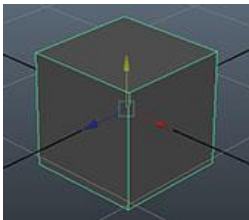## The Deep End

This chapter is going to throw you into the deep end of computer animation. We will take you through the process of building and rigging poly scorpion using Maya.

We are going to use very basic box modelling techniques to build a low poly scorpion. This will help you familiarise yourself with Maya and how it works. If you don't feel comfortable jumping straight in feel free to skip ahead to Chapter 1.
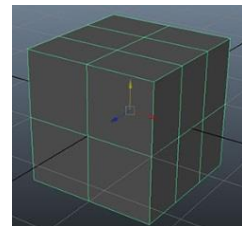
Modelling

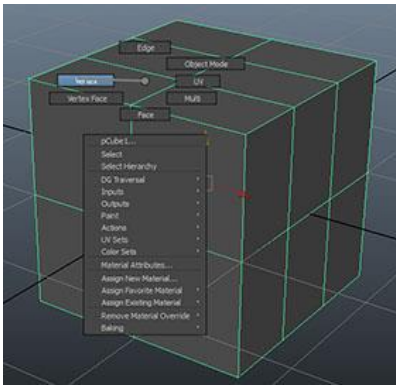We will begin by making the simplest of primitives; a cube. This will become the scorpion's body.

Create a cube in the centre of your scene (*Create > Polygon Primitives > Cube*). This is going to be the base for our scorpion.

So that we can give the body some shape we are going to increase the subdivisions in the cube input channel, we can also increase the size of the cube if we want.

To give the body some shape, hold and right click the cube and go into vertex mode.
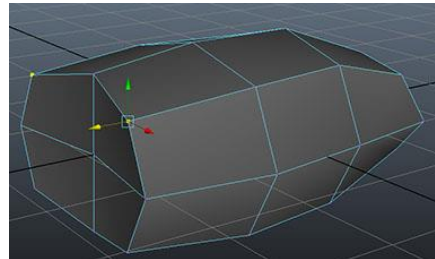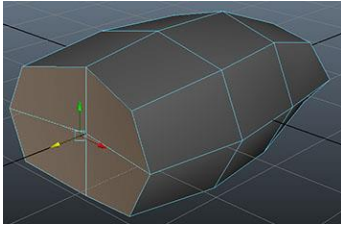
Chapter 0: Screw Conventions

In the *Tool Settings* panel, scroll to the bottom and enable reflection – this will ensure that as we modify the shape of the cube the other side will mirror our changes. It is important to maintain absolute symmetry throughout the Modelling process. If a-symmetry is required, we recommend you make these changes during the rigging process as an additional blend shape (which we will cover later).

Move the points to give your body some basic shape.

We are going to build the tail by extruding the rear four polys. Switch to *Face* mode and then select the four polygons. I have switched reflection back off, but you can leave it on if you would like.
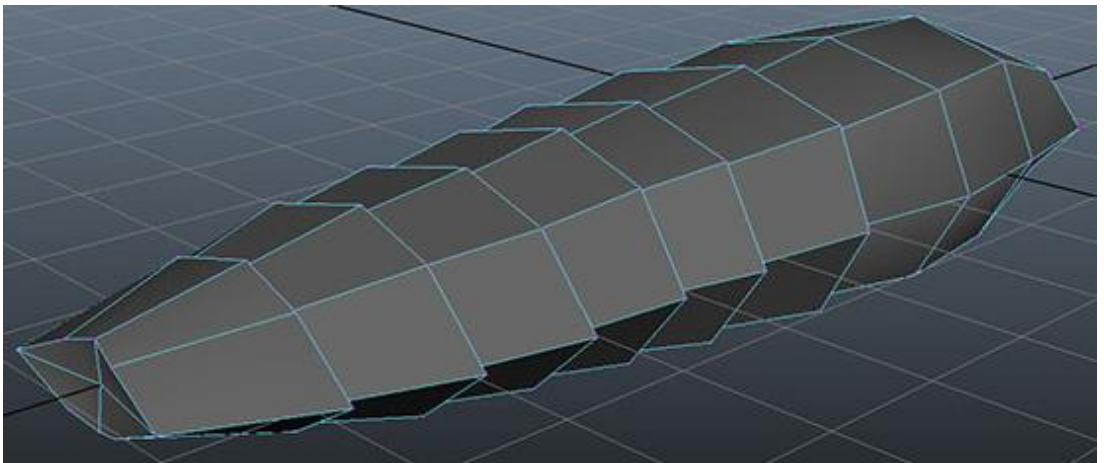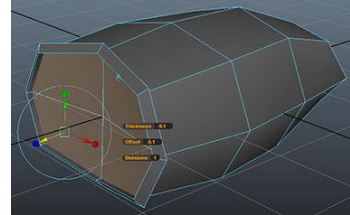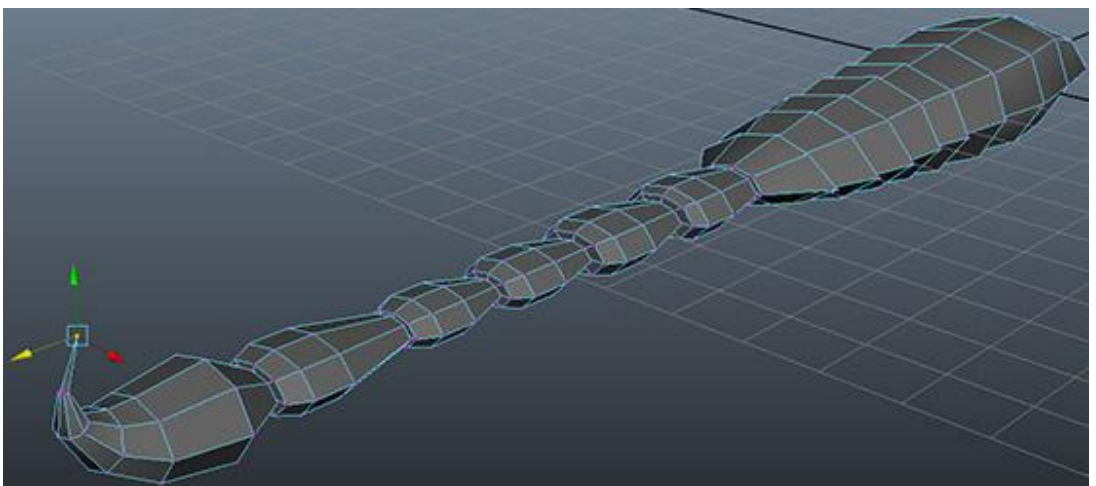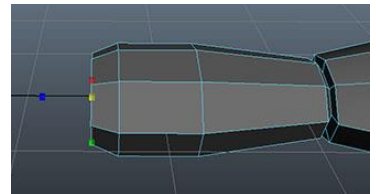
Chapter 0: Screw Conventions

Select the *extrude tool* and slide the *thickness* and *offset* values to *-0.1*. We are then going to repeat the extrude tool (press '*g*') a couple times

changing the thickness and offset values to give us our segmented tail shape. After the segments are created you may wish to

modify the shape in vertex mode.

