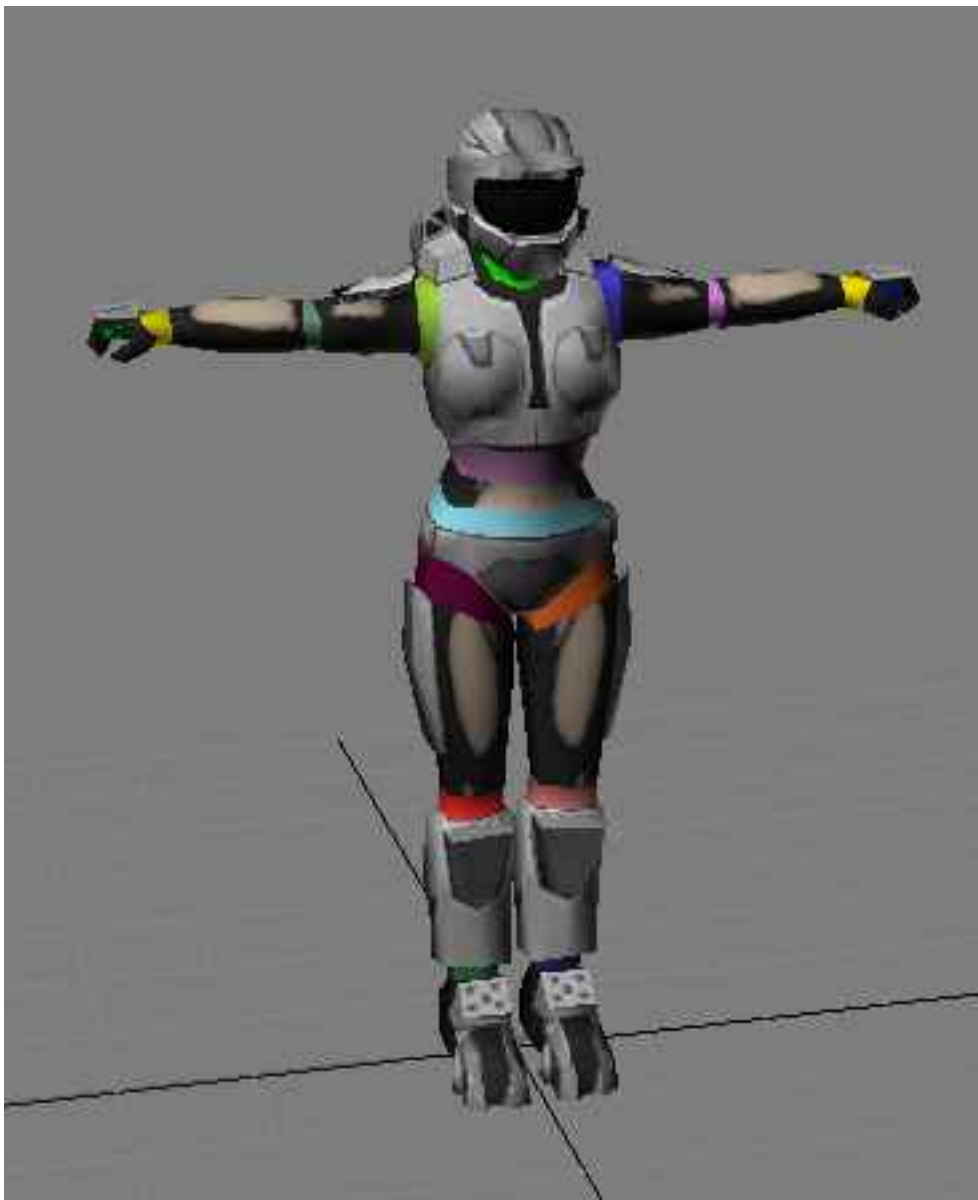
Now select the biped and click the wheel like thing on the right panel. there u will see a small little stick figure dude. Click it and now u can transform and manipulate the biped's body parts. you want to move scale and rotate the limbs into position to fit the model we built. dont forget to make sure that our model is to scale so it fits the size of the biped.



now that our model matches the biped we can start linking the body parts to there propper biped bones such as linking the chest to the chest bone, the head to the head biceps to upper arm bone, thighs to thigh bone etc all using the link tool. just click the model piece and drag it to the bone



Next We will make a new materail/submaterail group and assign 1 material ID to the model. Next apply different ones to the small section of polygons that surrounds each joint where we want the polygons to stretch while moving. This process helps group them all so when working in the tag editor we can work on each joint at a time while rigging.



here i have color coded all the different joints so that you can see where i want the joints to deform when in animation. Also keep them named so u can easily refer to which one is which in the tag editor we apply the same texture to all of them.

now make sure everything is properly linked up and saved. next we will import an exported txt file of the original MC GBX model using James. Model imported. delete everything except for the markers like head, left arm, right arm, etc. next place attach them to there designated hosts, left hand to left hand head to head, etc... dont worry about orrienteing them u will copy the data from the original MC in Kornman00.

Now you can Export as JMS and compile it in tool. If any errors popped up make sure u check that everything is linked properly and the jms file in in the right model folder that u typed in tool. dont forget to complile your bitmaps as well. and creat your shaders.

Now u can go back-up then open your cyborg_mp tag and replace the gbx model with the exported one and complile and test your map. Your model should be able to work ingame but its not done yet as your may notice the edges around the joints are disconnected, the next task is to manually bind those verts to the partent nodes. this will be the longest and most pain staking process of all time but it is the final step towards getting your model fully rigged.

