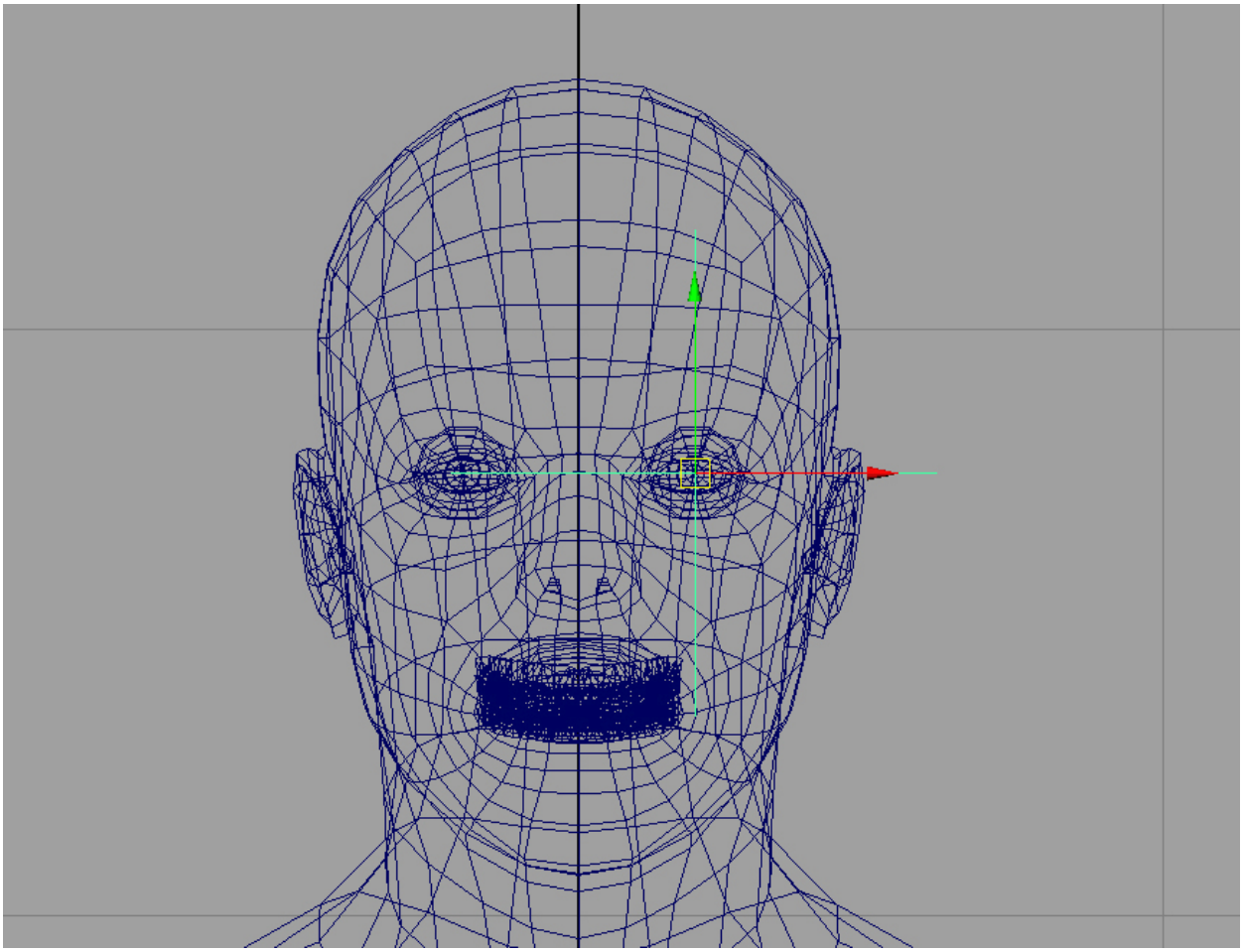
Eye Rigging Tutorial

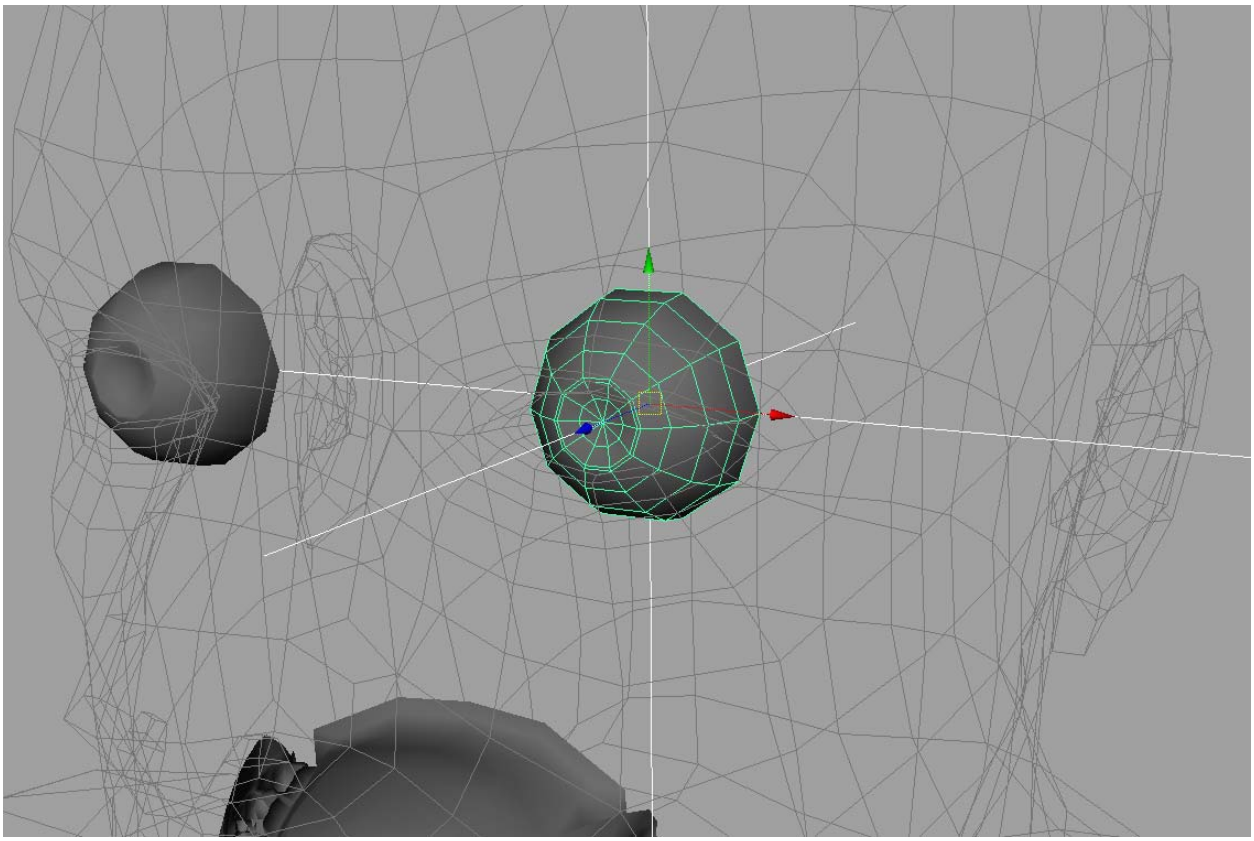
This tutorial will cover how to set up an eye controller to allow the eyeball and eyelids to move together. This will allow a more natural look to the eye when it moves in its socket.

First, you will need a character model with an eye ball inside the socket. The eyeball should be a perfect sphere to allow rotations to be even. This may not work with eyeballs that are elliptical in shape. For this tutorial, I will be using half of the left eye. Since it's spherical in nature to begin with, it won't serve much of a problem as we will not see the back of the eye.