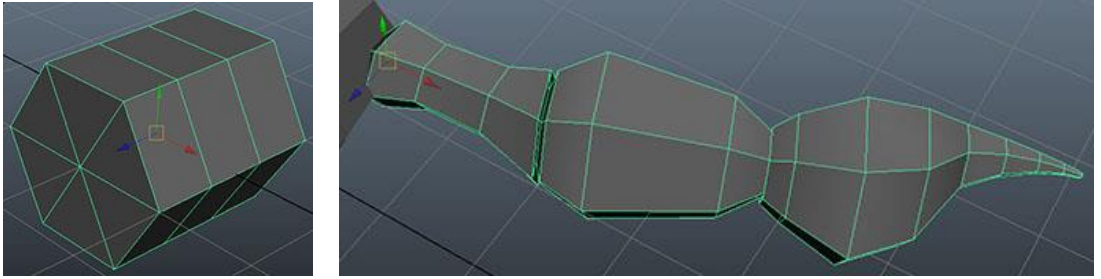When the flat segments are built, we will use the extrude tool again to create the rest of the tail but

instead of the thickness and offset values we will switch to the *move/scale* tools to modify the shape of the
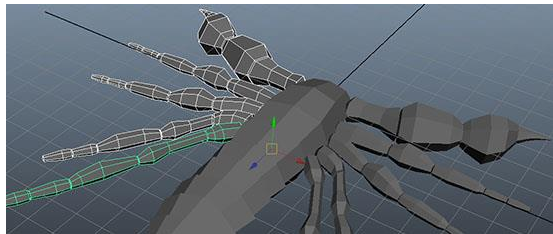
extruded part each time. We want a small gap in between the segments to allow them to bend when we rig and animate the character later. We then need to rotate the final extrusions to form the end spike.

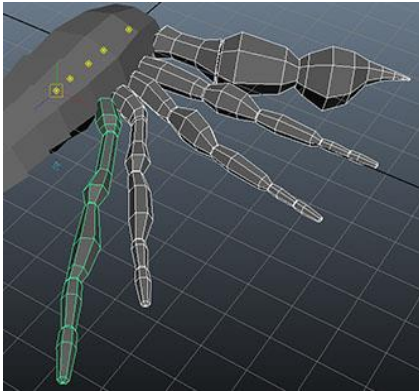With the main body complete, we can delete the construction history (*shift-alt-d*) and name the geometry.

For the legs we will use similar techniques but we will start off with a cylinder, changing the subdivisions in the input channel and rotating the x-axis by 90°.

Chapter 0: Screw Conventions

When you have finished building the legs, select all of them (and the pincers) and press '*insert*' to allow you to move the pivot. Hold down '*x*' to snap to the grid and move the pivot on the *X Axis* to the centre of the scene.
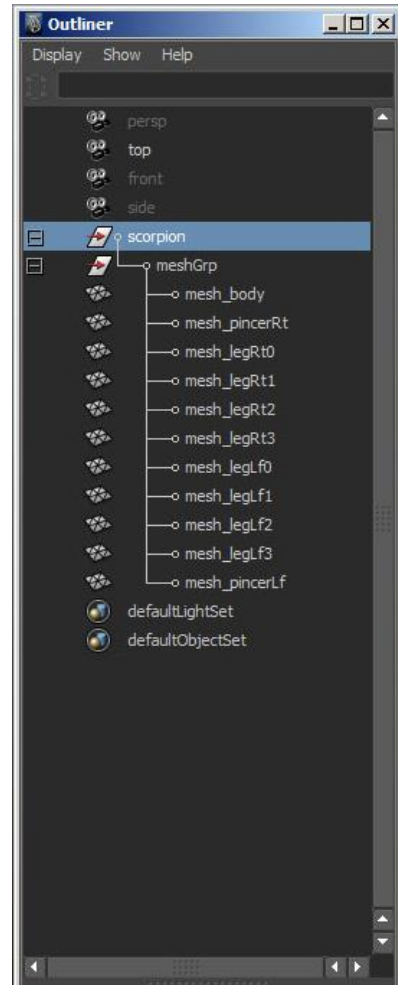
Duplicate the legs (*Ctrl-d*) and change the *scale X* value to *-1*. We would advise you to name all of your objects at this point.

