TD Matt

Maya character rigging for games, scripting, Unreal Engine.

Sunday, 13 February 2011

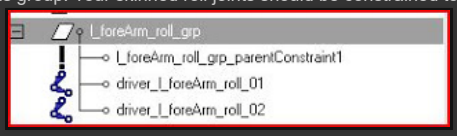
Setting up a forearm roll joint - method 2

Here is a second method for setting up forearm roll joints. It's a follow on from this post which covered how to set it up using the rotateY attribute of the wrist joint. This second method is the way I implement a forearm roll system, but I wanted to post the other version simply because it's a great way of looking at how different rotation orders can have implications for your overall rig set up, stability and usability.

So, to set up the forearm roll:

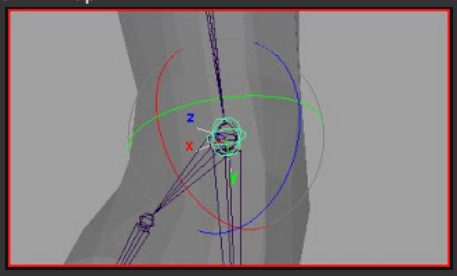
The basis for this is a single **aim constraint**, and is pretty quick to implement.

Start off, as usual, by duplicating the skinned roll joints and prefixing them with *driver_*. Create a group called *l_foreArm_roll_grp* and **parent constrain** with no offset to *l_elbow*. Parent both your driver roll joints to this group. Your skinned roll joints should be constrained to these driver joints.



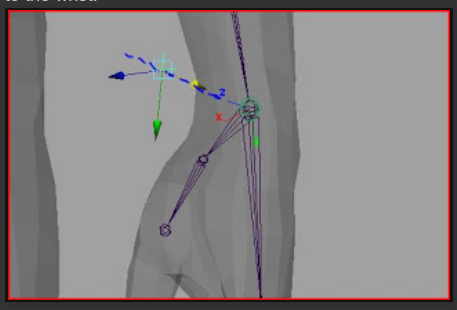
The **aim constraint** will be set up on *driver_l_foreArm_roll_02*. This will inherit the twist rotation only from the wrist joint, then a percentage of this will be used to drive *driver_l_foreArm_roll_01* which is located mid-way between the elbow and the wrist. You can see how the skin weighting is distributed across these joints in the previous post.

Here you can see the axis displayed for *driver_l_foreArm_roll_02*. We need two vectors for the **aim constraint** - an **aim** and **up**.



The **aim vector** is straightforward, in this case it will be -Y pointing towards *l_elbow*. If your joint orientations are different to mine it may well be a different axis, you should be able to figure out the correct one by displaying the local rotation axis as I have here.

The **up vector** will be controlled by a locator called *driver_l_foreArm_roll_02_upVec* which will be **parent constrained** to the wrist.



The position of the locator is important. You want it positioned at the side of the wrist, directly in line with the local wrist axis for that direction. For my skeleton this is +Z axis, and this will be the **up vector** for the **aim constraint**. Once you've got it positioned correctly, **parent constrain** it to *l_wrist*. Here is a quick capture as it's difficult to figure out the position from a still picture.

About Me

Matt

Senior Technical Artist for Ninja Theory Ltd.

View my complete profile

Links

- Keith Lango
- Chris Goodall - animator @ NT
- Jason Schleifer
- Animation Nourishment
- Rigging 101 - a good starting point
- Ninja Theory
- JB - Character artist @ NT
- tech-artists.org

Blog Archive

Search This Blog

Search

Total Pageviews

2399

71

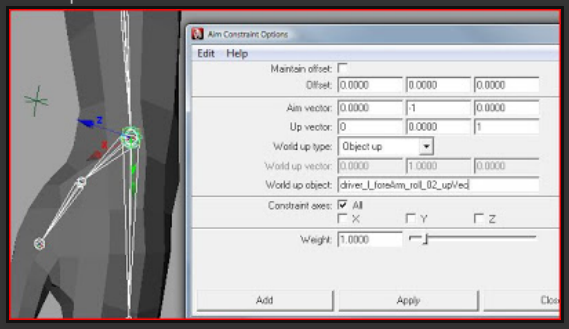
Followers

Subscribe To

Posts

Comments

So, create the **aim constraint** from *l_elbow* to *driver_l_foreArm_02*. The locator *driver_l_foreArm_roll_02_upVec* will be the **world up object** and we'll use the vectors we figured out in the previous steps.



After creating the constraint, check the rotation values for *driver_l_foreArm_02*. If your locator was correctly positioned and the vectors correct you should have 0 0 0 rotation on this joint. If not you'll need to retrace your steps and figure out what the problem is. A small rotation value would likely point to a slightly mispositioned locator, while large 90 or 180 values would suggest the **aim constraint** vectors are incorrect.

The final step is to drive *driver_l_foreArm_01* by taking a percentage of the roation from *driver_l_foreArm_02*.

Create a **multiplyDivide** node called *foreArm_roll_divide*. Set **operation** to **divide** and **input2x** to **2**. Connect *driver_l_foreArm_02.rotateY* to *foreArm_roll_divide.input1X* and *foreArm_roll_divide.outputX* to *l_foreArm_roll_01.rotateY*.

Posted by Matt at 15:00

Labels: forearm, roll joint, twist

2 comments:

karlijn 3 October 2012 at 13:38

Hi Matt,

I think there is a small error in the naming. At one point you change from " driver_l_foreArm_roll_02" to "driver_l_foreArm_02". Atleast that's what I think is happening. I could be wrong. :)

Gr, Karlijn

Reply

Unknown 3 November 2015 at 10:10

This comment has been removed by the author.