Screw Conventions

Contents

Contents

Contents

Contents

CHAPTER 0: SCREW CONVENTIONS

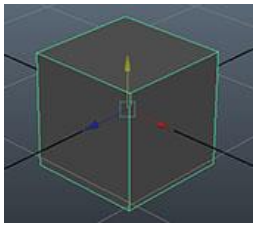
The Deep End

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.

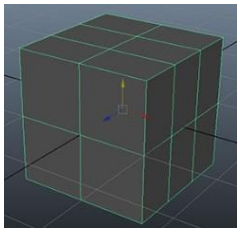
We are going to use very basic box modelling techniques to build a low poly scorpion. This will help you familiarise yourself with Maya and how it works. If you don't feel comfortable jumping straight in feel free to skip ahead to Chapter 1.

Modelling

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

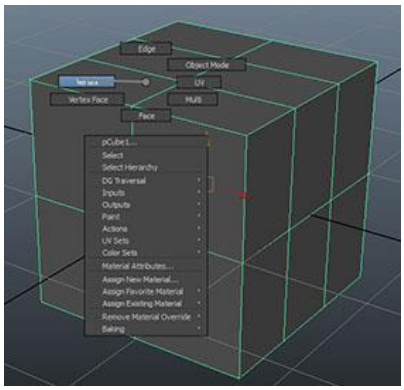


Create a cube in the centre of your scene (*Create > Polygon Primitives > Cube*). This is going to be the base for our scorpion.



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.

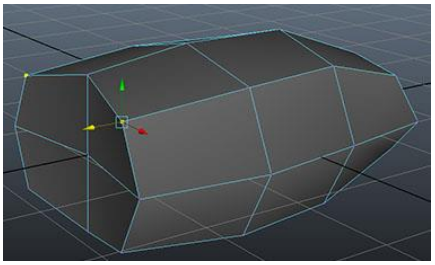
To give the body some shape, hold and right click the cube and go into vertex mode.

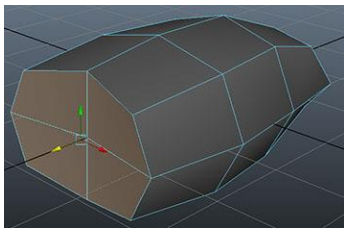


In the *Tool Settings* panel, scroll to the bottom and enable reflection – this will ensure that as we modify the shape of the cube the other side will mirror our changes. It is important to maintain absolute symmetry throughout the Modelling process. If a-symmetry is required, we recommend you make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

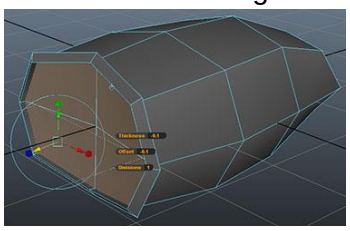
Move the points to give your body some basic shape.

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.

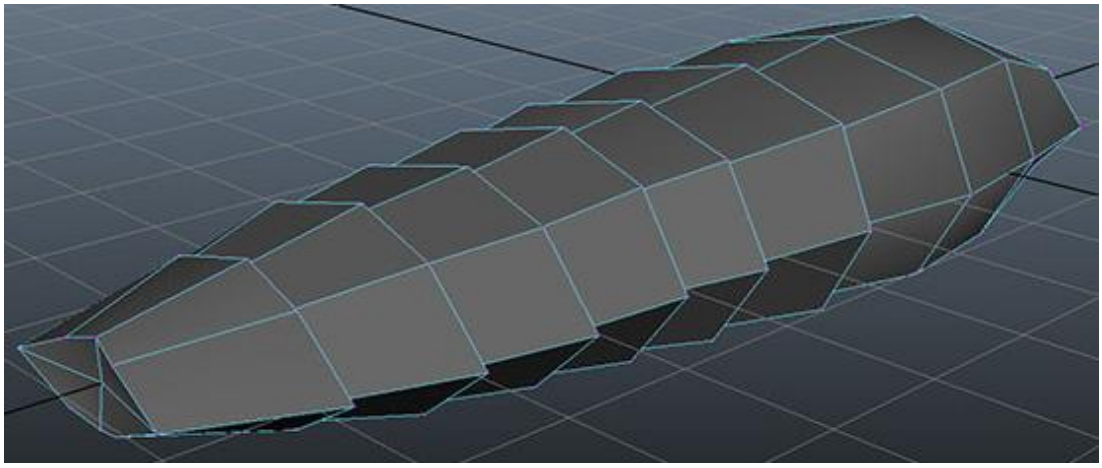




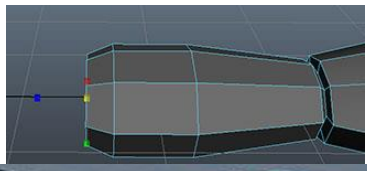
Select the *extrude tool* and slide the *thickness* and *offset* values to *-0.1*. We are then going to repeat the extrude tool (press '*g*') a couple times changing the thickness and offset values to give us our segmented tail shape. After the segments are created you may wish to modify the shape in vertex mode.

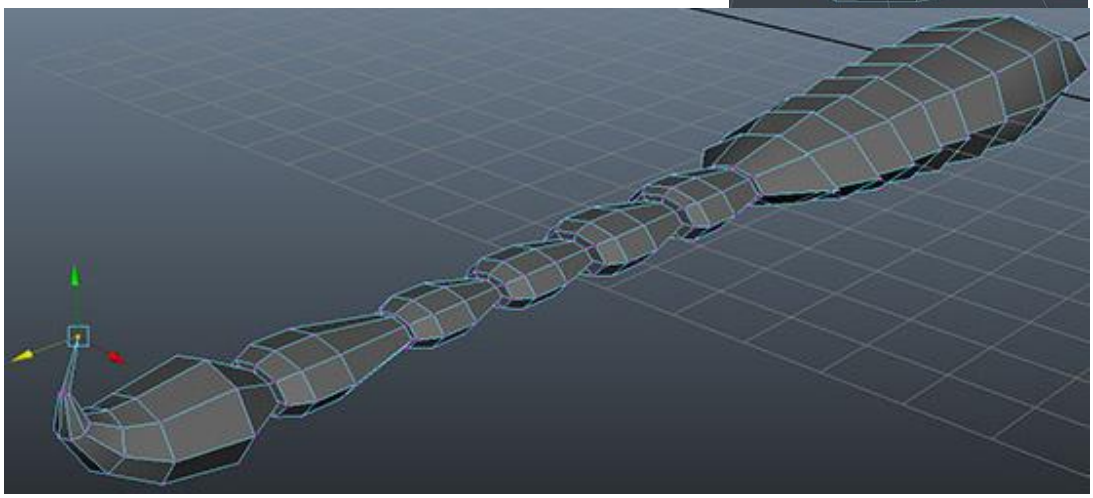


When the flat segments are built, we will use the extrude tool again to



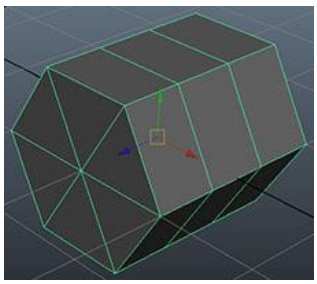
create the rest of the tail but instead of the thickness offset values we will switch to the *move/scale* tools to modify the shape of the and

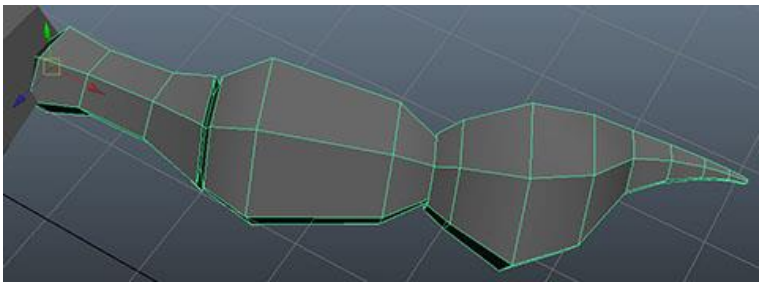




extruded part each time. We want a small gap in between the segments to allow them to bend when we rig and animate the character later. We then need to rotate the final extrusions to form the end spike.

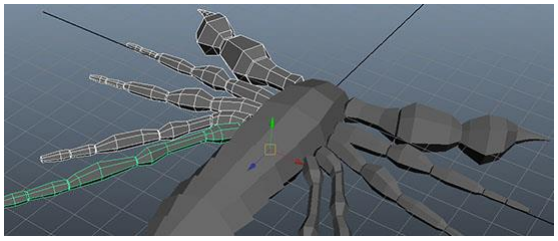
Chapter 0: Screw Conventions





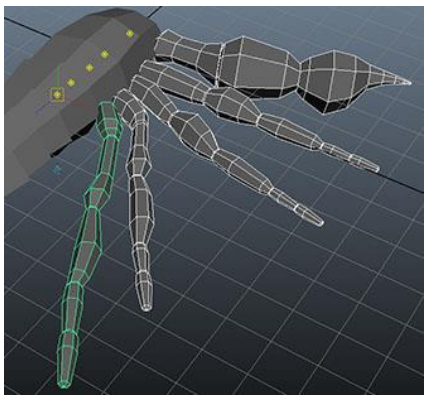
With the main body complete, we can delete the construction history (*shift-alt-d*) and name the geometry.

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°.



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.

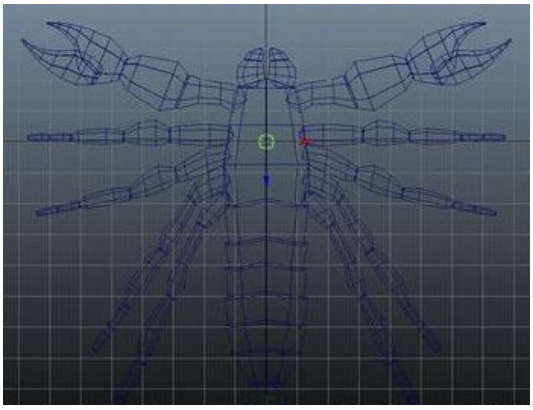
Duplicate the legs (*Ctrl-d*) and change the *scale X* value to *-1*. We would advise you to name all of your objects at this point.

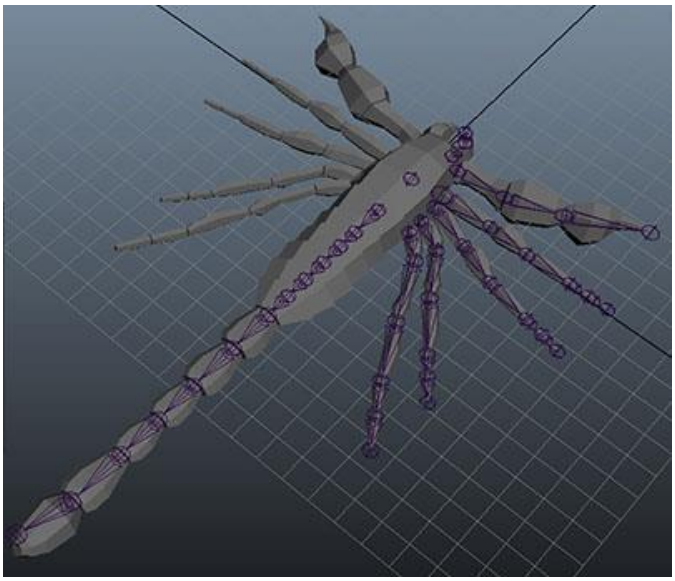


Select all of the objects and delete any history (*Shift-Alt-d).* You can also freeze transformations at this stage – this will reset any *move/rotate/scale* values to zero while maintaining the default position. With all objects still selected press *Ctrl-g* to group them. Name the group '*mesh_grp*'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

Rigging

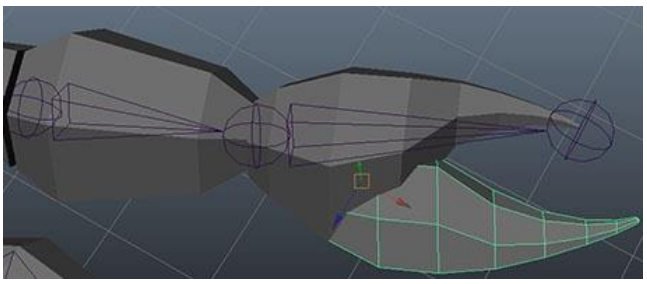
The first step of the rigging process is to create a skeleton to control our mesh objects. Using the *Skeleton* > *Joint* tool in top view, create a joint in the centre of the body (this will be the root joint), then create a joint chain for each of the limbs and the tail. You want to place a joint at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

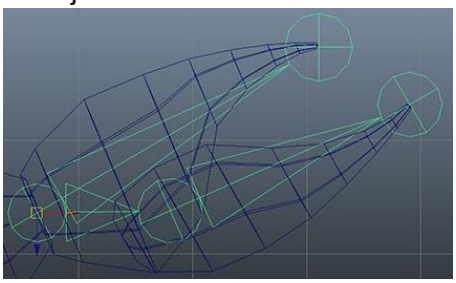




When the joint chains are all created, parent the limb chains to the root joint by selecting the joint chain, then the root and pressing '*p*'. If you want, you can also add the root joint to the current hierarchy in a '*ctrlGrp*' group or similar.

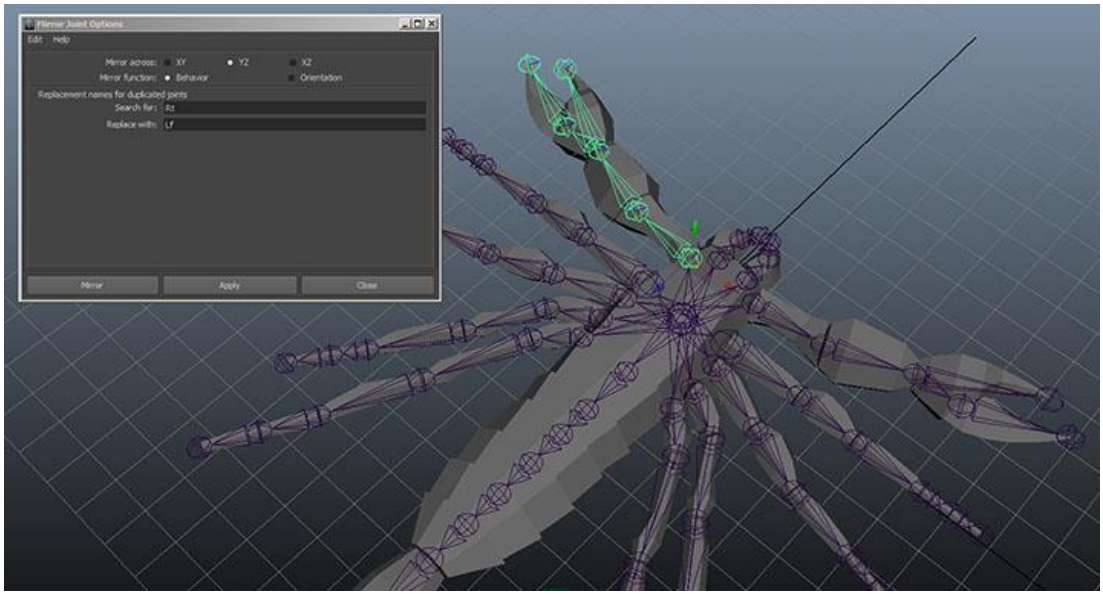
We need to repeat the techniques we used for modelling and joint creation for the other point of the claw – to add a joint in line with an existing rig, after selecting the joint creation tool select the joint you want to create the new join from.





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

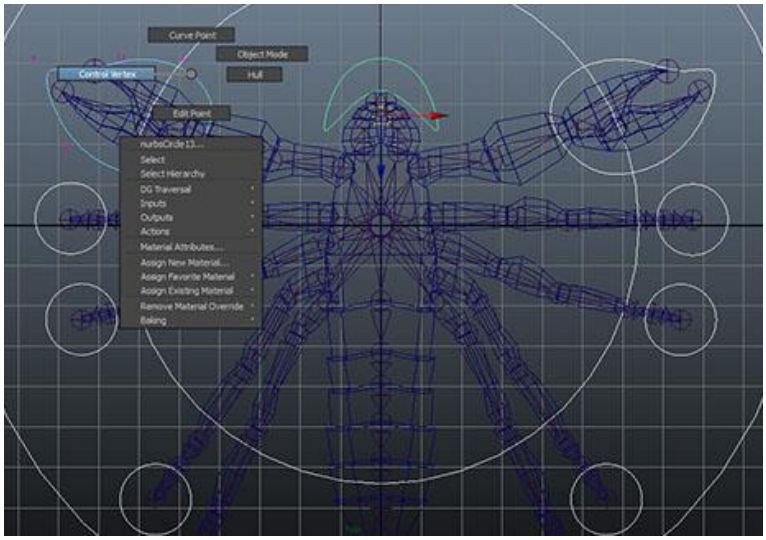
There are four different types of joint chains within our scorpion model which we can use different techniques to set up. First up we have the FK control for the teeth, and a single control to rotate each joint as one to make it really simple to animate.

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.

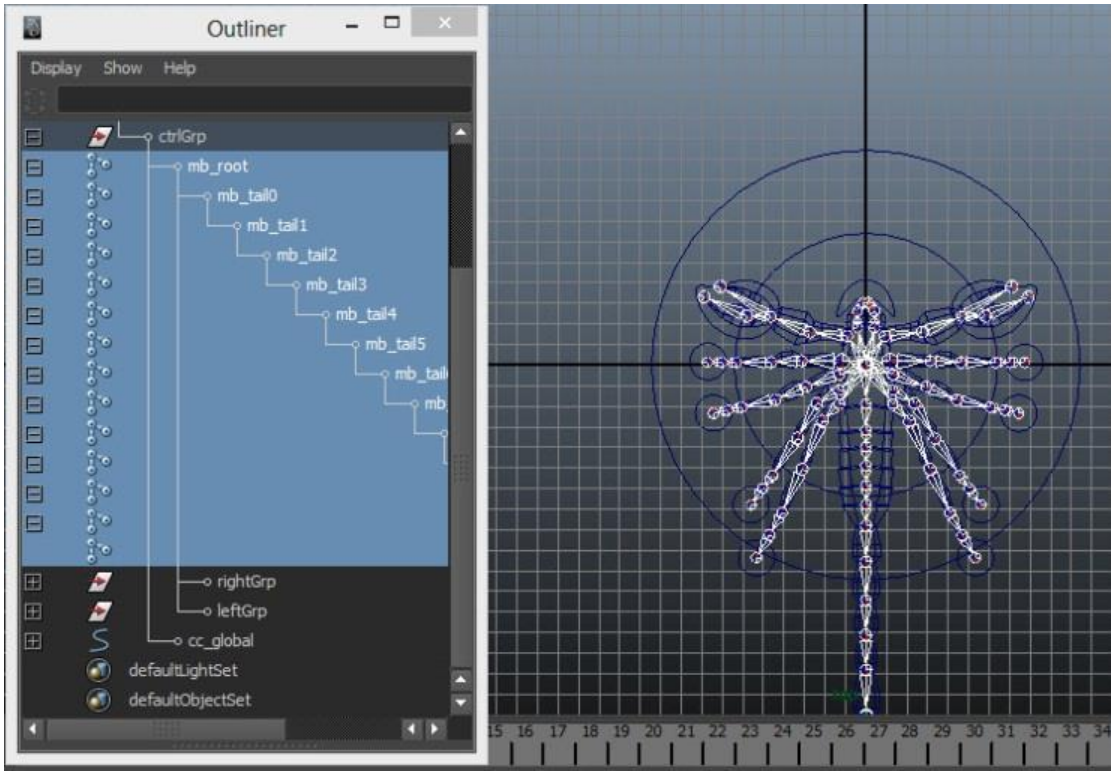
Finally, the claws are the most complicated. We will set them up with multiple IK handles and some FK control to give us the most flexibility. We will also add custom attributes for the pincers themselves.

These will be parented to a *global control*, which will govern the overall location of the scorpion.



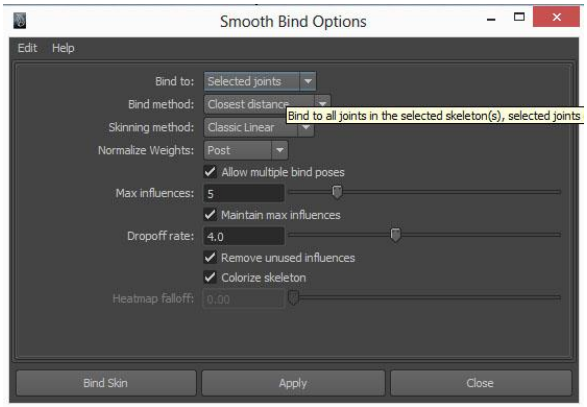
Using the *create* > *NURBS* > *circle*, we want to build a control layout in the top view. Our main points will be the global control, body, leg end and a control for the teeth. We also need a few to control the tail, which we will rotate in the side view. You can modify the shape of the circles if 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.

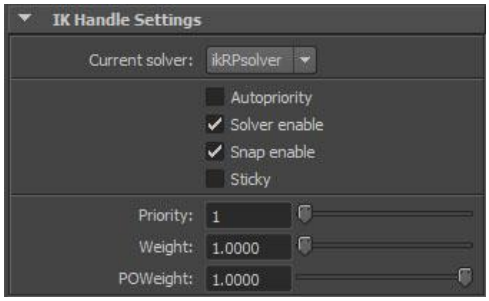


Chapter 0: Screw Conventions

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 is set to be interactive. Bind the skin.

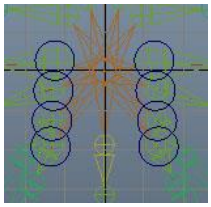


Repeat this operation for the legs, teeth and claws. You will want to select the root and the joints that will control the mesh in question.

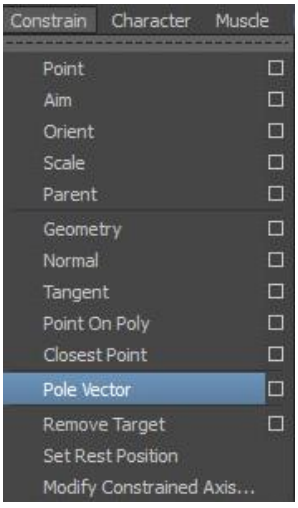


For the legs, open *Animation > Skeleton > IK Handle* tool options and make sure that the current solver is set to *ikRPsolver,* we need this to make sure that we can define the bend angle later on. With the tool selected, select the first and last joint in the leg chain. Using '*g*' repeat this process 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.

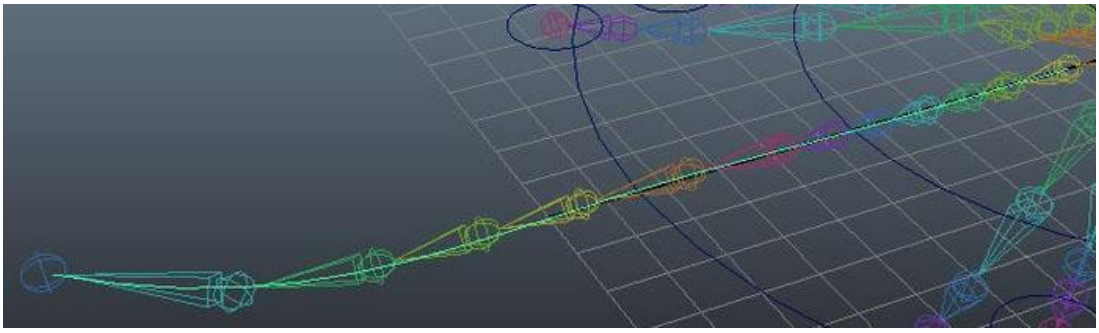


To assign these pole vectors, select the control and then shift select the IK Handel and go to *Constrain > Pole Vector*. Your legs may flip by 90° at this stage, if they do, don't worry!

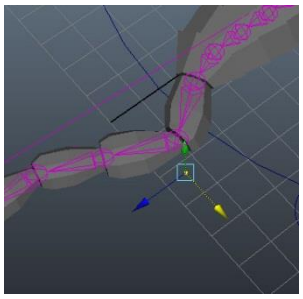


Chapter 0: Screw Conventions

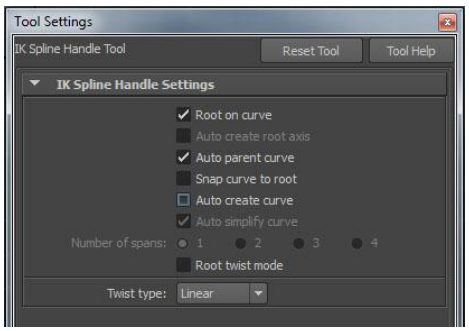
To rig the tail, first we are



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.

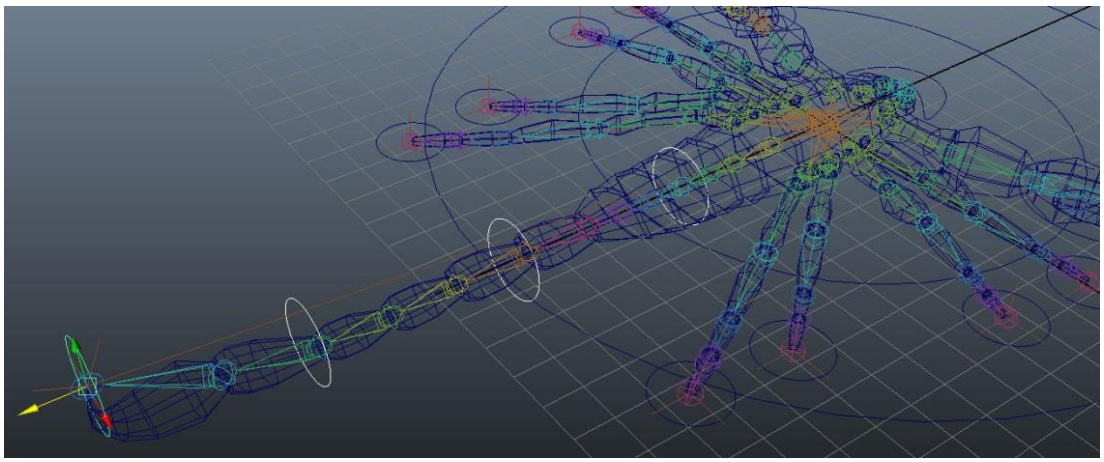


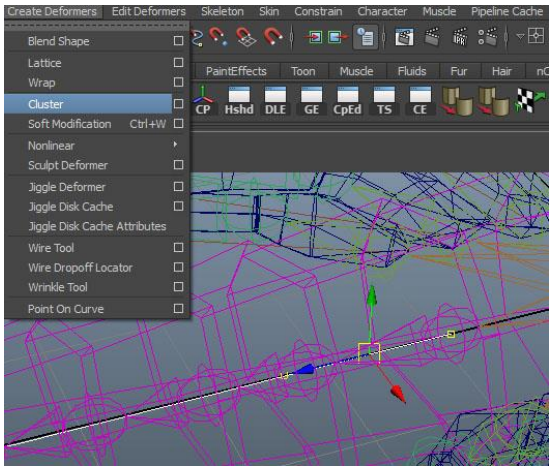
In the *IK Spline Handle Tool []*, tool settings; deselect the create curve. Now select the first and last joint, then the tail that we just built. The tool will build an *ikHandle* and the joints will conform to the shape of the curve (you can test this by selecting and moving the curve vertices).



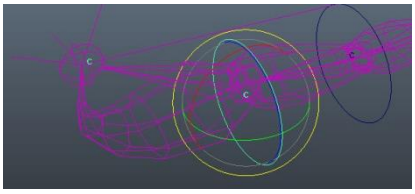
auto tail curve

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*.

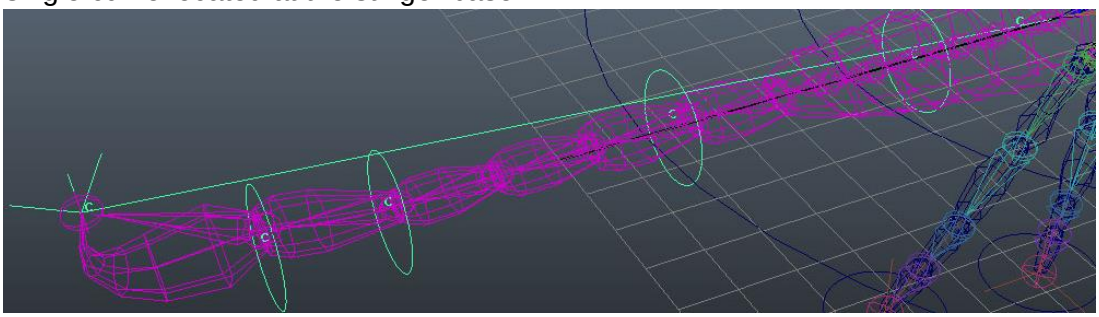




So that the tail is defined by the same number of vertices as we have controls, select the curve and go to *Edit Curves > Rebuild Curve*. Decide how many controls you would like and place that number minus three as the Number of Spans. In vertex mode, select the first two vertices (at the base of the 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.

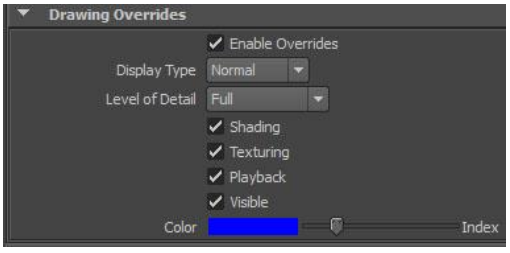


Parent the clusters too their respective controls, parenting the cluster at the base to the body control. If you wanted to control the rotation of the stinger by rotation rather than translation, you could parent the last two clusters to a single curve located at the stinger base.

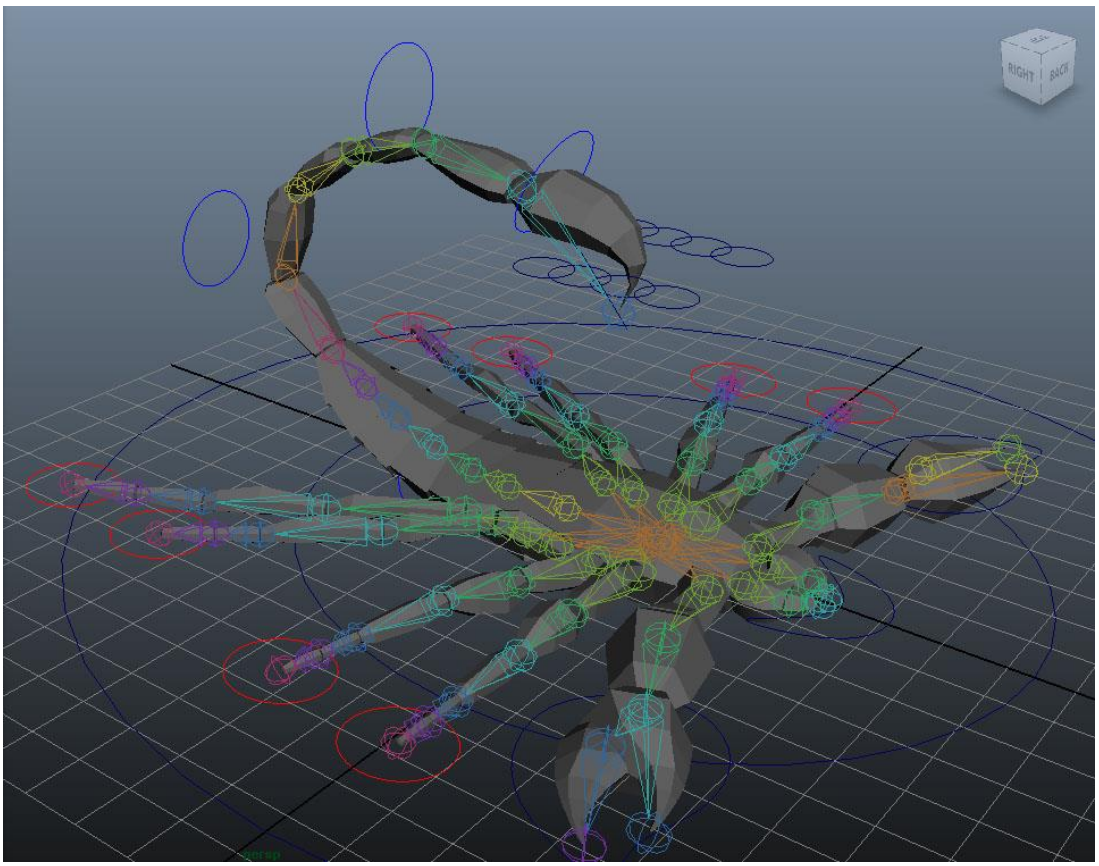


So that we don't get double transformation, make sure that the tail spline curve (*sp_tail*) is outside of the global control, I would recommend an '*extras*' group immediately below the overall scorpion group.

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.