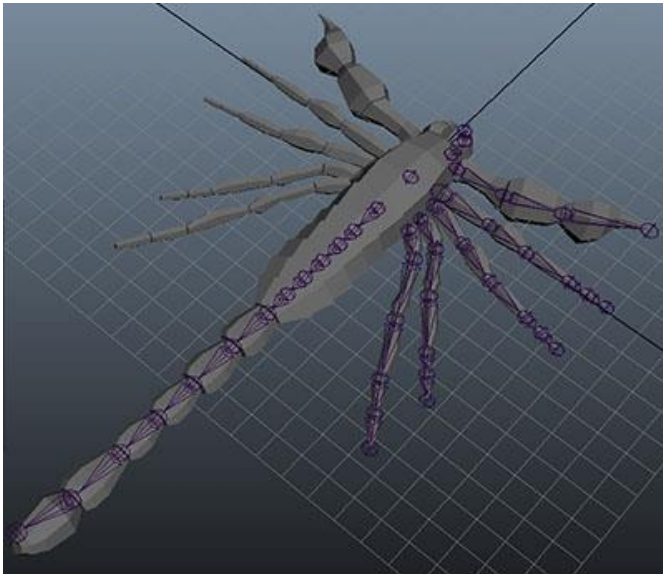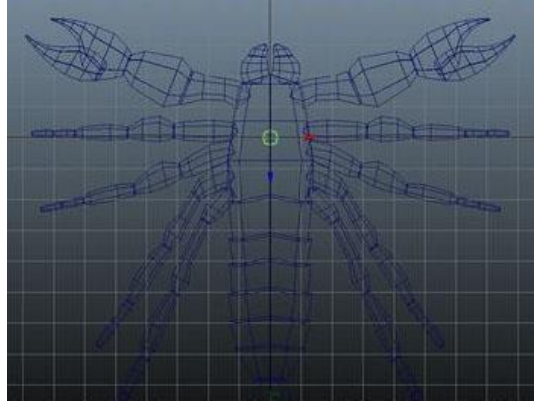Select all of the objects and delete any history (*Shift-Alt-d*). You can also freeze transformations at this stage – this will reset any *move/rotate/scale* values to zero while maintaining the default position. With all objects still selected press *Ctrl-g* to group them. Name the group '*mesh_grp*' or similar, then repeat and name the top group '*scorpion*' or similar, this will be useful for rigging later on.

Rigging

The first step of the rigging process is to create a skeleton to control our mesh objects. Using the *Skeleton > Joint* tool in the top view, create a joint in the centre of the body (this will be the root joint), then create a joint chain for each of the limbs and the tail. You want to place a joint at each point at which there is going to be a bend.

When the joint chains are all created, parent the limb chains to the root joint by selecting the joint chain, then the root and pressing '*p*'. If you want, you can also add the root joint to the current hierarchy in a '*ctrlGrp*' group or similar.

Chapter 0: Screw Conventions

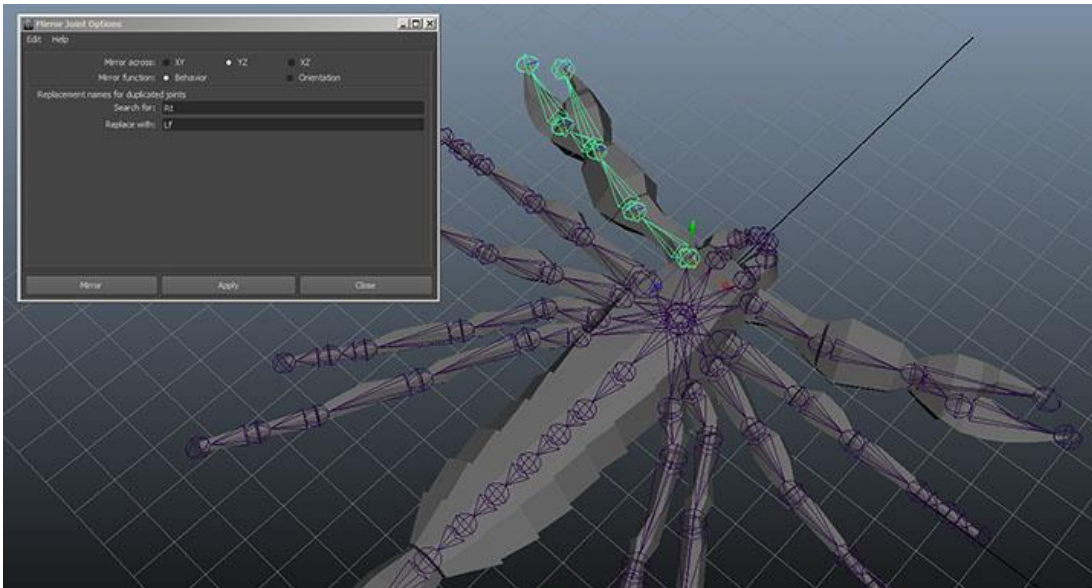We need to repeat the techniques we used for modelling and joint creation for the other point of the claw – to add a joint in line with an existing rig, after selecting the joint creation tool select the joint you want to create the new join from.

Chapter 0: Screw Conventions

We will mirror the right joints over to the left. In the *Mirror Joint* options box, ensure that *YZ* is selected. Apply this to each joint chain individually. If you have labelled your joints with '*right*' or '*Rt*' or similar, you can tell Maya to rename these modifiers and keep your naming conventions clean.

Joint Chains

There are four different types of joint chains within our scorpion model which we can use different techniques to set up. First up we have the FK control for the teeth, and a single control to rotate each joint as one to make it really simple to animate.

The leg chain is going to be a basic IK rig with a single IK Handle; all the movement will be controlled via a single foot control and the angle of the leg controlled by a leg pole vector control.
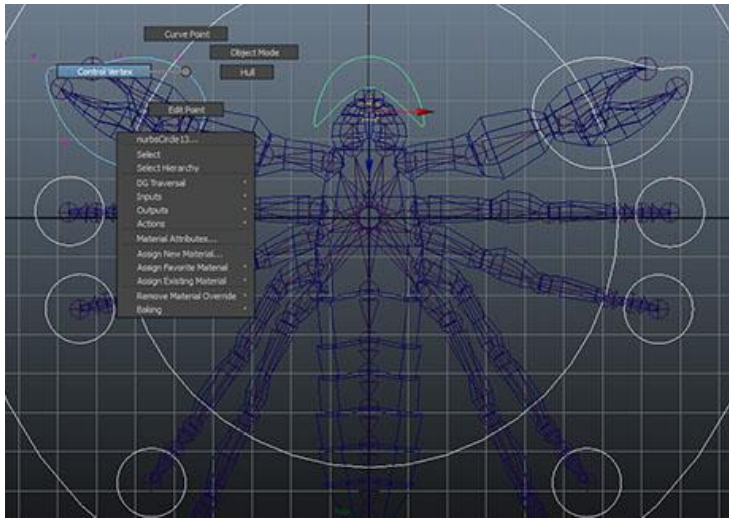
The tail will be slightly more complicated, we are going to set up a spline IK control. This will give the animator several controls to modify a curve that will define the tail shape. We will also set up custom attributes for tail twist.

Chapter 0: Screw Conventions

Finally, the claws are the most complicated. We will set them up with multiple IK handles and some FK control to give us the most flexibility. We will also add custom attributes for the pincers themselves.