-Binding individual vertices in Kornman00 (hacked guerilla)

The hacked HEK tools can be downloaded from

http://kornner.epgservers.net/mambo/index.php?option=com_remository&Itemid=42&func=selectfolder&filecatid=2

For those who don't know what Kornman00 is it's an altered version of guerilla that allows you to edit vertex data. In this we will only edit what is under the Uncompressed Vertices. Don't pay any attention to Compressed Vertices. Now make sure you have your gbx model tag opened in the tag editor and export it as txt. Now open this txt file in wordpad or notepad. You will use the txt editor to find the vertex by using the find tool to look for its X Y or Z location. Now that you have the txt file opened go to edit > find. This is where you will type in the vertex location and which node that you want to bind it to.

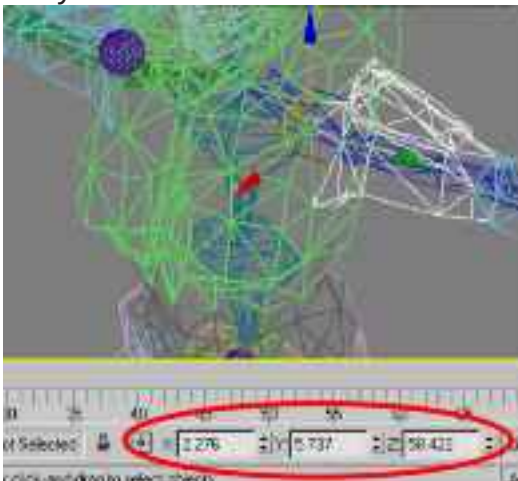
These are the translations of which node index is which bone
node index:

- 0 = pelvis
- 1 = Left thigh
- 2 = right thigh
- 3 = lower back
- 4 = left shin
- 5 = right shin
- 6 = upper spine
- 7 = neck
- 8 = left foot
- 11 = right foot
- 12 = head
- 13 = left upper arm
- 14 = right upper arm
- 15 = left forearm
- 16 = right forearm
- 17 = left wrist
- 18 = right wrist

Now when you find the vertex that you want to switch which bone it's attached to all you have to do is change the node0 index to whichever one you want. If you want a vertex to share its distribution between 2 nodes (node0 index and node1 index) just type the other node that you want it to share with in the node1 index. This can be done in both Kornman00 or wordpad/notepad. Make sure when sharing a vertex between the 2 bones that the weights for each node index add up to = 1 like 0.5 and 0.5, this will equally distribute the weight which the vertex is affected by, this is useful when you want a more smooth rig in a certain area.

-Finding Uncompressed vertices

Now to find a vertex you must search the txt document for the vertex's location and element number. The location of the vertex is found in the 3ds max document. Convert to editable mesh use the move tool and select by vertex. At the bottom you will see the vertex's x y z locations. I usually look for the y or z locations since these are the ones that are most unique.



Now these units are not the same as Halo's units there 100x larger in max so to get the Halo's unit conversion divide it by 100 and you get the locations in the tag.

In this case we selected the vertex on the left upper arm and noted that y z its locations are 0.0573 and 0.5842. Leave out the 3rd decimal digit because 3ds max tends to round it off.

Now in the txt file we will first type 0.0573 in the find window and click find next. You may come across this number several times but make sure it's in the position real point 3d line also make sure that the other numbers match up to the xyz locations in 3ds max. And also that it's connected to the proper node index it was binded to in 3ds max

the line should look something like this after it has been found

```
element 3
  position    real point 3d    0.022758    0.057366    0.584224
  normal      real vector 3d    0.737986    0.360577    0.570404
  binormal    real vector 3d    -0.699097   -0.009112    0.714968
  tangent     real vector 3d    -0.084133    0.927847   -0.363348
  texture coords real point 2d    0.063818    0.743388
  node0 indexshort integer      13
  node1 indexshort integer      -1
  node0 weight    real  1.000000
  node1 weight    real  0.000000
end element 3
```

now u go back to kornman00 and for the left upper arm block find vertex 3 and change the node index0 to 6 since we want this vertex to stick to the upper spine.

Note: Node0 indexs with (3x) by them are compressed vertecies which are note used in the halo for pc so just skip them and ignor them or delete them all.

Note: U can edit this information in Notepad or Wordpad but it can cause sapien to crash when loading it.

after that save it and begin looking for the rest of the vertexs in this area that u want to pin to the 6 node. also since this is the shoulder bone we may want some verts to be shared between nodes to do so just leave the node0 index as it is and put 6 in the node1 index then for the weights type in 0.5 and 0.5 in each or maybe 0.6 and 0.4, it all has to add up to 1 though. if its it 0.7 on the 6 node, then it will be stuck more to the that bone than it is to the arm bone. after your done save it and look for the other vertexs and bind them. Do this to all the vertexs on the bones that u want to warp. be patient this is allot of work, this can take anywhere from a few hours to even a week or 2. It depends on the ammount of polygons in your model and the ammount of quality u want your biped to have. Dont forget to regularly save and text your map to make sure they bind properly also look for ones you may have missed or leave a open whole in your biped. if u edited this info in wordpad or notepad u must save then import into the tag editor then save it again. Remeber to keep refering to which node index is which bone and make sure you bind your vertex to the right bone. this is the final step towards rigging your model and once you have completed this step on all of the transforming vertecies, you are DONE!!!!

Congradulations u have just completed the hardest task to do in HEK! enjoy playing as your own custom character in halo ce!

Tutorial Produced by
Jahrain

Special thanks to Taimat for figuring out that custom biped are possible in halo ce and for teaching his words of wisdom to me which helped me also figure this out, now with this tutorial anyone should be able to make there own custom bipeds in halo.