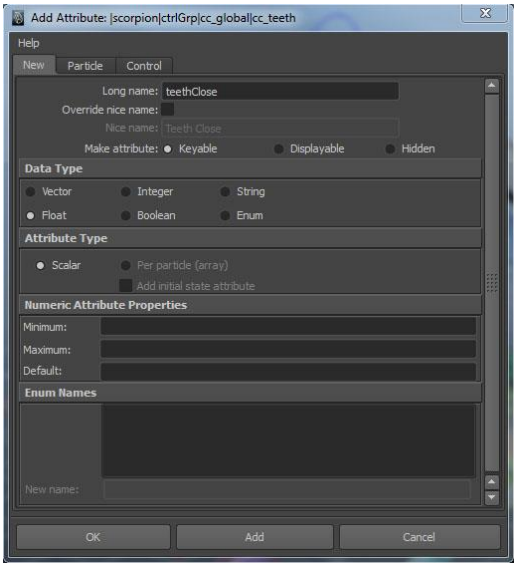


We will use the

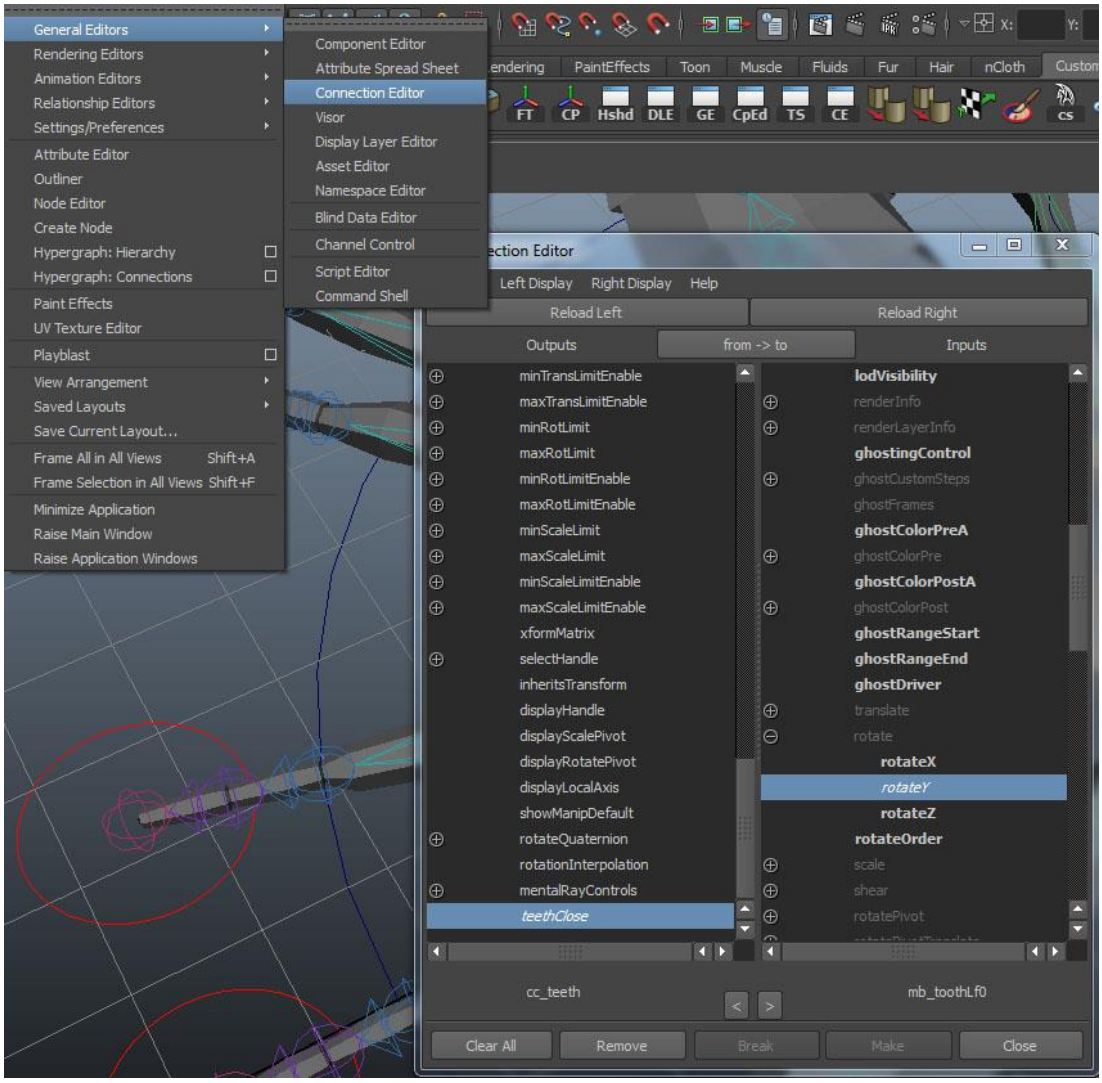


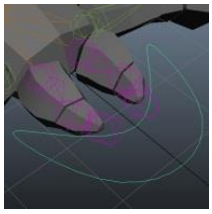
same technique to hide the *ikHandles* of the feet we created earlier and make them a different colour; you can group all of the feet in a new group (*ctrl-g*).

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*.

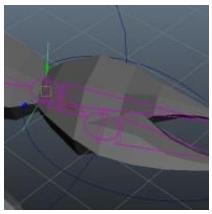


We are going to connect this to the teeth joints' rotate values. Open the connection editor, the teeth control should be loaded into the left side, but if it is not select the control and click 'reload left'. Select one of the teeth joints and click reload right. We want to connect the *teethClose* value to the *rotate value*.

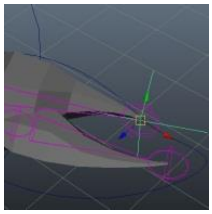




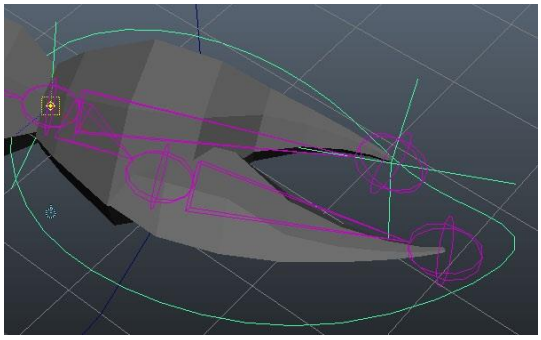
Use the above technique to connect the rest of the teeth joints (*right* and *left*) to the same control. You can test the control by changing the *teethClose* value and both teeth should now open and close together.



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.

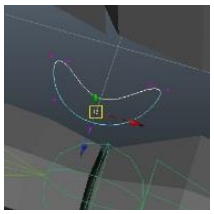


Create another one from this joint to the end of the claw (make sure you connect the end of the attached claw, **NOT** the pincer).



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.

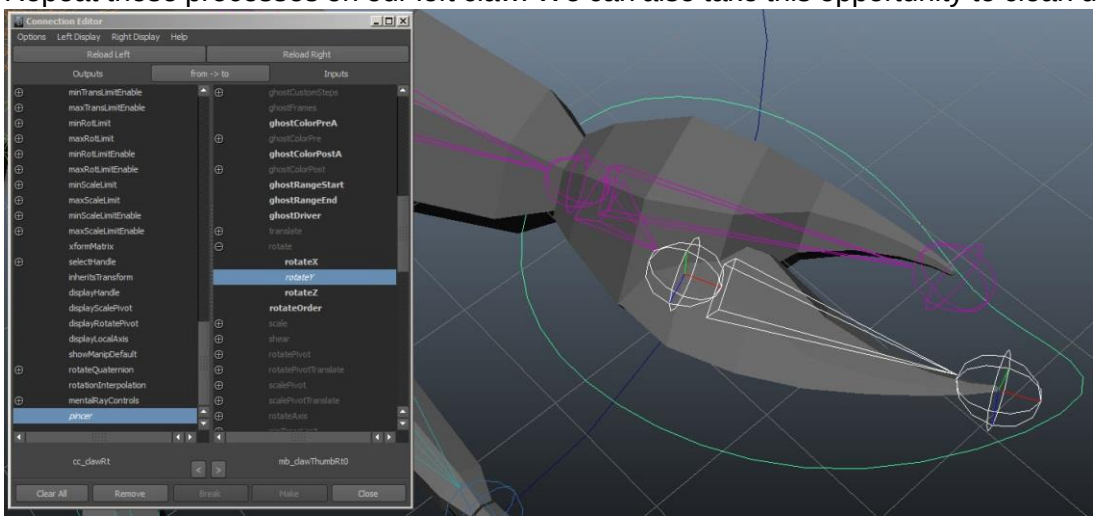
So that the claw rotates around the correct pivot, press *insert* to enter pivot mode and holding '*v*' to point snap, move the claw control's pivot to the centre of the claw joint.



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.

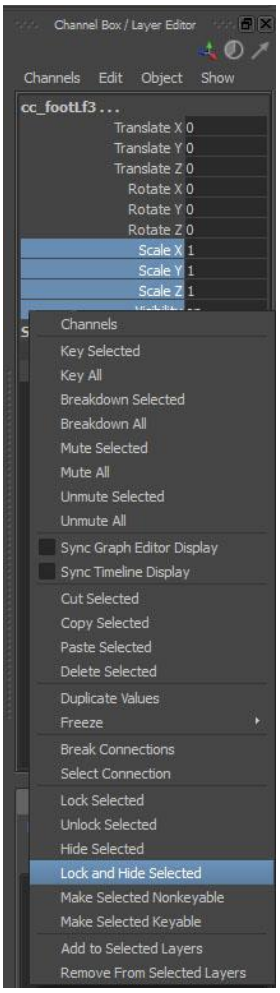
To finish off the claw, we are going to create a new attribute to control the pincers. Like we did before with the teeth, add a new attribute called '*pincer*' or similar and using the connection editor again, connect the new pincer attribute to the joint rotate attribute.

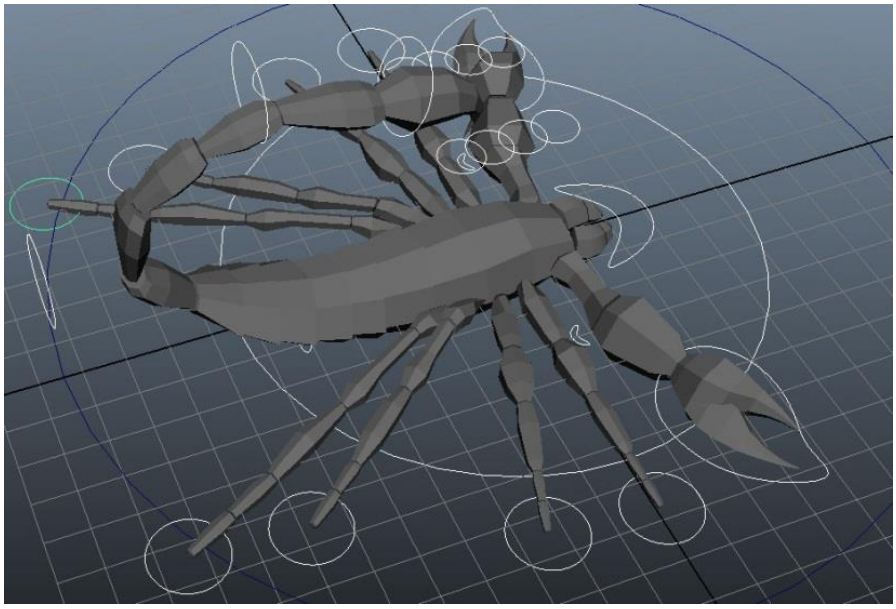
Repeat these processes on our left claw. We can also take this opportunity to clean up some

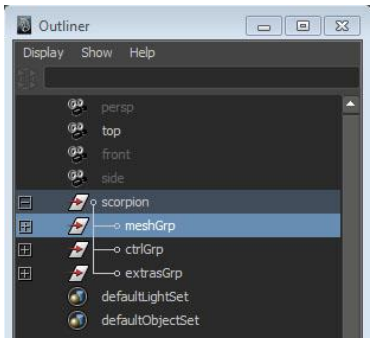


of our rig, making sure that our items are named correctly and appropriately laid out in our outliner. Remember that we can colour code groups of controls to make them easier to see, and hide things like *ikHandles* and joints that we do not need to see anymore.

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*'.

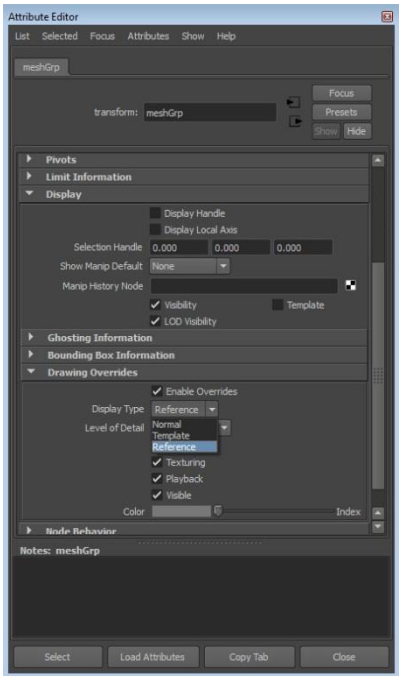






If we go around the rig, we will find a lot of other attributes we do not need. The rotation on IK and pole vector controls for example (legs and tail) and all attributes on the mouth control apart from our custom controls.

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.



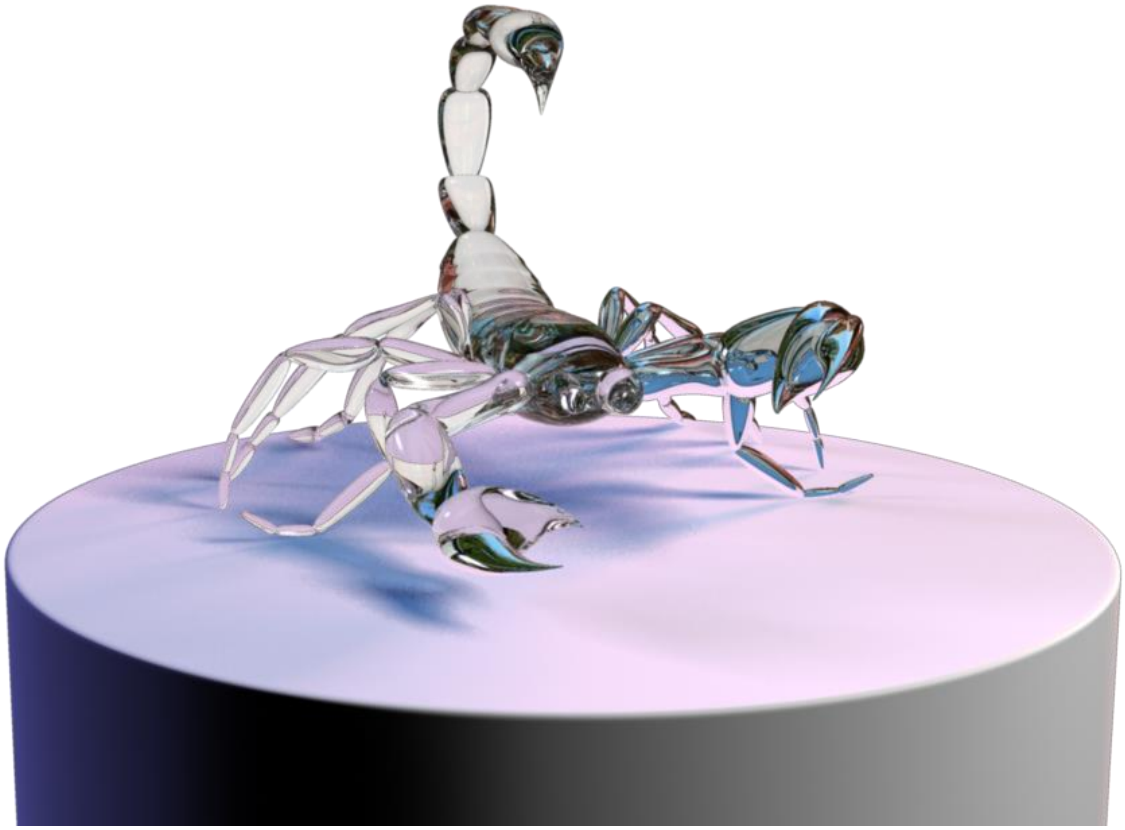
Animation

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

Turn to page 203 for an overview of an arthropod walk cycle.

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.*

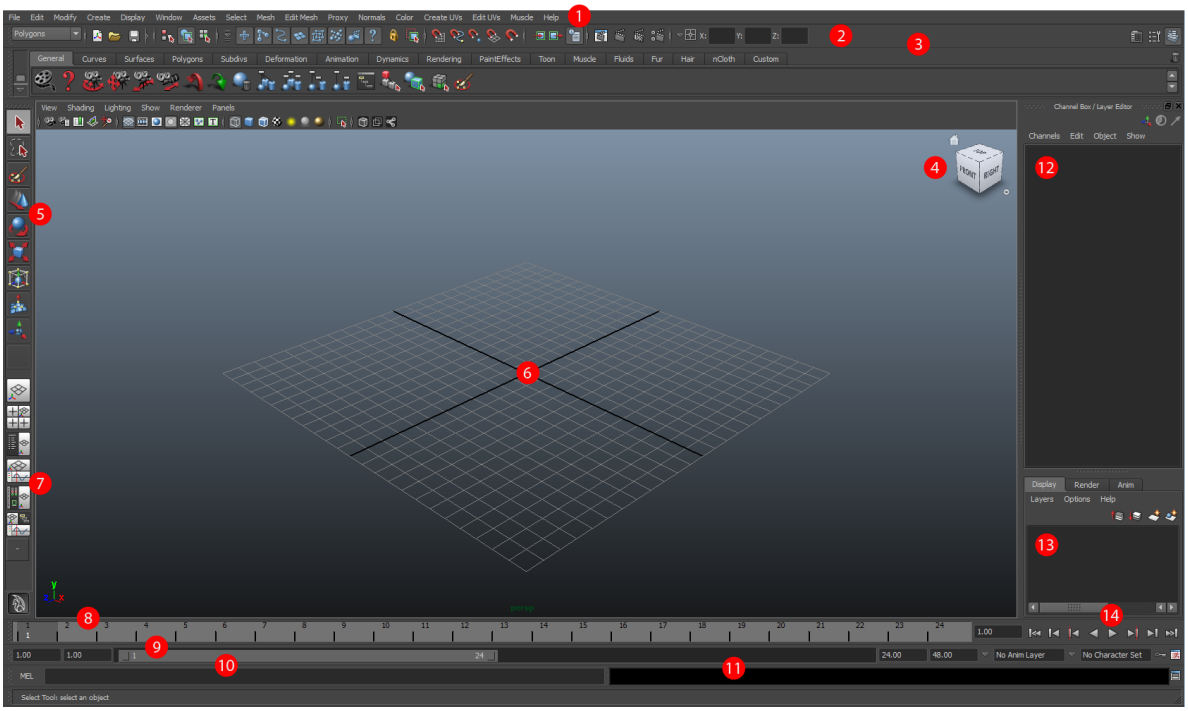
Keeping everything referenced ensures that if we have multiple shots using the same environment or the same characters, they are all using the exact same version.



CHAPTER 1: GETTING STARTED

Maya Interface

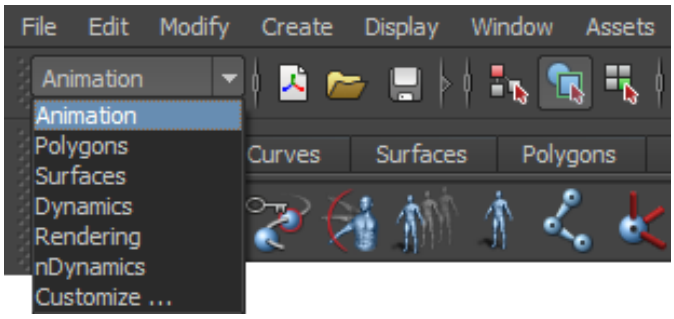
As you can edit the interface to your own preferences, we will need to ensure the correlation between this book and what you see on the interface in front of you. To bring Maya back to its default settings navigate to *Window > Settings/Preferences > Preferences* (In the dialogue box) *Edit > Restore Default Settings.*



1. Main Menu Bar
2. Status Line
3. Maya Menu Set
4. Global Controller/Manipulator
5. Tools
6. Workspace
7. Workspace Pre-sets
8. Time slider
9. Range Slider
10. Command Line
11. Help Line
12. Channel Box
13. Layer Editor
14. Playback Controls

Maya Menu System

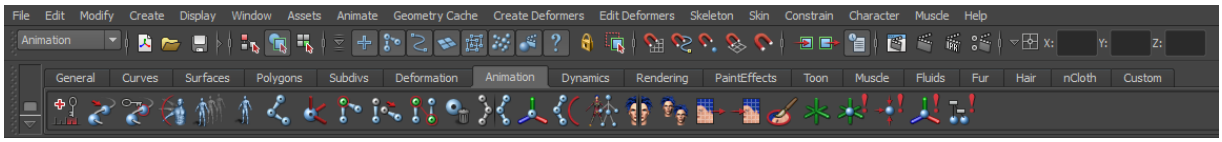
By default Maya opens up on the *Animation Menu Set*, this will present you with the following options:



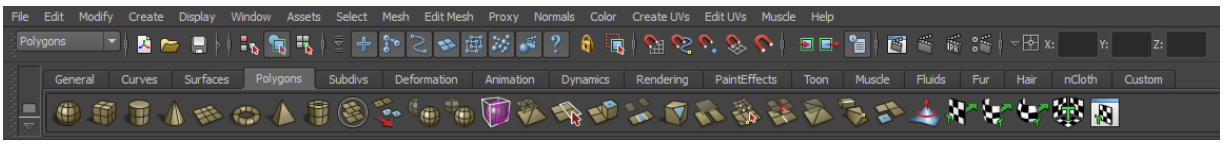
By selecting from the menu set it changes all icons apart from the first 7 (which are constant). This means whatever stage you are undertaking you have to ensure you're in the correct menu set, because you may find yourself looking for a tool you cannot access.

The below will illustrate examples of available options when selecting the different menu sets and the variety of tools that are available:

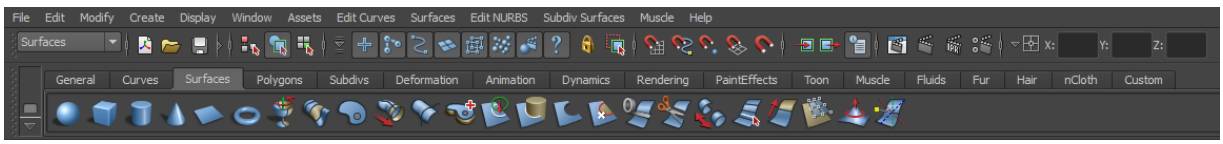
Animation:



Polygons:



Surface:



Menu Shortcuts

F2 – Animation Menu F3 – Polygons Menu F4 – Surfaces Menu Set
F5 – Dynamics Menu F6 – Rendering

Overview of Processes and Terminologies

Workflow

Maya workflows may vary between you, so don't be alarmed if you end up doing things a little differently. Workflows are the order in which you would go through a project, but certain steps may be missed entirely depending on your goal. Our best practice would go something like this:

1. Concept Art and Storyboarding

Concept art is the first stage where any character is born and should always be used to relate back to during the design stages. Storyboarding is a sequence of illustrations with the sole purpose to pre-visualise your final animation.

2. Modelling

The 3 very different geometry types (Polygons, NURBS & Sub-Divisional Surfaces) each have their own unique attributes that once utilized will enable you to create models with ease. Each type can be converted into one another at any point throughout the modelling process and is often useful to do so as they each have key advantages that once utilized could lead to increased productivity.

Construction History: Every action you do will create a footprint how many sequential footprints that are recorded (i.e. how many times you can press *ctrl-z*) are editable within Maya preferences, the default is set to 50 actions. However, history is applied to a specific object every time something is changed, be it moved, stretched or interacted with by another. This enables you to tweak settings that have been previously applied, creating a kind of dynamic undo ability. It does however take up additional memory and unfortunately the more history that ties onto an object the more likely things will go wrong at later stages such as rigging. Therefore, once you're satisfied with your model and before moving onto the next stage save your file with the entire history (optional), delete the history *File > Delete by Type > History* and save using the sequential naming structure discussed on page 41.

3. Textures & Shading

Shaders are what we use to provide our models with colour; contributing to their final appearance. Early on, you will most probably add default colours to objects but as your understanding increases along with your knowledge of the various Shader types, you will be able to create more elaborate and realistic effects depending on your requirements. Texture mapping adds additional levels to the standard Shader options; it allows you to import image files on various attributes allowing the manipulation of effects such as transparency or bump.

4. Lighting

There are various types of lighting used within Maya and arranging lights within your scene in order to complement models can often be overlooked by individuals or even missed off all together. Taking some time to learn the methodologies attached to this section will pay dividends when it comes time to render.

5. Rigging | Binding

Joints are forms of hierarchies, and once a skeleton (rig) has been created it allows you to bind with geometry in order to create deformations. They can be used to perform actions such as page turns in a book just as well as creating a fully functional character.

This section can also include:

- Painting Weights
- Creating Blendshapes
- Hair and Fur Rigs

Once your character is bound to the rig you will be able to facilitate movement.

6. Animating

This is where you can breathe life into your character. Traditionally, you will begin with key framing, however you'll soon learn this isn't the only way to meet your goals.

7. Rendering

Quintessentially, rendering is the process of compiling all (or selected) aspects of your scene into one. By default it will take into account all the characteristics such as applied textures, lighting, shadows, reflections, and transparency.

Transform Hierarchy and Relationships

In Maya an objects relationship to another creates what is known as the transform hierarchy. A parent/child relationship would link attributes such as translate, rotate, and scale. These attributes can be linked to its parent, its parents' parent and so on and so forth.

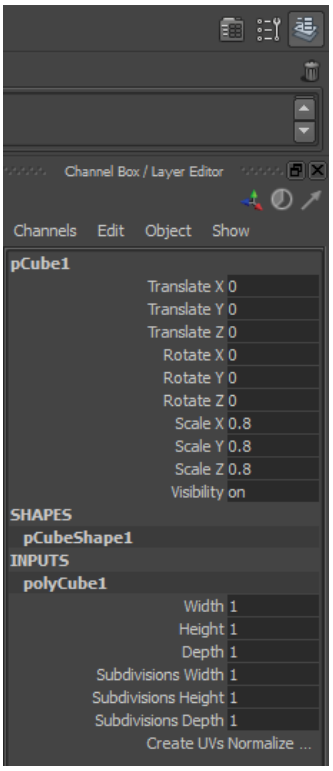
Scripting

Almost every aspect of Maya can be manipulated using MEL script, based on C++. This allows a user to directly interact with Maya and write their own scripts. This is particularly aimed at technical animation and rigging, for example moving a large number of objects with a random variable, or automating a long drawn out task to be executed at the press of a button.

Maya has recently also added support for Python, a more powerful and cross compatible language, however, I would recommend this only to programmers who are already comfortable with Python as Maya will not break down the commands for you to reference like it does with MEL.

We will delve into an introduction to MEL scripting a little later and as part of this book, you will be eligible for a free Scripts via the Spotlight Studios website.

Channels Box and Nodes



The channel box is located on the right hand side of your screen and provides access to the most commonly used attributes. It has a coloured key (opposite) for quick reference illustrating if an attribute has been keyed, locked, or linked with another. The nodes on the right hand side (RHS) are split into three sections:

- **Transform:** General positioning information, affected if you translate rotate or scale your model.
- **Shape:** Holds all generic information to do with the overall shape appearance.
- **Input:** Generic creational input node, holds information on height, width and length as well as options such as how many subdivisions or sections, an item will have.

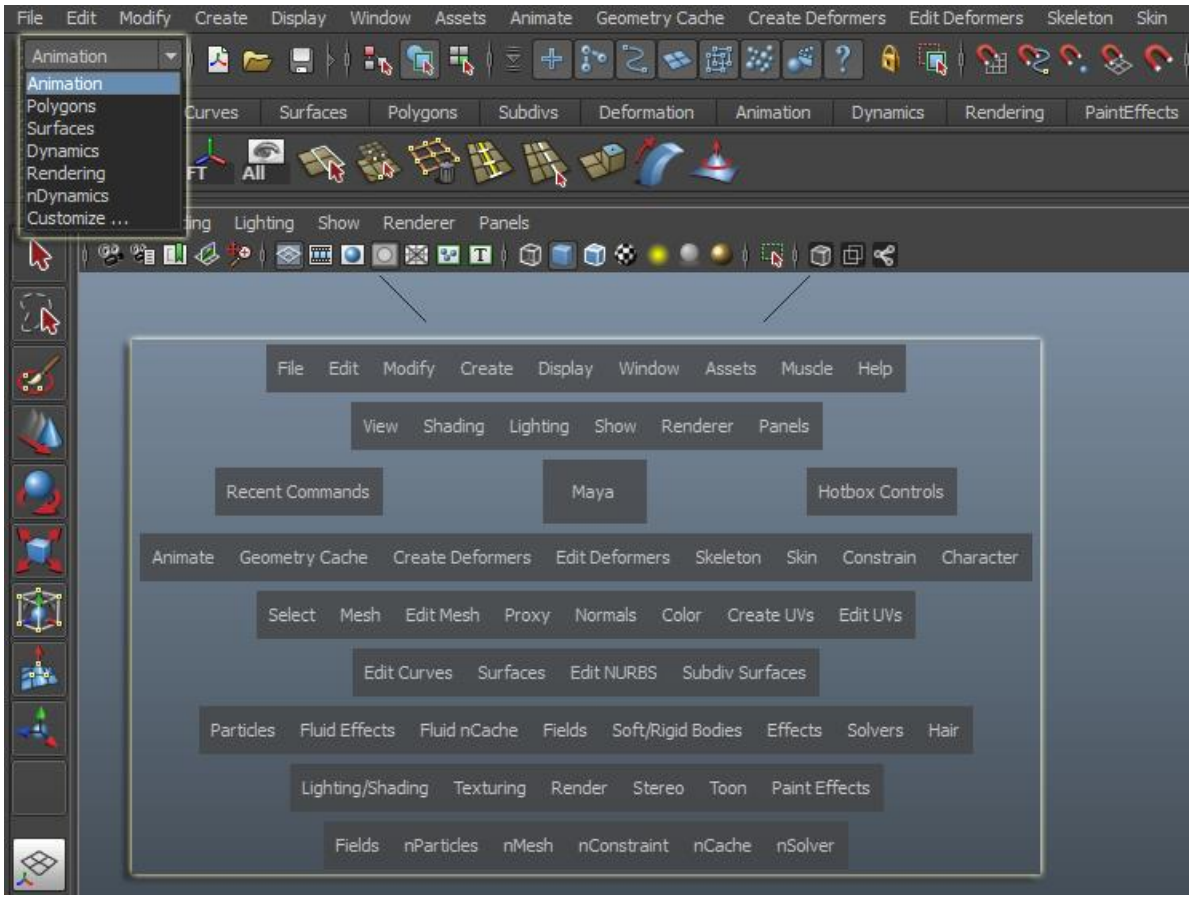
Note: *The input node can only be used for initial modifications as later usage will result in anomalies if you have edited your shape in any way. Should you delete the history at any point (on your model) this option be removed.*

Connection	Colour	Description
Muted Animation	Brown	The attribute has been animated but it has been muted, therefore playback will not commence.
Constraint	Blue	This attribute is linked to an attribute of another.
Locked	Grey	Un-editable whilst in this state.
Non-keyable	Grey (Light)	Cannot be animated.
Blended	Green	Blended with another attribute.
Keyframed	Orange	A keyframe has been added to this attribute.
Expression Control	Purple	An expression regulates this attribute.
Connected / Dynamic	Yellow	Shares identical values as another, or is dynamically influenced.

Using the Hot-Box

The Hot-Box is another often untouched tool that we find incredibly useful. Holding down the Space Bar in any of you viewport windows will give you a series of white menus that duplicate the menu structure of Maya. This saves time changing menu modes and gives you all of Maya's controls at your fingertips; you can use this as well as, or instead of the standard menu, and can also display pane specific and custom menus.

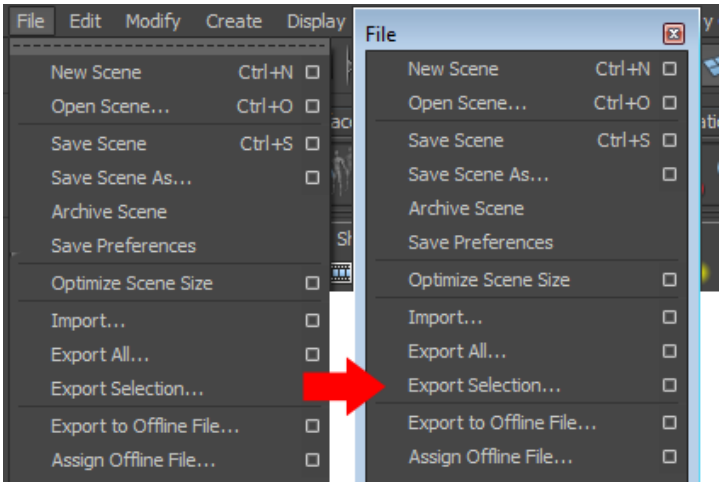
The Hot-Box also gives you quick access to difference layouts, masks, UI elements etc.



Note: *The hierarchical structure correlates directly from the menu at the top left (discussed earlier) and the Hot-Box controls that appear when pressing the space bar.*

Using the Menus efficiently

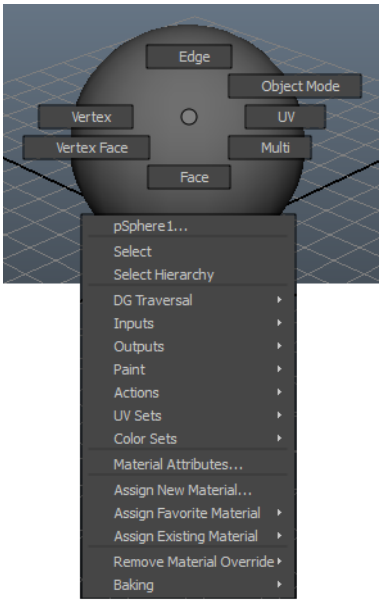
Wherever possible, get into the habit of using the small option box on the right hand side of the dropdown menu, even for mundane tasks like creating a cube. This provides access to an additional dialogue box, thus allowing you to see additional parameters, whilst ensuring the default settings are applied, unless you require otherwise. Many of these settings or parameters can be altered at a later point using the *Attribute Editor* or *Channel Box* so if you do forget initially this can be rectified.



Maya provides the ability to “tear off” nearly all menus and sub-menus to give you instant access to certain tools. To do this click on the dotted line at the top of a drop-down menu and this will create a pop-out version, very useful for circumstances such as Painting Weights.

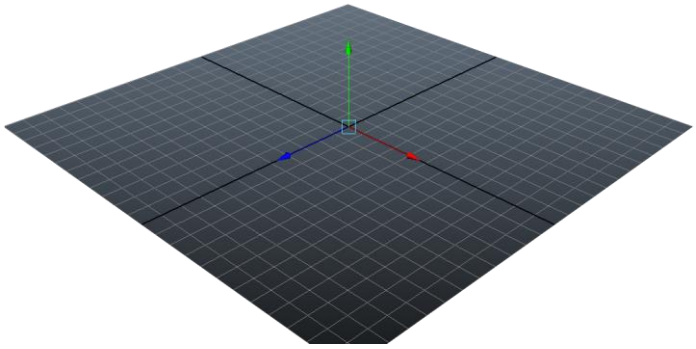
Marking Menus

These are selection sensitive, which means dependent on the item in question the menu will adapt accordingly. To access a marking menu hold down your *Right Mouse Button* (*RMB*) on an object; then by dragging it around and interactive line will emanate from the centre to illustrate what option you currently have selected, by releasing the *RMB* you will select the relevant option.



Working Within a 3D Environment

If you are not accustomed with 3D packages you should familiarise with their basic workspace principles. As you would expect the three dimensions are represented by axes which are:



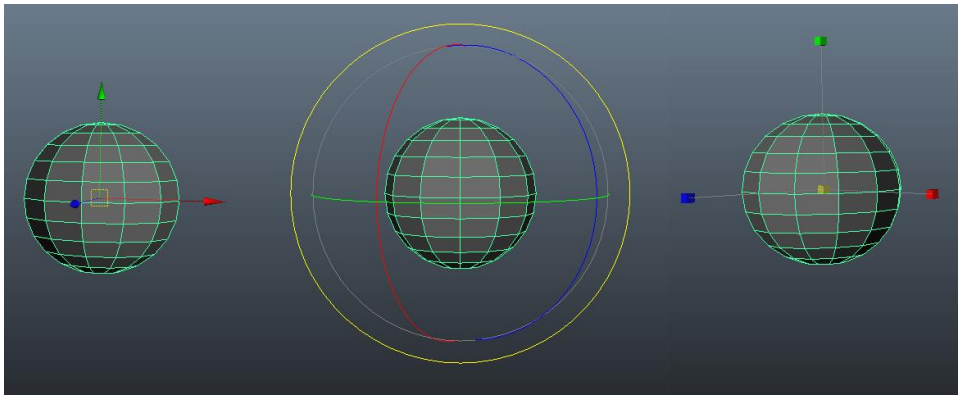
Length	(X – Red)
Height	(Y – Green)
Width	(Z – Blue)

These axes will always serve as you reference points and will be given specific co-ordinates based on the objects location, the default location of an object being 0,0,0.