These will be parented to a *global control*, which will govern the overall location of the scorpion.
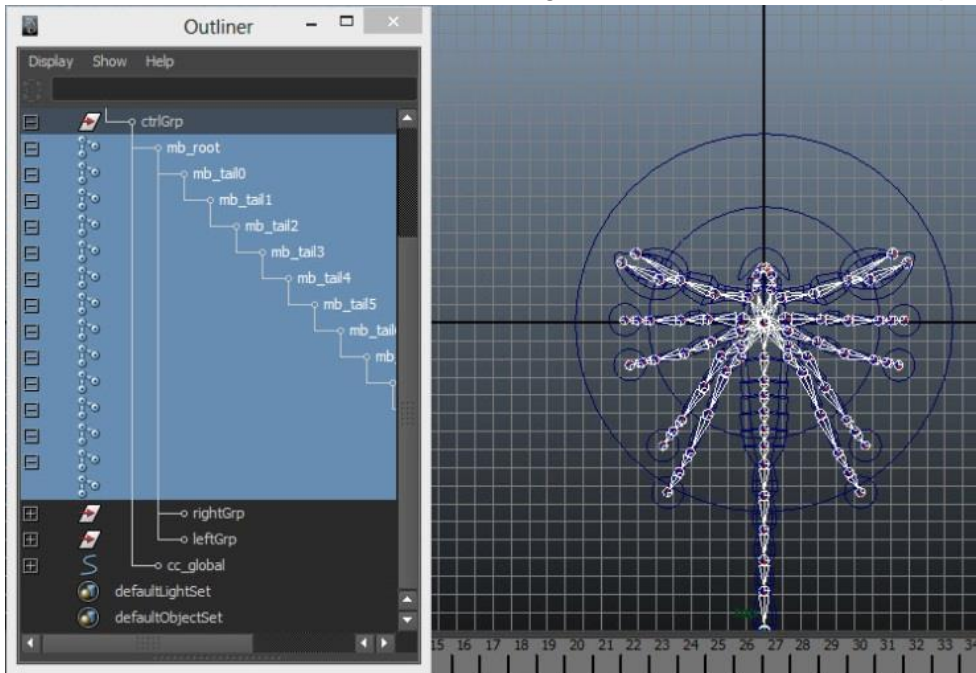
Using the *create > NURBS > circle*, we want to build a control layout in the top view. Our main points will be the global control, body, leg end and a control for the teeth. We also need a few to

control the tail, which we will rotate in the side view. You can modify the shape of the circles if you wish by going into '*Control Vertex*' mode.

Parent the legs, claws body and teeth controls under the global control. Select all of the legs, claws, the body, the teeth and either select the global control and press '*p*' or *MMB click and drag* them under the global control in the outliner. You will then want to parent the global control under the '*ctrlGrp*' or
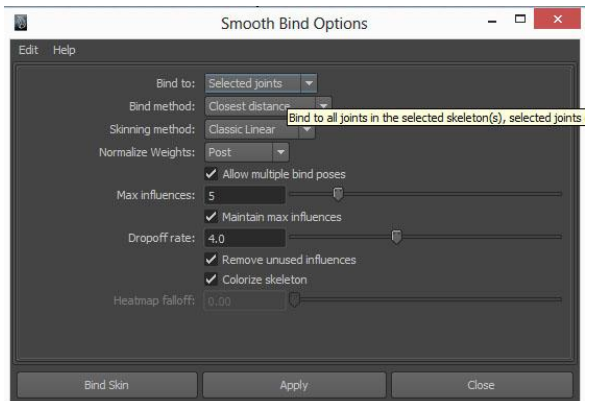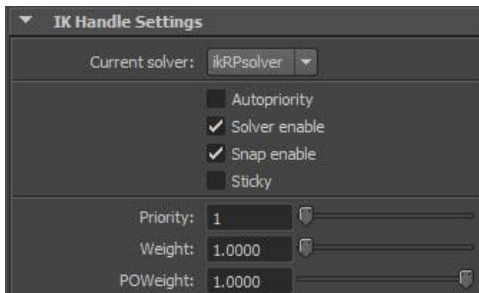
similar that we created earlier. We also want to group the entire join chain (from the *mb_root*) to the body control curve.

Chapter 0: Screw Conventions

Before we setup our IK handles (and while the joints are still in the exact place we want) we are going to skin our scorpion mesh to the joints. Select the root joint and all of the tail joints, then shift select the body/tail mesh. *Go to the Skin > Bind Skin > Smooth Bind □* (you may need to make sure you are in the animation menu set by pressing '*F2*'). Make sure that the operation is set to use selected joints and that normalise joints is set to be interactive. Bind the skin.
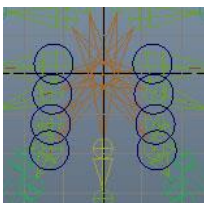

Repeat this operation for the legs, teeth and claws. You will want to select the root and the joints that will control the mesh in question.
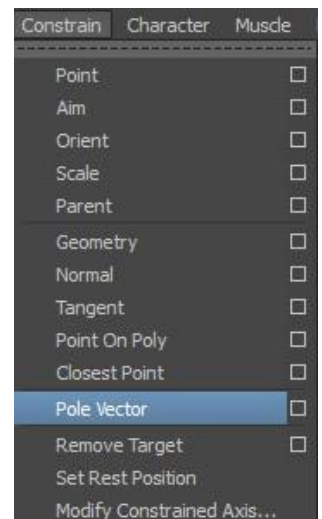

For the legs, open *Animation > Skeleton > IK Handle* tool options and make sure that the current solver is set to *ikRPsolver,* we need this to make sure that we can define the bend angle later on. With the tool selected, select the first and last joint in the leg chain. Using '*g*' repeat this process for all of the legs. This will create *ikHandle* nodes in the root of the world, which we are going to parent to the control curve of each respective leg.