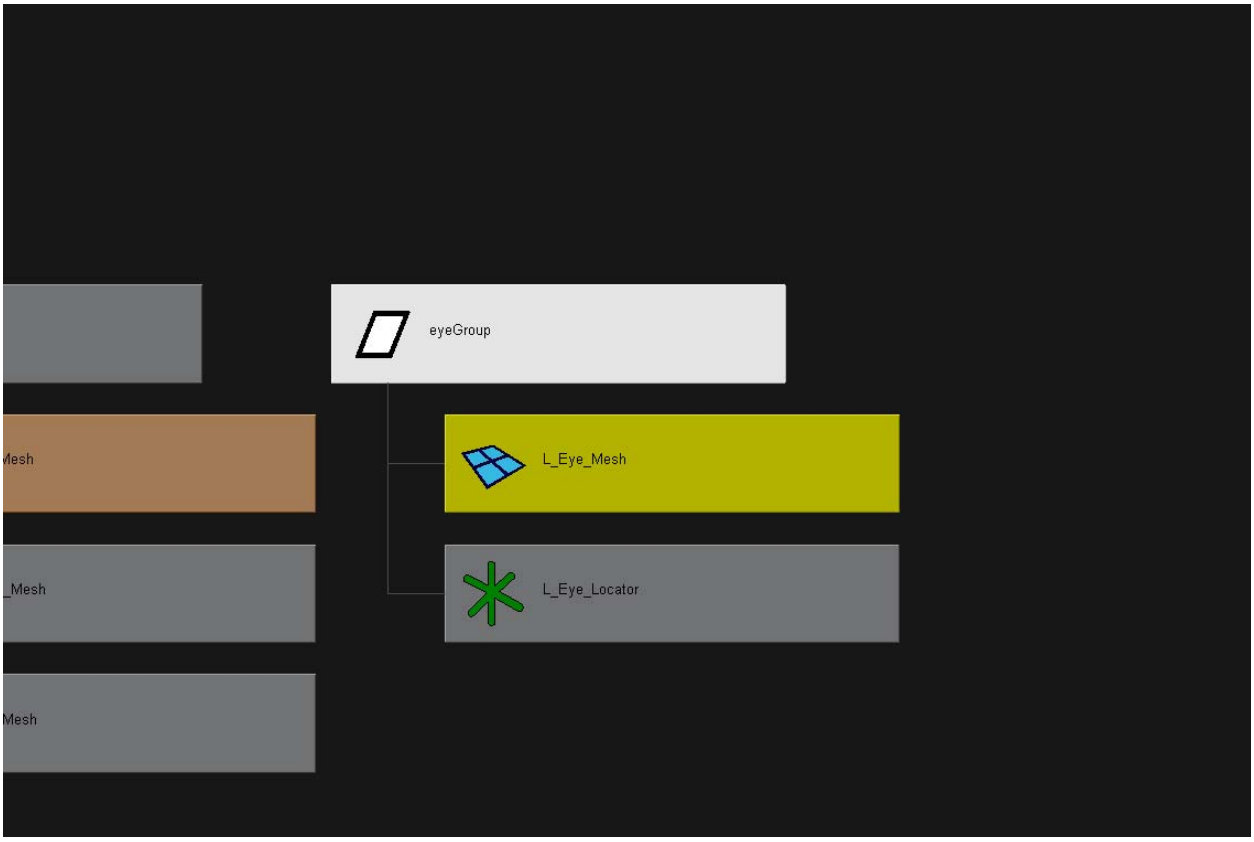
To start off, go to *Create* > *Locator*. Name this left eye locator. Take the locator and move it exactly inside the eyeball so that its rotation axes match those of the eyeball. To do this, go to the front view and hold down the "v" button to snap to vertex. Move the locator so that it's center with the eyeball.



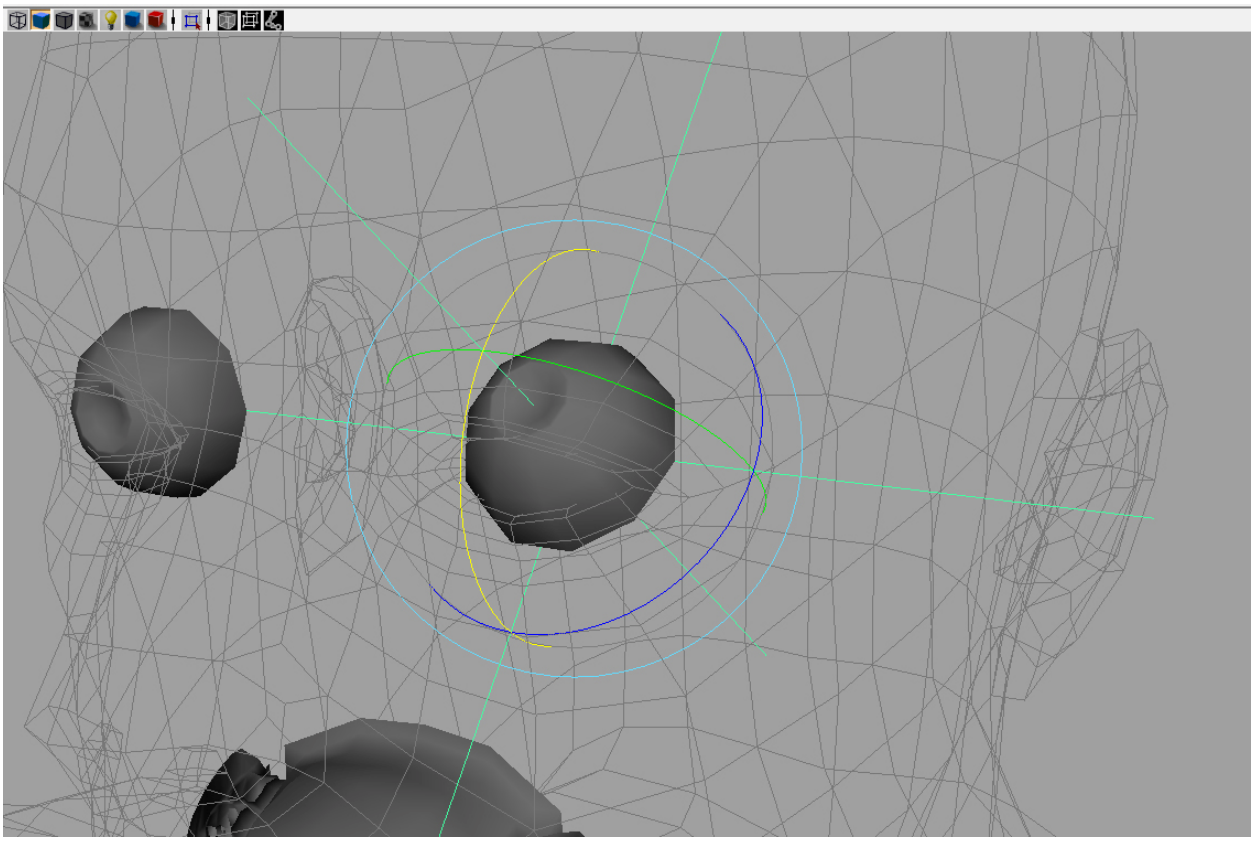
Next, go to the side view and move the locator to the center of the eyeball with "v." Your locator should now be centered with the eye and have matching rotation axes. Freeze the transformations of the locator when it is in place.



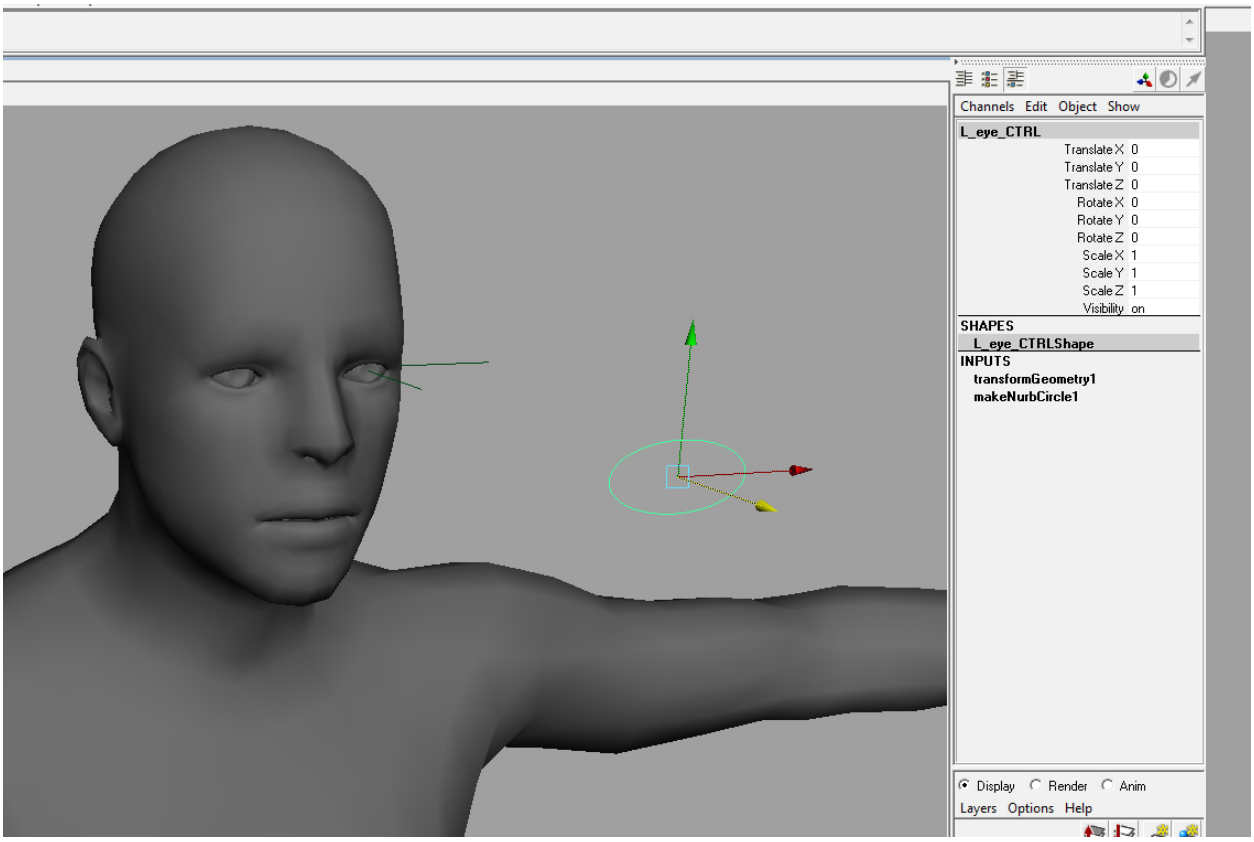
In the Hypergraph Hierarchy (*Window > Hypergraph: Hierarchy*), create a group for the eyeball and locator and name it appropriately (ie. eyeGroup). This will hold some of the components that we will use later.