Manipulators

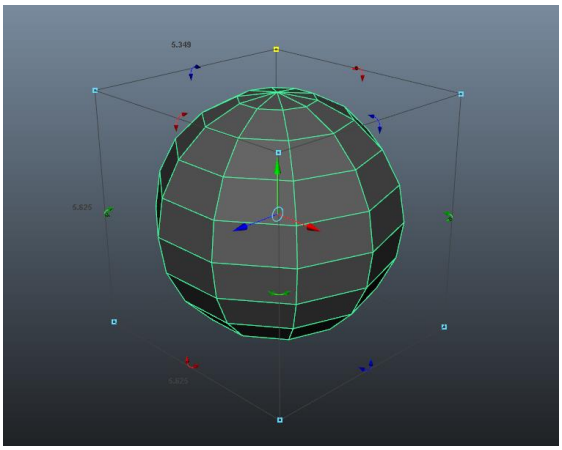
There are three main types of manipulators that are used constantly throughout a project and these are:

Translate	**Rotate**	**Scale**
Shortcut: w	**Shortcut: e**	**Shortcut: r**



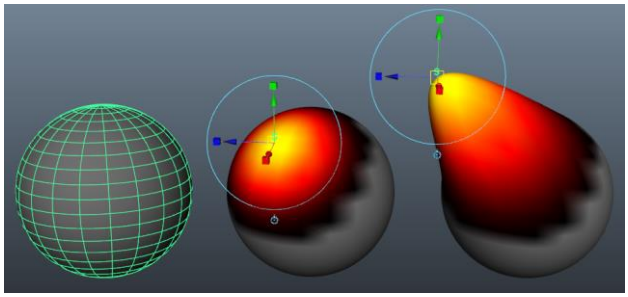
The manipulator handles that appear when the relevant tool is used are self-explanatory, nonetheless we recommend using them a little to fully understand their usability. Once you begin to use keyboard shortcuts (such as those highlighted above) it will considerably speed up your workflow.

Universal Manipulator



This tool is a combination of all three of the above actions and allows for simultaneous manipulation without the need to change tools. Some users find this quicker and useful others prefer to edit each aspect individually ensuring errors are not made by selecting the wrong manipulation type.

Soft Select



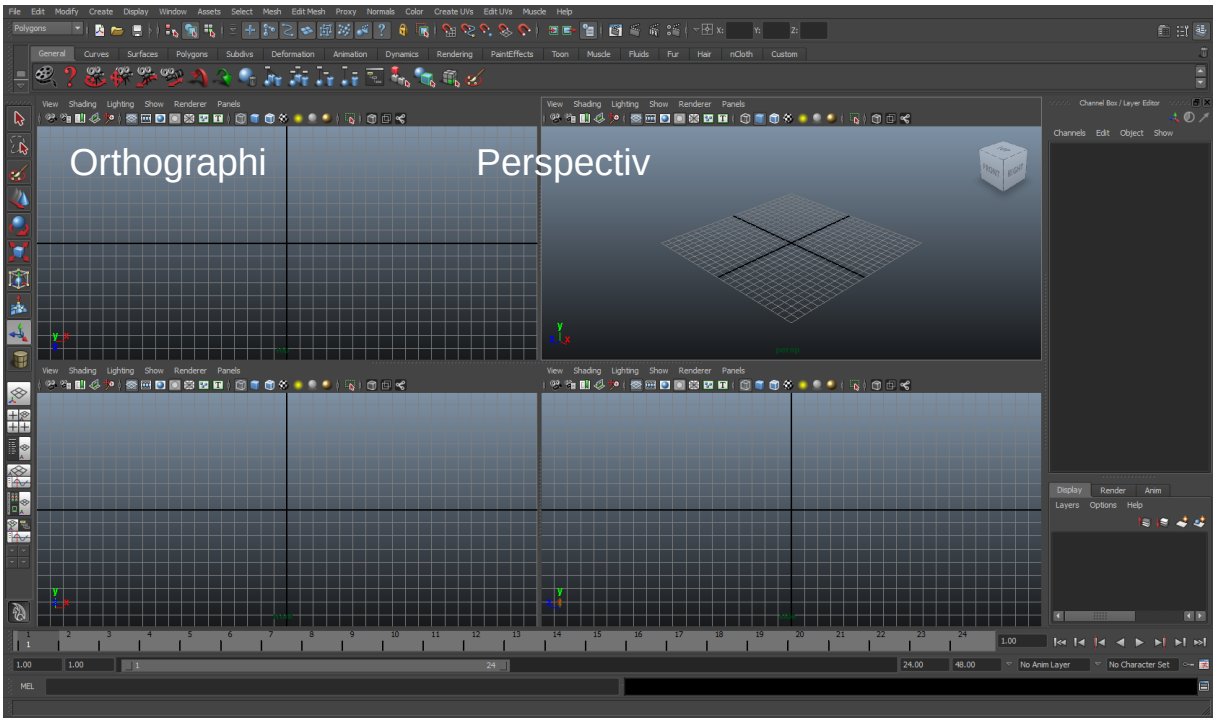
Checking this box within the tool settings pane (or pressing '*b*') for your selected manipulator (*translate, rotate or scale*) allows you to add a gradual falloff from the selected points. When active, you are presented with a colour coded selection area (Yellow / Red / Black) indicating the influence that will be affected. This is useful to give smooth deformation of an object, especially useful in creating and modifying organic shapes such as faces.

Note: *This is not to be confused with the* Soft-Modification tool *which we will get too later in the deformers section.*

Workspace and Cameras

In order for you to be able to view a scene, you will always be looking through a camera. There are two main camera types in Maya, these are:

1. **Perspective** – These cameras provide depth illustrating that objects further away from the camera appear smaller than those closer to it, just as if you perceived the objects through a camera lens.
2. **Orthographic** – These will not add perspective in any way but will allow parallel viewing from any side, or the top of a scene. This is usual for comparing the actual length of various parts of a shape as their displayed size is irrespective of their distance from the viewer.



Switching Cameras / Workspace

By default you will see the (top right) perspective camera view screen upon launching Maya. To alternate views, press the spacebar to launch this screen, and simply ensure the mouse pointer is hovering over the workspace you require and tap the space bar again to select an alternate view.

Navigation

The majority of you will be using a mouse and keyboard to navigate around within Maya, there is another popular form of navigation which is the graphics tablet, which people will often customise to their own personal preference once they have had experience within Maya. If you are a first time user we would recommend the traditional keyboard and mouse at this point, but feel free to experiment in the near future.

Camera movement and control (Standard 3 button mouse):

- Left Mouse Button (LMB) - Selection tool.
- Alt + LMB - Rotates camera.
- Right mouse button (RMB) – Marking Menu.
- Alt + RMB - Dolly In and out (not zoom as it actually moves the camera).
- Middle Mouse Button (MMB) Scroll - zoom (changes focal length).
- Alt + MMB – Panning, horizontal or vertical depending on mouse movement.

1 or 2 Button Mouse controls

For Mac and Laptop users, we would highly recommend purchasing a 3 button mouse (Mac OSX is configured to work with them out of the box). On a Mac the Option button replaces Alt, and the Command (Apple) key will replace the windows Ctrl key.

However, if you are set on not having a 3 button mouse you can navigate to *preferences* > *Interface* > *Devices* and you can toggle between one, two, or three button mouse controls with or without the option of a scroll wheel.

Creating a Personalised shelf

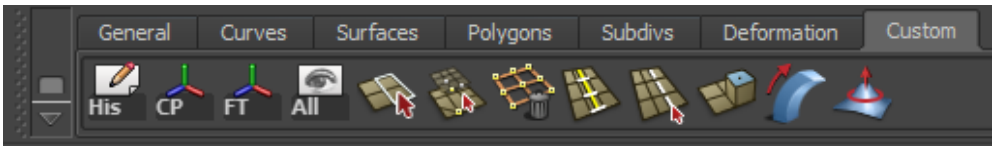
The shelf is an invaluable tool at your disposal. Essentially it is a customisable shortcut bar to give you access to all of your favourite tools or scripts quickly and easily. Maya comes set up with a number of preset shelves for different areas of animation, such as Polygons, Rendering and nCloth.

These provided a very useful basis for getting started, but it becomes much more valuable when you start creating your own shelves. You can change shelves by selecting the tab, or if you have a lot then you can click and hold the tab on the far left and select it from a drop down menu. The button under that gives you a menu where you can create, delete, load, save and modify the hierarchy of your shelves.

Create a new shelf and give it a name. To add a button to a shelf you simple shift-ctrl-click the menu item, for example, to add to create poly button, go to *Menu > Create > Polygon > Primitives* and *Shift-Ctrl-LMB* the Cube option box.

You will see the button added to the end of the shelf. You can re-arrange shelf buttons by middle click and dragging the button, and remove shelf buttons by dragging them to the bin icon on the right hand corner. Now click the button and the Cube Create tool will be activated and the tool settings window will also appear.

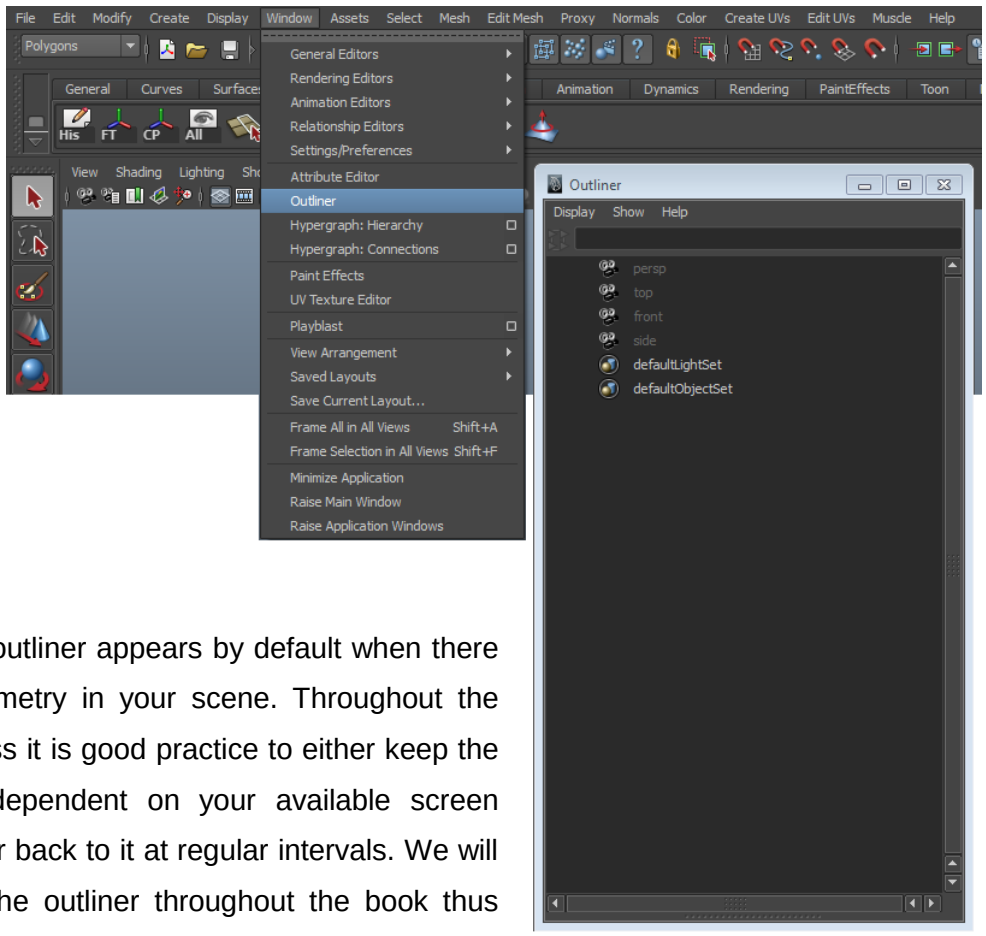
Example Shelf



From Left to Right: Delete by type History, Centre Pivot, Freeze Transformations, Show All, Split Polygon Tool, Merge Vertices/Edges, Insert Edge loop, Select Continuous Edge, Extrude, Bend, and Sculpt Geometry Tool.

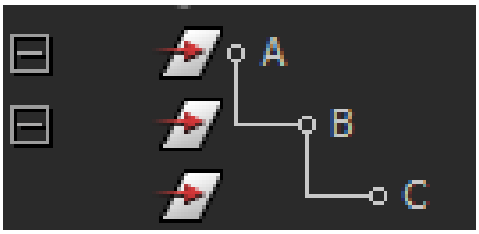
Introduction to the Outliner

The outliner is one of the key areas when it comes to the organisation of your scene. Maya represents every object you create as a node (as well as a few defaults), these nodes can then be organised within the outliner to ensure organisation and continuity is kept throughout. This feature allows you to re-arrange the nodes into groups, thus creating hierarchies enabling them to be controlled as one. You may also use this to create and edit parent/child relationships explained previously.



This is how the outliner appears by default when there is no other geometry in your scene. Throughout the modelling process it is good practice to either keep the outliner open (dependent on your available screen space) or to refer back to it at regular intervals. We will be referencing the outliner throughout the book thus gradually introducing you to new aspects as a when you need them.

Outliner Hierarchy



The outliner hierarchy is structured as parents and children and the transform manipulations are passed from parent to child and so on. For example, if a parent node ‘A’ is moved along the X-axis by 2 then the child ‘B’ will also move along the X-axis by 2 and so will it’s child, ‘C’.

One parent can have many children, a child can become a parent but will only ever have one, although can be influenced by its parents parent; known to the child as a grandparent node.

Viewing your Scene

Smooth Shaded: *Shading > Smooth Shade All* - This enables you to display the items in your scene as solid objects, if you have assigned colours they will be displayed accordingly (they can be disabled by selecting, *Shading > Use Default Material* very good for gauging what your image will render like.

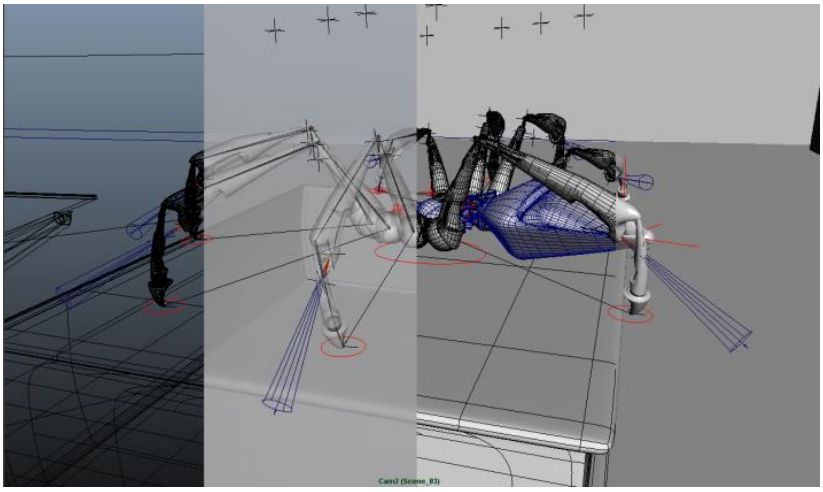
Wireframe: *Shading > Wireframe* - Often used when editing vertices, enabling you to see the individual make-up of an item, whether this be in poly’s, NURBS or sub-D’s.

Wireframe on Shaded: *Shading > Wireframe on Shaded* - A combination of the two shading options above. You’re able to see the make-up of an object whilst keeping it in solid form, very useful when tidying up geometry.

X-Ray: *Shading > X-Ray* - As you will probably guess this is gives you a semi-transparent view of any object within the scene.

Variations:

In the panel options there are various *Shading* options you can apply, the ones above are the most commonly used but there are others, so have a play and see what suits you.



Scripting

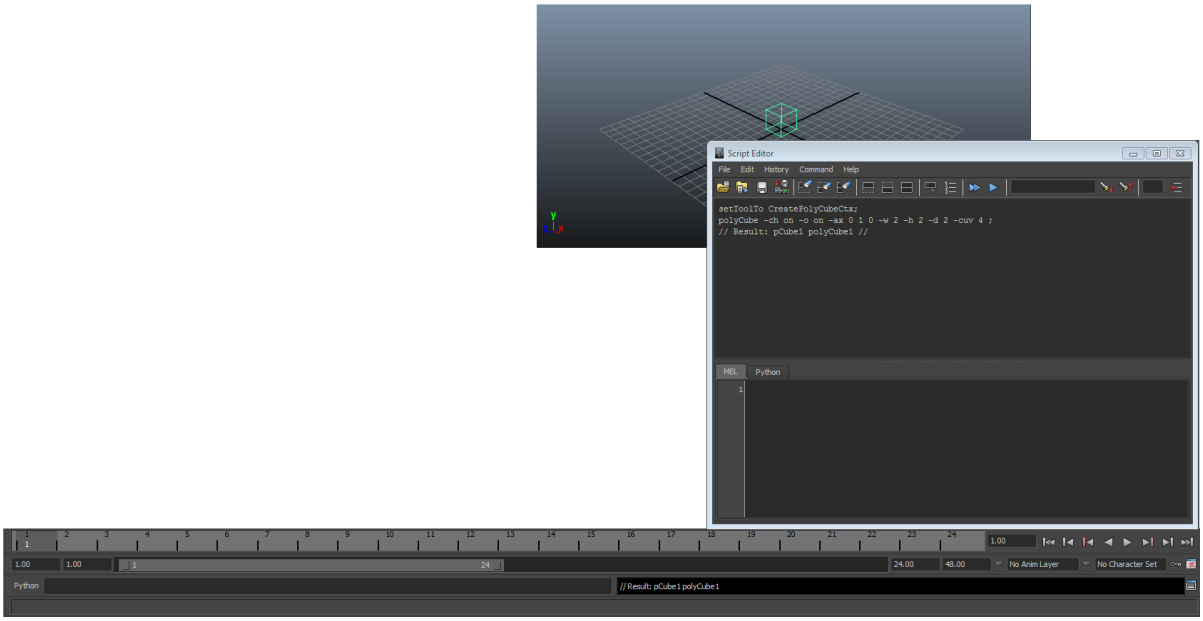
Scripting is using code (specifically MEL script) to directly interact with Maya. This can involve everything from automating tasks to tool development.

MEL (Maya Embedded Language)

For every action a piece of MEL code is created and implemented, you can use this to write scripts or execute simple commands. The image accompanying this text illustrates how the creation of a polygon cube generates MEL code and if we were to highlight the code in the top section, copy it to the bottom section and execute the command we would be presented with a second polygon cube. We will delve into MEL scripting a little later in the book and will introduce you to some of our scripts (available from our website) which you are welcome to use at your discretion.

Python

Some programmers who migrate over to Maya will be able to implement their own tools using this coding language. Maya has only allowed the default built-in commands to be accessed through Python scripting.



Hot Keys

Note: *Maya is case sensitive so if you are getting an unexpected action or nothing at all ensure Caps Lock is off. Also the Option key on a Mac replicates Alt.*

Menu Selection

Hotkey	Action
F1	Help
F2	Animation Menu Set
F3	Polygons Menu Set
F4	Surfaces Menu Set
F5	Dynamic Menu Set
F6	Rendering Menu Set
F8	Toggle Component Type / Object Type Mode

Display

Hotkey	Action
1	Low Polygon (base)
2	Medium Polygon (Combination of base and smooth)
3	High Quality Polygon (Smooth Preview)
4	Wireframe
5	Shaded
6	Textured
7	Use Scene Lighting
Alt-b	Cycle Workspace Background Colour
Space	Hotbox Controls
Ctrl-h	Hide Selection *(Display > Hide > Hide Selection)*

Hotkey	Action

Ctrl-Shift-H	Show Last Hidden *(Display > Show > Show Last Hidden)*
Alt-h	Hide Unselected *(Display > Hide > Hide unselected Objects)*
Shift-I	Isolate *(Show > Isolated > View Selected)*
Shift->	Increase Vertices Selection
Shift-<	Decrease Vertices Selection

Navigation

Hotkey	Action
Alt-LMB	Tumble / Rotate
Alt-MMB	Track / Pan
Alt-RMB	Dolly / Zoom

Moving Objects

Hotkey	Action
Alt-Up	Move Up One Pixel
Alt-Down	Move Down One Pixel
Alt-Left	Move Left One Pixel
Alt-Right	Move Right One Pixel

Tool Operations

Hotkey	Action
Return	Complete current action
Insert	Edit Pivot Point
Q	Select Tool

Hotkey	Action

W	Translate (Move) Tool
E	Rotate Tool
R	Scale Tool
T	Show Manipulator Tool
Y	Select Last Used Tool
+	Increase Manipulator Handle Size
-	Decrease Manipulator Handle Size

Snapping

Hotkey	Action
C	Snap to Curve
X	Snap to Grid
V	Snap to Point

Editing

Hotkey	Action
Ctrl-c	Copy *(Edit > Copy)*
Ctrl-d	Duplicate *(Edit > Duplicate)*
Ctrl-Shift-D	Duplicate Special *(Edit > Duplicate Special)*
g	Repeat Last Action
Ctrl-g	Group *(Edit > Group)*
p	Parent *(Edit > Parent)*
Shift-P	Unparent *(Edit >Unparent)*
Ctrl-s	Save
Hotkey	**Action**
Ctrl-v	Paste *(Edit > Paste)*

Ctrl-x	Cut *(Edit > Cut)*
Ctrl-z	Undo *(Edit > Undo)*
Shift-Z	Redo *(Edit > Redo)*

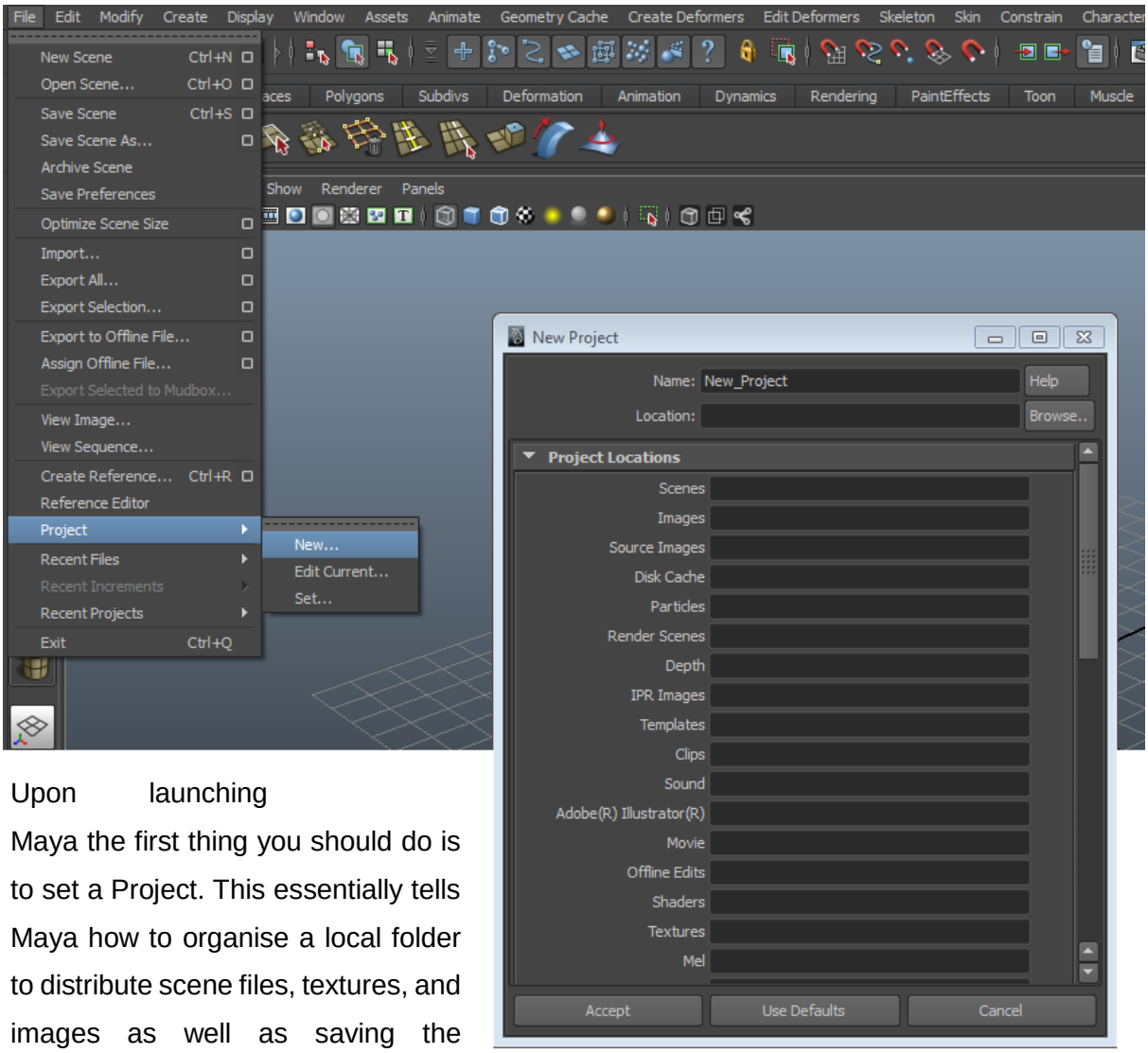
Animation Operations

Hotkey	Action
s	Set Key *(Animate > Set Key)*
Shift-E	Set Key for Rotate
Shift-R	Set Key for Scale
Shift-W	Set Key for Translate

Note: *To edit or create a new Hotkey navigate to Window > Settings / Preferences > Hotkey Editor*

Scene Setup

Creating a New Project

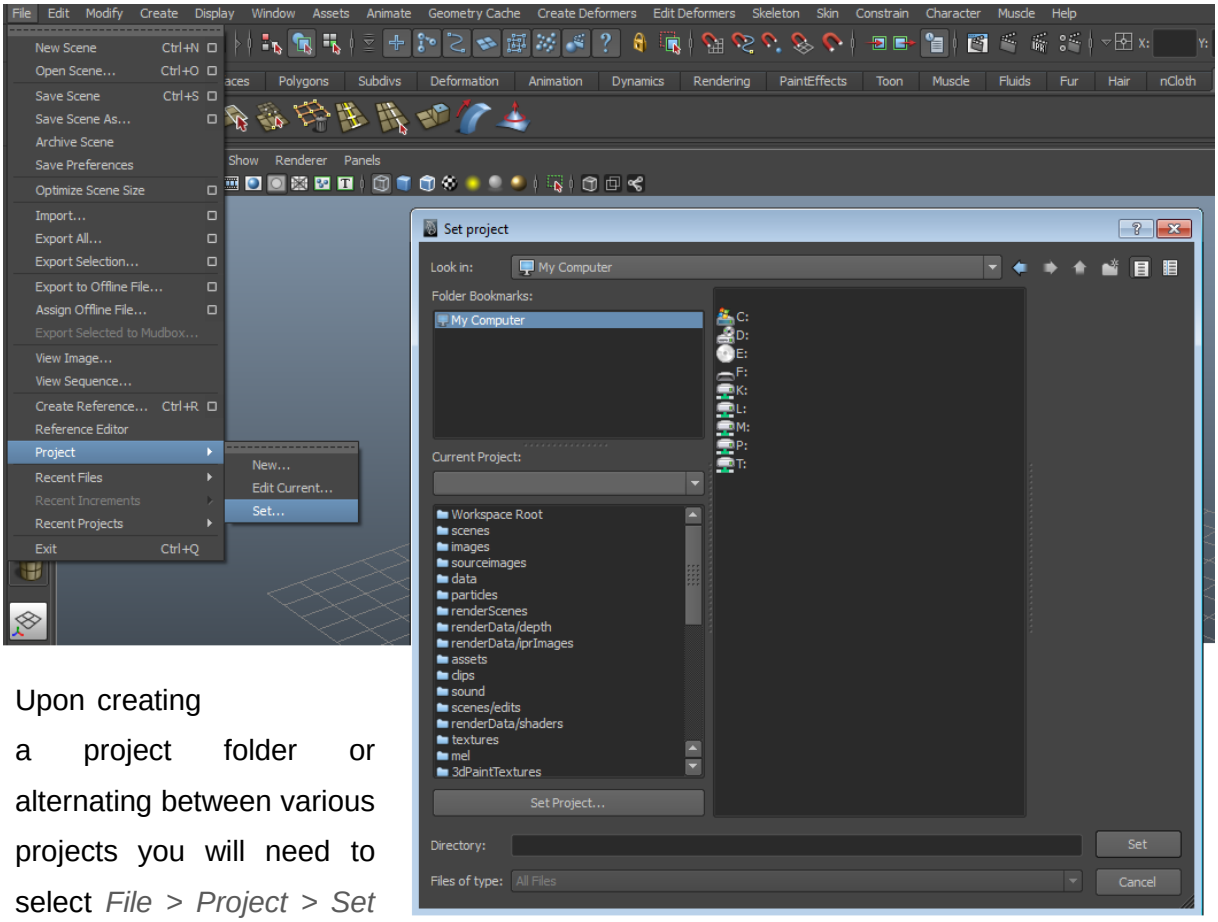


Upon launching Maya the first thing you should do is to set a Project. This essentially tells Maya how to organise a local folder to distribute scene files, textures, and images as well as saving the workspace information. This allows for easy integration within the project and allows the project to be distributed while maintaining links to assets, textures and other referenced scene data.

This ultimately means working from multiple workstations is effortless as you can just copy the project folder to a different location (or load the project from a network location) and images etc. will be referenced relative to the project location rather than an absolute path such as *C:\Documents and Settings\My Pictures*\.

To create a New Project click, *File > Project > New,* then enter a project name and click *use defaults* at the bottom (recommended). If you do however wish to fill them in manually you can write your folder names you require in the relevant empty fields. After either option you will need to select *Create*, to finalise the process.

Setting an Existing Project



Upon creating a project folder or alternating between various projects you will need to select *File > Project > Set* and select the appropriate project file (remembering Maya looks at this folder for references)

Scene Preparation

We recommend Modelling to scale where appropriate because it helps with realism and consistency across projects. To alter the scale of your grid navigate to *Window > Setting/preferences > Preferences > Settings > Working Units > Linear > then select "Meter"* (or your preferred scale).

Clipping Plane

In the cameras attribute editor *Window outliner > (camera in question) > attribute editor > Far clipping plane* change the *Far Clip Plane* from 1000 to a higher value such as 5000. Without this setting when you zoom out to view your model you may find some of it gets cut off or disappears entirely.

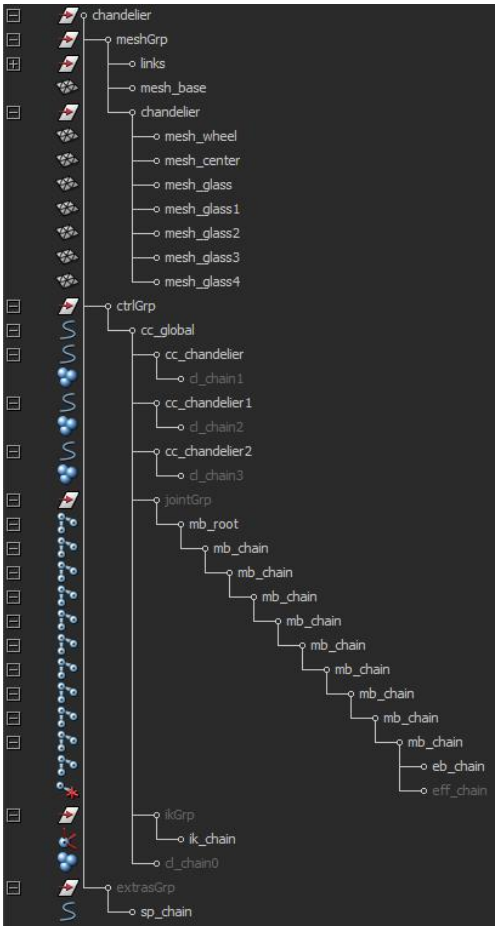
Camera Bookmarking

Although we aim to create a setup that can be rendered from any angle you might have a preferred one or in our circumstances purely a reference for progression. To create a bookmark put your camera into position, within your camera viewing pane, select *View > Bookmarks > Edit Bookmarks…* type in a name and a description (if required) then press enter. Now move your camera somewhere else, jump back to *View > Bookmarks > Your Bookmark* and you should notice your camera jump back into position

Useful Notes

Naming Conventions

It is extremely important to have good naming conventions and group structure when you are building your scene. It is a practice we would recommend you get into a habit early on so that it becomes second nature. The advantage of this is twofold; it makes it easy to find a specific object (you can simple type the prefix into the *Outliner* or your *Select by name* field) and it also helps minimise grouping problems such as double transformation. On the right is an example of an asset we have created (the chandelier from our saloon scene) to give you an example of good naming and grouping practices.



It is worth noting that Maya likes objects to start with a lower case letter and for multiple words to be defined by capitalising the first Letter. For example Maya will read "*oneTwoThree*" as "*One Two Three*".

Save Sequentially

You should structure your saved files in generations, so that if you need to return to a previous point for whatever reason, or you incur file corruption, only the time between file generations is lost. This can be done daily, weekly, or for every save generation depending on how important the project is, how much the project changes and, ultimately, how much space you have available.

The simplest method is to simply use Maya's built in incremental save feature. Clicking the box next to *File > Save* will bring up a few options, turning the feature on and off, and also allowing you to automatically limit the number of generations that is kept on the computer at any one time.

An alternative manual method is to simply number each File <Scene_01_001.ma, Scene_01_002.ma etc.> alternatively if you're feeling adventurous you could incorporate the date, time, place of creation (home/labs/work) etc.

It is also worth mentioning here, that it is important to back up your work to an **external** drive or an online storage alternative as often as you feel necessary.

Hiding Un-Used Items to free up real-estate:

You can hide nearly every part of Maya's UI in order to dedicate a greater portion of the screen to the viewport. Everything is entirely subjective to an individual's way of working, but combined with the hotbox this can be a very powerful and visual way to work.

The Outliner is your friend:

Everyone works in different ways, but in our experience we have found that working in a very 'clean' and organised way tends to make the process much faster and more efficient in the long run, this makes problems that crop up much easier to fix and group work a lot less painful.

The Outliner is simply a Hierarchal list of your scene, like a simplified Hypergraph. You can select and move objects around between hierarchal levels and provides a very easy way of renaming (double click) and organising (*MMB and drag objects*) your scene. (Pg 32 for more details)

Display Layers

You can use display layers to limit what is visible and selectable in your scene. These work in a very similar way to Photoshop layers.

For each display layer, objects that you put in the layer can act as normal, be referenced, or become a template.

When a layer is set to reference, the objects will appear as they would normally in the viewport, but they will not be selectable. This is very useful if you have background objects or geometry that may get in the way of you selecting your controls.

When a layer is set to template, the objects are neither selectable nor are they renderable. They will show up in your viewport, but only in a greyed out wireframe. This could be used to either limit the clutter on the screen (while still showing the objects' locations) or to show the position of an animated proxy object while you animate the main character over the top.