To control the angle of the IK bend, we must create control curves and constrain them as pole vectors. In the top view, create some small circles around the start of each leg and then move them up in the side view. Parent these to the 'body' control curve. This will keep the legs
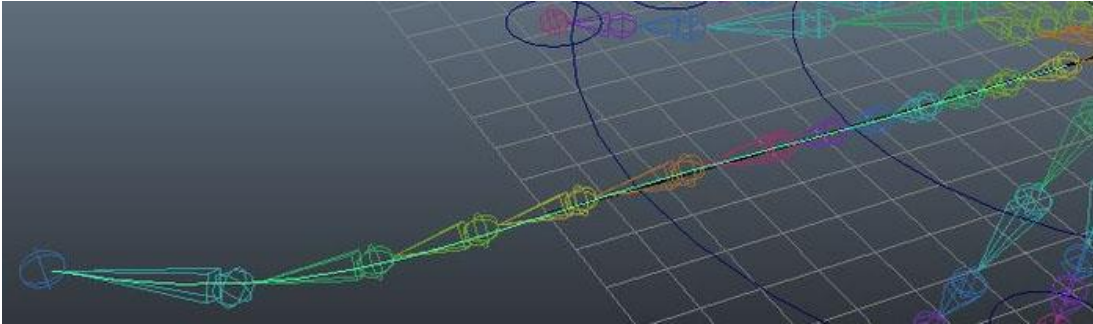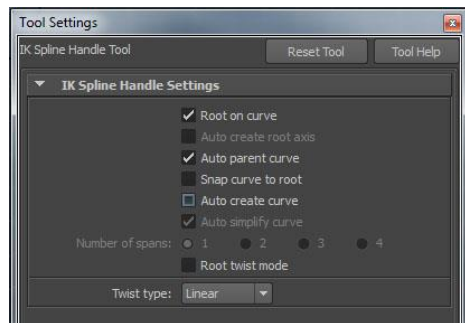
pointing inwards while giving us full control over the angle if we want to change it. Rename the circles *pv_legRt1* or similar to keep your scene clean.

To assign these pole vectors, select the control and then shift select the IK Handel and go to *Constrain > Pole Vector*. Your legs may flip by 90° at this stage, if they do, don't worry!
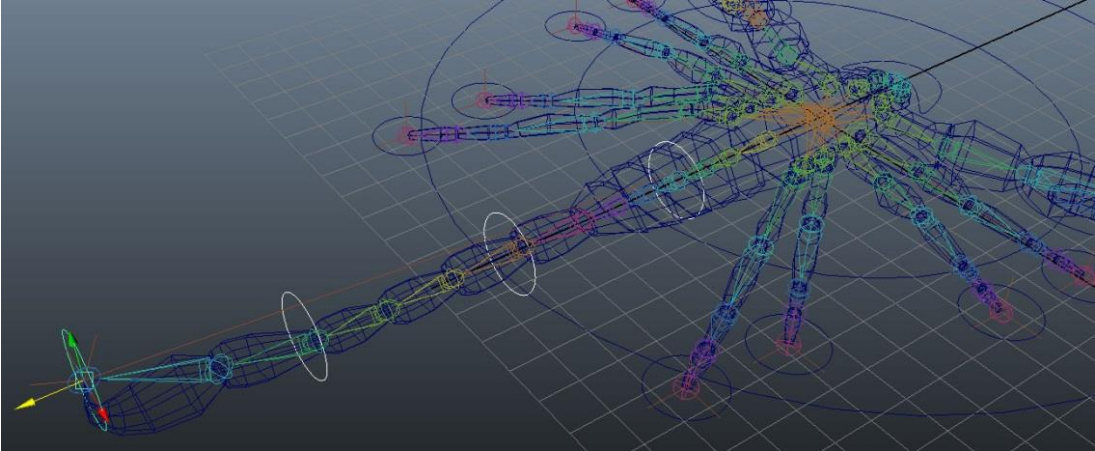
To rig the tail, first we are going to build the curve that is going to define the shape. The easiest way to do this is to use the create *cv curve tool* and snap each point to the centre of each of the tail joints (hold '*v*' for vertex snapping) you may get more accurate results if you also hide the polygons from the show menu. Name this curve '*sp_tail*' or similar.

In the *IK Spline Handle Tool*, deselect the auto create curve. Now select the first and last tail joint, then the tail curve that we just built. The tool will build an *ikHandle* and the joints will conform to the shape of the curve (you can test this by selecting and moving the curve vertices).
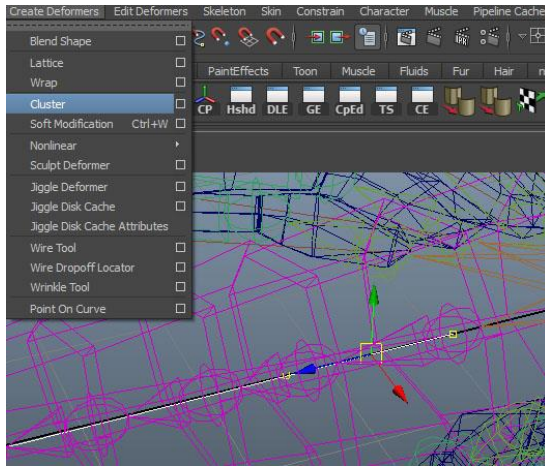
Chapter 0: Screw Conventions

To control the shape easily, we will create some curves along the length of the tail. Create a circle in the front view, and then duplicate it about three or four times in the top view spreading them across the length of the tail. When you are finished, make sure you *delete any history* and *freeze the*

transformations.
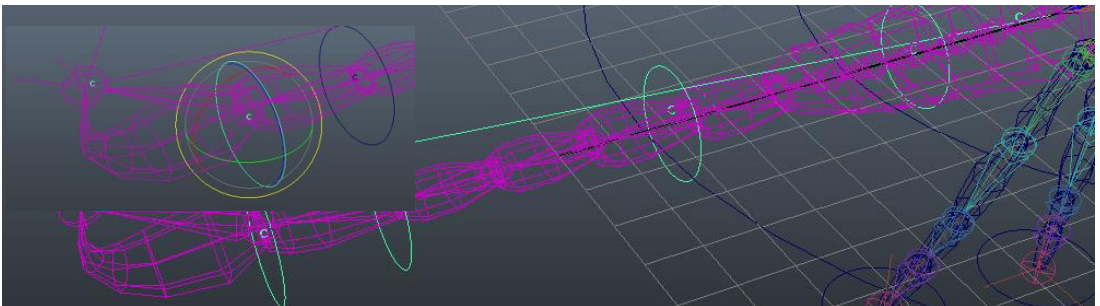
Chapter 0: Screw Conventions