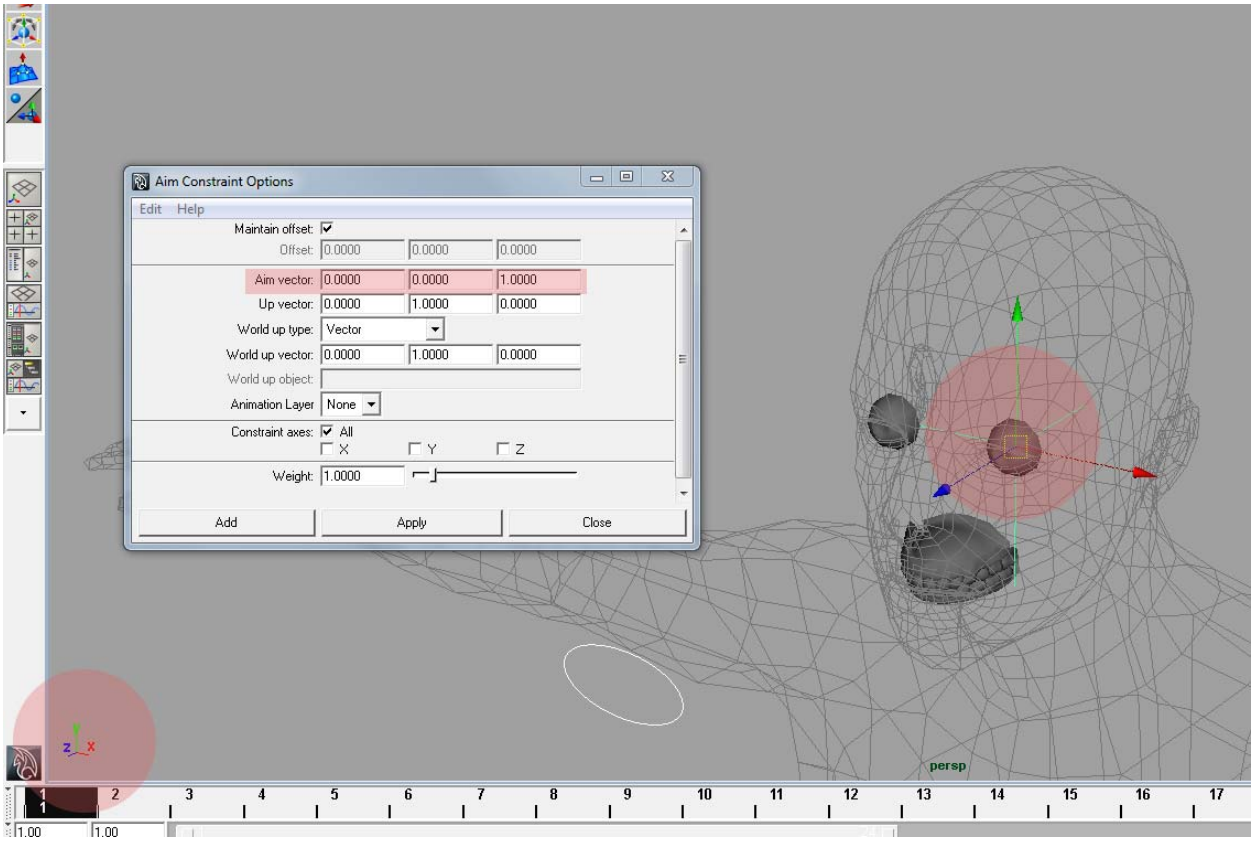
Next, select the locator and then the eyeball. Go to *Animation > Constrain > Orient* to add an orient constraint. To ensure this worked, rotate the locator to see if the eyeball follows along with it. If it does, you're good to go. If the eyeball flips upon adding the orient constraint, be sure to freeze the eyeball's transformations before adding the constraint.



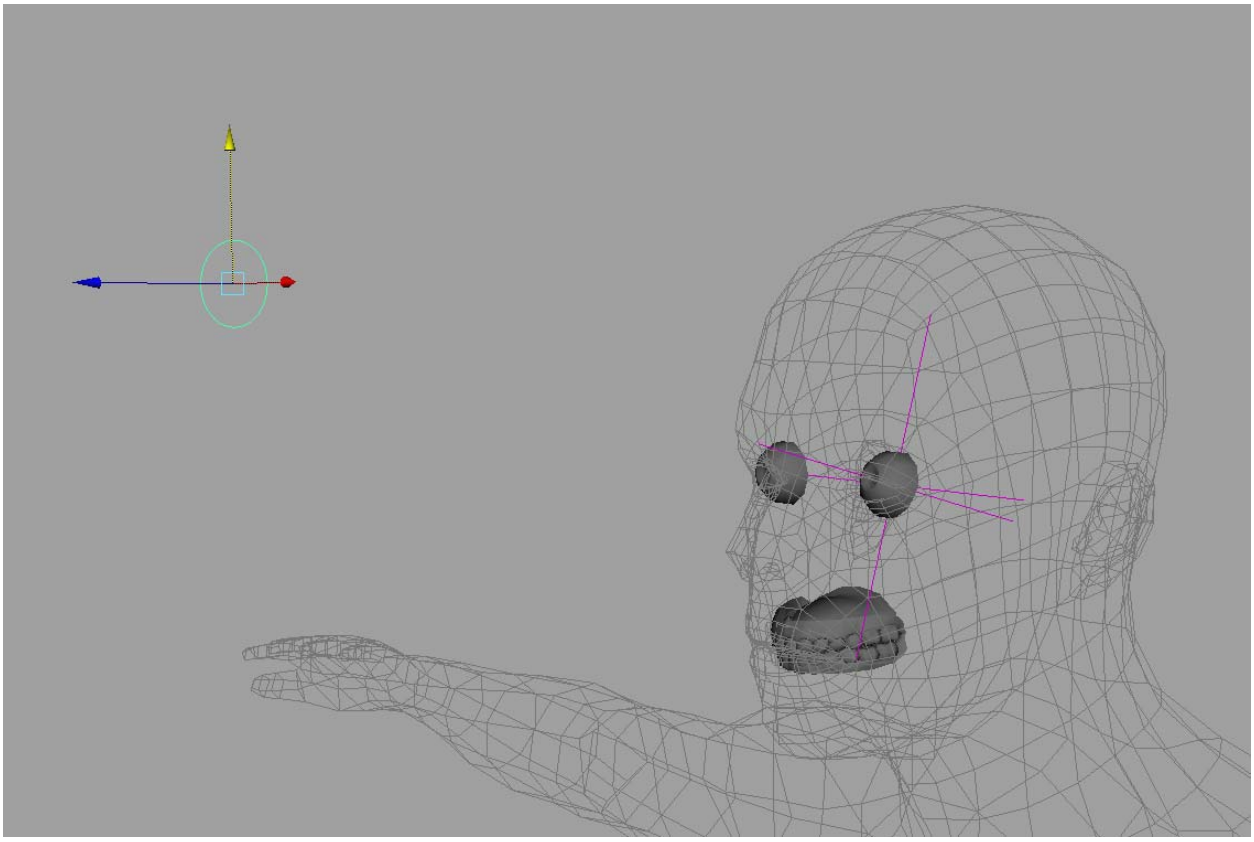
To control the eye's rotation, we will need a controller. Create a NURB curve by going to the Curve shelf and selecting the circle curve. Name this circle curve as the left eye controller. Like the locator, go to the front view and hold "v" to move and snap the controller to the eyeball's center vertex. It's recommended to change the curve circle into an eye shape (or anything close enough to one) to match that of the eye's shape. Go back to the perspective view and move the eye controller out a good distance away from the eyeball. Freeze the transformations afterwards.