Large / Group Projects / Referencing

When working on large or group projects, organisation becomes even more important, and a clear file naming and storage should be designed and maintained very early on.

The more 'stuff' in your scene, the slower Maya will run and the more difficult it becomes to navigate, select and modify parts of the project. You may also have the same content in different scene files, what happens if you need to change the main characters dress? You can help to get around this to some extent managing layers, but it doesn't quite solve the problem.

The best way to properly address this kind of issue is to Reference in your different parts. This means that at the top level, you will not actually have any objects in the scene, but rather be changing the values of the objects you are referencing. This means that at any stage you can go back and edit the referenced objects without worrying about what you have used the objects in. This also means that in some cases you can animate a character in a scene whilst other people continue to work on the files, each time you open the scene or reload the references everything will be updated to the latest versions, but any changes you have made within your animation will stay the same.

With specific reference to speeding your scene up, you can animate your character to a very basic 'proxy' environment, and then when you have completed the animation, you can unload the environment and save the file as an animated asset. When you are ready to put the scene together, referencing this file will automatically follow through and also reference in the original character model, too.

This kind of structure, as long as it is well managed allows you to be much more flexible with your animation.

To import an asset as a reference, simply click *File > Create Reference* and choose the file you wish to reference. You can use the Reference Editor (next button down) to manage and reload existing references.

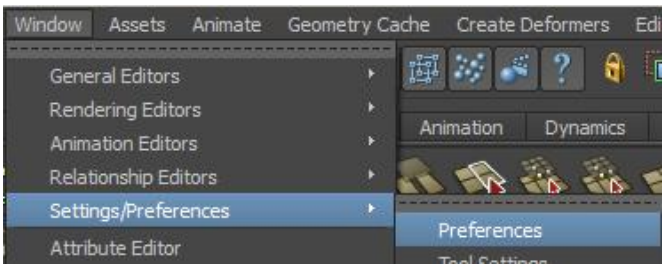
It is worth noting that the organisational structure of the asset will stay unchanged, and referenced nodes cannot be moved or deleted. For this reason, it is strongly recommended by us that your assets always stem from a single named Node.

CHAPTER 2: MODELLING

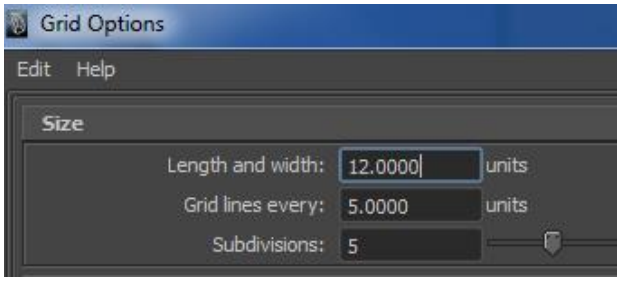
Introduction to Modelling

This is how your characters and scene objects are created. Anything imaginable can be built within Maya and by using a variety of techniques explained below.

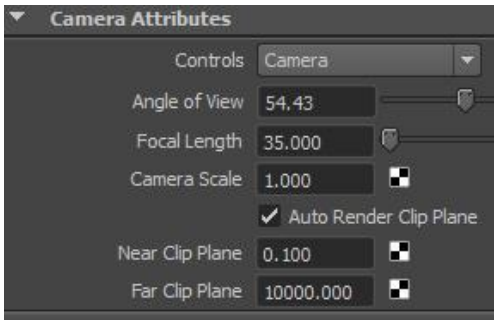
Modelling to Scale



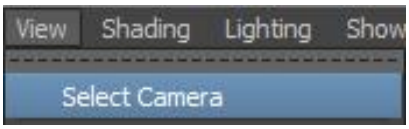
It is important to model to scale where possible. Especially if you plan to integrate realistic lighting and/or dynamics into your scene – these both rely heavily on accurate physical models. The first step is to decide what units to work with within Maya – In your *Settings/Preferences > Preferences* window go to the *Settings* category and choose your working units based on what will be most suitable for your current scene.



If the Maya scene is set to metres, then a box that is one metre cubed in real life should be one unit long. Likewise if we are working in cm then a person should be 170 units tall. This may seem outrageously large when we view it in Maya, especially compared with our grid which is now tiny in comparison. We can easily go and fix this however, by going into our Grid Options box *Display > Grid* □ and adjusting the Length and Width of our grid, and more importantly (because this will affect your top, front and side panels also) how often you want Maya to display grid lines.

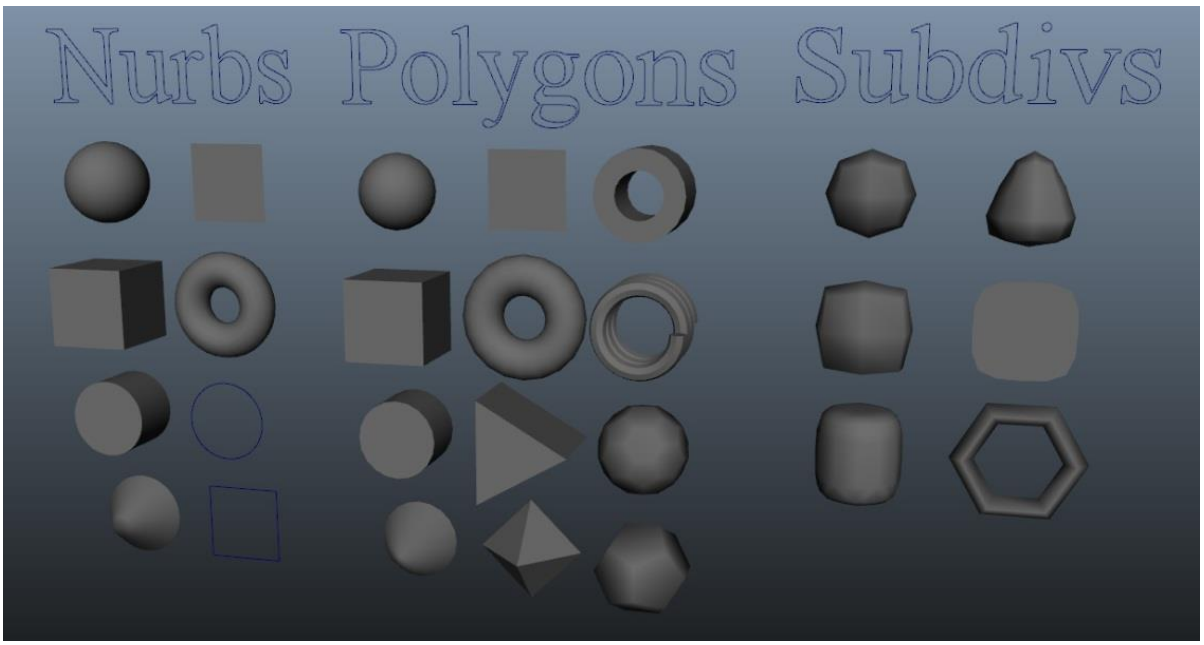


One note to make is that if you are modelling to scale, you are likely to 'outgrow' the camera's range of view. The camera has a limited distance that it can view objects which is called the *Far Clip Plane* and can be adjusted in the attribute editor with the camera selected (remember you can select the current view camera in the viewport menu by selecting *View > Select Camera*).

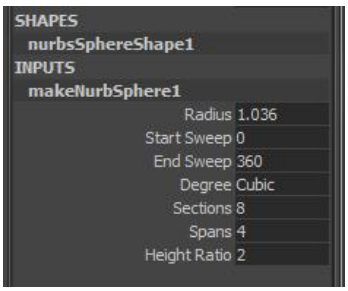


Primitives

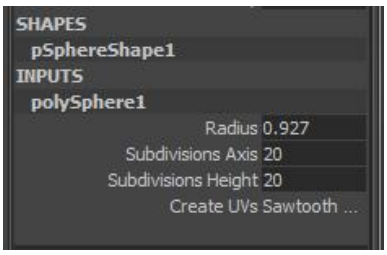
A Primitive is often the starting block of any model and will take a relatively simplistic form to be later sculpted into something more aesthetically appealing. The initial level of detail and starting geometry can be altered offering the modeller a greater level of control. Using the channels box (when initially created) you can edit aspects such as divisions, sections and spans, however once the geometry has been edited this feature cannot be used as you will notice some abnormal behaviour



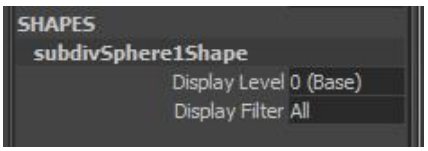
NURBS



Polygons



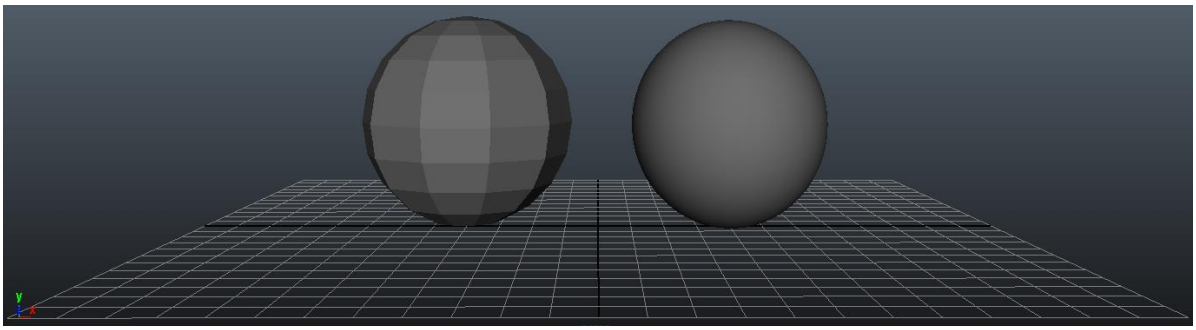
Subdivs

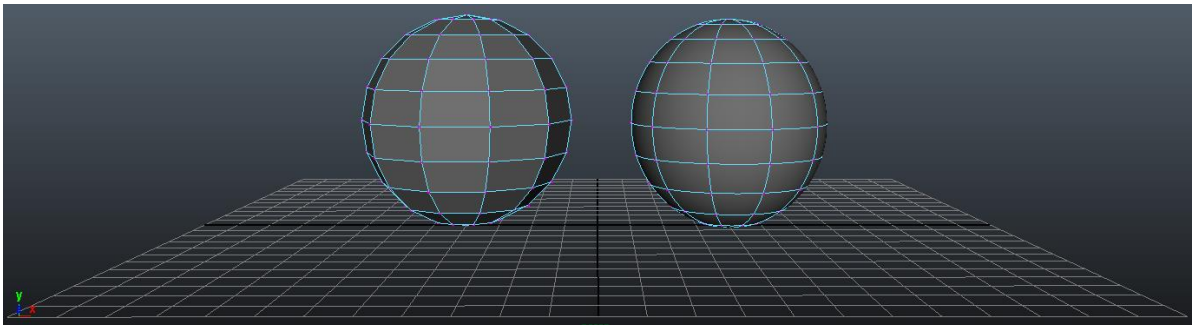


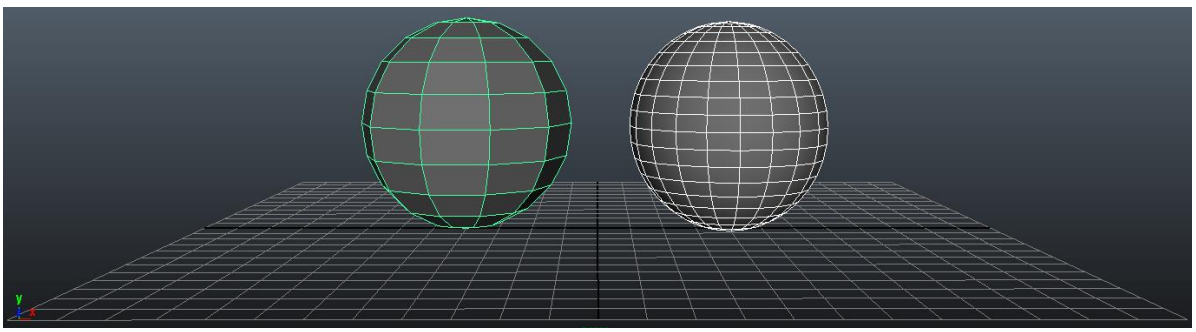
The Basics of Modelling

Polygons

Polygons allow for a simplified overall Modelling process, you can make fantastic looking models with only a handful of key tools. Its universal Modelling form often allows a simple interaction with external packages like Zbrush and Photoshop. Creating "*hard edges*" is relatively effortless and often requires considerably less geometry than other means. Unlike NURBS, polygons will retain their form until rendered where at this point you can create a smoother appearance (first image). The biggest challenge arises when used within an organic model as each individual polygon is independent and just one edge or vertex out of alignment can become noticeable. It makes for good practice to keep polygon models in their lowest possible resolution and only to smooth (last image) in the final stages otherwise your scene can become unnecessarily large.



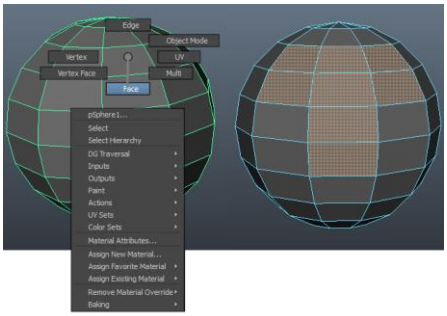




Common Polygon Modelling Tools and Techniques

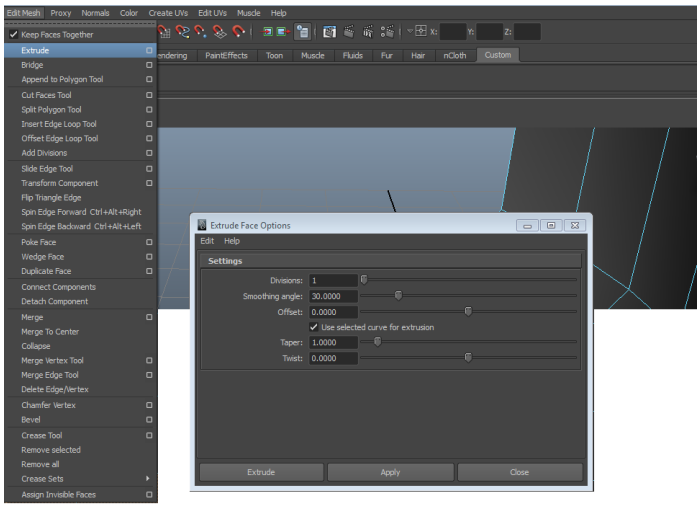
Extrude

Possibly the most commonly used tool within polygon modelling which allows you to create additional faces and manipulate them accordingly.



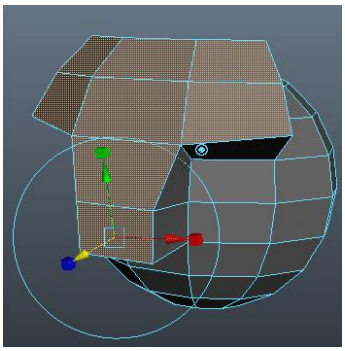
Using the Tool:

- *RMB* (hold)
- Select Faces
- Highlight appropriate faces



- Navigate to (*Polygon Menu set*) *Edit Mesh > Extrude* □

You will then be presented with a type of universal manipulator



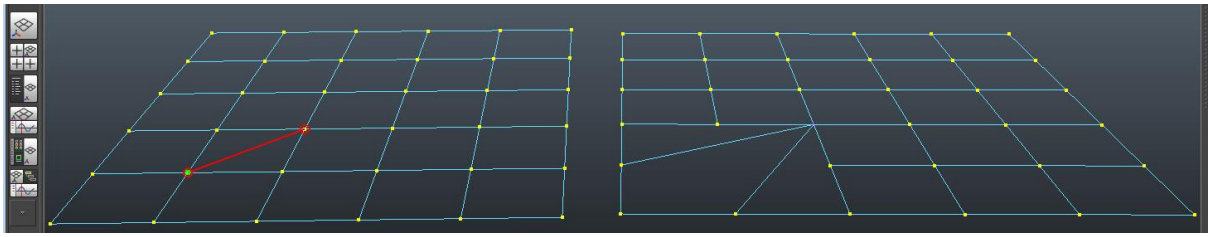
You can use the Translate, Rotate, and Scale attributes of the manipulator to move the extrusion in relation to the object. If however you would like your manipulator to act in accordance to the scene defaults (i.e. be able to extrude perfectly along the Y axis) before you click on the manipulator you will notice a little circular icon just outside the rotation ring. By pressing this you can alternate between the two, we recommend you try this so you have a better understanding before moving forward.

Merge Vertex / Edges

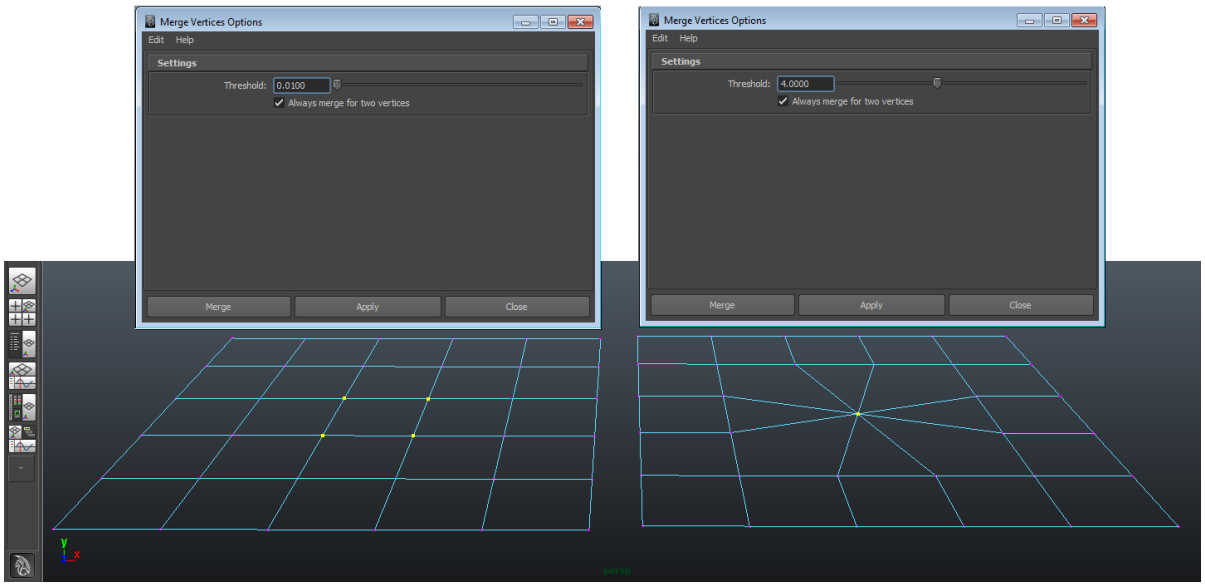
Vertex

Firstly we'll take a look at merging a vertex as they have two varied ways to get the same desired effect.

1. **Interactive Merge Tool** – As you would assume is allows you to merge them interactively, to do this select your object then navigate to *Edit Mesh > Merge Vertex Tool*. You will notice the selected object will transfer into Vertex mode and you now have a + as a curser. To merge a vertex to another click on the one you want to move initially then select the target vertex illustrated below. If you click on the tools option box you can chose if it merges to the target or the mid-way point between the two vertices...



2. **Merge Vertex Tool** – By default this tool will merge a vertex to a central position amongst those selected prior to the merge. You are also able to merge multiple points at once by shift selecting or highlighting a large section. To perform this action, navigate to Edit Mesh > Merge □. The slider represents the tolerated difference where it will actually perform a merge, if a vertex is too far away from the target to nothing will happen, if nothing happens increase the range and merge again.

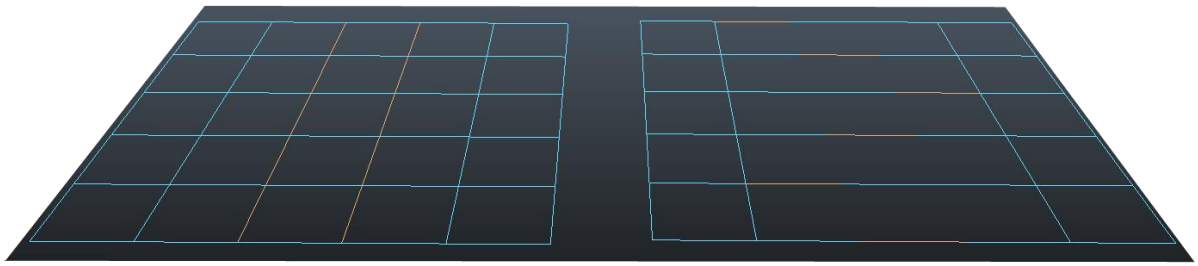


Edges

Merging an edge is less commonly used and to do so is a very similar technique to that above but instead of selecting a vertex you select and edge *Edit Mesh > Merge Edge Tool* □.

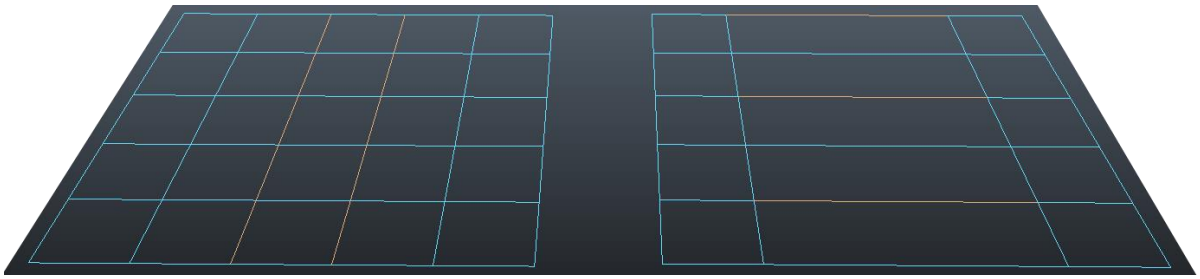
Deleting a Vertex or an Edge

Polygon geometry is made up of edges, where those edges cross paths you will find vertexes and faces make up the gaps in between. By removing a face there are no further implications as the surrounding geometry is untouched. If you remove an edge or vertex there are potentially side effects such as those illustrated below. You can clearly see that if the two long edges are removed along the centre by pressing delete, they leave behind the intersections making one line appear as three.



Broken Lines (Vertex points still

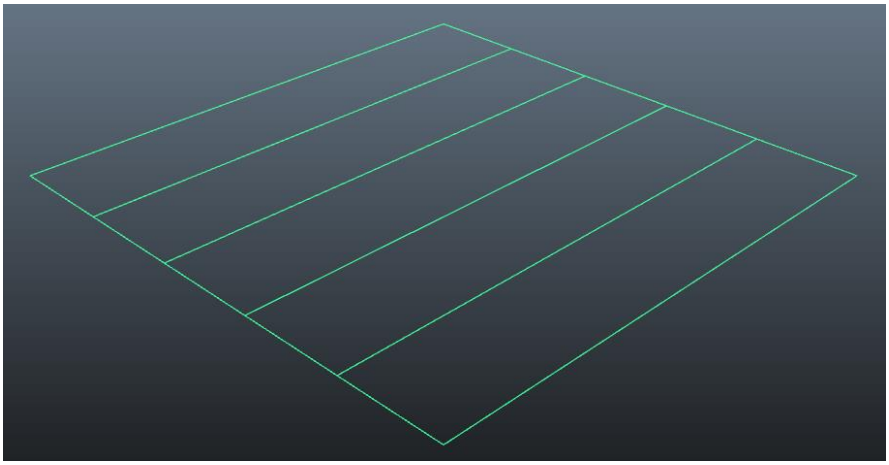
The resolution to the above issue is simple, you select the edges in the same fashion but instead of using the delete key you have to use the *Delete Select Vertex / Edge Tool*. This way you will not leave behind any unwanted geometry, keeping your model as clean as possible.

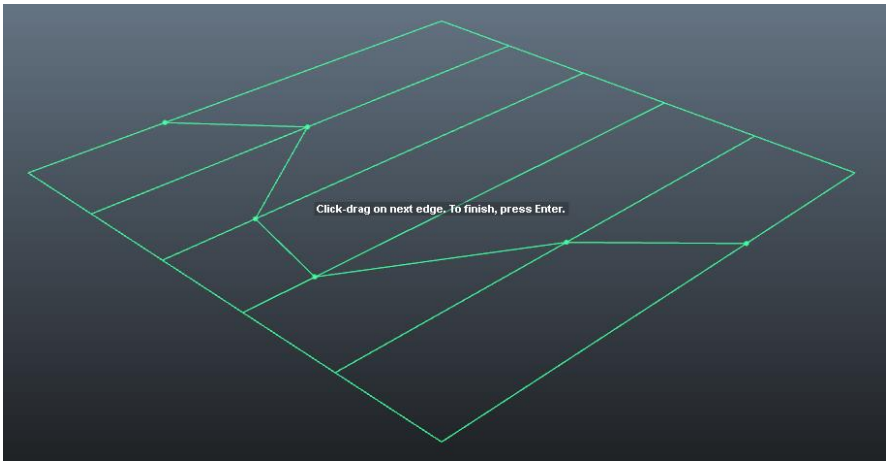


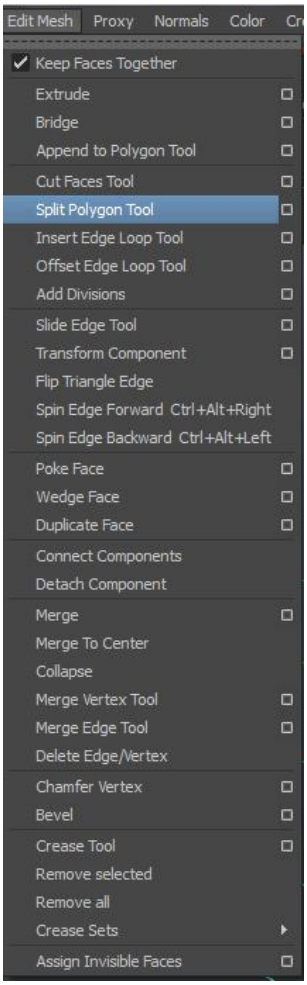
Complete Lines (Vertex points

Split Polygon Tool

This tool is used for splitting individual faces of polygons, often used when altering the topology (edge flow) of any character/model. We will delve further into its uses at a more appropriate time. If you wish to access this tool, navigate to (Polygon Menu Set) *Edit Mesh > Split Polygon Tool*, once you are happy with your split; press the enter key to finalise.

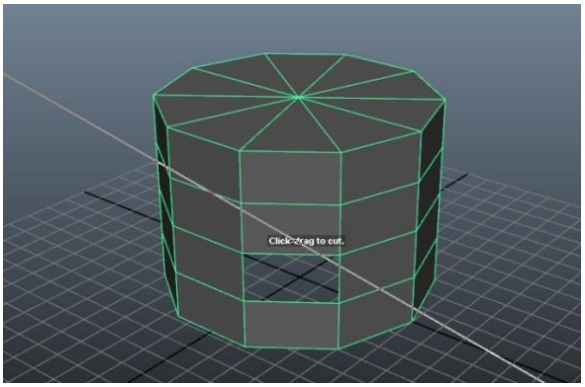


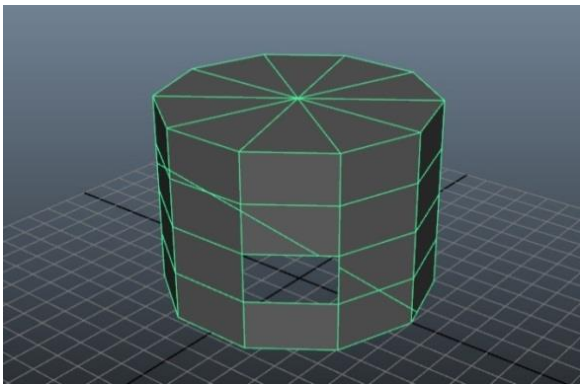




Cut Faces Tool

This tool quite literally allows you to slice through your geometry paying no particular attention to edge flow, shape sides or holes in geometry:



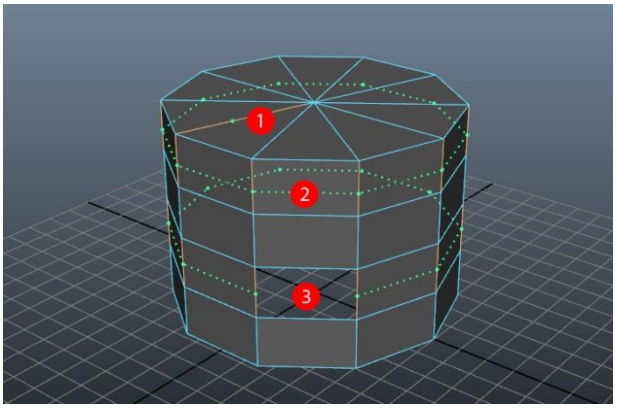


Edge Loop

An "*Edge Loop Tool*" as the name would suggest adds a complete looping edge around your geometry, whilst taking into account edge flow unlike the "*Cut Faces*" tool. The loop will complete full circle (2) unless it meets one of the following issues:

- Poly face that doesn't equate to having 4 sides (1)
- Geometry ends or vertices/edges may not be joined (3).

Note: *If you were expecting it to loop around and it doesn't, investigate the area in which the flow seize to locate your issue.*



Common NURBS Modelling Tools and Techniques

The initial few techniques (Loft, Revolve and Boundary) involve creating curves then using them to create the required geometry. Therefore before we delve into Nubs familiarize yourself with the functionality of curves and how they need to be constructed.

CV Curve Tool

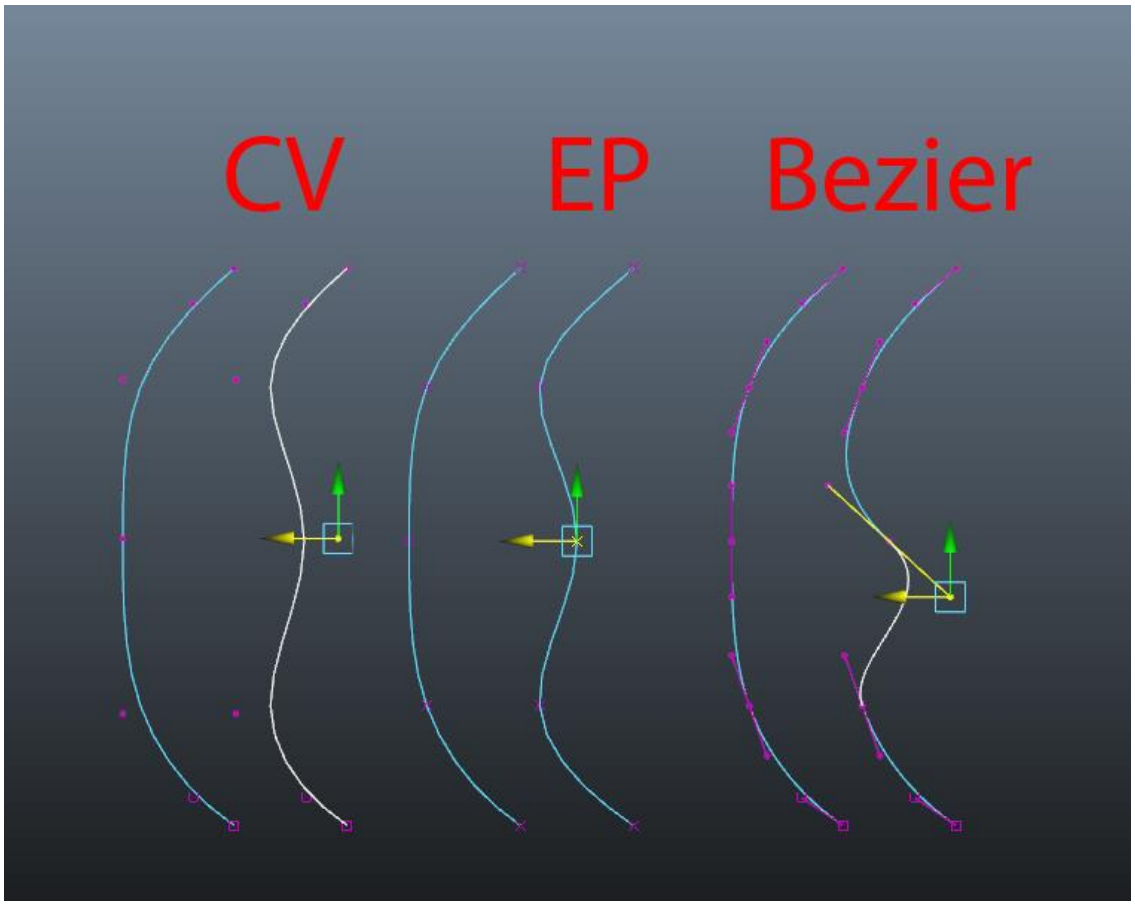
CV Curves in essence are NURBS Curves with editable *Control Vertices*; each vertex however will not lie directly on the curve, as each of them represents points on a control lattice that encompasses the entire curve. You can however enter Edit Point mode to adjust vertices within the curves *Marking Menu*.

EP Curve Tool

Edit point curve tool will create curve points to follow exactly, you can however select CV's within the *Marking Menu* which will then alter the way in which the curve is interact with.

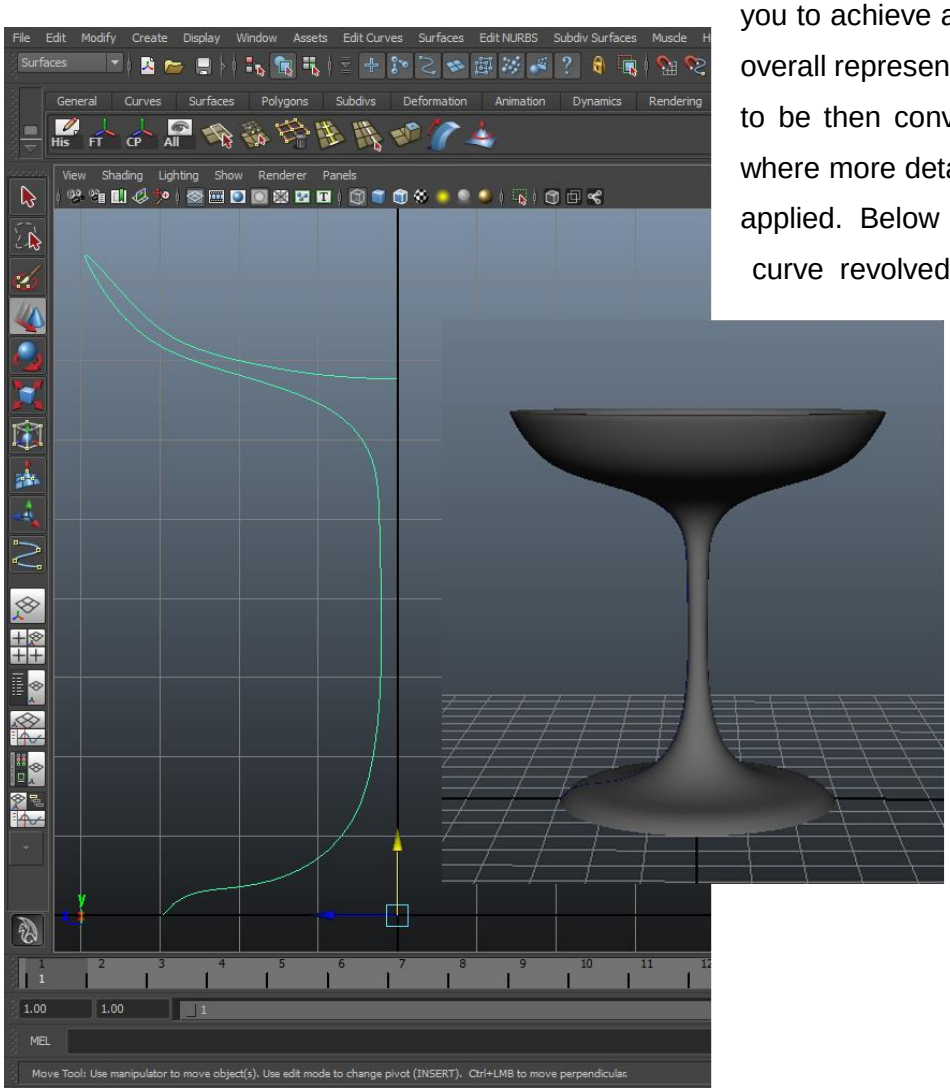
Bezier Curves:

Bezier curves have always been used within Mayas Hypergraph however these have been recently transitioned to the surfaces toolset. Bezier Curves are most commonly used in computer imaging software for smoothing out curves, but gained their notoriety when dealing with vector images as they provide the ability to be scaled indefinitely.