So that the tail is defined by the same number of vertices as we have controls,

select the curve and go to *Edit Curves > Rebuild Curve*. Decide how many controls you would like and place that number minus three as the Number of Spans. In vertex mode, select the first two vertices (at the base of the tail) and create a cluster. Repeat this process on each of the rest of the clusters individually. You should end up one extra cluster as controls.
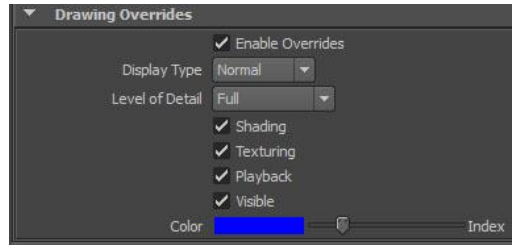
Parent the clusters too their respective controls, parenting the cluster at the base to the body control. If you wanted to control the rotation of the stinger by rotation rather than translation, you could parent the last two clusters to a single curve located at the stinger base.
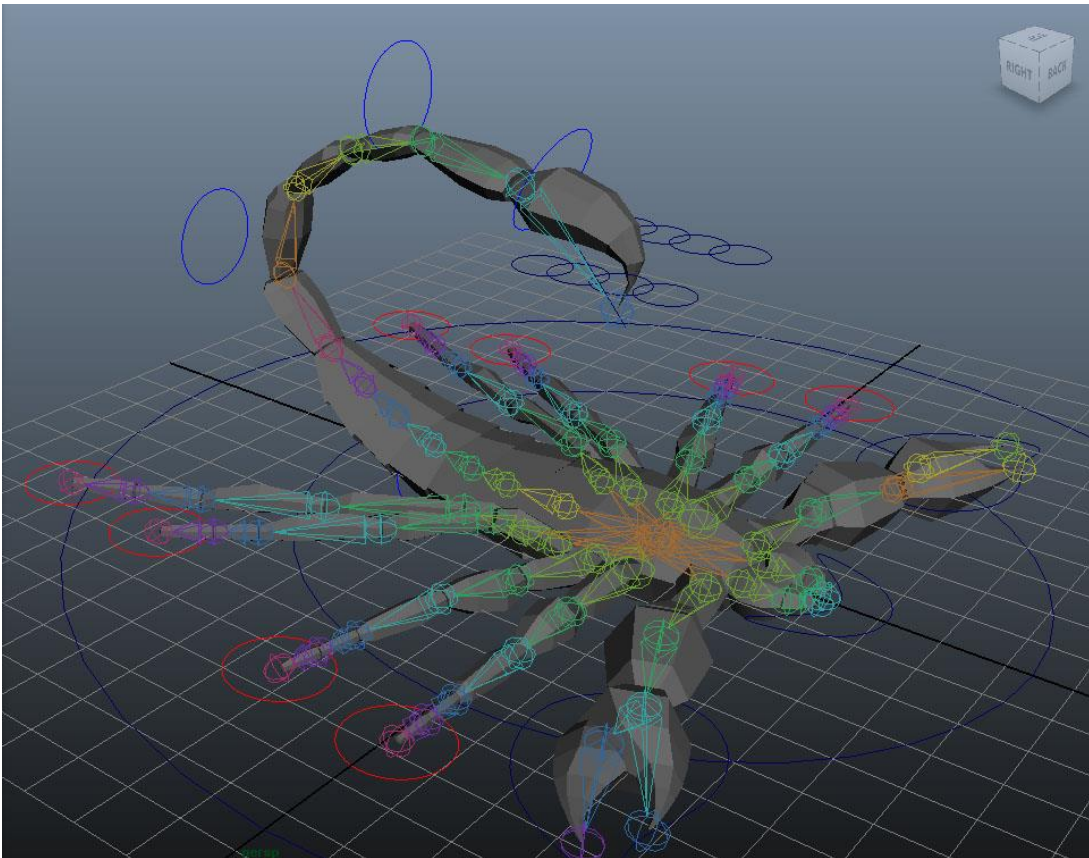

So that we don't get double transformation, make sure that the tail spline curve (*sp_tail*) is outside of the global control, I would recommend an '*extras*' group immediately below the overall scorpion group.

