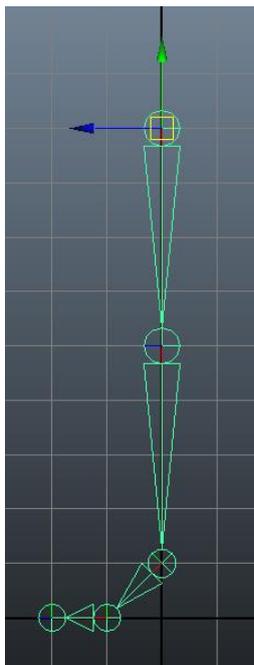


Reverse Foot Lock

The reverse foot lock is a method of rigging the foot that allows us to simulate the way a foot 'rolls' from the heel, to the ball of the foot and to the toes. The most common method of doing this is to use extra joints that double back on themselves to give you this motion, however I much prefer this technique using groups.

The advantages are increased flexibility in your rig, and a much cleaner workflow when you have your joints visible.



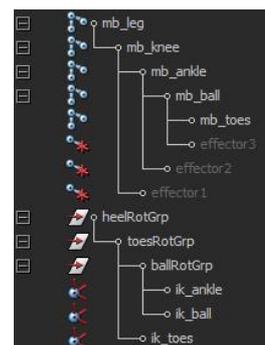
First we need to build our leg joints. Using the *Joint Tool*, create and name your leg joints in the side menu. We need a *leg*, *knee*, *ankle*, *ball* and *toes* joint.

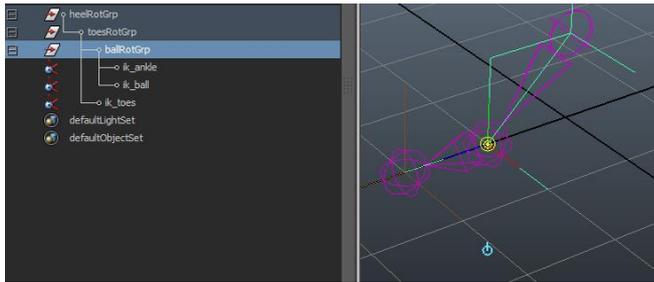
Using the *IK Handle Tool*, create an IK between the *leg* and *ankle* joint, the *ankle* and *ball* joint and the *ball* and *toes* joint. This will give us three IK Handles that we will use to control the way our foot moves.

We will name these *ik_ankle*, *ik_ball* and *ik_toes*.



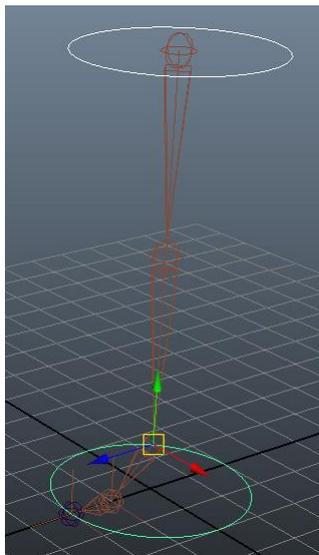
Group *ik_ankle* and *ik_ball* handles together in a *ballRotGrp* by selecting them and pressing *Ctrl-G*. *Shift-select* (or *Ctrl-select* in the outliner) the *ik_toes* handle and press *Ctrl-G* again, rename the new group *toesRotGrp*. Now press *Ctrl-G* one more time and rename the group *heelRotGrp*. Your outliner should look something like this:





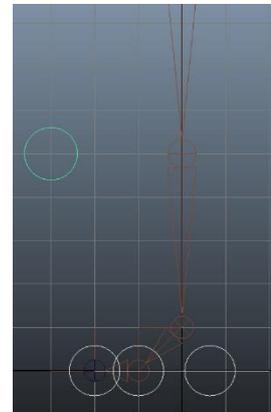
Now that our groups are built we need to pin the pivots to the right points. Using *insert* to toggle or by holding the 'd' key combined with the 'v' key, point snap the pivots of the groups to joints as follows:

- ballRotGrp* - *ball* Joint
- toesRotGrp* - *toes* Joint
- heelRotGrp* - pin to the *ball* Joint and then move back to where the heel will end.



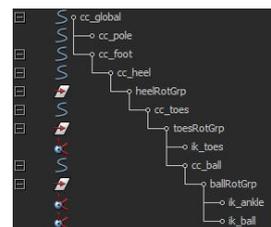
Using NURBS circles we are going to build a control system. First we need a global control to be built at the root of the leg up joint, then an overall foot control where the pivot is set to the ankle joint.

In the side view create three more circles and create them on the toes, ball and heel location.

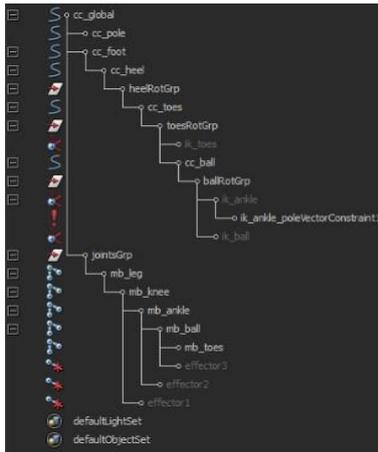


Finally we need to build a circle to act as a pole vector for our leg. Make sure you name your controls appropriately (I would recommend using the prefix *cc_*). After our controls are built, we need to delete our history (*Edit > Delete by Type > History*) and freeze our transformations (*Modify > Freeze Transformations*).

Re-arrange and group the controls to match the following Hierarchy: Remember you can use *middle click and drag* to move objects around the outliner.