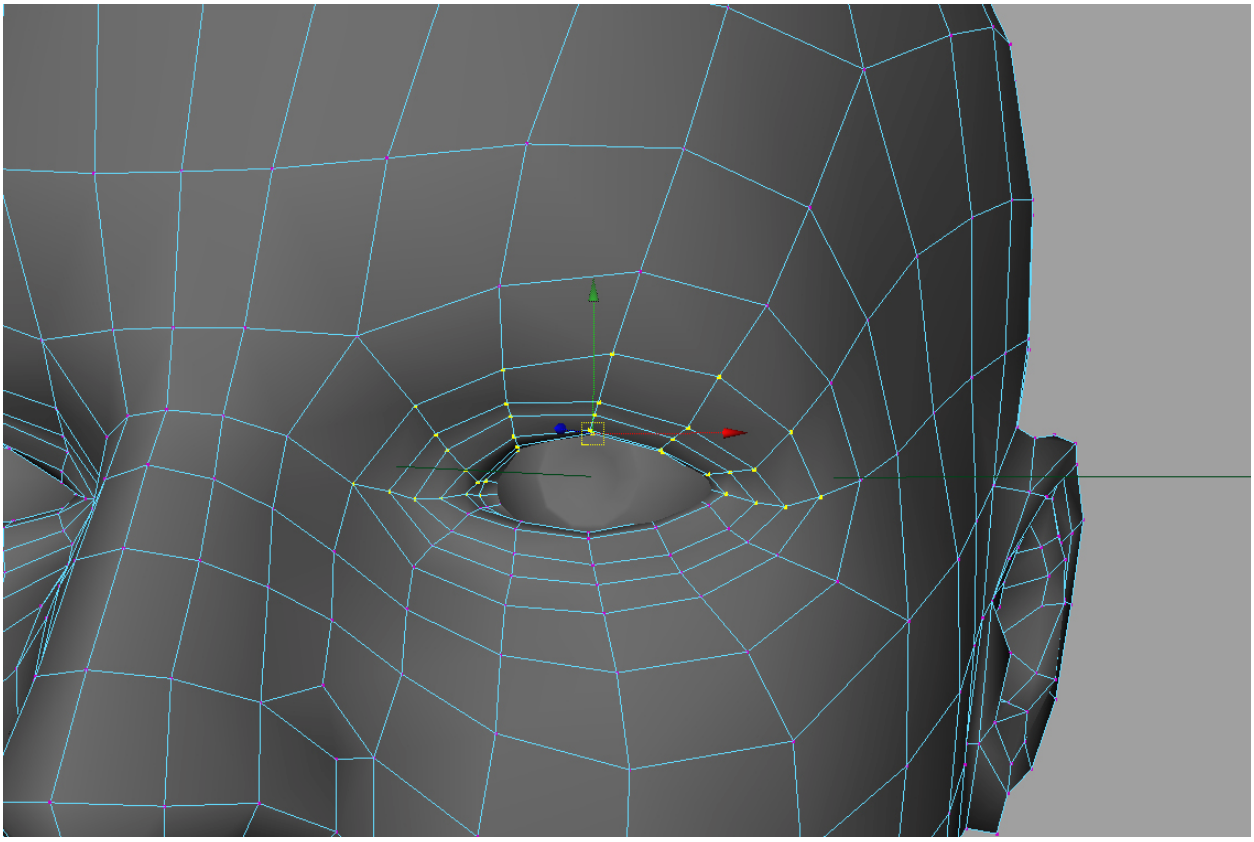
Now select the left eye controller and then select the locator. Go to *Animation > Constrain > Aim > [Option box]*. Now pay close attention to how your space is set up. On the lower left, you'll see the cross shape axis with a blue Z, a green Y, and a red X. Compare this to the color arrows of the Move tool to know where they are in accordance to the space's axis. For example, the character model I am using is facing the blue Z axis. Now let's look at the Aim Constraint Option Box.



Where it says Aim Vector, there are three values you can change. The first one is for the X axis, the second one is for the Y axis, and the third one is for the Z axis. By default, the first column should have a 1.000 value. This determines where the eye locator will point to depending on the axis you give a "1" value to. Since my locator is pointing towards the Z axis, I want to change the first column to "0" and the third column to "1". Remember that this will vary depending on where your locator is positioned in the space, so if your eye locator is pointing towards X, you will want the first column to have a value of 1. When you have everything set, add the aim constraint. When you move the controller up and down, the eye should now follow the controller.

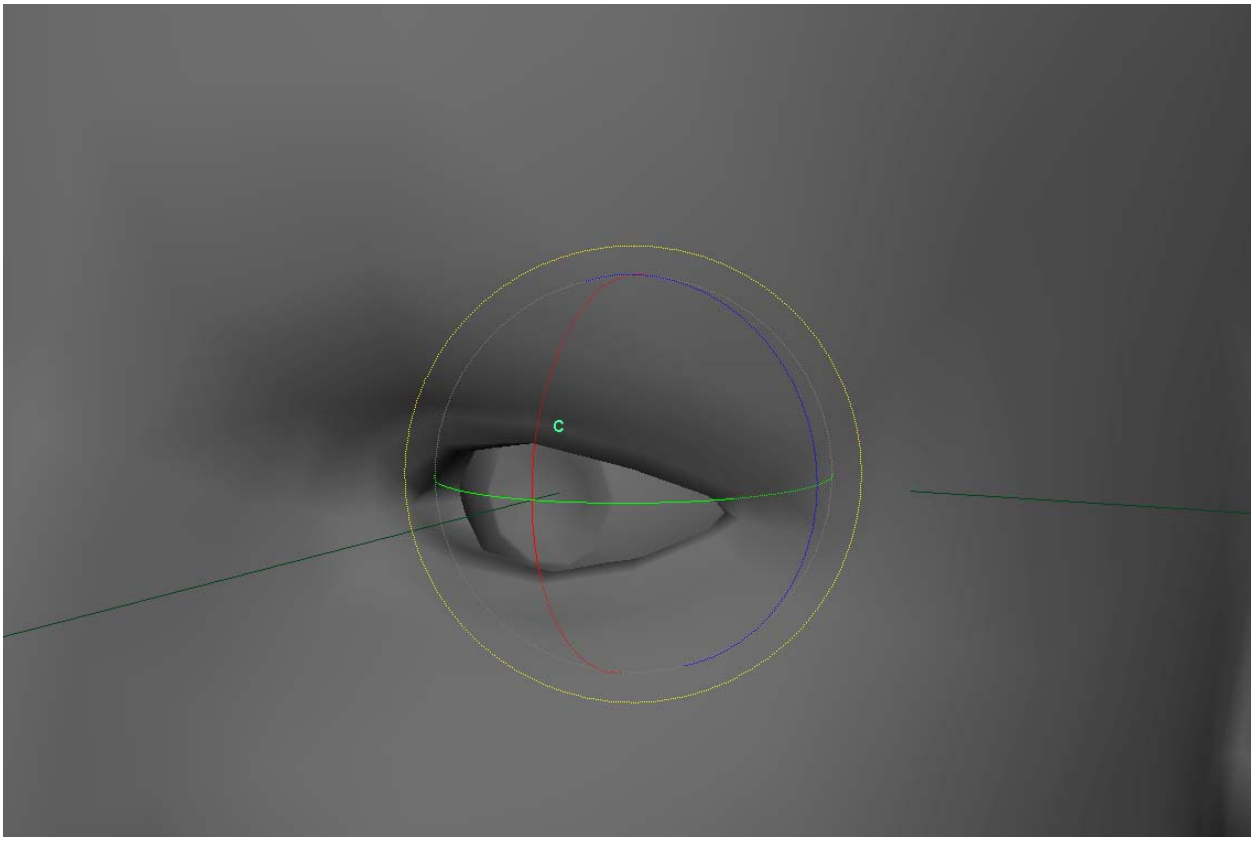


To add the next element of realism to the eye movement, we will have the eyelid move with the eyeball as it rotates. Select the top vertices above the eyelid in which the eyes would blink.