NURBS & Curves

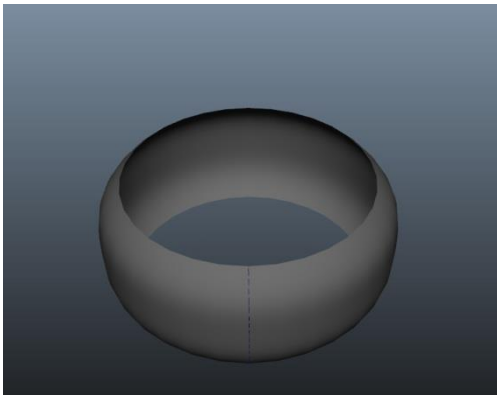
NURBS (Non-Uniform Rational B-Splines) are very different to your conventional modelling tools; whereas polygons are rigid; NURBS create objects using curves and surfaces allowing for an interactive interpolated surface and are often used to re-create organic objects such as people and detailed mechanical objects. A popular modelling technique using NURBS it patchwork modelling, whereby you create sections of a model separately and joining them at them end (If necessary). They are also used in the early stages of Modelling as they enable you to achieve a quick and accurate overall representation of your model, to be then converted into polygons where more detailed aspects can be applied. Below you will see a CV curve revolved into a wine glass, and can hopefully appreciate how quickly some shapes could be achieved using this and similar methods.



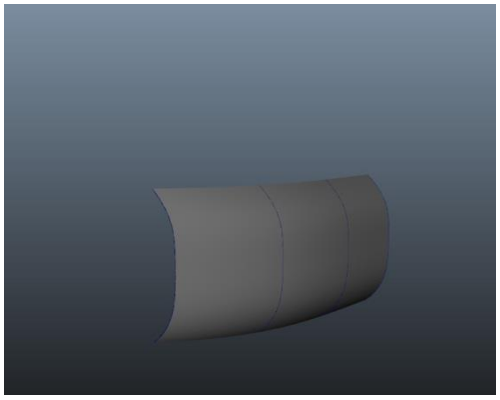
Creating a Curve

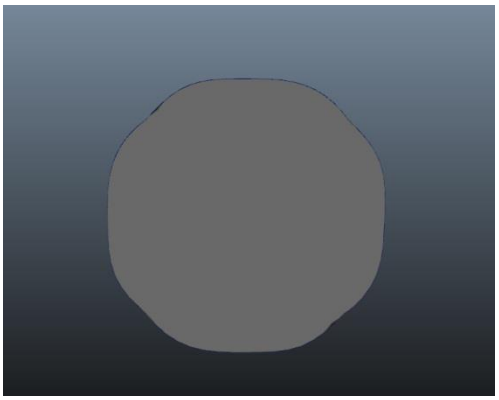
To simply select either (CV or EP) curve, navigate to the Create menu ensuring you are (generally) in an orthographic camera (perspective camera not recommended for drawing curves), and repeatedly click until you have created the shapes outline. It makes for good practice that whenever you have a corner you apply at least 3 curve points to define it.

Loft, Revolve, and Boundaries

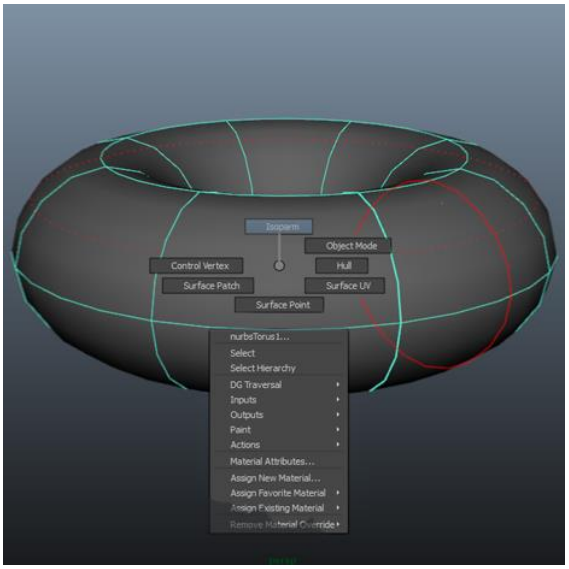


These three common techniques all stem from the creation of one or more EP/CV curves with the purpose of creating editable geometry. A revolve will use only 1 curve and in order to create its geometry it will "*revolve*" around its centre pivot (by default this will be the centre of the grid 0,0,0) creating elliptical objects. A loft uses two or more curves to create a surface panel and will subsequently pass through any amount of specified curves which will ultimately define the overall panel shape. The final one is the boundary which requires a closed group of 4 Curves (selected in a clockwise formation), creating geometry within the specified area.





Isoparms



These allow you to add additional geometric lines providing more control over the specified area. When you have your NURBS shape, hold RMB to enter the marking menu, and “isoparm” (being the top option) will allow you to start placing them. Click on an already existing line to illustrate which curve you would like it to follow (horizontal/vertical) then drag appropriately, holding shift if you would like to enter more than one. To illustrate all pending isoparm placement you will see dotted yellow lines, to finalise the process and insert the isoparms navigate to *Edit NURBS > Insert Isoparms*.

Control Vertex

The same as a CV point on a single curve, you can manipulate each CV for each curve that makes up a NURBS surface.

Hull

A Hull is essentially a row or column of CV’s that makes up a NURBS surface.

Surface Point

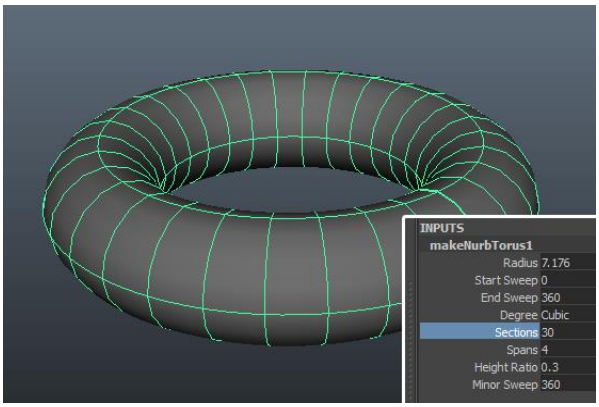
As simple as it sounds, this indicates a specific point on a surface. Although this cannot be manipulated, you can select points on the surface in order to insert isoparm’s for example.

NURBS Primitive Settings

NURBS primitives by default are created with history based on a few common attributes. We will cover the most useful of these below. By creating the object with history, this also means that we can change the settings, such as the radius, even after the objects has been created, moved and modified.

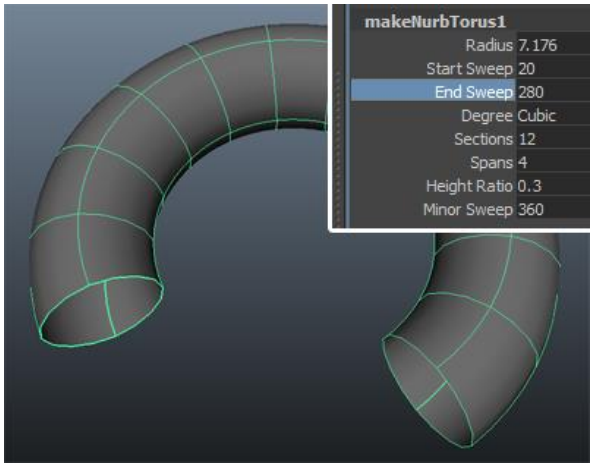
Sections

Sections are used in spherical objects to control the resolution used; i.e. the number of isoparms to define the shape as demonstrated on the right.



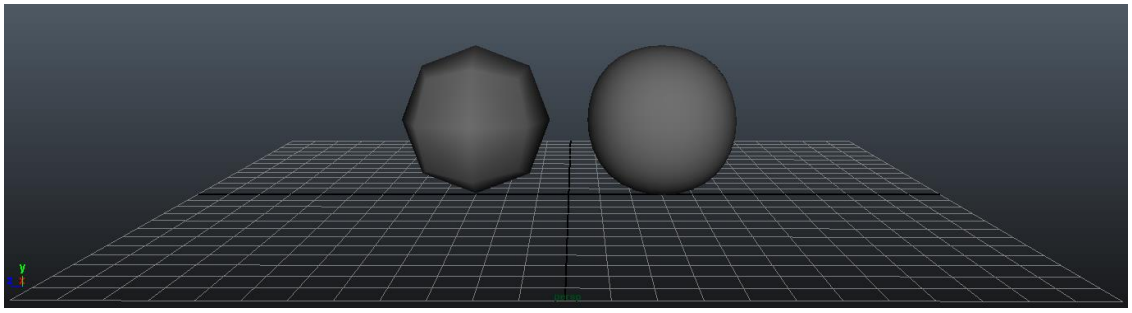
Start sweep / End sweep

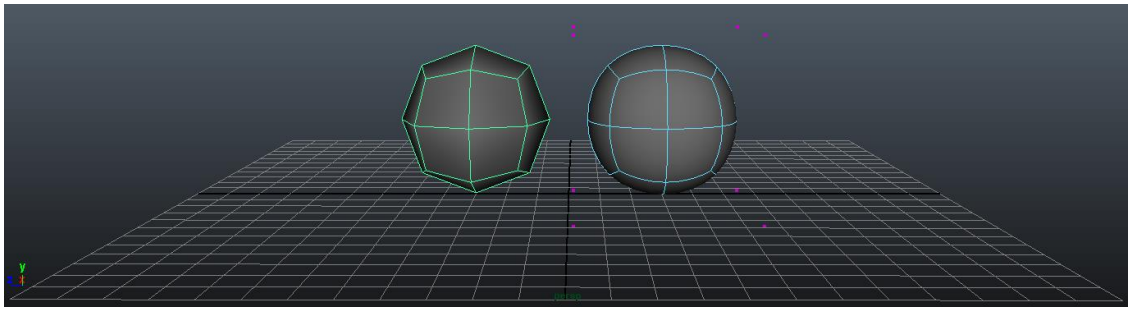
Spherical objects also have start and end sweeps. This is the angle at which the loft that creates these surfaces starts and ends as demonstrated on the right.

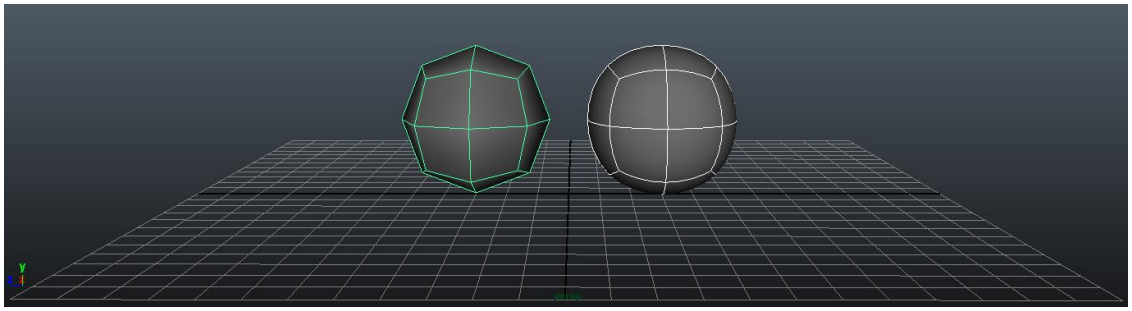


Sub-Divisional Surfaces

At a glance sub-divisional surfaces can often appear to offer the best of both worlds; allowing you access to the simplified polygon modelling form but the manoeuvrability of a NURBS surface, thus making it a popular method for creating organic models. A big advantage is you can work in hierarchies from “0” (Base level) upward, but it’s wise not to enter above level 2-3 if you wish to convert to polygons at any point. You have fewer vertices to manipulate the object but the hierarchical structure provides you with the ability to add detail without adding geometry, but they are not without their limitations. You will notice a vast reduction in tool set in comparison to polygons, but the biggest discrepancy of this method is undoubtedly the increase in render time, some results may be worth the increase but if you have a busy scene and limited time it could be worth considering converting them to an alternate form.







Render Layers

Similar to the above organisational structure this feature allows you to split up your renderable objects onto various layers or what is often referred to as render passes. Rendering in layers is an essential tool especially when it comes to the compositing stage as we will discuss in greater depth

Layers give you a visual hierarchy which is useful in all areas of animation. This can be used to minimise the geometry in the scene, hide the rig etc.

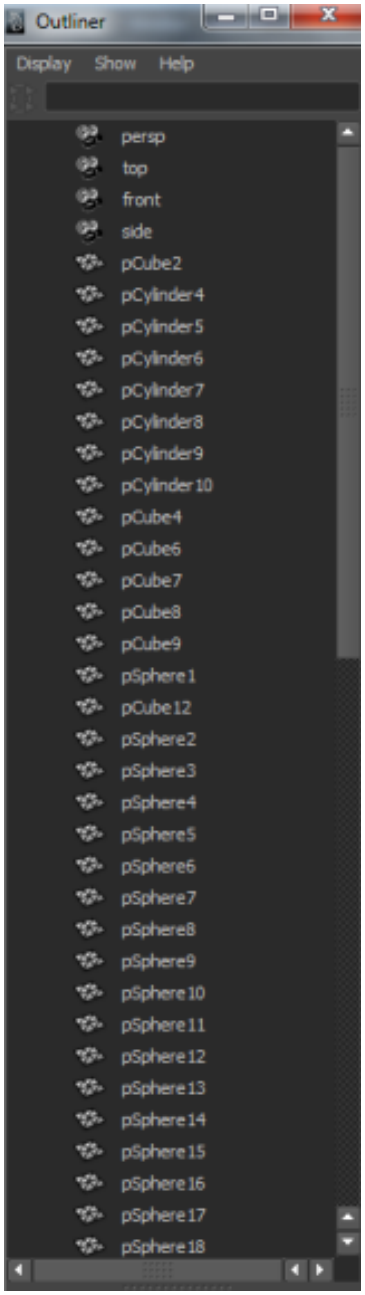
How to use the outliner:

Task	Details
How to access	*Window > Outliner*
Select a node (object)	LMB click on the node
Select multiple nodes	LMB drag selection or Ctrl/Shift+LMB
Create a group node	Select group contents then press Ctrl+G
Reorganise objects between groups	MMB drag the node to the desired node
Remove a group	Highlight group and Ctrl+Shift+G
Parent a node to another	MMB the child under the parent
Un-parent a node	MMB the child away from the parent
Name / Rename a node	LMB Double Click, then type in new name

As a continuation from the introduction chapter on page 32, the outliner will help you to ensure you're organised and structured. Initially this may seem pointless and mundane but by using simple naming techniques and hierarchies like those demonstrated below it will prove invaluable to you and colleagues as you progress towards more complex scene files.

The below examples are taken from the same scene, hopefully you will understand why we are stressing the importance of this aspect.

Untouched



Organised

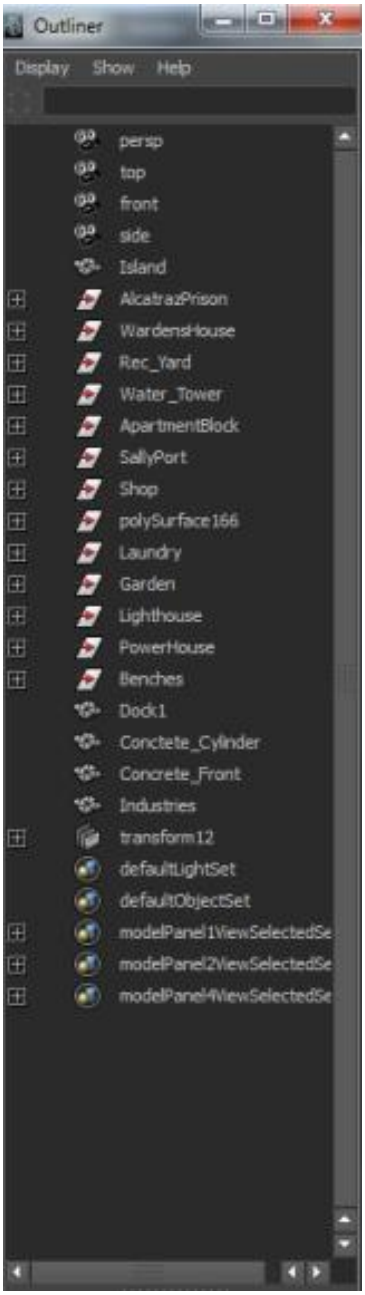


Image Planes

Image planes can be very useful at the modelling stage; they allow you to keep the model accurate to their original drawings and maintain consistency throughout. An excellent website for acquiring these for free is www.the-blueprints.com here you will be able to download image planes for vehicles, electrical items, weapons, and various other miscellaneous items.

On the other hand if you are solely working on a project as opposed to within a client based production team you may find it easier to have a few reference images of similar models, then use them as well as your creativity to model something truly original.

Image Plane Setup

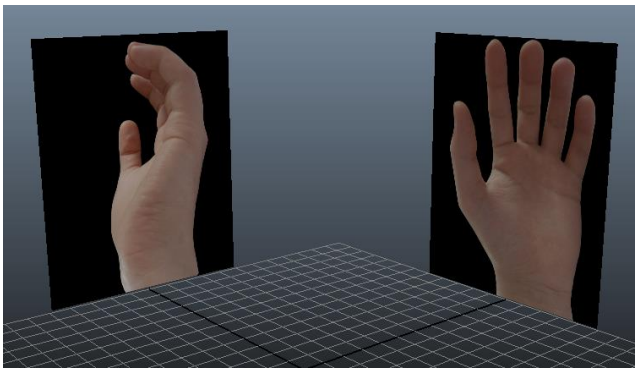
Note: I*f you need assistance with preparing an image plane in Photoshop for use in Maya please refer to page 109*

Once your image has been prepared we will be looking at two methods of setting up an image plane.

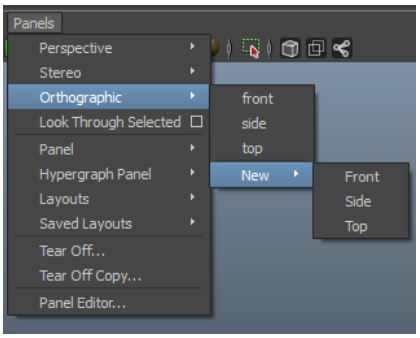
Importing an image onto a plane

1. Create polygon Plane
2. RMB > Click and drag down to Assign new Material > Lambert
3. *Common Material Attributes > Colour > Chequered box > File > image name* (select file by clicking on the folder icon)

Note: *you may need to alter the UV settings (pg 81) or update the size of the image plane (*Attribute editor > PolyPlane > PolyPlane History > width/height*), to replicate exact dimensions.*



Importing an image onto a camera



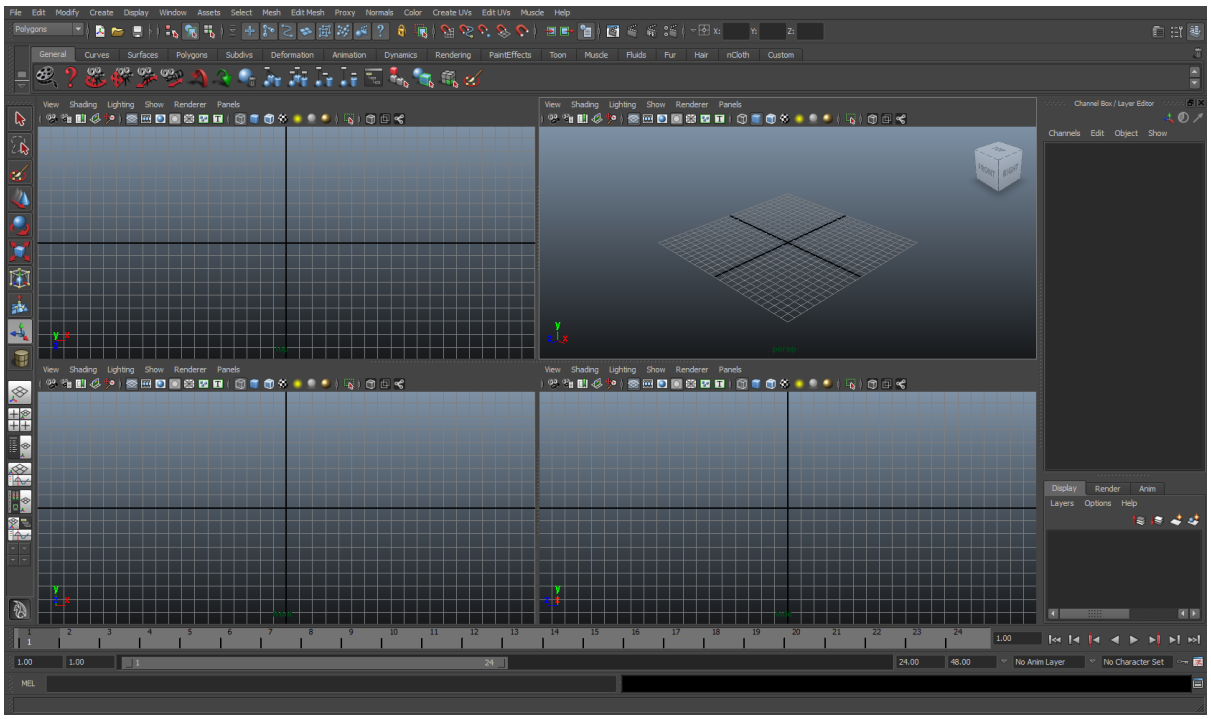
For good practice you can set up new orthographic cameras to hold reference image, however this comes down to personal preferences. To set up new cameras: **In view pane:** *Panels > Orthographic > New...*
1 x Top (bottom may be required in some circumstances)
1 x Side (only 1 pane is needed for symmetrical models)
2 x front (you will need to rotate one of these 180° as well as adopting a suitable naming strategy [Img_Front, Img_Rear etc.).

Import your images onto the camera: *View > Image Plane > Import Image*

at this point in the attribute editor check the *looking through camera* option as opposed to "all views" (you might want to do this bit last so in the perspective view you can see the theory behind what happens next.

Change 2 viewpoints to the new front and rear cameras (leaving the persp cam): *Panels > Orthographic.... (Select Cam)***:**

By doing this you should have a similar setup to before, only some of your viewpoints are the new cameras



View > Image Plane > Image Plane Attributes... Placement Extras > Centre... (Adjust the value in the last box (*Z* axis) to conflicting values one as 20, and one -20 (this may need to be adjusted depending on your grid size.

Summary

What this has effectively done is put a camera and image plane on either side of the grid. You will now only see the relevant image planes from the appropriate camera.

Anatomy and Physiology

Having an understanding of the basics of anatomy and physiology will undoubtedly assist you when it comes to creating your own characters; as allowing your edge flow to mimic human muscle structure is pivotal if you wish to achieve true to life deformations.

You should research anatomy images and pay particular attention to muscle structure if you want to achieve realistic deformations.

Tips before we begin Modelling Anatomy

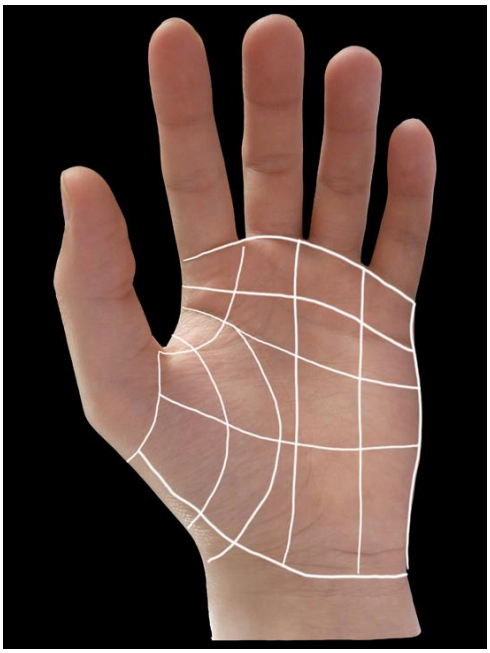
- Ensure you feel comfortable with the basics of polygon modelling as terms such as, extrude, merge, combine, and edge flow are used but not explained in this area of the book. Please refer to pages 49 - 59 for more information.
- Familiarise yourself with the physiological makeup of the particular aspect you wish to model, as it will help you pre-visualise edge flow.
- Having the correct edge flow from the outset will reduce the need for the split polygon tool.
- Remember that image references for organic models may not always align perfectly like you can often get with blueprints. Therefore be aware of variances in angles, size and perspective; using them only as a guide to help maintain overall consistency.
- Have real life images at the ready, if you're using dual screens; plaster one side with reference images, as they will all assist you.
- Remember once you have your base humanoid mesh you can manipulate this to adhere to the requirements of your specific model proportions (i.e. gangly arms)
- Remember when using polygons, to use your smooth preview (using numbers 1,2 or 3) to help pre-visualise your model you can model in any mode but we would always suggest referring make to the base mesh to "keep thing tidy".
- Finally and possible one of the key points in modelling **Keep the geometry as low as possible for as long as possible**.

Human Hand

Arguably one of the most challenging aspects to master within Maya is the creation of the human hand. There are a multitude of “best practices” out there and to be honest we’ve seen professional results achieved from many of them. The most common techniques reside around a box, having the fingers extruded from the front and the thumb from the side; however hands do not have this physical makeup as in reality your thumb takes up over one third of your palm. Therefore the more time you spend modelling; you’ll discover these techniques (as with ours) are open to interpretation and what you learn from one technique, book or tutorial may not provide you with everything you need. As stated earlier the approach we will take is based around edge flow from the outset meaning the general physiological makeup is there from the very beginning making the process seemingly more logical.

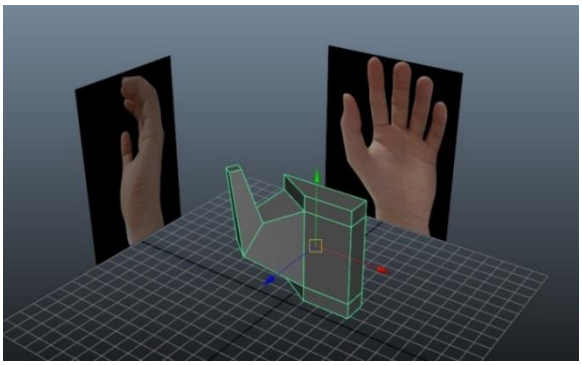
Before we begin:

- Familiarise yourself with the edge flow.
- Ensure you have set up image references within your scene (pg 62).
- Use multiple real-life images.
- No hands are identical.
- The thumb **isn’t** stuck onto the side of a hand. Try to visualise the edge flow to help you during the modelling process (image Right).
- Moving individual vertices are imperative to achieving a true to life organic representation.
- Fingers should subtly vary in size and shape.
- Throughout the modelling process ensure you smooth preview to help you manipulate your geometry.



Chapter 2: Modelling

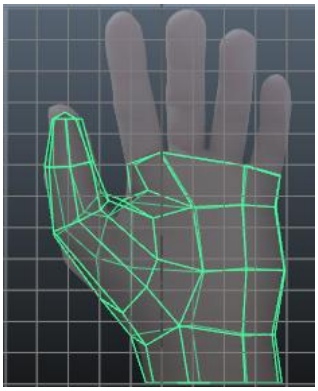
Step 1: Creating the base edge flow (palm and thumb)

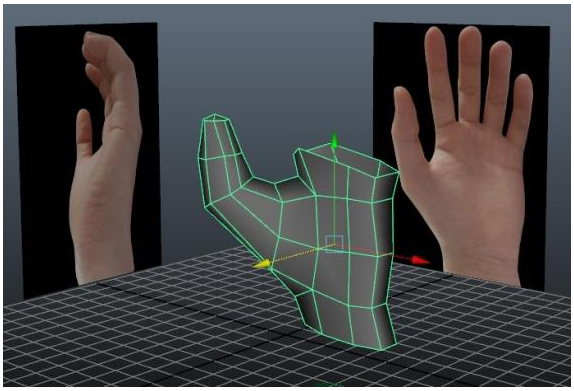


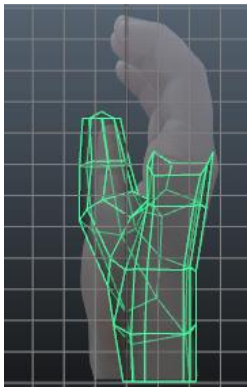
- *Create* > *Polygon Cube* (1 x 1), *Extrude* the top and bottom faces then position the cube to cover the palm.
- Take the front LHS row of vertices, move right towards the centre.
- Extrude the LHS centre section (thumb), scale in, rotate up, and position slightly to the left
- Extrude in the same point, again scale in and position higher (being the tip of the thumb

Step 2: Shaping the Geometry

- Remove the top and bottom faces then perform a smooth, which will enable you to shape the hand further, whilst ensuring basic shape of palm and thumb are achieved.
- Manipulate your geometry whilst remembering the image references are there to assist you during this early phase.

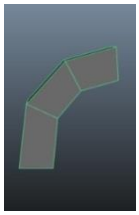
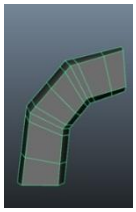
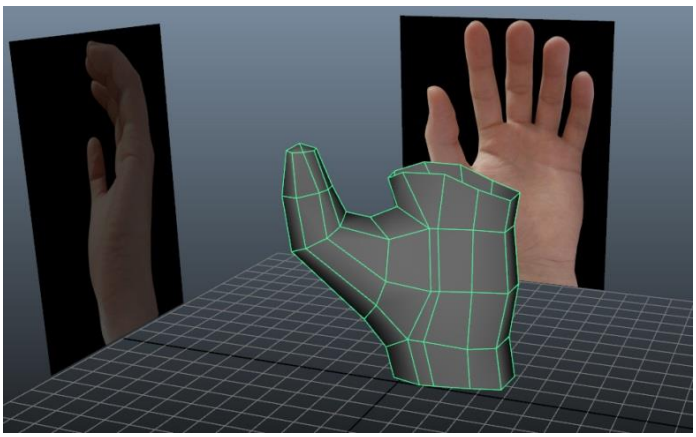
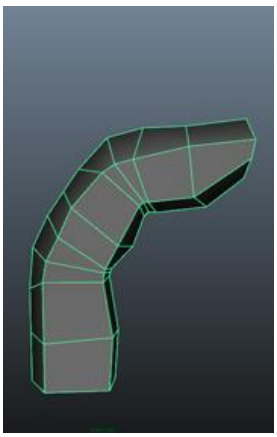






Step 3: Creating the finger

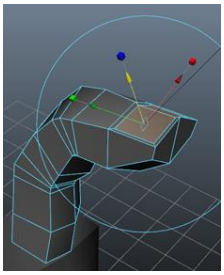
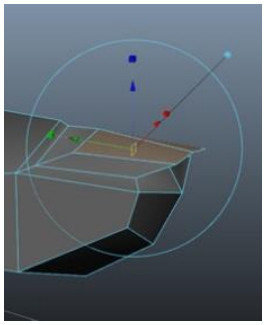
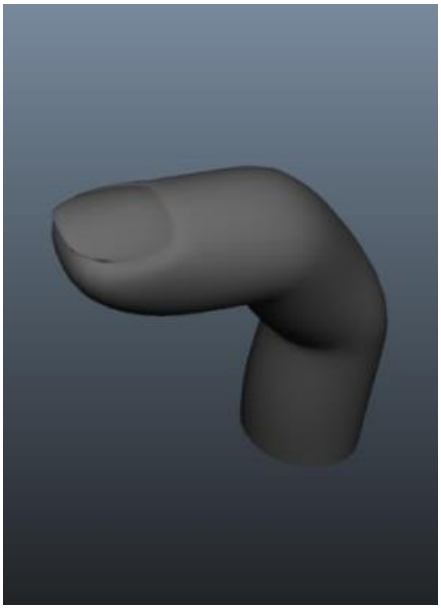
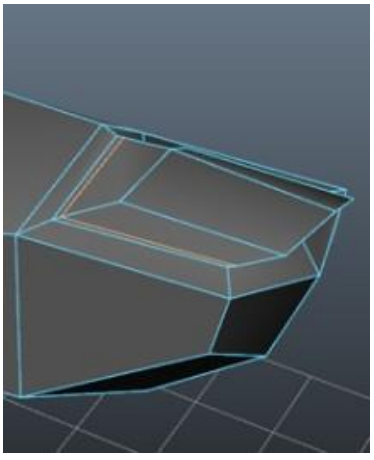
- *Edit Mesh > Insert Edge Loop Tool* to create the vertices required in the centre of the palm.
- *Create > Polygon Primitives > Cube* add 3 divisions to create the finger.
- Add no more than one edge loop down the centre and on each side of both knuckles, ensuring you keep the underside geometry close and the upper geometry spaced out.

- To create the fingernail, select the top two faces and extrude them inward and then down slightly. Move the foremost 3 vertex (of the finger) down and forward to assist in shaping the fingertip.