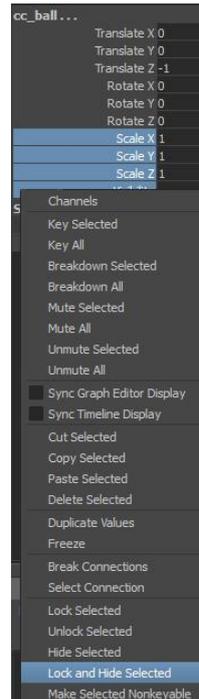


Select the *cc_pole* control and then the *ik_ankle* handle and build a pole vector constraint (*Constrain > Pole Vector*).



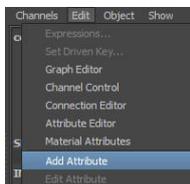
Our final step for this configuration is to clean our scene up a little and limit the control of our controls. First group the joints into a *jointGrp* and parent that group under the *cc_global*. We can also hide all of our IK Handles while we are in the outliner.

We can lock the scale and visibility attributes in all of our controls apart from the *cc_global*.

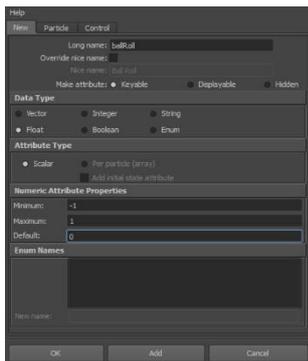


In the same way we can lock and hide the *translation* attributes in your *channel box* for the *cc_heel*, *cc_ball* and *cc_toes* controls and the *rotation* attributes on our *cc_pole* control. For the *cc_heel*, *cc_ball* and *cc_toes* controls you could also lock and hide the Rotate Y and Z to make it simpler, although some user may appreciate the ability to rotate the heel and toe controls. We now have a reverse IK control that is as powerful or simple as we would like.

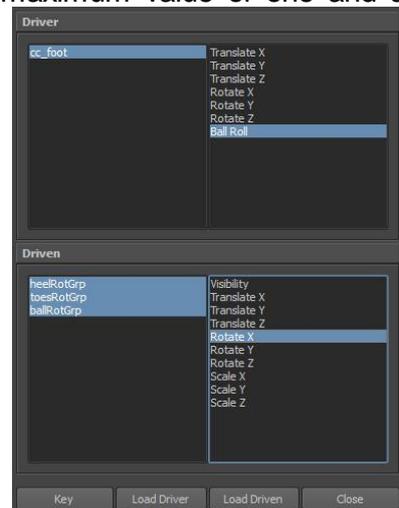
As an extension to this control we could add a ball roll attribute in our *cc_foot* control curve and use that attribute to drive our groups in a specifically timed way to give us a nice rolling animation as the value changes from zero to one.



With the *cc_foot* control selected add a new float attribute with the name *ballRoll*, a minimum value of minus one, a maximum value of one and a default value of zero.



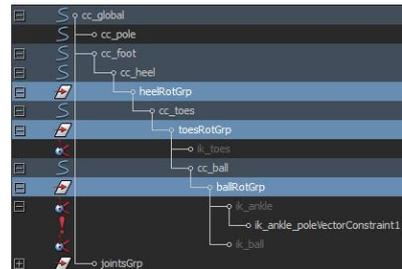
Select the *ballRoll* attribute in the *channel box* and open the *Set Driven Key* tool. We will use this to set up the connection between the *rotate X* attributes for our reverse foot lock groups and the *ballRoll*.



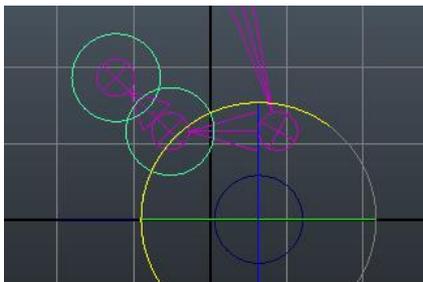
With the Set Driven Key tool open, click on *Load Driver* and make sure the *Ball Roll* attribute is selected. In the outliner select the *heelRotGrp*, *toesRotGrp* and *ballRotGrp* and then press *Load Driven* to set them as your driven objects. Select them again in the *Set Driven Key* tool and then select *Rotate X* on the right.

While everything is zeroed out press the *Key* button to lock the values. This ensures that the default value of zero brings out foot back to a resting point.

Select the *cc_foot* control from the *Set Driven Key* tool and change the *Ball Roll* attribute to *-1*.

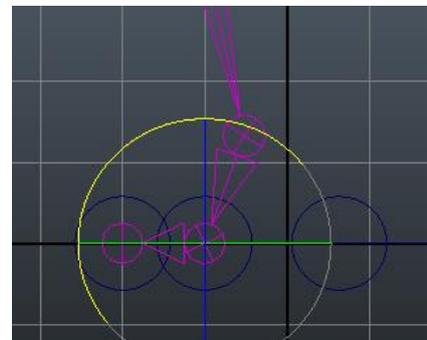


This will be the start position for our ball roll, which is when the heel is being planted. Rotate the *heelRotGrp* back to -45° in the *X-axis* and then press *Key* in the Set Driven Key toolbox.



Select the *cc_foot* control again and change the *Ball Roll* value to *0.5*. Set the *rotate X* value of your *ballRotGrp* to *25 degrees*. Make sure both the *ballRotGrp* and *toesRotGrp* is selected and press the *Key* button to lock the values.

Our final step is to set the *Ball Roll* value to *one* and key the *toesRotGrp* with a *Rotate X* of 45° and the *ballRotGrp* back to *zero*.



This concludes our reverse foot lock tutorial. You can test the foot roll by selecting the *Ball Roll* attribute and *MMB click dragging* the mouse in the view pane. As you scroll from *-1* to *1* you should see a nice rolling foot simulation that we can use to speed up animation, whilst also having full control over the foot with our control curves.