Chapter 0: Screw Conventions

We can start cleaning the tail up now, hide (*ctrl-h*) the clusters and *ikHandle* of the tail. When the rig is finished, we want to leave only the control system visible and selectable. Select the tail group and open the attributes editor. In the display tab, if you like you can move the tail into a more natural position.
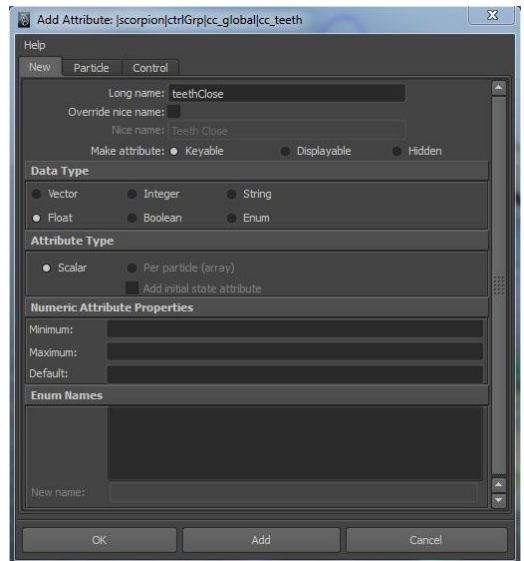
We will use the same technique to hide the *ikHandles* of the feet we created earlier and make them a different colour; you can group all of the feet in a new group (*ctrl-g*).
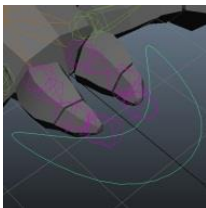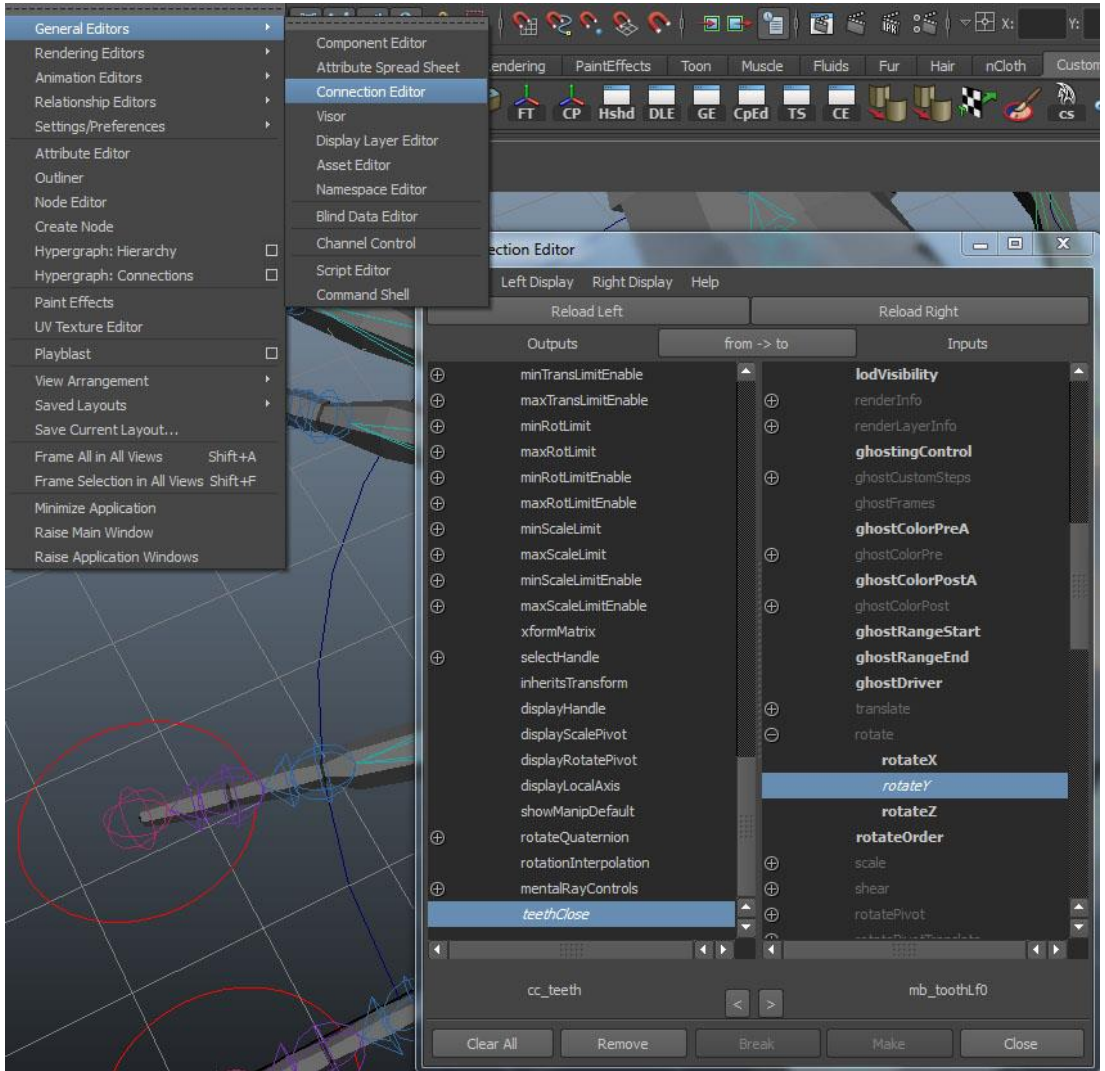
Chapter 0: Screw Conventions

We will now tackle the mouth and what is a very basic controller for the teeth. Select the teeth control and in the Channel Box's edit menu add a new attribute called *teethClose*.

We are going to connect this to the teeth joints' rotate values. Open the connection editor, the teeth control should be loaded into the left side, but if it is not select the control and click 'reload left'. Select one of the teeth joints and click reload right. We want to connect the *teethClose* value to the *rotate value*.
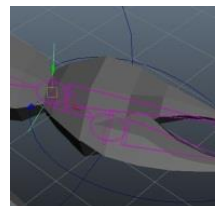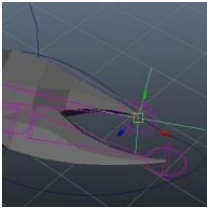
Use the above technique to connect the rest of the teeth joints (*right* and *left*) to the same control. You can test the control by changing the *teethClose* value and both teeth should now open and close together.

It is now time to tackle the claw. First we will build the IK handles that will govern the limb. As you did with the legs, create an ikHandle between the first joint and the base of the claw.
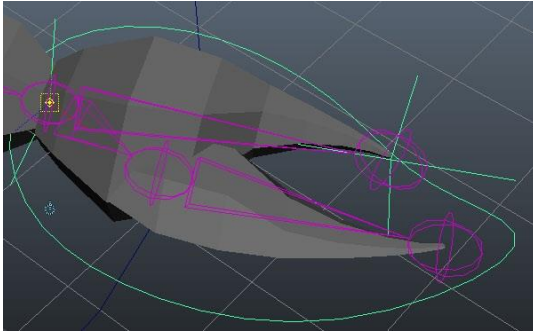
Chapter 0: Screw Conventions

 Create another one from this joint to the end of the claw (make sure you connect the end of the attached claw, **NOT** the pincer).
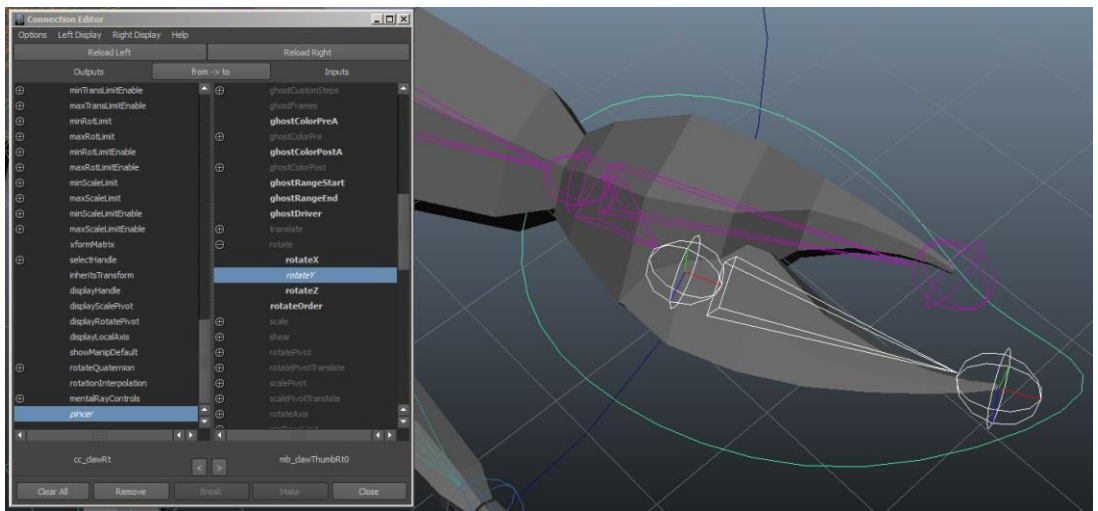
Group both of these handles to the claw control. This will mean that moving the claw will set the arm bend using IK, but rotating it will still affect the claw end like FK.