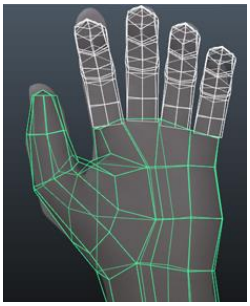
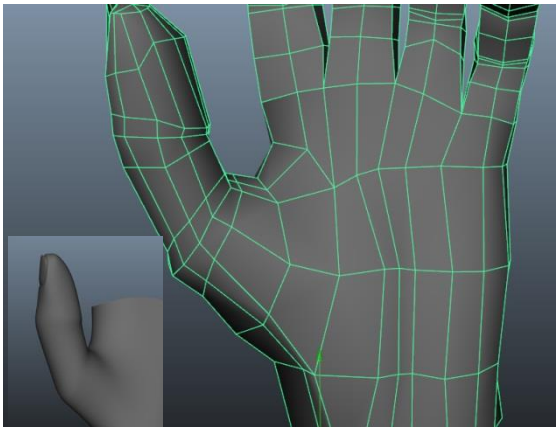
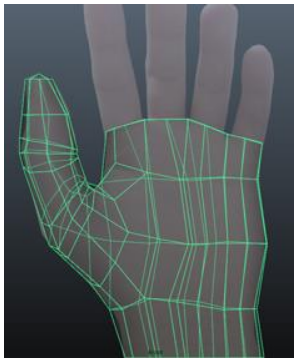
Note: *Nails can be made as separate geometry if you would like.*

- Then select the same two faces and extrude them up, in and forward, creating the nail. Depending on the look at this point you may wish to bring in the edges slightly and add in an edge loop to define the nail.

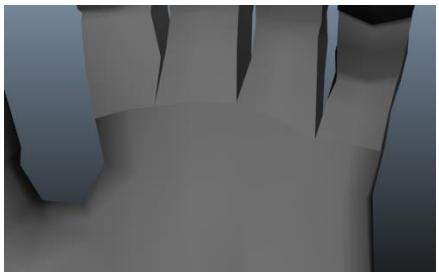
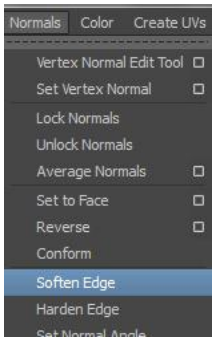





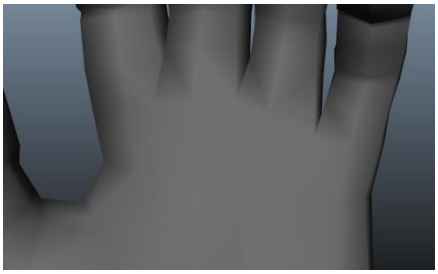
Step 4: Combining the Geometry

- You can (for a uniquely organic feel), replicate step 3 for each individual finger or for convenience duplicate it (*Ctrl-d*) three times.
- Spend some time applying the skills just learned to the thumb.
- Once you are happy with your thumb, and have all your fingers aligned add in extra edge loops on the palm to compensate for the ones we added in the centre of each finger.
- Before joining the geometry, ensure the scale has been adjusted appropriately spend a little time altering the fingers vertices independently to uniquely differentiate them from one another.
- Pick one finger, snap the corresponding vertices together (*v-MMB*). Combine them one at a time (*Mesh > Combine*) then merge the vertices (*Edit Mesh > Merge*) repeating for the remaining fingers.

- You may come across the below issue when combining geometry. To rectify navigate to *Normals > Soften Edge*.

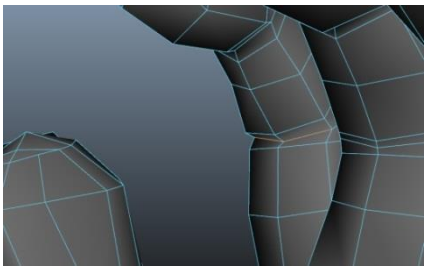
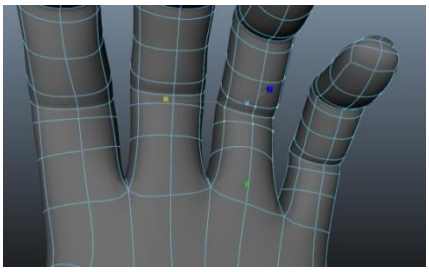
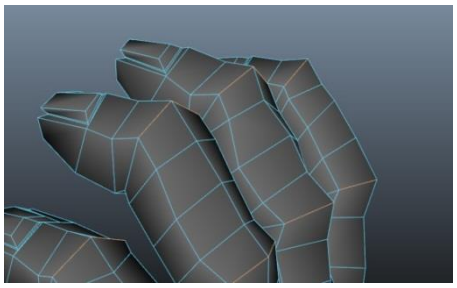
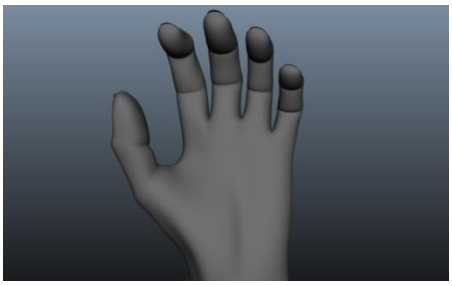





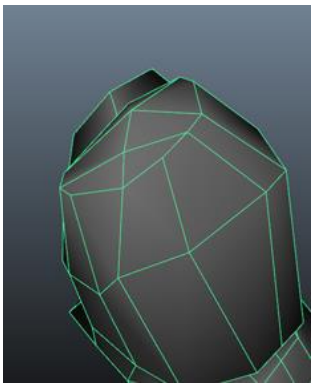
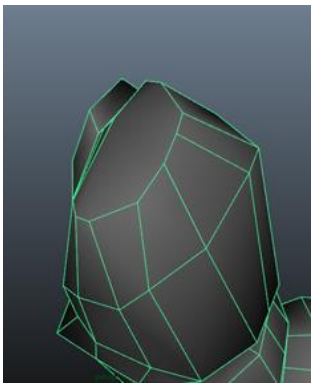
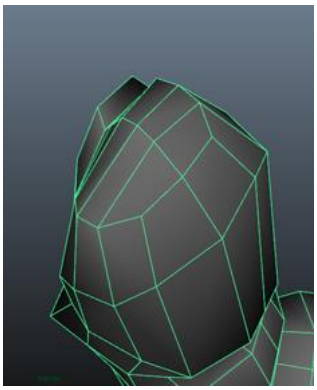
Step 5: Tweaking the Fingers

The most common issue when people model hands in the concept of “Bendy Fingers”, which means they appear to have little or no joint definition and curve rather than bend. The reason we’re addressing this now rather than before we duplicated the fingers is because this section really helps give individual character, especially if you duplicated them from the original.

- In the crease use the edge tool to move the centre edges inward.
- The knuckles:
    - Use the Edge tool to loft them above the rest of the geometry
    - Use the scale tool to increase the width around the knuckles.

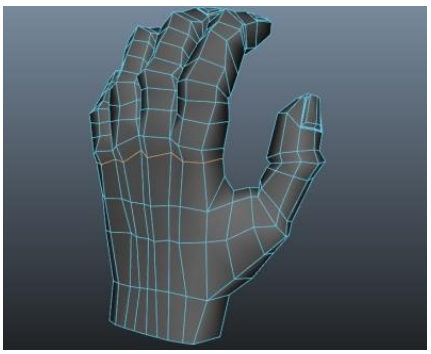
- Fixing the thumb Geometry: It appears that on our model when we added an edge loop for the fore finger to be attached we arrived at some unusual geometry at the tip of the thumb. To rectify RMB click *Edge* then *delete all edges* on tip, then using the *split polygon tool*, rebuild them.

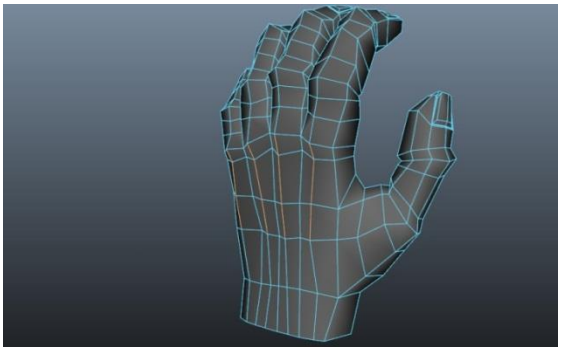




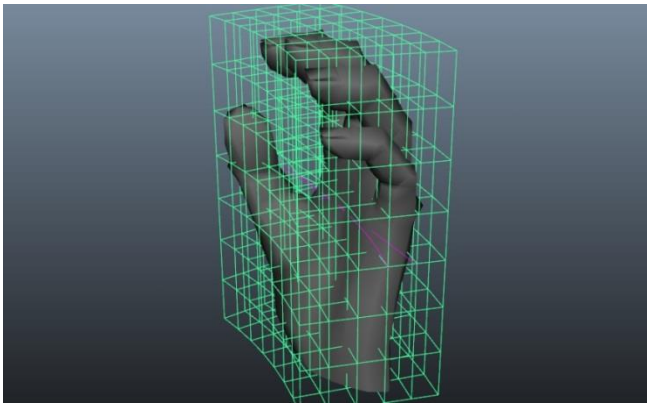
Step 6: The Back of the Hand:

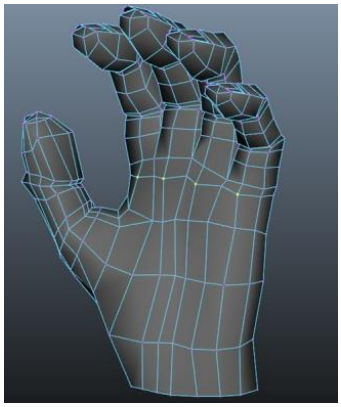
The last major section to address is the rear of the hand

- Add in an edge loop at the top of the palm
- Raise the edges that run down the back of the hand
- An optional step is adding a little extra curve to the model. To do so select the Animation Menu Set then highlight the hand:
    - Delete History – *Edit > Delete by Type > History*
    - Create Lattice –*Create Deformer > Lattice*
        - *S Divisions **8** T Divisions **8** U Divisions **6***
    - Add Bend deformer - *Create Deformer > Nonlinear > Bend*
        - *Rotate X&Y **90** Curvature **-0.3***
    - Delete history (once happy)





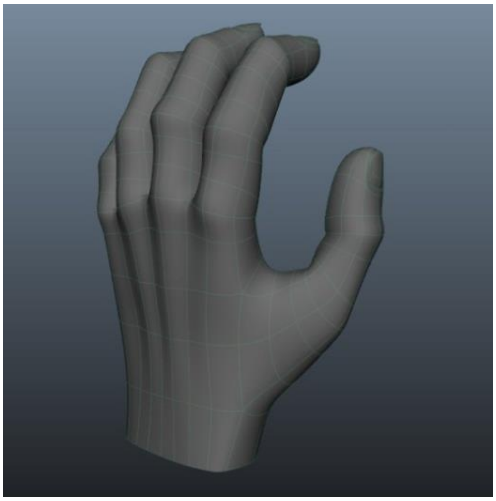


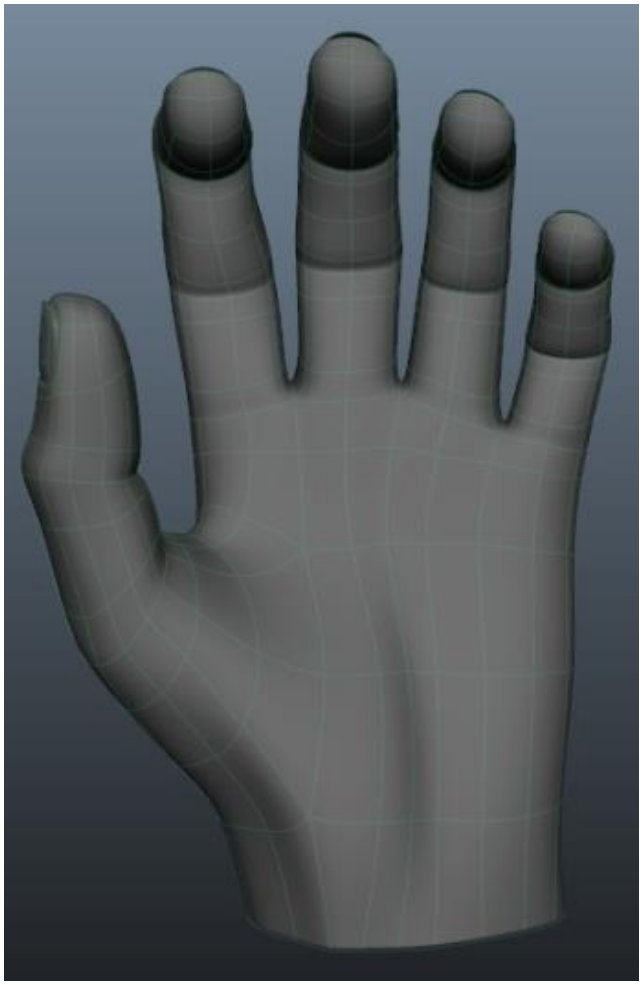


Conclusion:

Currently the model is relatively low resolution and no re-arrangement of edge flow was required, however using the skills you have just learned you can spend some time perfecting your geometry by either manipulation or the addition of more edges (if required).

Note: *To add more acute detail you should consider using textures. Jump to page 97 where we take you through setting up a basic skin shader.*





Human Ear

This is an extremely difficult section and will need some time before it's truly mastered, we recommend spending time between sections to ensure you're happy with your geometry. The ear at first glance appears to have geometry cropping up all over the place and can be difficult to gauge a starting point.

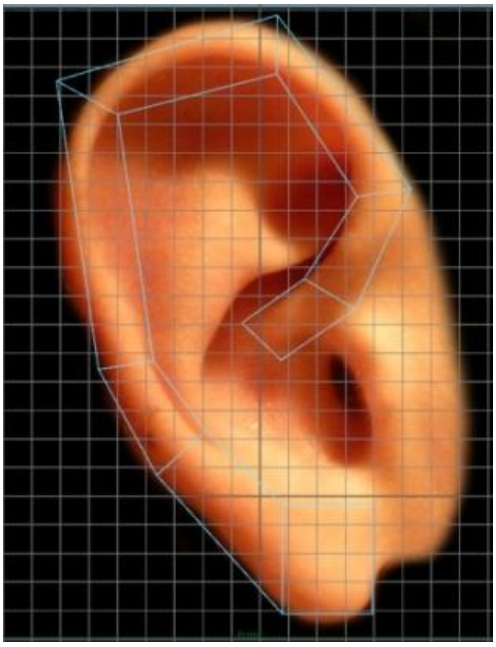
Before we begin:

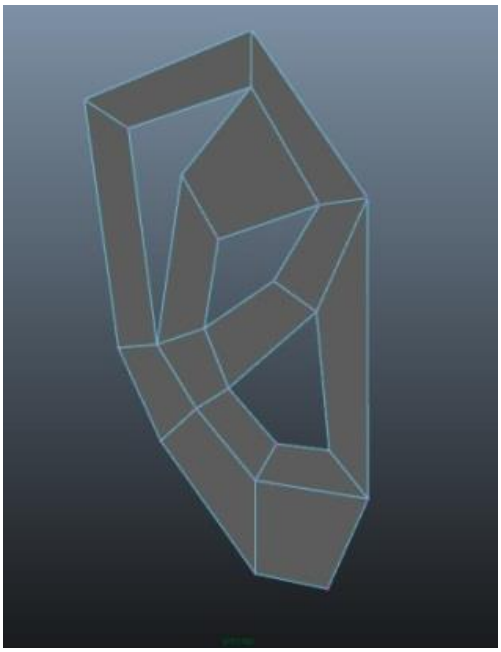
Familiarise yourself with the starting edge flow we are working towards (two interlocking curves almost like an *e & c*)

Either acquire some pictures or take some of your own for use as references. It is important that you are confident with basic modelling techniques for this section in particular.

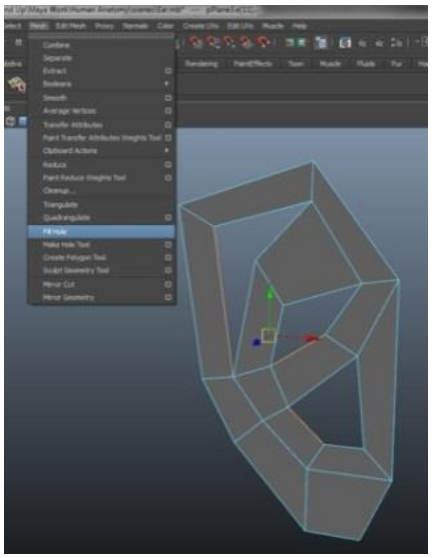
Step 1: Base low-res geometry

- *Create* > *Polygon Plane*, position it upright over the earlobe base.
- Extrude the LHS then rotate and position them in accordance to the below illustration.
- Extrude the top RHS into the centre (creating similarities to above) and then down alongside the geometry you just created, then up around what will be the ear cannel.
- Snap the vertices to the ones they run alongside and merge together.





Step 2: Depth and shape

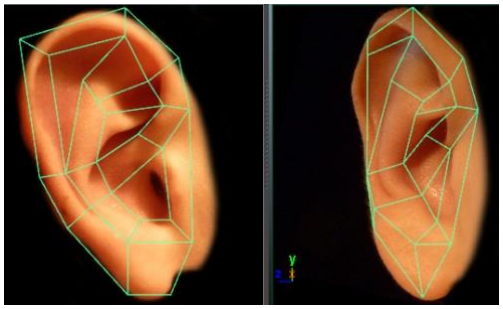


Select one edge from each of the three sections that contain a hole *Mesh > Fill Hole*

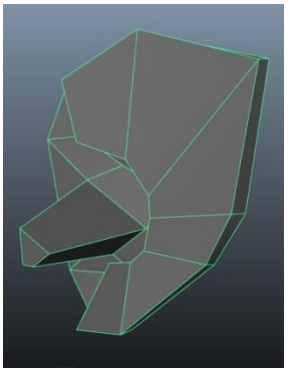
- Split the inner ear poly to create the fork like shape illustrated on the edge flow picture.

Note: *At this point it leaves a 5 sided shape but this will be addressed in step 3.*

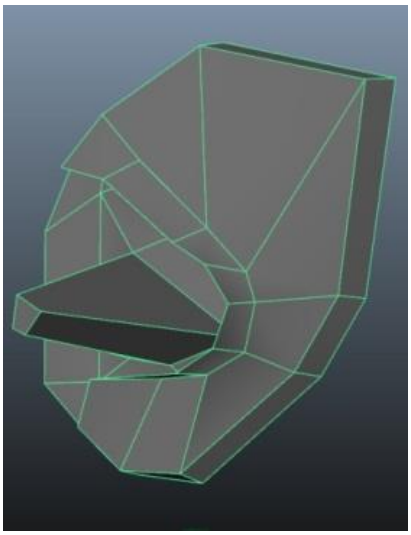
- Align your vertices in the side panel to help shape your ear and select the face over the ear cannel and extrude in and back.
- Starting at the bottom select all edges until you reach the split that you recently added.

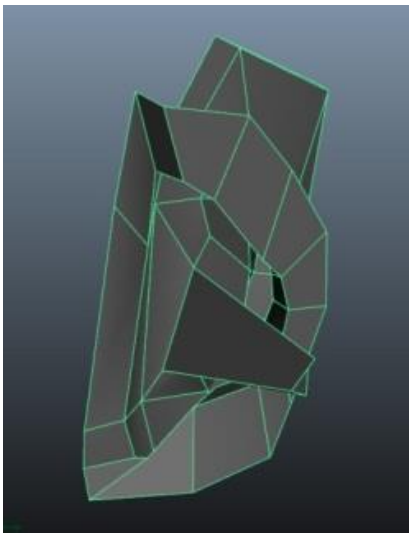


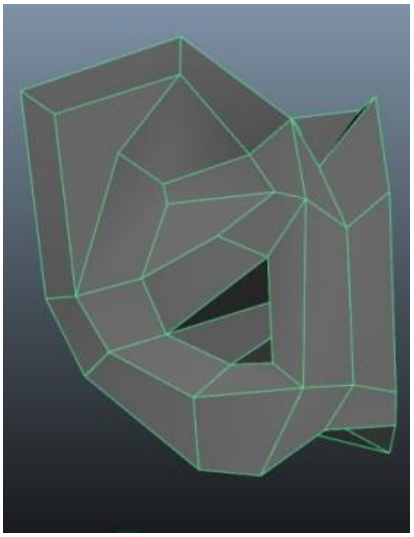
Extrude back then repeat (g) and extrude inward giving it depth and creating the back of the ear



- Continue by selecting the edges you missed of last time as well as all the edges you've just made on the back of the ear and extrude again
- Spend some time shaping the extrusions one at a time using both the front and side perspectives.
- Select the Edge tool and double click on any of the inside edges to select the entire loop, extrude one last time, and fan out the edges.

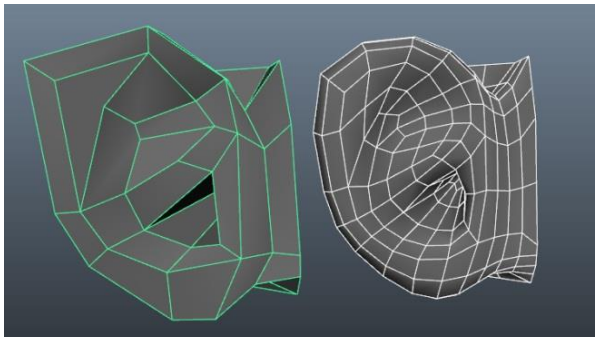


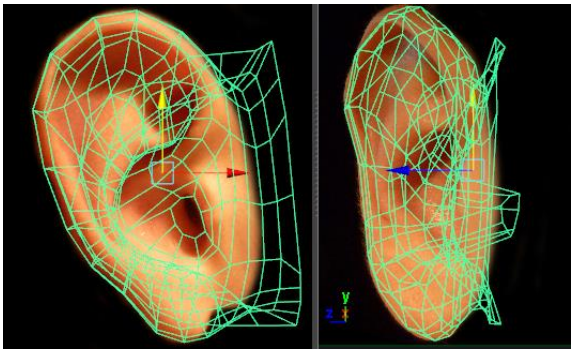




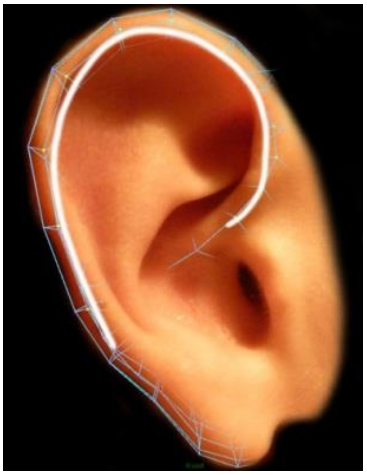
Step 3: Defining the ear and accentuating the lip

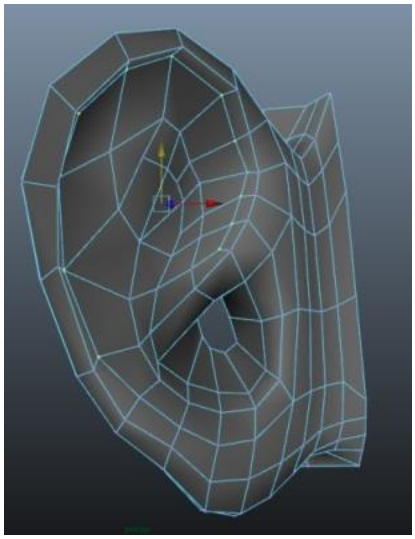
- The first issue we'd like to address is the rectification of our 5-sided polygon. Using low base meshes initially provide us with a little leniency to create such shapes due to the fact when we smooth (effectively cutting every polygon in half) the issue is automatically rectified. Now we have achieved a suitable edge flow as well as a sufficient amount of geometry, it allows us to go back in and tweak our ear into a more accurate representation of our images
- Spend a little time aligning your newly smoothed mesh to the outer contours of your image reference before we begin to shape it internally.

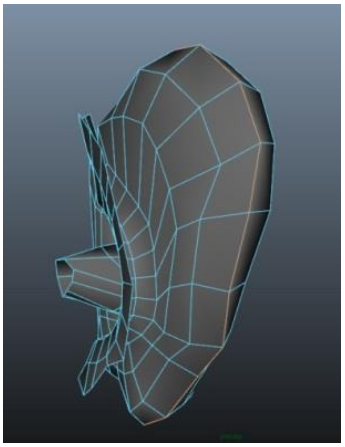




- The next step would be to accentuate the inner lip at the top of ear, therefore select the appropriate vertices, pull out and scale inward, as far as you feel necessary.
- Next move the topmost row of vertices towards the ones you have just scaled to help define the edge and move the next row of vertices located on the back to be the new highpoint.
- Select the next row of vertices in, enlarge the section and sink back into the ear as far as it will go without protruding out of the back.

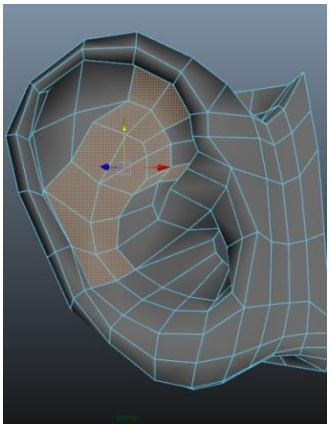
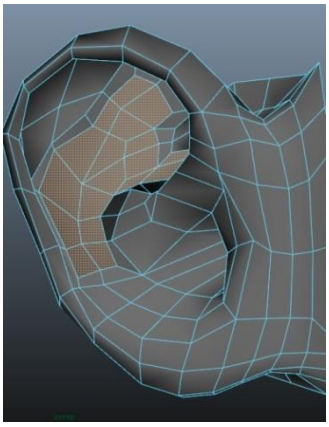
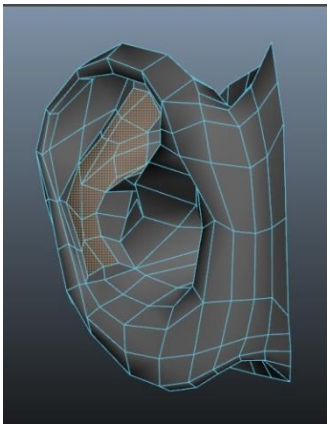






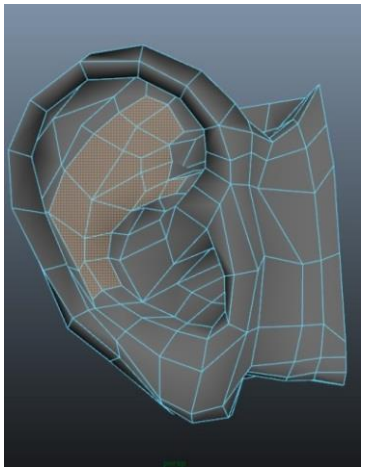
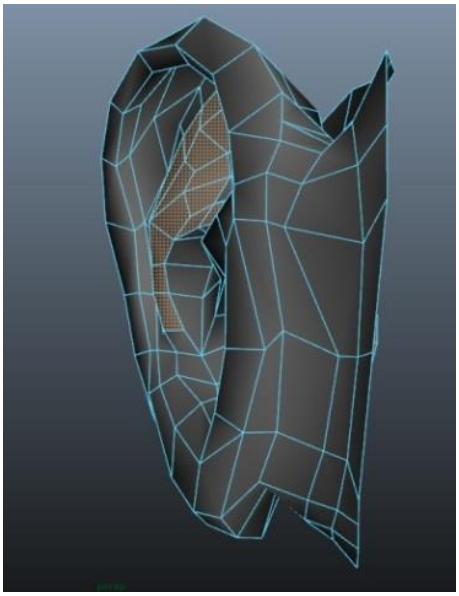
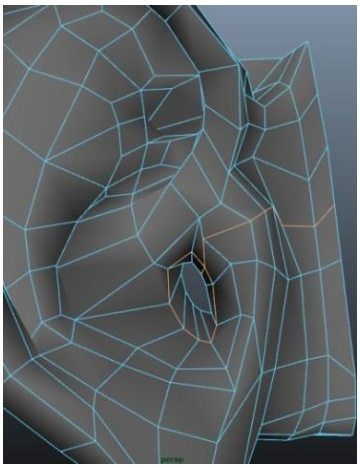
Step 4: Push, pull and re-shape:

- Pay particular attention to where the indents are in the ear and manipulate what geometry you have to position it as best as possible.
- Select the faces of the inner ear that appear raised and extrude them up, and inward slightly.

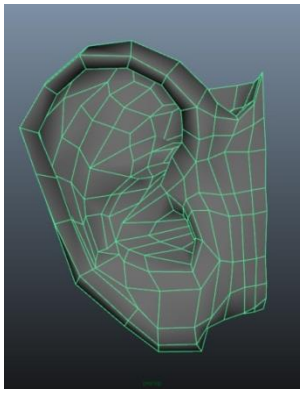
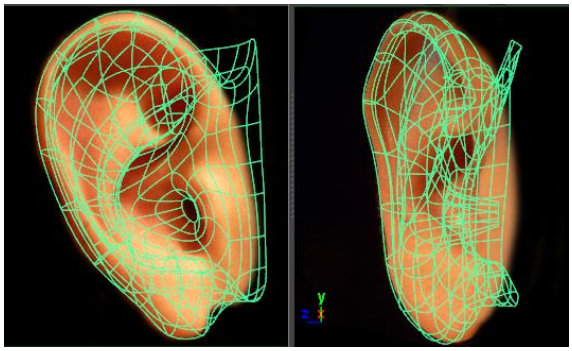
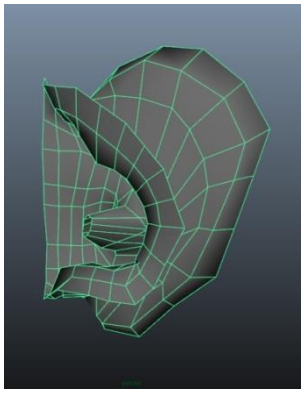
- The next step will involve you levelling out the edges of the extrusion so it flows into the geometry we had before.
- We also added in an edge loop in the ear cannel and on the upper RHS of the ear cannel to aid in deformations around that area, more specifically to help differentiate (from the original drawing) the outer and inner ear curve

Note: *Don't worry if it's taking a long time, ensure you save incrementally and you can always jump back to a previous version should you need.*

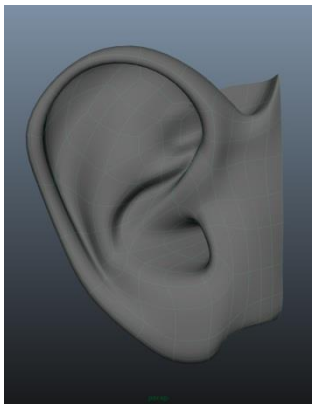
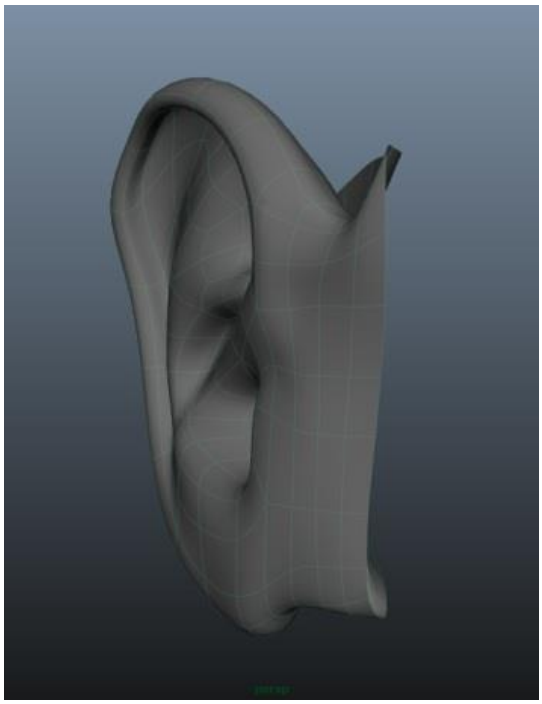
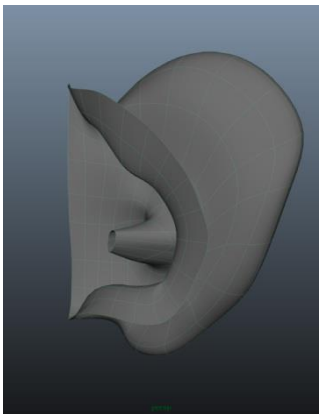
Step 5: The earlobe

- To finish off we're going to restructure the earlobe slightly to give it the round padded appearance they usually have.

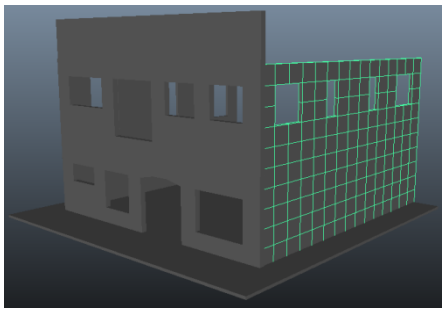
Conclusion:

The technique which we have just guided you through has a relatively straightforward structure as it derives from the initial fundamental edge flow we outlined from the introduction. The appearance of the ear at this stage may still need some modification especially when aligned with your own image reference and/or characters head, but these adjustments will be minimal and amendments will literally be tweaks.

Modelling a Saloon

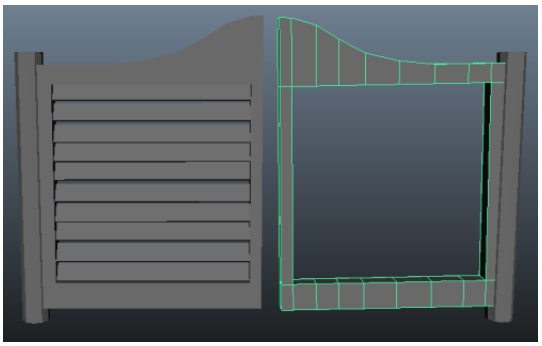
In this section we will take you quickly through the process of modelling the Saloon scene we will be using for the texturing and lighting section. We will be keeping the scene fairly simple but we would encourage you to spend a little more time on some extra detail or if you're feeling confident have a go at creating something completely different.



Step 1: Outer Building

Firstly ensure you are modelling to scale (try to do from memory if not flick back to pg 46 and take a look how). Create the outer walls and the floor. I used Polygon cubes against the grid to help me gauge an accurate / realistic building height.

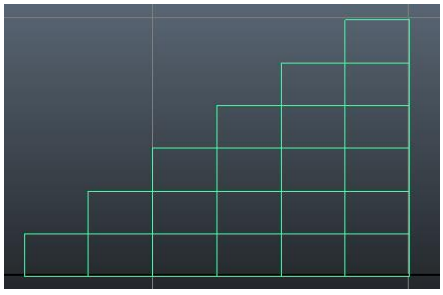
Step 2: Saloon Doors



I created a *Polygon Plane*, added in the relevant subdivisions and shaped the door. Highlight the faces you no longer need and delete them, to leave a hole. Select the remaining *faces* and *Edit Mesh > Extrude* them back to add width to the door. The centre planks are elongated cubes.