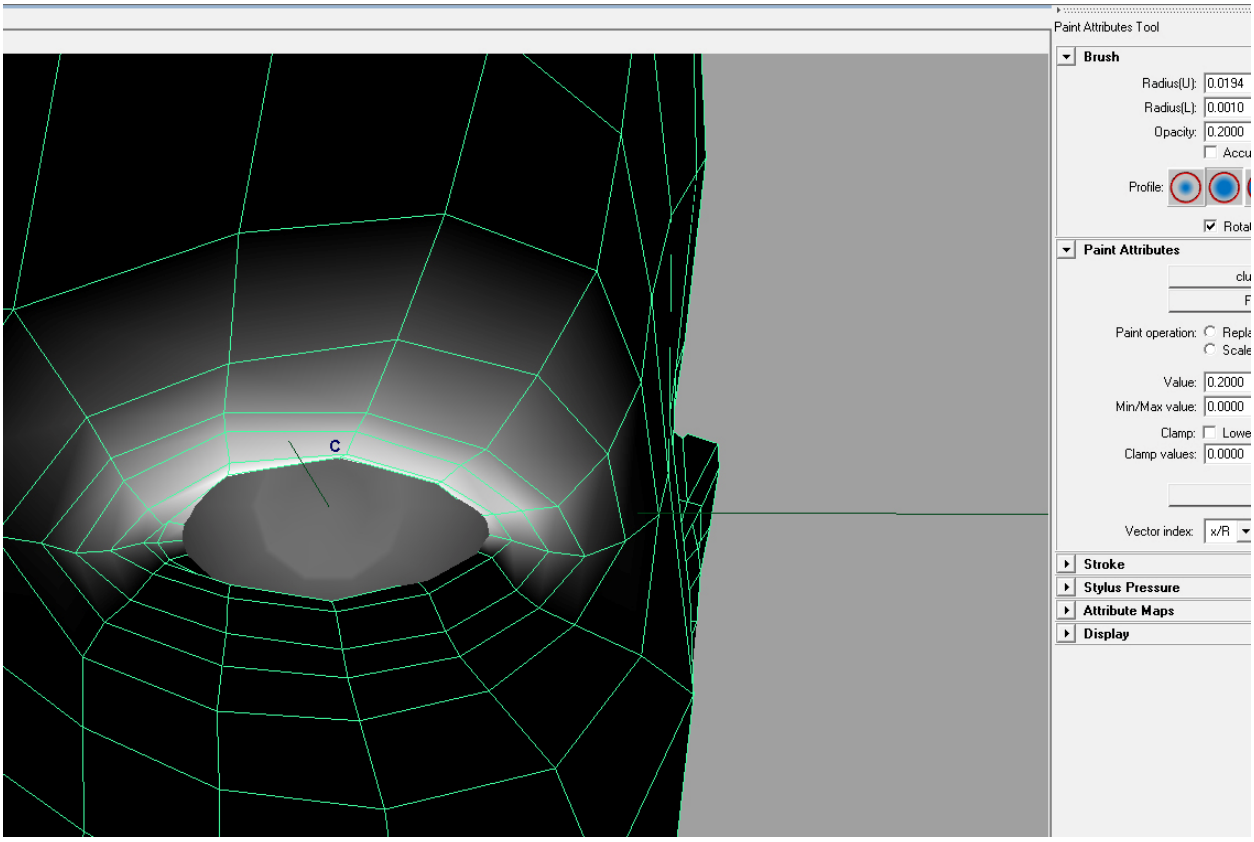


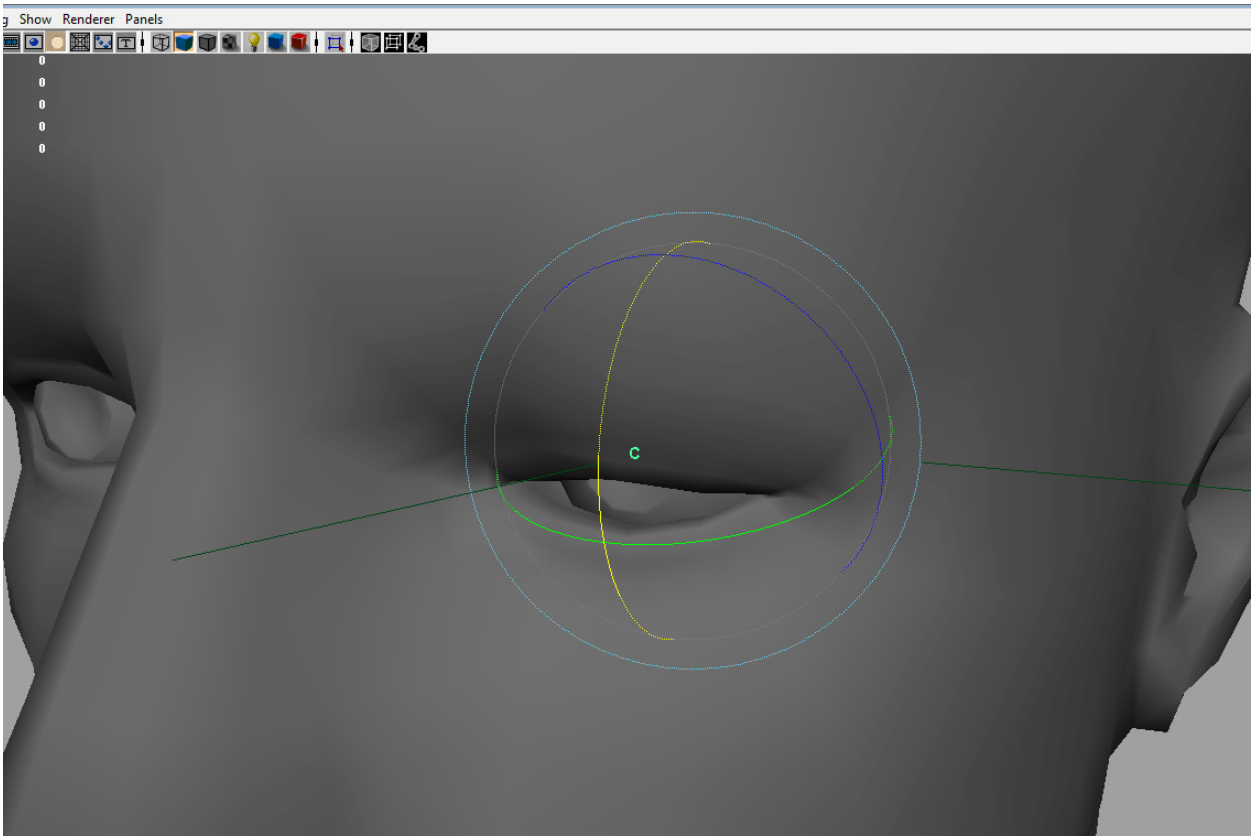
Now go to *Animation > Create Deformer > Cluster > [Option Box]* to apply a cluster deformer. In the option box, be sure to check mark the Relative box so that when the character moves, the cluster deformer will move with it.

You should now have a floating "C" that will move the vertices you have previously selected. However if you rotate them now, it will not match up with the eyes. To move the cluster deformer's rotation axis (not the deformer), press the INSERT button and move its pivot into the same location as the left eye locator. When it's in place, hit INSERT again to set the pivot of the cluster to the same location as the left eye locator. In the Hypergraph Hierarchy, place the cluster into the eye group.