So that the claw rotates around the correct pivot, press *insert* to enter pivot mode and holding '*v*' to point snap, move the claw control's pivot to the centre of the claw joint.

We still need control over the angle of the claw's bend, so create a circle (change the shape if you wish) to be used as a pole vector. Make sure that you freeze any transformations and delete any history on the curve.

To finish off the claw, we are going to create a new attribute to control the pincers. Like we did before with the teeth, add a new attribute called '*pincer*' or similar and using the connection editor again, connect the new pincer attribute to the joint rotate attribute.
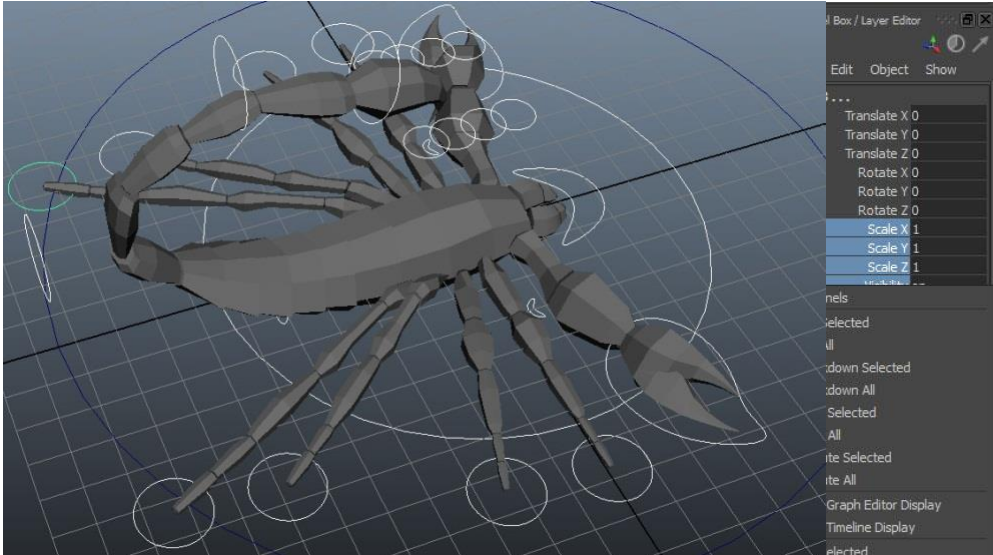
Chapter 0: Screw Conventions

Repeat these processes on our left claw. We can also take this opportunity to clean up some of our rig, making sure that our items are named correctly and appropriately laid out in our outliner. Remember that we can colour code groups of controls to make them easier to see, and hide things like *ikHandles* and joints that we do not need to see anymore.
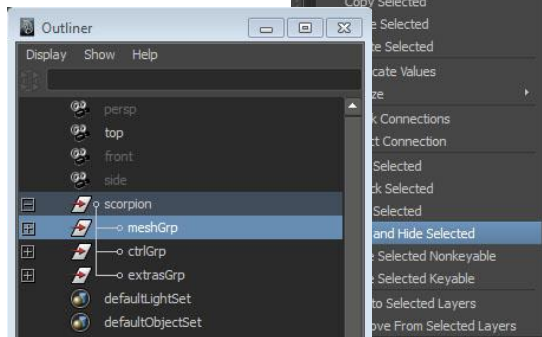
Chapter 0: Screw Conventions

Another way we can clean the rig is to hide attributes that we do not need; for example we do not need the scale attributes on any control other than the global. Select all controls other than the global, drag select the scale (we can
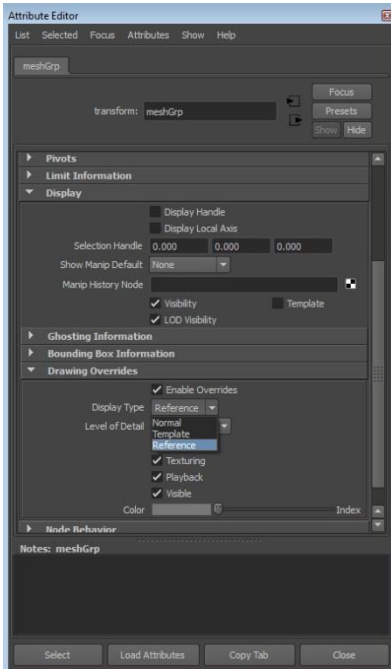

also add in the visibility attribute) and holding the right mouse choose '*lock and hide selected*'.

If we go around the rig, we will find a lot of other attributes we do not need. The rotation on IK and pole vector controls for example (legs and tail) and all attributes on the mouth control apart from our custom controls.

Our scorpion asset is nearly complete; all we need to do now is make the mesh group un-selectable to make it easier to select our control curves. Select the mesh group from the outliner and similar to how we changed the colour of our curves, after enabling display overrides change the *Display Type* to *Reference*. This will keep the mesh visible and renderable, but not selectable.

Animation

Now that the asset is complete, it is ready to be used for animation.

Note: *When we are setting our scene up for animation, we want to reference as much (if not all) of our assets and environment as possible. This means our characters, props and even the environment and camera systems themselves where possible.*

Keeping everything referenced ensures that if we have multiple shots using the same environment or the same characters, they are all using the exact same version.