Note: *Before you duplicate centre planks and the doors themselves you may want to consider arranging your UV's (pg 81).*

Step 3: Stairs



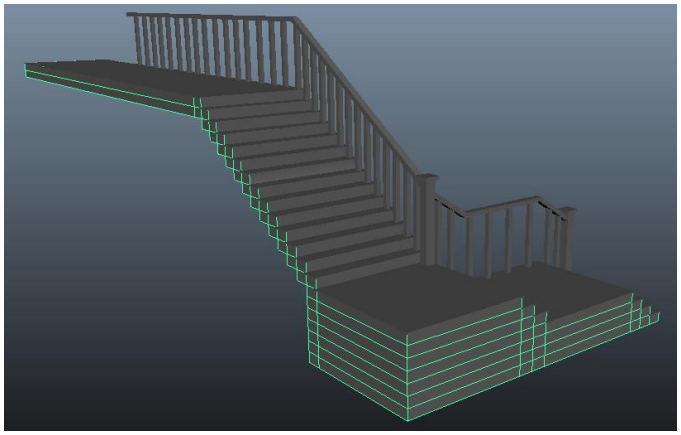
Stairs look relatively easy but they can trip you up if you're not careful. The biggest issue generally comes with scale so as a rough guide:

- Rise: 0.17m
- Run: 0.28m

Luckily for us these values fit very closely in with our grid. The Image is using a scale of meters, so 1.5 grid squares equals six complete stairs with values of 0.17m rise and a 0.25m run, which is close enough for me.

So to actually create the stairs *Create* > *Polygon* > *Plane* enlarge (using the grid as a reference) to the height you need. Split the plane using subdivisions, again using the grid as a scale. Delete the faces so you have a flat staircase (previous image), then select the faces and extrude them to any width you require creating your staircase.

However we are going to give you a little challenge as the staircase we are using in the saloon has some elongated platforms and a 90 degree turn. Using the above methods replicate this below staircase in the corner of your saloon.

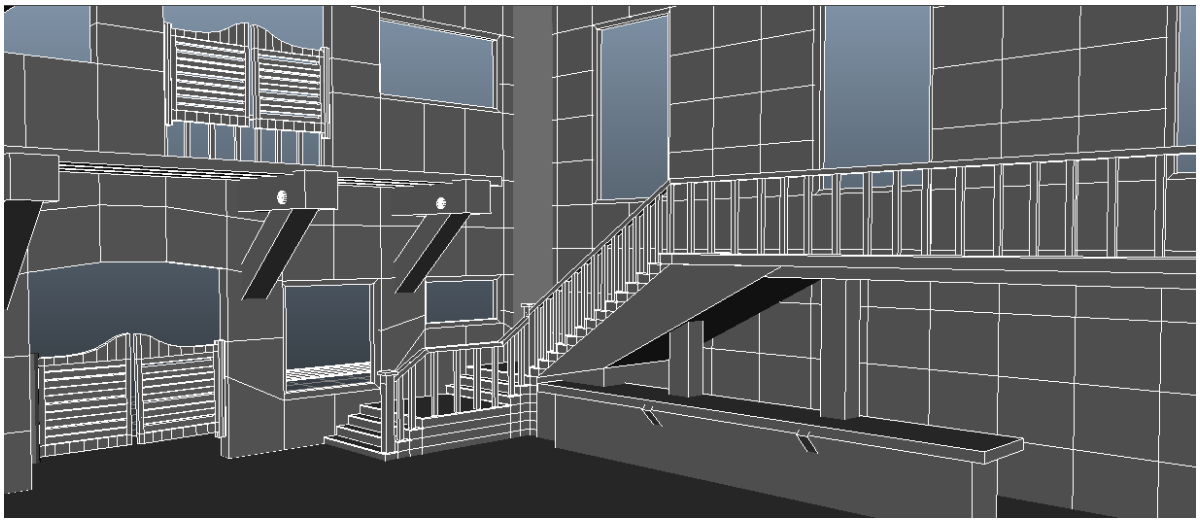


Note: You may need to merge some geometry depending on how you proceed. Also try adding in a stair rail and stepping plates (a thin piece of geometry) on each step to give it a nice finish.

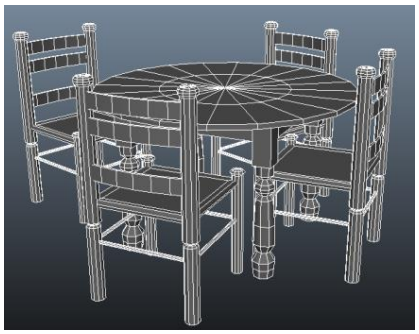
Step 4: Adding in some finishing touches

We went through and added some very simple low poly additions, we are sure you are more than capably of replicating without further instructions.

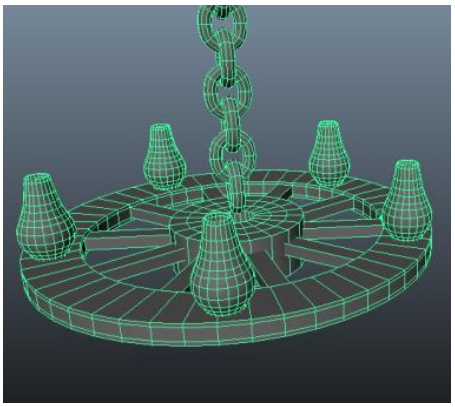
- Bar Area
- Window Frames
- Beams + Balcony (Three wooden planks)
- Duplicated the doors



Step 5: Props



It is important you spend a little bit of time populating your scene with additional items that complement the environment you have just created.



Finished Scene

The rendered image is a taster of what we will take you through in the coming chapters.

CHAPTER 3: TEXTURING

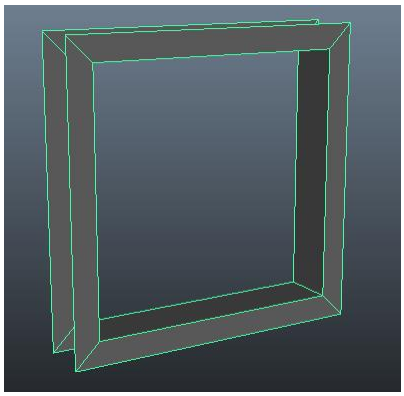
Preparing an object for a Material

UV Unwrapping Basics

UVs are a series of 2D points that reside within the vertex component information. In essence they provide co-ordinates that allow us to map a 2D texture onto a 3D image. UV's are always done after the Modelling process as further manipulation of an objects vertices will manipulate the UV's also.

Mapping UV's

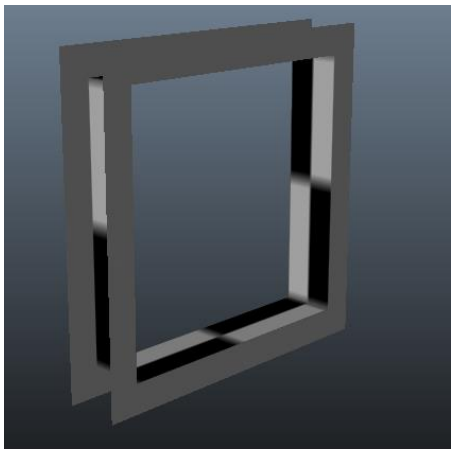
There is no glossing up this section as it's probably one of the most boring and tedious parts of texturing but also pivitol to your success. If you plan on spending some time getting that pefect texture you are going to want your Model to be prepared correctly.



I'm going to make a very quick low polygon that I would like to be a wooden window frame.

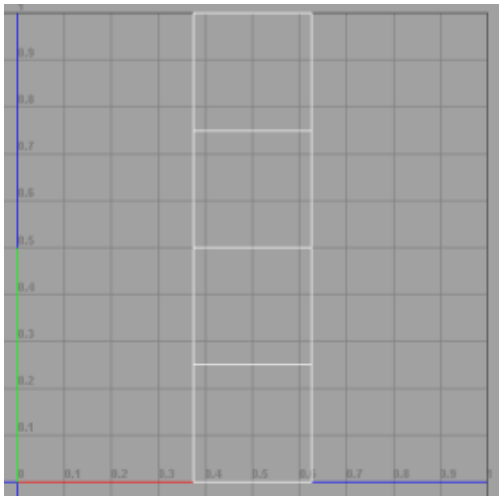
For reference I made a polygon cube, sized it accordingly, deleted the front and back face, extruded all the side faces then deleted the faces I selected to do the extrude).

Now as this is a wooden window frame I am going to want my wood grain to flow with the shape as if the were created out of a piece of wood (seems logical).

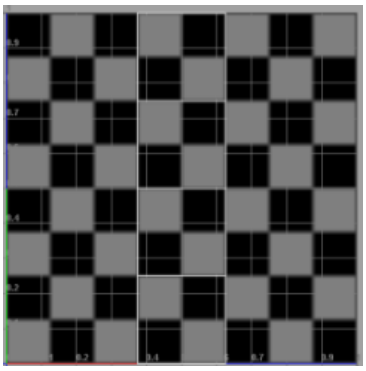


Next it's time to test the object with a chequered texture (Right click and drag down to *assign favourite material > Lambert > Colour □ > Checker*

I'm sure you will agree that's not what you expected to happen. So it needs a little work to ensure the texture is correctly spread out over your objectd.

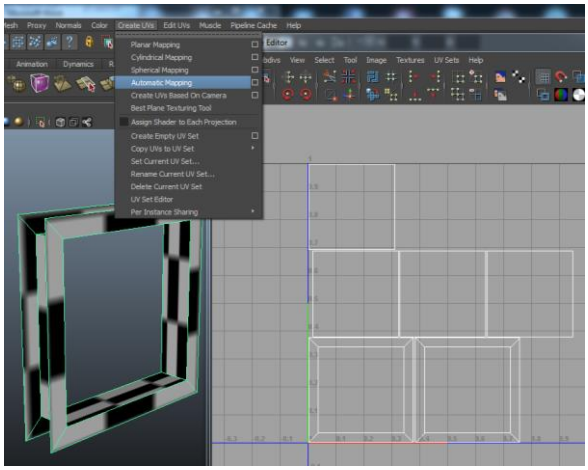


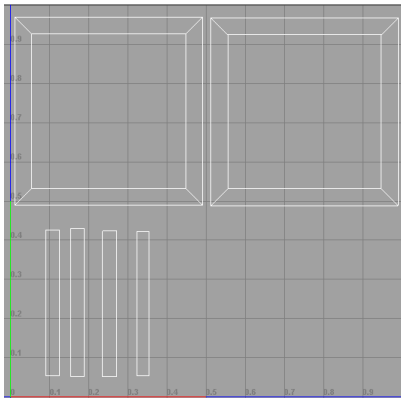
So to find out why we need to take a look at the UV Texture Editor *Polygons (menu set) > Edit UV's > UV Texture Editor*



The images to the left are of the same thing only with the display image on (and dimmed) or off (to toggle these option look for a picture of a face within the UV Texture editor). From here its very obvious why the centre section is the only aspect that has a texture, because it's the only four sides we can see.

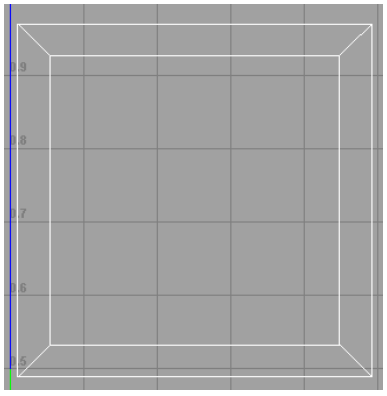
From here we are going to select *Create UV's >Automatic Mapping* instanstly you will see your texture map all over your image and something a little more familiar looking within your UV Texture Editor. You will notice some sections are more closely packed with the texture then others.

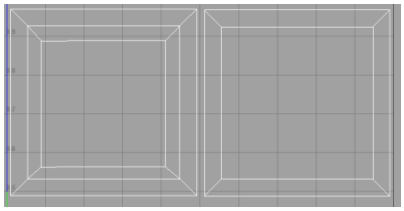




You can spend a little time cleaning things up to make them more accurate, and depending on how fundamental they will be to your shot will depend on the level of accuracy you may require.

To move the UV's you have *to right click and drag to UV*, then the same principles apply as when you are in the workspace (move, scale and rotate). You can separate or join edges and/or UV's, or snap them to one another by holding the "v" key. Don't' be afraid of overlaying one area with another if you want the same bit of the texture to be applied (i.e. both sides of an object).

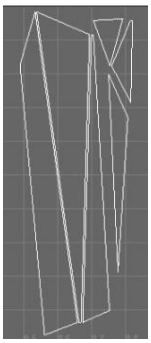


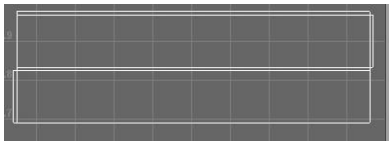


Once you are generally happy with the layout you can either make a generic texture and import it or you can save an image to use as a reference in Photoshop (or equivalent) by going to *Polygons > UV snapshot*. Adjust the sliders accordingly but as a general rule textures should be square and are usually 1024 or 2048 (again dependant on the image quality you require). We will continue with wooden texture creation a little later on.

Transferring UV Attributes

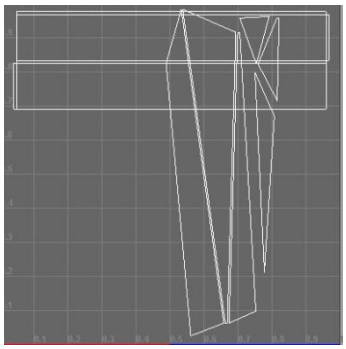
You may come across circumstances where you have to transfer the UV's of one object to an identical counterpart.



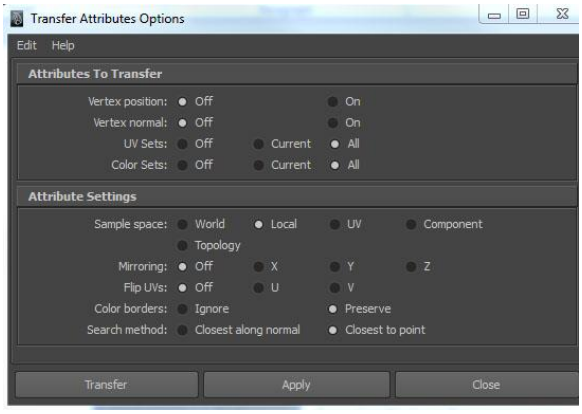


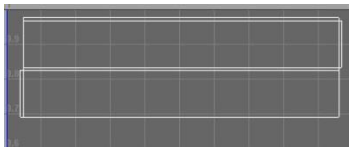
I'm just going to be using a simple wood plank in this example:

I would like to transfer the UV's of the object on the left to the UV's of the object on the right.



Select the source object (left), then shift select the target (right). Within the polygon menu set select *Mesh > Transfer Attributes* □ and apply the following settings.





With both objects still selected you will see that the UVs appear like one set because they are now exactly aligned.

Note: Until you clear your history your geometry will inherit all changes made to the source.

Overview of Maya Materials

Understanding the fundamentals of Materials *(Window > Rendering Editors > Hypershade)* is important and should not be overlooked, therefore when you do have to create a material it will ultimately replicate what you initially envisioned.

Diffuse: *The spread or distribution of light across a surface*

Specular: *Characteristics referring to the shaders reflective capabilities*

Lambert: *Most Commonly used for matt surfaces such as wallpaper*

There is no specular highlight on this shader as the light is diffused across the surface of the material in every direction.

Blinn: *Often used for glossy/metallic surfaces such as chrome*

The surface of a Blinn diffuses specular a lot more gradually than that of a Phong. Eccentricity controls highlight size giving a more versatile surface that can be easily adapted to various materials.

Phong: *Often used to replicate metals, plastics and glass*

This shader has a very definitive focal point. Cosine Power (Low value big highlight) affects the “Shininess” in turn leading to a specular highlight that drops off sharply. Less complex than a Blinn as it fails to account for angular changes when looking at the specularity.

Phong E Shader: *Similar to above, and are used for metals, plastics and glass*

It provides the user with more control over using two attributes being Roughness and Highlight Size together to affect the “shininess” whilst maintaining a sharper appearance than Blinn.

Ramp: *Very often used for editing other shader attributes*

Enables transition between multiple colours on a single shader. Commonly used to edit the appearance of other shaders; enabling the adjustment of attributes such as transparency, specular highlights, incidence, incandescence, light, reflectivity and ambient occlusion.

Layered*: Often used when a material is composed of multiple components*

Enables different textures types to be applied to the same piece of geometry. For the more advanced user it allows for shader stacking which can produce some very elaborate and spectacular results.

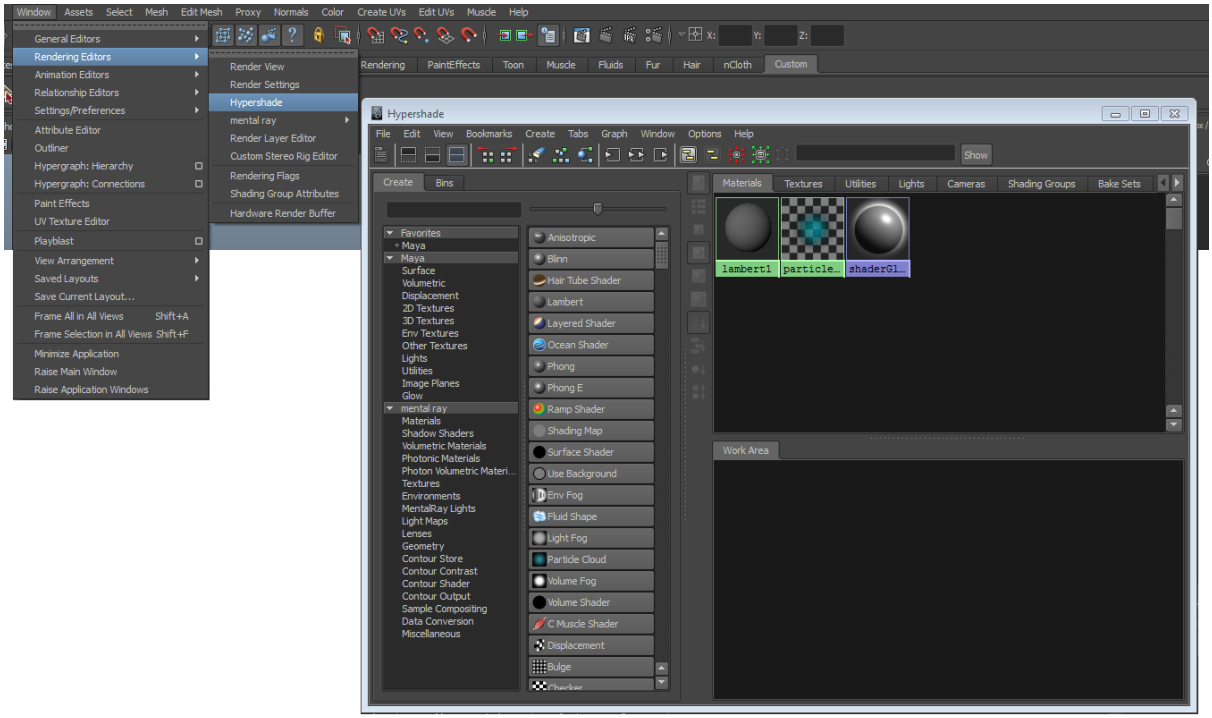
Anisotropic Shader*: Often used for blemished objects such as crumpled foil*

Used for materials where light reflectivity differs according to direction as it will reflect asymmetrically.

Hypershade

Ways to Access the Hypershade Menu

- *Window > Rendering Editors > Hypershade*
- Inside a Panel select *Panels > Panel > Hypershade*
- Hotbox: *LMB click* underneath all other options and *drag left*



Maya's Hypershade provides you with the ability to quickly create, edit and delete render nodes. If you're thinking "What is a render node?" then the simple answer is that they are iconized characteristics of elements within Maya, such as Materials, Textures, Lights, Utilities, and Cameras. Below you will see the default Hypershade menu and what there functions are.

Hypershade Tabs & Nodes (Left Column)

The tabs and structure are organized so you can easily access the specific node you require. One of the few ways to create nodes is directly within the Hypershade, to do this you firstly click on any heading node under the “create” section then click the appropriate node (right hand column) to add it to the work area.

Hypershade Menu

As within most aspects of Maya you can create customised tabs and choose a layout of your choice. Many of the menu options are self-explanatory but we will be covering the more ambiguous ones later in this section.

Hypershade Work Area

As you might imagine the “Work Area” tab is where you can create relationships, or edit attributes and characteristics of multiple nodes at any one given time. Although designed for building shaders, rigging techniques also use this area.

Navigation and Controls within the Hypershade

You can navigate and access features within the Hypershade menu the same way in which you do within your scene.

Hot Key	Action
Alt + LMB	Tumble / Rotate
Alt + MMB	Track / Pan
Alt + RMB	Dolly / Zoom
A	Frame all Nodes
F	Frame an individual node
Ctrl + A	Attribute Editor

How to Work on a Texture Node

MMB click and drag from the top window to bottom (Workspace).

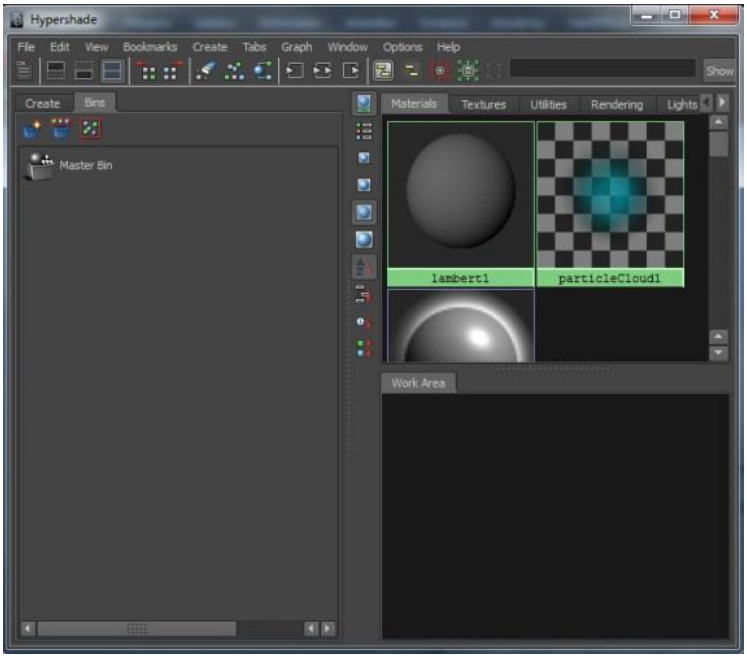
Clearing the Workspace

Navigate to the icon bar and click on the clear graph Button which resembles an eraser tool this will (like eraser tool), the nodes will not be deleted just removed from the workspace area.

Renaming a Node

On the node, right click and drag down to rename.

Grouping



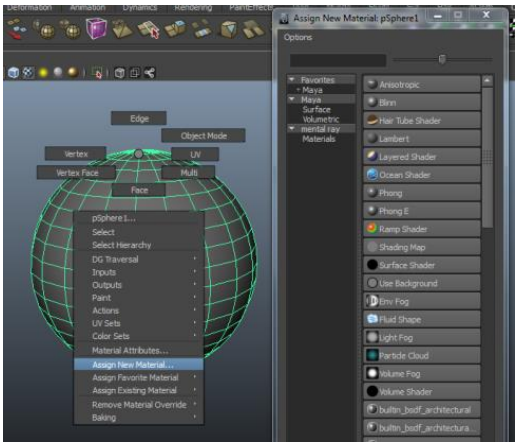
Like within the main scene you can create layers to help organise your scene, the same principle applies to the textures only instead of layers they use bins.

Within the Bins Tab you can create an empty bin, create a bin from selected or select all unsorted items.

Removing a node from the bin is just as easy. Select the node, right click on the bin and select *remove selected node*.

Assigning a Material to an Object

There are a few ways to assign a texture to an object so which ever method you use will have the same effect.



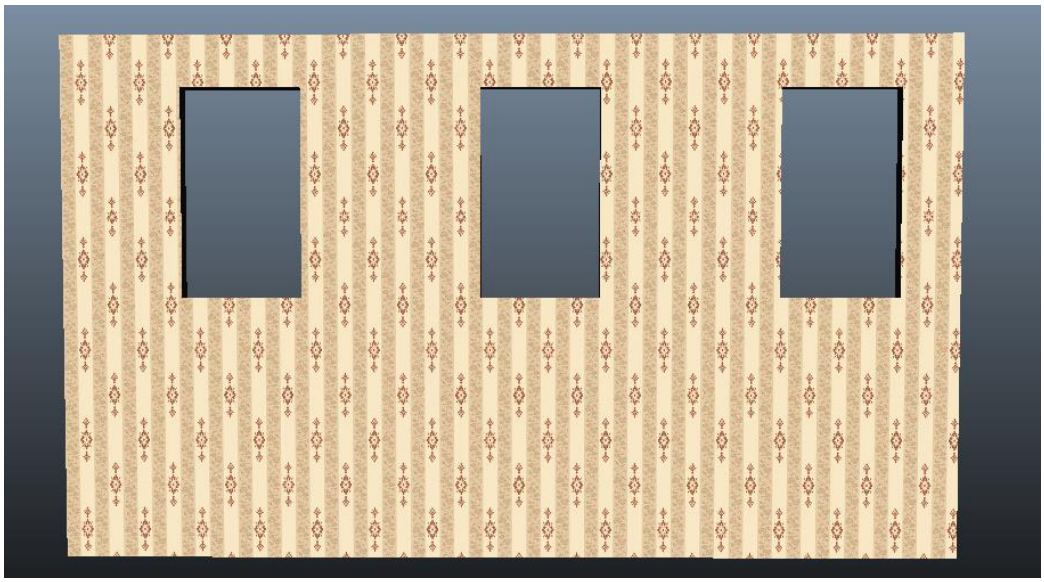
- Open the *Hypershade* > Click on the Material > *MMB* drag from the texture/shader node onto your object
- Right Click on your object, scroll down to assign a texture, either choose a created texture from the list or create a new one.
- Select your object(s), open the Hypershade, right click on the texture/shader node and select *assign material to selected*.

If you haven't already got a texture assigned to your object, follow one of the above methods.

Adding custom features to your Material

Adding a custom image to a Material

- Select *lambert* (or suitable alternative) *> (Name it) > Colour □ > File*
- *Select image*



Normal, Bump or Displacement Maps?

A question that arises frequently as well as making regular appearances on forums; are the differences between each of these maps and how/when to use each of them.

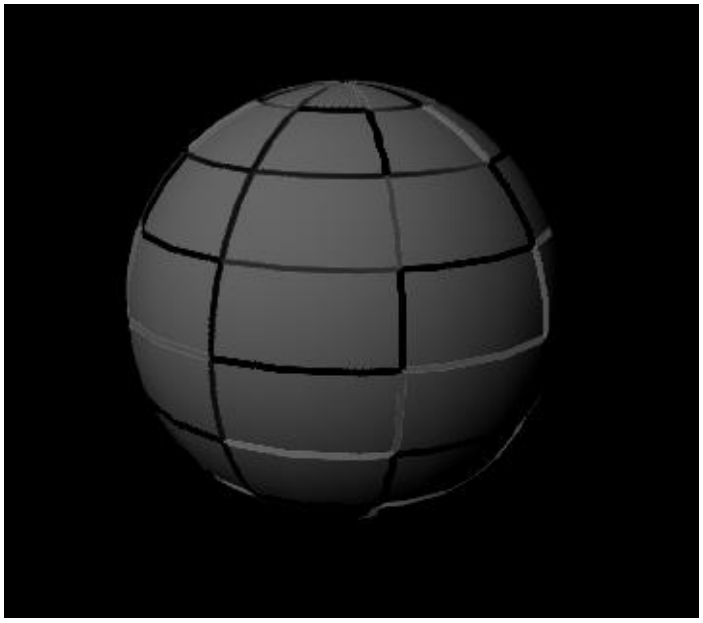
Bump Maps and *Normal Maps* do have their similarities in so far that neither of them affects geometry in a physical form however they do hold a vastly different amount of information and are used for a very different purpose. A *Bump Map* holds only single axis (height) information and will be used primarily to “shade” the surface and create the subtle feeling of depth. A *Normal Map* has the ability to hold information on all 3-axis allowing it to retain detail from higher mesh models to be latter recreated on lower geometric representations. *Displacement Maps* have a similar makeup to that of a *Bump Map* displaying a white to black colour range from 0-255 nevertheless their rendered results vary considerably. They do by all accounts rely heavily on near perfect UV’s as they affect the geometric position of vertices over the objects surface in relation to their value.

Mipmapping

In this process (designed primarily for bitmaps) Maya stores a colour value for sequentially decreasing image sizes i.e. 1024 x 1024, 512 x 512, 256 x 256 and so on all the way down to 1 x 1. Therefore during the rendering process the renderer determines the most appropriate mipmap for the relevant pixel, thus if the pixel is in the background a low resolution mipmap is used and vice versa. Maya can also use an average of two mipmaps for a more accurate representation of a pixel. Mipmapping is designed to reduce the rendering load as it will not have to filter through large image sizes to determine a specific pixels colour.

Assigning a Bump Map

- Select your existing material (i.e. lambert1) > *Bump Mapping* □ > *File*
- *Select image* (Bump Map)



Note: *This image has been assigned a black and white chequered texture into the bump. Creating the feeling of depth that additional geometry wasn't required to make.*

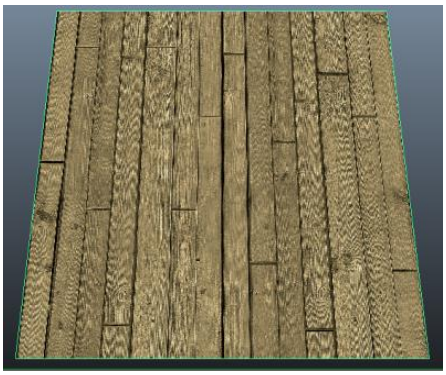
Repeating Textures

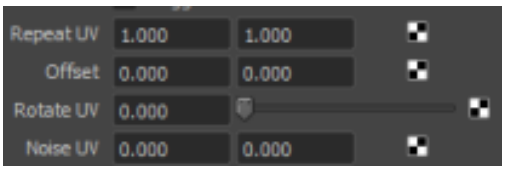
As the title suggests you can make a small material repeat over a larger area. For this to be effective it is advisable you use a seamless square texture, or learn how to create one yourself on page 118.

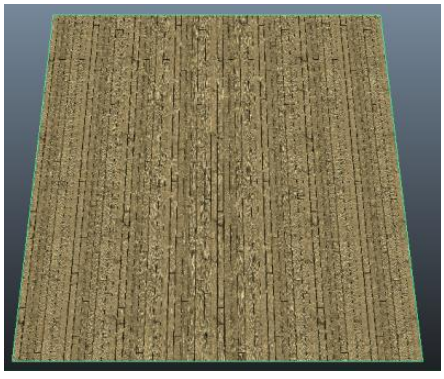
Once you have your image file ready, create a Polygon Plane and stretch it out over your grid

- Press 6 to go into shaded mode (this way you can see the changes)
- Select your existing Material
    - *Object > Material (i.e. Lambert1)*
    - *Window > Rendering Editors > Hypershade > Lambert1*
- Press the arrow going into the box either next to the material name or next to the colour (this will take you to your texture file)
- Press the same arrow again next to the material name or on the tabs you will see a *place2dTexture* Tab
- Repeat UV's
    - 1x1 are the default settings
    - 2x2 = 4 instances of your image
    - 3x3 = 9 instance

      and so on…
- Rotate Frame – If the image if it has come in a different way that what you have set up for your UV's use this setting to correct the rotation.







Texture Examples

Mia_material_x is probably one of the most versatile textures within Maya and because of this we find ourselves using it a large proportion of the time as it's easy to create anything from wood, metal or glass within a few simple steps.

We will take you through a couple of them but will leave the remainder for you to play around with.

Wooden Floor

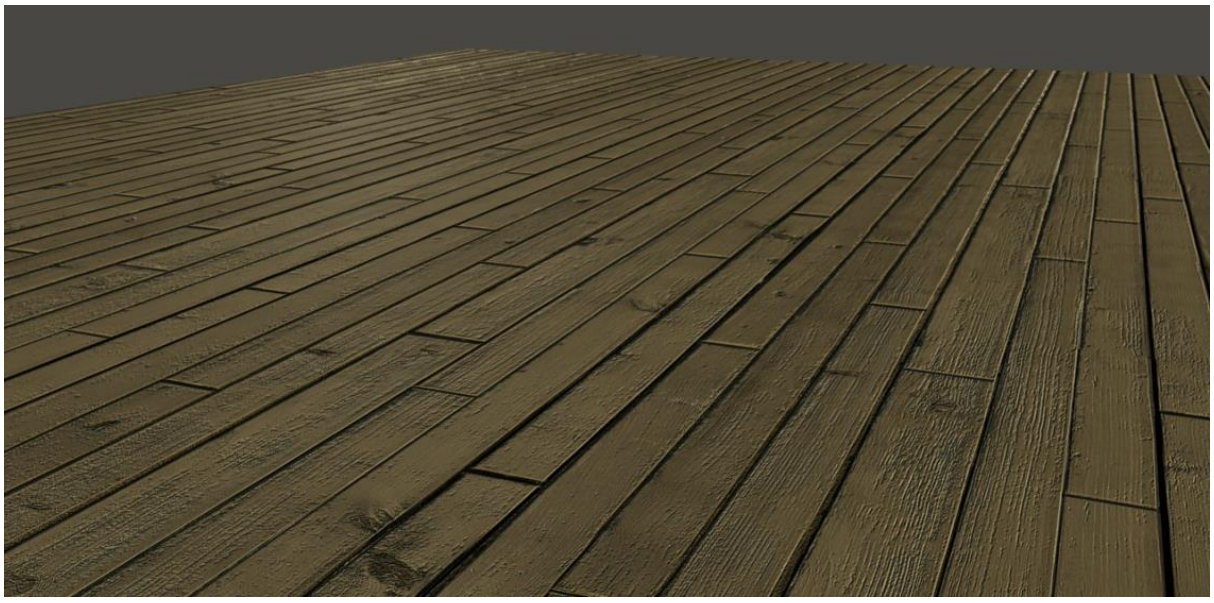
- *Create a floor plane > RMB Click* and Drag down to **>** *Assign New Material*
- *Select Mia_material_ x > (Name it…WoodenFloor) > Select Colour □ > File*
- *Select wooden floor image* (you had already prepared)
- Drop *Reflectivity* down to around > *0.1- 0.2*
- *Glossiness > 0.2 – 0.3*

Bump

- *Attribute Editor* > *Bump* > *Standard Bump* □
- Import file
- *Bump depth* = *0.15*

More definition (points further away from the camera)

- File attributes
    - *Filter Type* > *Mipmap*
    - *Mental ray* > *Advanced Elliptical filtering*



Glass Window

We are going to create a very simple glasslike texture that we can apply to our window geometry.