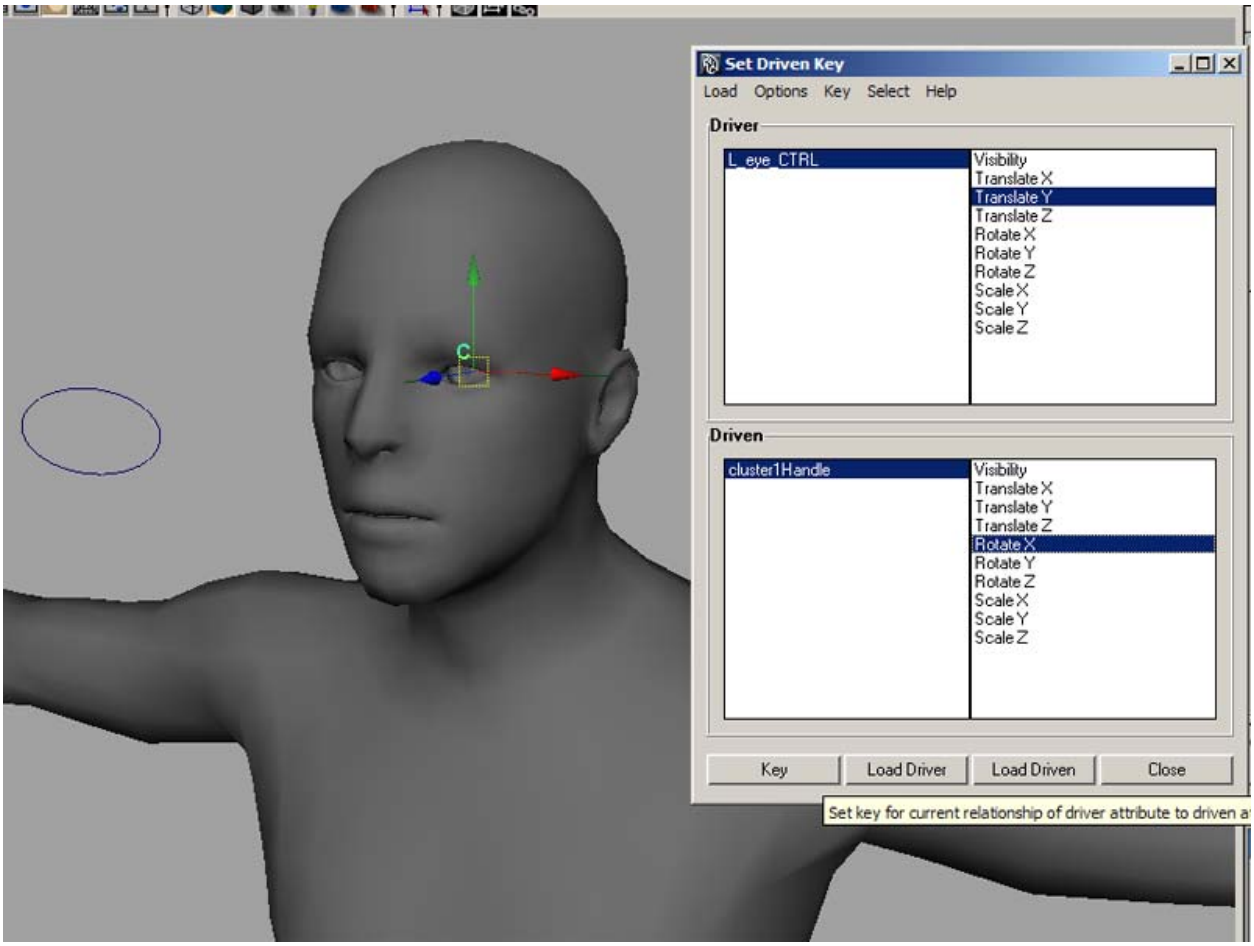


The weighting for the cluster deformer needs to be edited as it is too strong. Rotate the eyelid so that it is in the closed position. Select the body mesh and then go to *Animation > Edit Deformers > Paint Cluster Weights Tool* to paint weights onto the vertices. Select the Replace radio button, set the Value to 0, and press the Flood button to get rid of the default weight. Select the Add radio button and lower the Value to around 0.200. Start from the lower lid of the eye moving up to the top with gradation of weight. Near the side of the eyelids, the weights should start at around 60% and diminish faster as it expands to the sides. Unlike painting weights for a skeleton rig, you can smooth these weights without worrying about weight displacement. Continue to paint weights until you get that right balance of the eyelid following the eye.


Paint Attributes Tool
Brush
Radius(U): 0.0194
Radius(L): 0.0010
Opacity: 0.2000
Profile:
Paint Attributes
Paint operation:
Value: 0.2000
Min/Max value: 0.0000
Clamp values: 0.0000
Vector index:
Stroke
Stylus Pressure
Attribute Maps
Display
C

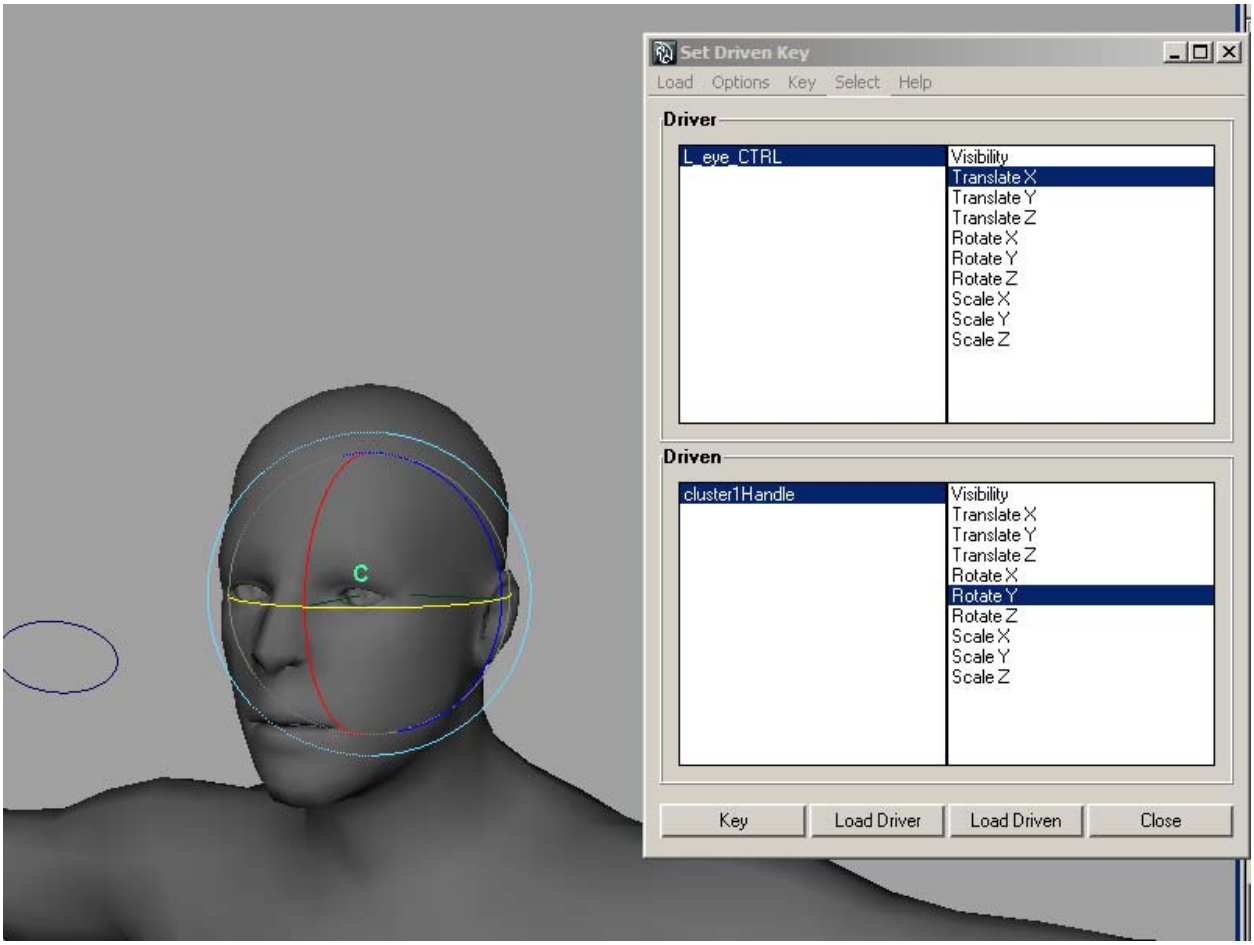

Show Renderer Panels
C

Now that the eyelid works properly, we will use set driven keys to have the lid track with the eyeball's rotation. Go to *Animation > Animate > Set Driven Key > Set* to open a dialogue box. Select the left eye controller and press the Load Driver button. Select the cluster deformer and press the Load Driven button. This means that the left eye controller (the driver) will control the cluster deformer (the driven). For the controller, select the Translate Y in the Driver box. Then select the cluster's Rotate X in the Driven box.



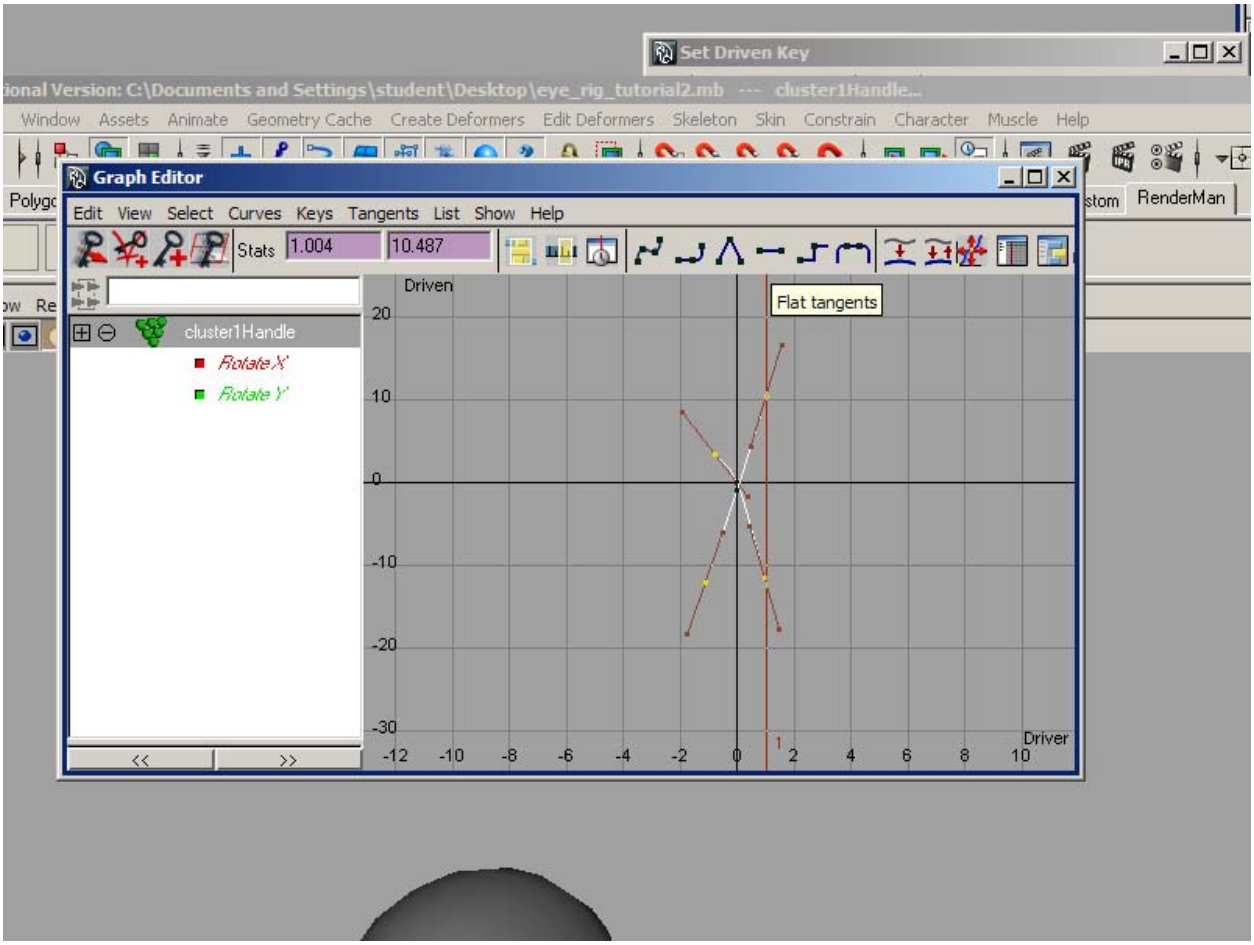
Key these positions when the controller's Translate Y and the cluster's Rotate X are both at "0." Afterwards, move the controller up to have the eye look upwards a bit. Select the cluster and rotate the lid slightly up. Key this position. Move the controller down so that the eye is looking slightly down and rotate the cluster down slightly to have it cover a bit of the eye. Key this position. Now we have the lid following the eye up and down with the controller.

For horizontal movement, reset the values of the controller and the cluster at "0." In the Set Driven Key dialogue box, select the left eye controller's Translate X and the cluster deformer's Rotate Y. Key this position.

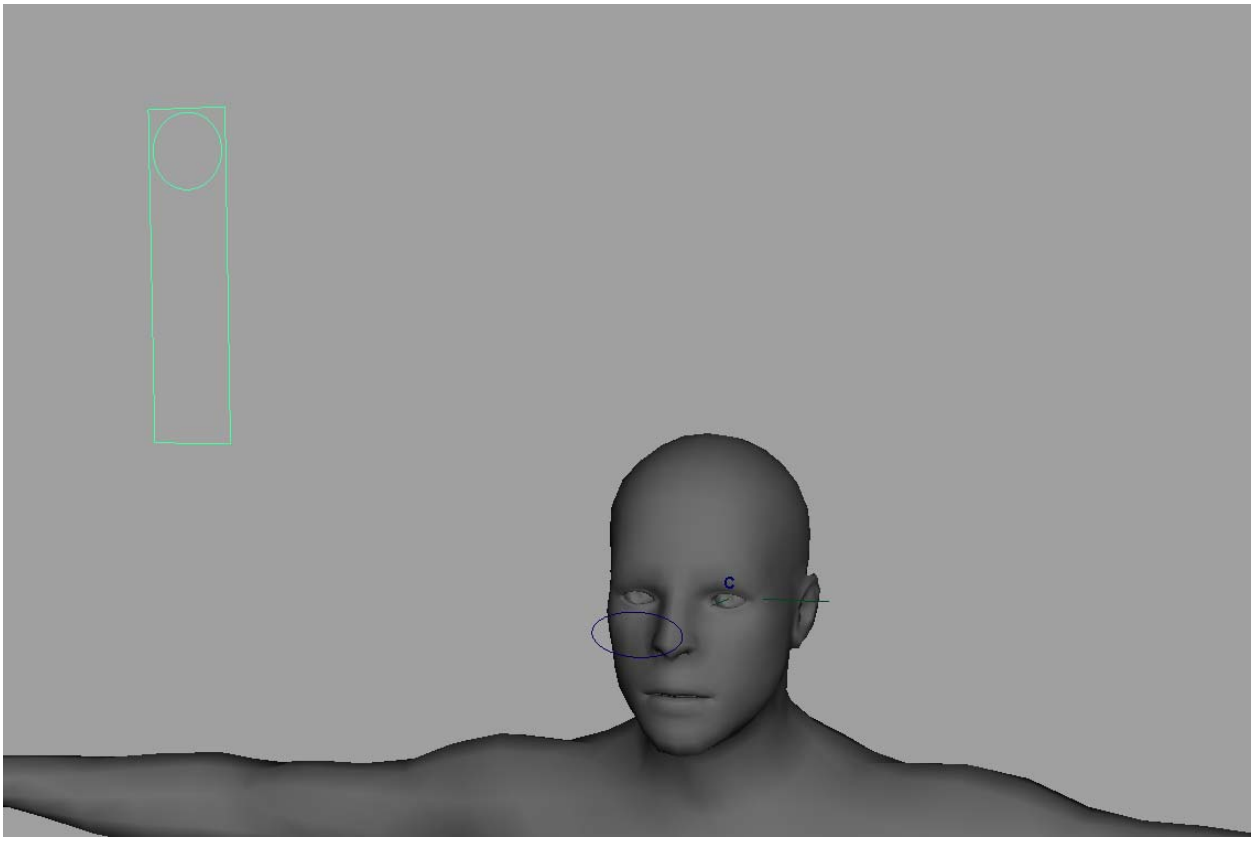


Just like the vertical movement, you will move the controller slightly to the left and right while rotating the cluster respectively, keying each new position. Once this is complete, the eyelid should track with the eyeball's rotation.

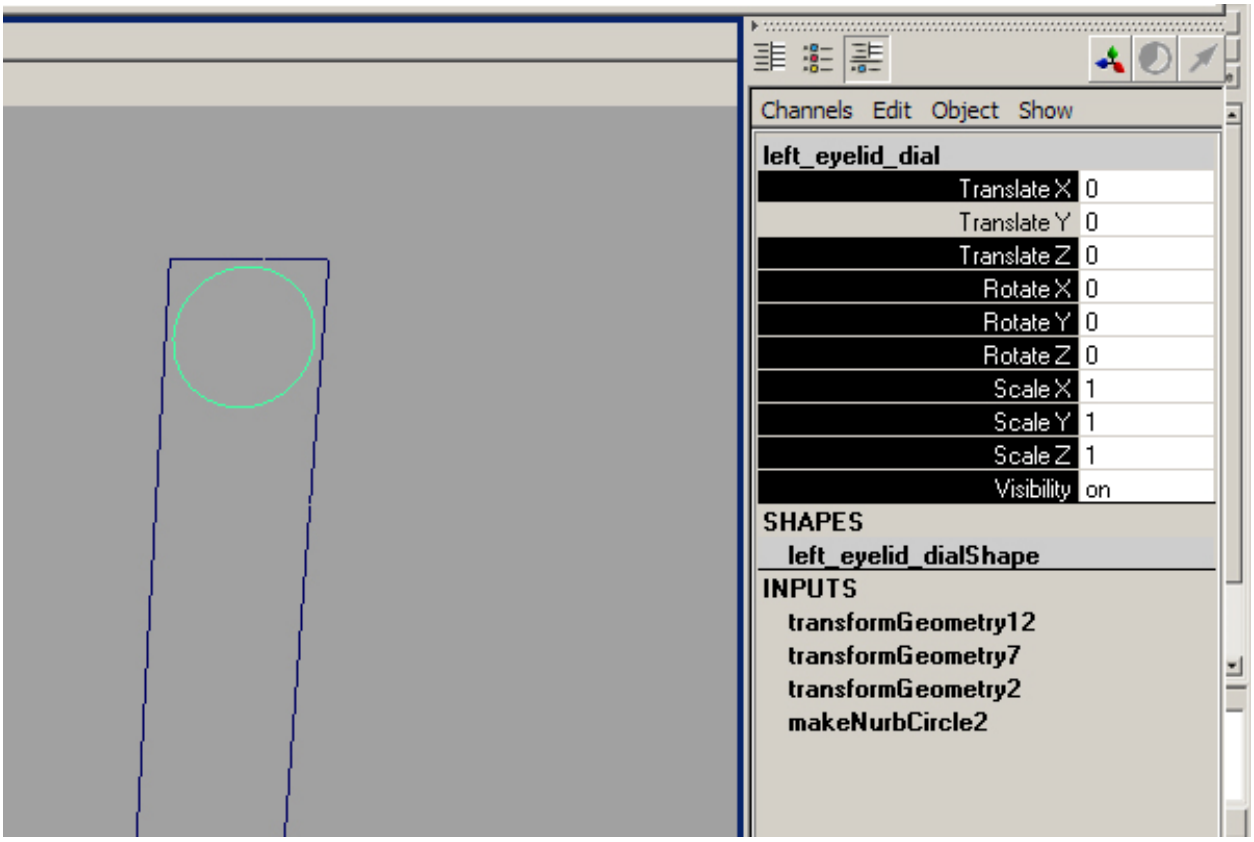
If the rotation is a bit too powerful, you can change this through the Graph Editor. Go to *Window > Animation Editor > Graph Editor* to edit the keys. Select the cluster deformer to find its keys in Editor. You should see a green line and a red line crossing over. Take both ends of both curves and hit the Flat Tangent button to flatten them out. This will allow the eyelid to gradually move into place while tracking the eyeball. By using the middle mouse button, you can also move points around to further improve the animation.



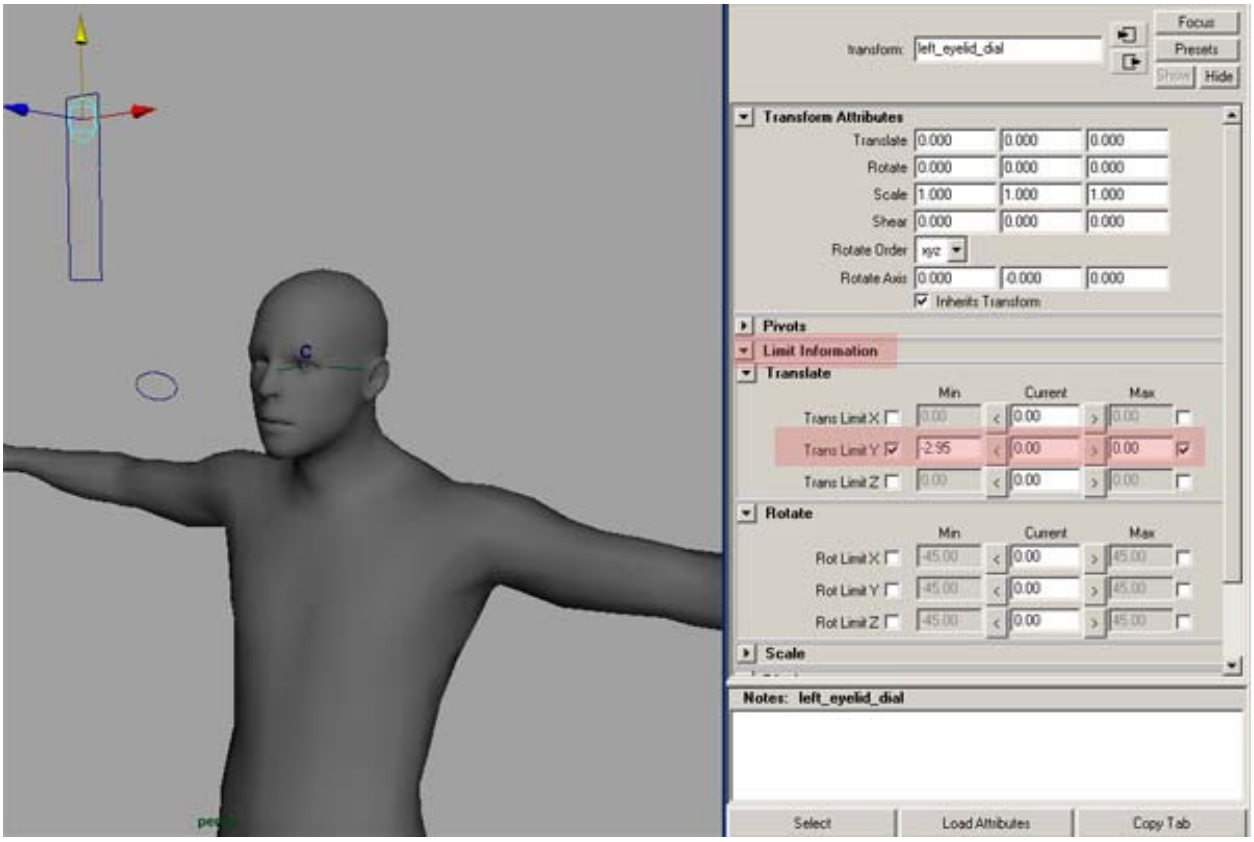
To manually control the eyelid with the eye moving, we'll need to create a control panel to control the eyelid's movement. Create a rectangle curve with a greater height and a circle curve that will serve as the dial for the controller. Name the rectangle left eyelid controller and the circle curve the left eyelid dial. In the Hypergraph Hierarchy, make sure the dial is in the same group as the left eyelid controller. Place these controllers next to the face and freeze the transformations (including the group). Make sure the dial is at the top inside the rectangle.



To make this work, we'll need to add some limitations to the dial's translates. Go to the Channel Box Editor and lock everything for the left eyelid dial except the Translate Y by right-clicking these attributes and selecting "Lock and Hide Selected."

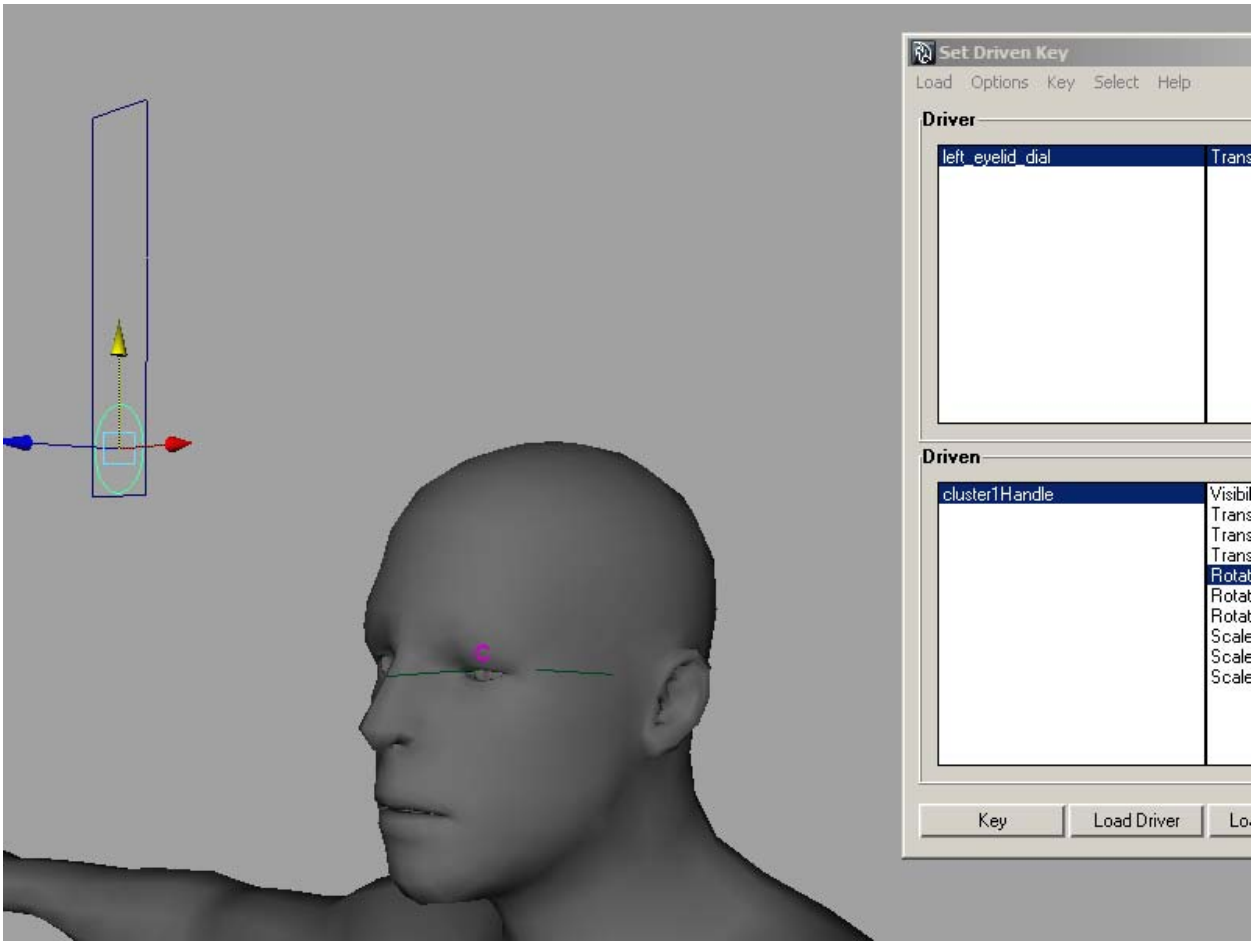


Go to the Attribute Editor expand the Limit Information option for the left eyelid dial. Find out how low the dial can go before touching the bottom of the rectangle controller. For my dial, it reaches the bottom at -2.95. In the limits, select the Trans Limit Y checkbox with the minimum values at the lowest the dial can go (for me, -2.95). Select the Max checkbox and change this to "0." Now the dial can only be moved up and down within the rectangle and not outside of it.



Next we will set driven keys so that this dial will control the eyelid's rotation. Open up the Set Driven Key dialogue box, select the dial and press the Load Driver button. Select the cluster deformer and press the Load Driven button. Make sure the dial's translate Y is at 0 and the cluster deformer's rotation X is at 0. Select the dial's Translate Y in the Driver box and the cluster deformer's Rotate X in the Driven box. Hit the Key button to set the key.

Next, select the dial and put its Translate Y at the lowest value (mines is -2.95). Select the cluster deformer and rotate it so that it’s halfway closed. Go back to the Set Driven Key dialogue box and key those same attributes. Now whenever you move the dial down, the eyelid will rotate as well.



This concludes the tutorial. Apply these techniques to do the lower eyelid and the right eye.