- *Windows > Hypershade > Create > Mental Ray Materials > Mia_material_x*
- Add file texture colour (window)
- Reflectivity = 0.3
- Index of refraction = 0.8
- Refraction Colour > add in same texture file as above

Or

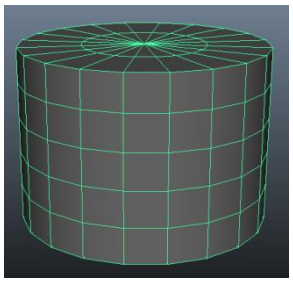
- *Windows > Hypershade > Create > Mental Ray Materials > Mia_material_x*
- *Presets* > GlassThin > Replave*
- Add file texture to *Diffuse Colour*

Note: *Useful to link to the same file rather than creating multiple instances of the same texture.*

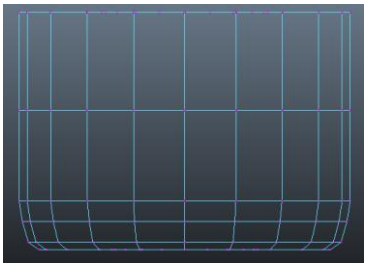
- One quick way to “dirty up” the window would be to
    - Transparency > Chequered box > Noise/fractal/file(custom)

Glass - Whisky Tumbler (and contents)

Create > Polygon > Polygon Cylinder

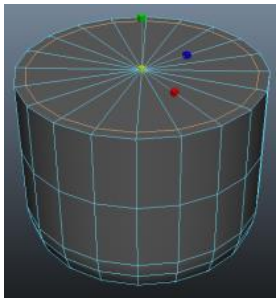


- Subdivisiosn Axis 20
- Subdivisions Height 5
- Caps 2

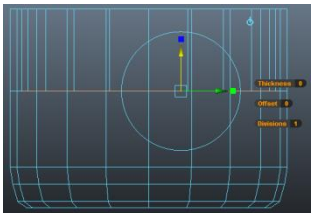


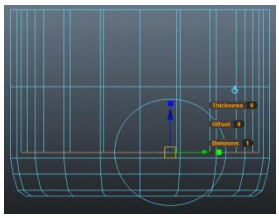
There are 4 rows of vertices in the middle of the tumbler. Switch to side view and out of those 4 select the bottom 3 rows of vertices and scale them in the Y Axis. Drop them to the bottom of the so they are close to the base and gradually scale them inward creating the curve at the bottom.

Select the remaining row of Vertices and move them down to a height where you would like the glass to appear filled, (and tweak the glass height if required).

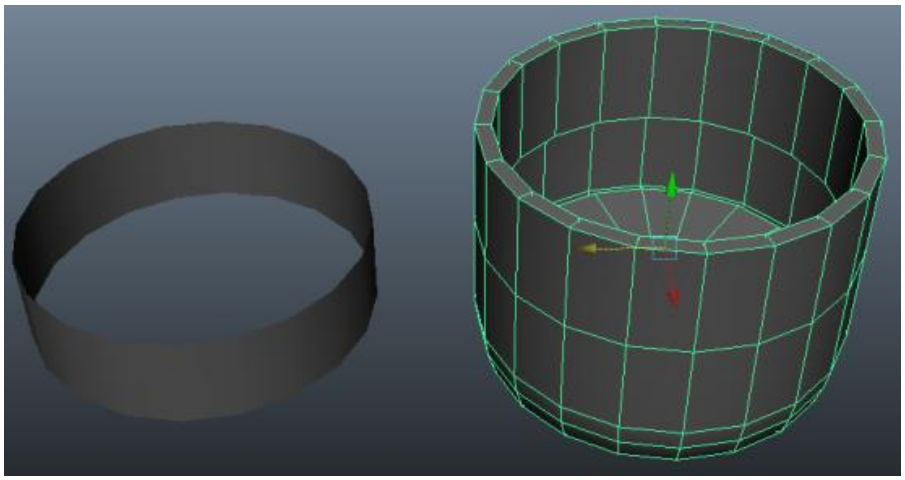


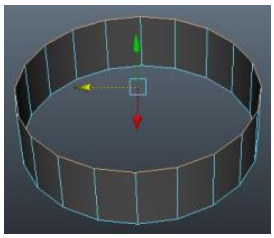
Switch to *top view* (or *perspective*) and select the top inner edge (*RMB > Edge > Double click one edge to select continuous edges*) and scale them outward towards creating the glass thickness. Then select all the centre faces and *Edit Mesh > Extrude* switch to side view, hold down the "*v*" key and snap the extrude to the row of vertices that determines the fill level, press "*g*" to repeat the extrude and bring the vertices down towards the bottom of the glass leaving a thick base.



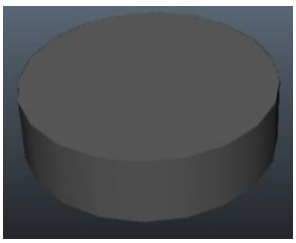


Switch to perspective view and select the faces below the fill line but not the base (you should have one row of faces selected at this point). *Mesh > Extract* will separate the geometry from the Tumbler, then move the geometry to one side.

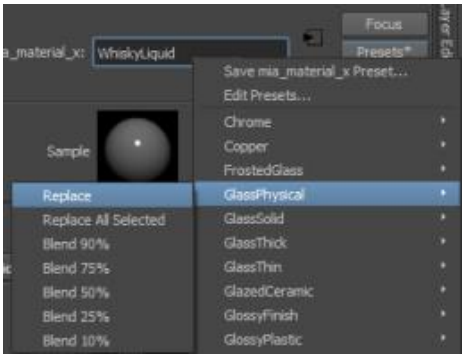




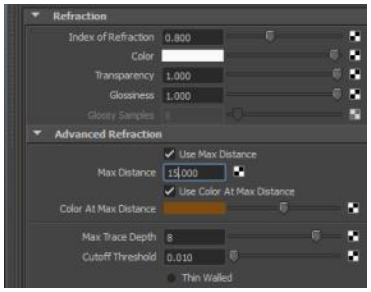
Select the upper rim of edges *Mesh > Fill hole* then *Edit > Delete by Type > History*. Assign a *mia_material_x* and rename it to *WhiskyLiquid* (or similar), select the *Presets** button and select *GlassPhysical > Replace*.



Note: If the mental ray materials are not available in the Hypershade menu we need to make sure the *mayatomr.mll* plugin is loaded in the Plug-in Manager (*Window > Settings/Preferences > Plug-in manager*)



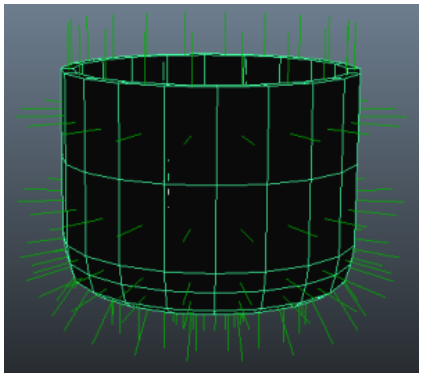
- *Refraction > Index of Refraction* 0.8 -1
- *Advanced Refraction > Max Distance* 15.0
- *Advanced Refraction > Colour at Max Distance* Chose liquid colour



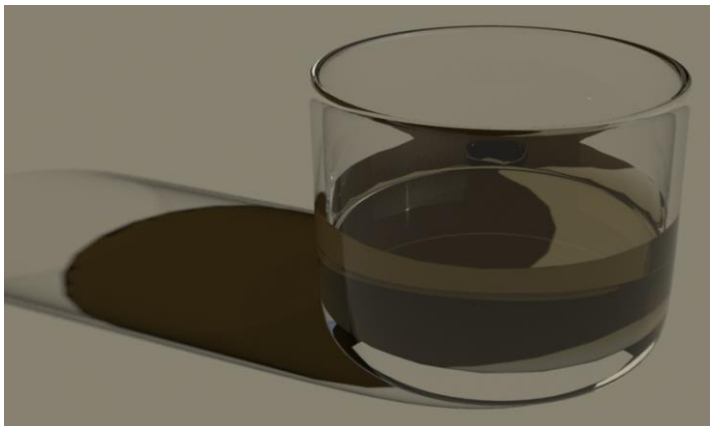
Put your liquid geometry back in place (zero out your translation attributes) then select the top face you created earlier by filling the hole. *Mesh > Extract* to separate the top face from your geometry and assign a *mia_material_x* and name it *WhiskySurface*. Select the *Presets** button and select *GlassPhysical > Replace*.

- *Refraction > Index of Refraction* 1.33 - 1.36
- *Advanced Refraction > Max Distance* 15.0
- *Advanced Refraction > Colour at Max Distance* Chose liquid colour. This works best with an exact match so if you just take the HSV values from your *WhiskyLiquid* texture this will be fine.

Lastly, if you select your tumbler assign a *mia_material_x* and name it *GlassTumbler*. Select the *Presets** button and select *GlassThick > Replace*.



Note: *We need to ensure all normals (Display > Polygons > Face Normals) are facing outward to ensure the effects are calculated correctly. If not go to the Normals > Reverse option to flip them around.*



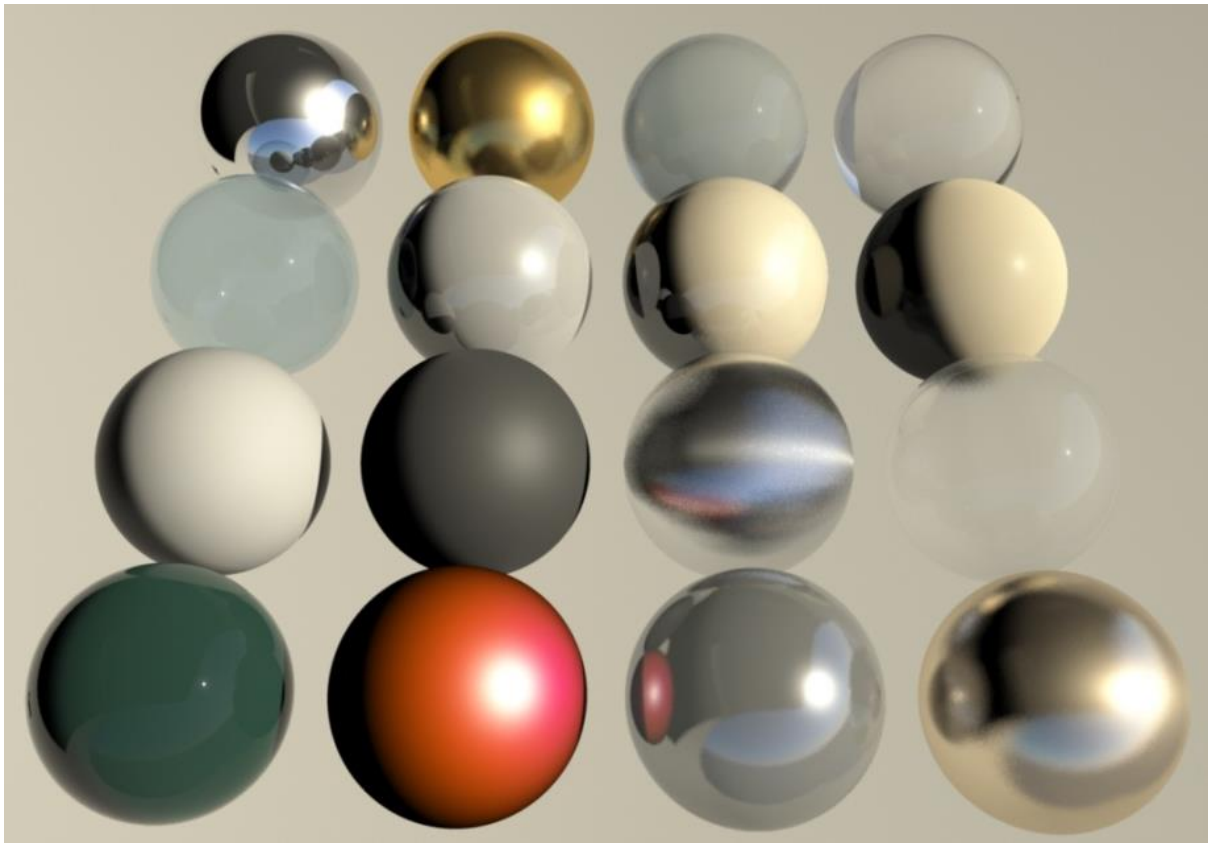
We will not be delving into the render and lighting in too much detail, but in order for you to render this effect you need a light and some improvements from the base render settings. We will be using two of Mayas pre-sets.

- **Light:** *Physical sun and sky* (*Render Settings > Indirect Lighting > Environment > Physical Sun and Sky > Create)*
- **Render settings:** *Quality > Quality Presets > Production*

Note: *You can tweak the* Max Distance *settings in your liquid material to lighten and darken the appearance.*

Here is another example of the above technique within a finished scene.

Mental Ray Presets



(Top left to bottom right) Chrome, Copper, Frosted Glass, Glass Solid, Glass Thin, Ceramic, Glossy Plastic, Matte Plastic, Pearl, Rubber, Stained Metal, Translucent Plastic Film, Water, Car Paint Shader, Glass/Pearl (Preset Blend), Copper/Stained Metal (Preset Blend).

Image Optimisation

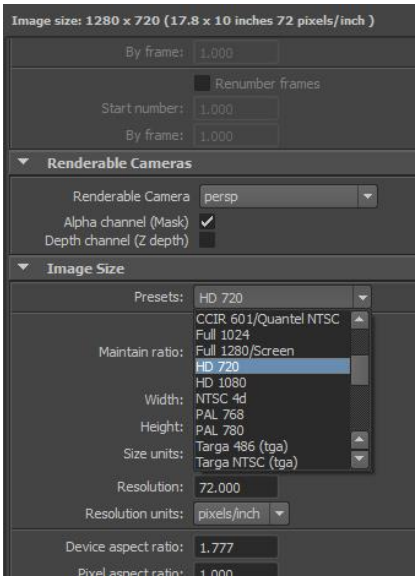
Maya has its own proprietary image format that it likes to use, the *.iff format. Converting your files to this format will actually allow Maya to work more efficiently, thus speeding up render times. There are two main ways to convert images to .iff's, Fcheck can save images out as .iff's or you can download a plug-in for Photoshop from high-end 3D.

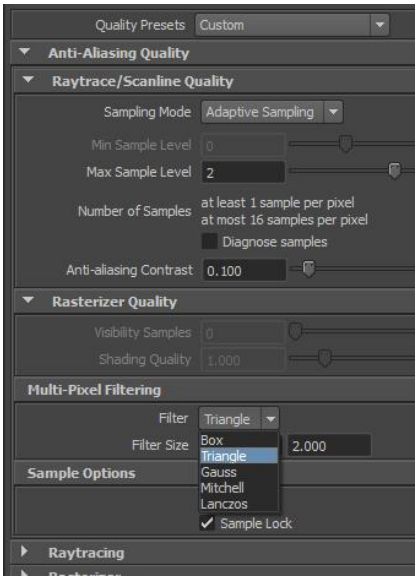
Another useful way of speeding up Maya's rendering is to set the texture filtering to 'off' or 'mip-map'. The texture filter is essentially a blur applied to smooth textures, but in large image files and heavy scenes in general this can substantially increase render times and memory load. Obviously there is a trade off in quality, but you will have to find what works best for your project.

Subsurface Scatter Skin Material

Amongst the Mental Ray materials that are built into Maya are a few subsurface scatter nodes. The one we will be using today is the *misss_fast_skin_maya shader*, which has a few extra options than the *misss_fast_simple_maya* shader to give us a really nice skin shader, really quickly.

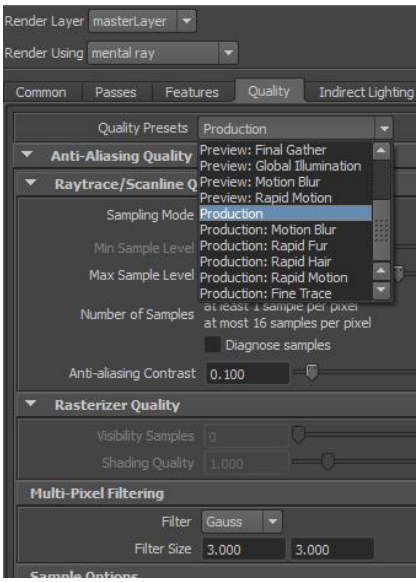
Open a new scene and let's set up our rendering settings. First change the resolution so something more appropriate such as *HD720*, and then make sure we are using mental ray as our renderer.



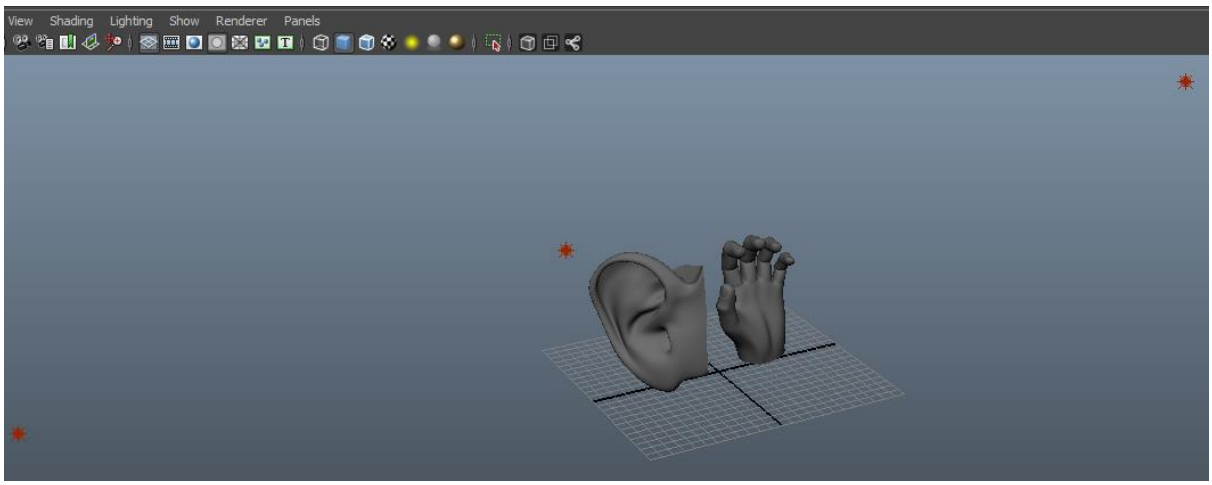


Note: *If the mental ray option is not available in the drop down menu we need to make sure the mayatomr.mll plugin is loaded in the Plug-in Manager (Window > Settings/Preferences > Plug-in manager).*

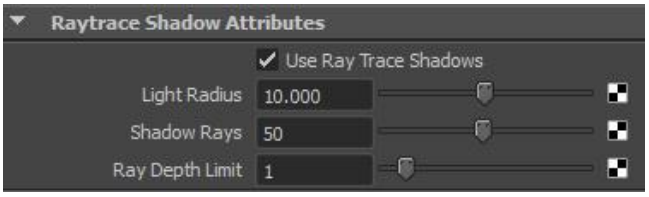
In the *Quality* Tab we can use the preset *Production* to give us a good start on our settings, and for now we will just change the filter to *Triangle* to give us sharper edges and speed up rendering.



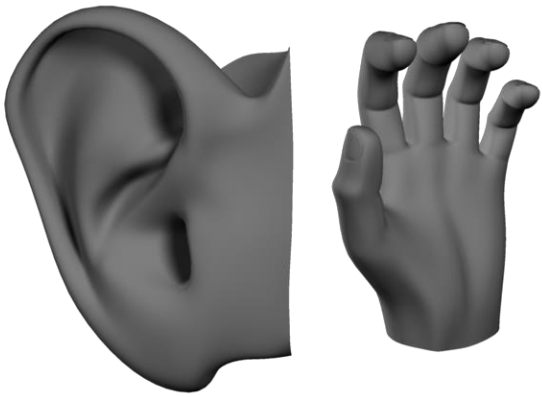
For the scene lighting I have used three point lights, a front light a side light and a back light. The side light I have reduced the intensity to 0.5, otherwise all settings are left at default. The models I have chosen to import for this tutorial are the ones we built in the modelling section earlier, the ear and the hand. These should give a good representation of how skin shaders will look on your own models.



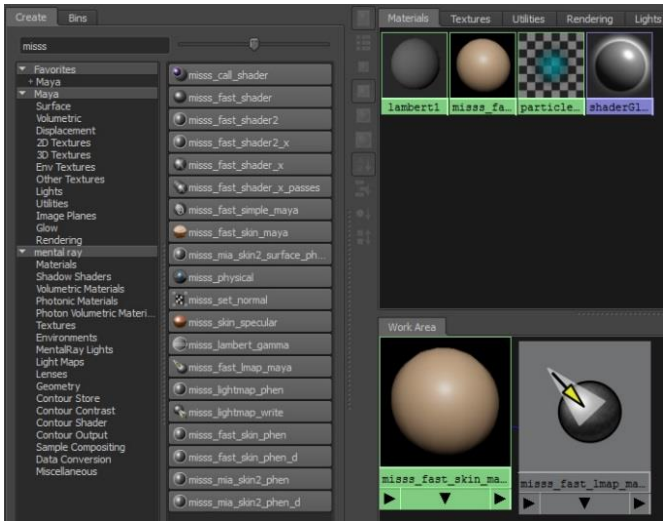
I have enabled ray-traced shadows on the front light with the following settings to give the scene some depth.



If we hit render then, we should see something like this:



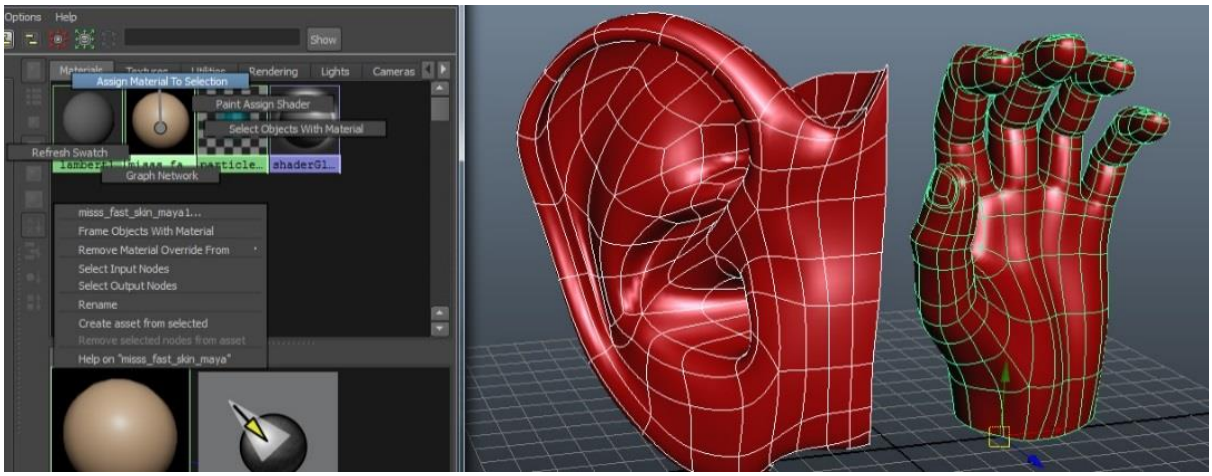
You can see the influence of the soft shadows being created and we can also see a little of the backlight breaking across the edge of the hand.



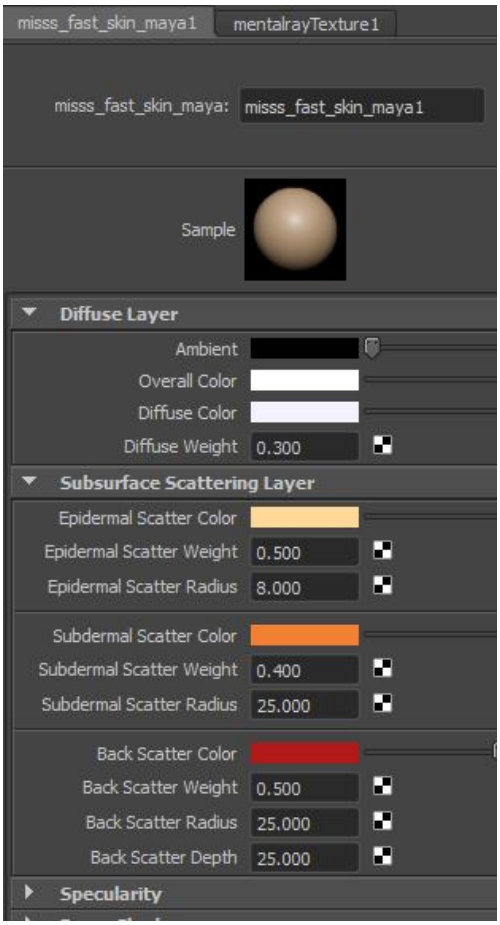
Open your Hypershade window (*Window > Rendering Editors > Hypershade*) and in the search bar type "*misss*". This will filter all of the subsurface scatter nodes. We are going to use the *misss_fast_skin_maya* node so click on that. (A node will appear in your materials window and Work Area).

You will find that another node, the *misss_fast_lmap_maya* node has been automatically created and linked to the skin node for us.

Select both of the objects and right click hold the skin shader and select Assign Material to Selection. You will most likely see the objects turn bright red, this is ok – it is because Maya's viewport cannot render subsurface scattering.



Render your scene. You can see that although the shading is nice, there is a long way to go to make it realistic. It is too pale and orange and not enough backlight is filtering through.



Double click on the material in the Hypershade to open the Attributes Editor and expand the *Subsurface Scattering* Layer panel. This is where most of our adjustment will come in.

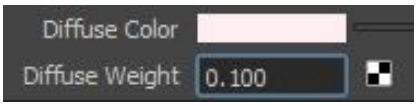
The *Diffuse Layer* carries the main colour attributes.

The *Epidermal Scatter* is our primary focus, this is the colour of front light reflecting off the surface.

The *Subdermal* is the '*flesh*' part of our skin.

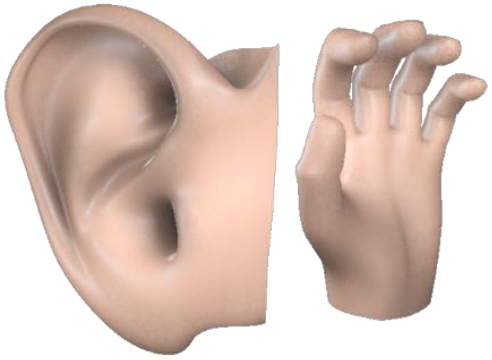
Finally, the *Back Scatter* is the light passing in from behind, which is why we have the extra attribute *Depth*. This controls how far into a surface the light can penetrate.

The *Radius* value controls how much the scatter will spread over our model and finally the *Weight* value controls how much influence each colour will have over our final skin material.



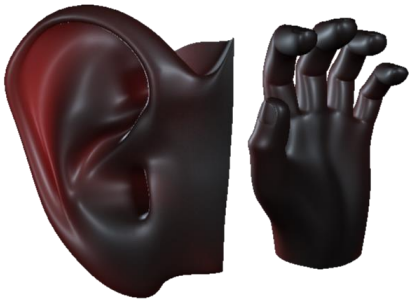
The Diffuse colour is going to have very limited influence, change the value to 0.1 and change the *Diffuse Color* to a very light pinkish tone.

The easiest way to see exactly what we are getting with each of the scatter colours is to work with them one at a time. Set the *Scatter Weight* for the *Subdermal* and *Back Scatter* attributes to zero.

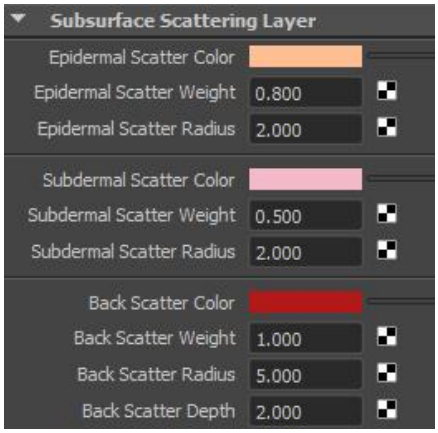


We will work on the *Epidermal* first. This is going to be our primary colour so set the *Scatter Weight* to one and The *Scatter Radius* is far too high so bring that down to about two. We should get a nice, lit skin colour coming from our model when we hit render.

For the *Subdermal*, (remember to set the *Epidermal Weight* back to zero first) we want a more pinkish 'flesh' colour and we don't want the influence to be too high, a *Scatter Weight* of about 0.5 should suffice. We will set the *Scatter Radius* to two again.



For our *Back Scatter* colour we can keep the deep red colour and set the *Scatter Weight* to one. We want a slightly wider Scatter Radius somewhere around five and the *Scatter Depth* Value depends on your scene and model scale, for me a value of two is sufficient. You can see that especially the thin tip of the ear, the red comes straight through, and on the hand the edges are highlighted.

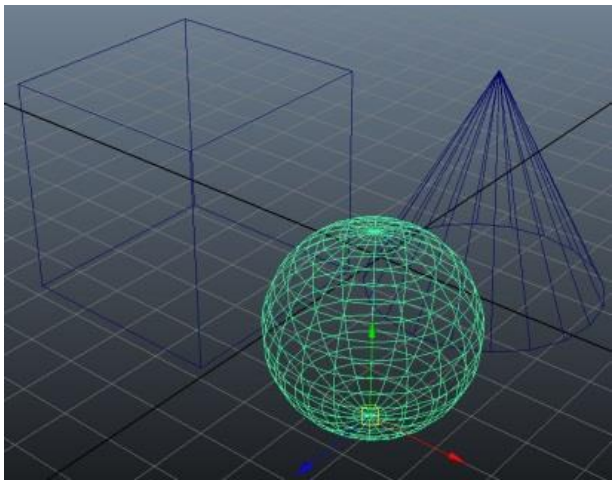


Finally, we want to combine these settings. Set the *Epidermal Weight* to 0.8, the *Subdermal Weight* to 0.5 and the *Back Scatter Weight* to one and hit render.

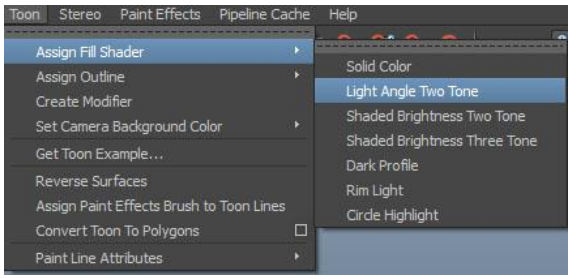
Now we have a simple skin material that you can customise and tweak for your own specific needs.

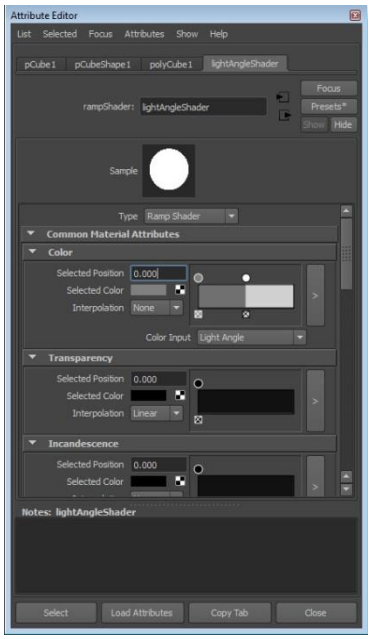
Toon Shading

There are two main components to toon shading; on the one hand there is the flat shading itself and on the other there is the solid outline on the objects themselves. For this tutorial we are going to focus on the two tone shader.



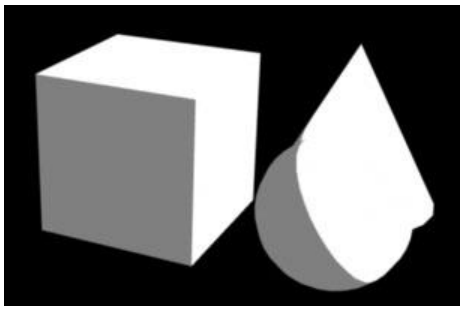
Create a point light and a few objects in the viewport. With one of the objects selected, and in the Rendering menu set, click on *Toon > Assign Fill Shader > Light Angle Two Tone*.

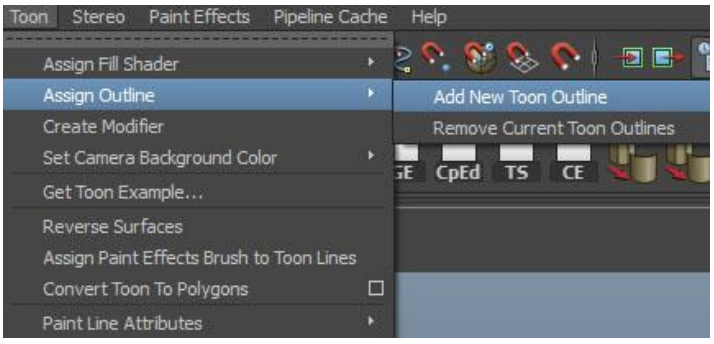


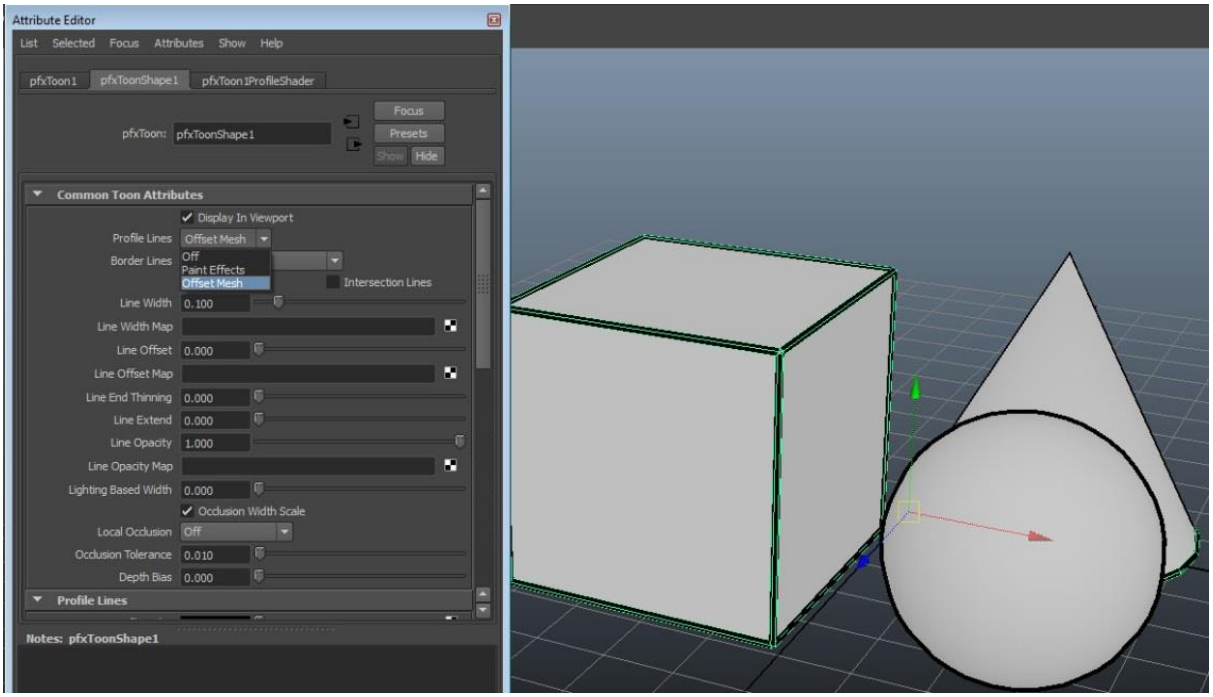


The attribute Editor will open with a focus on a newly created *lightAngleShader*. If you render the scene you will see the shader at work, splitting objects into two clean shades of grey.

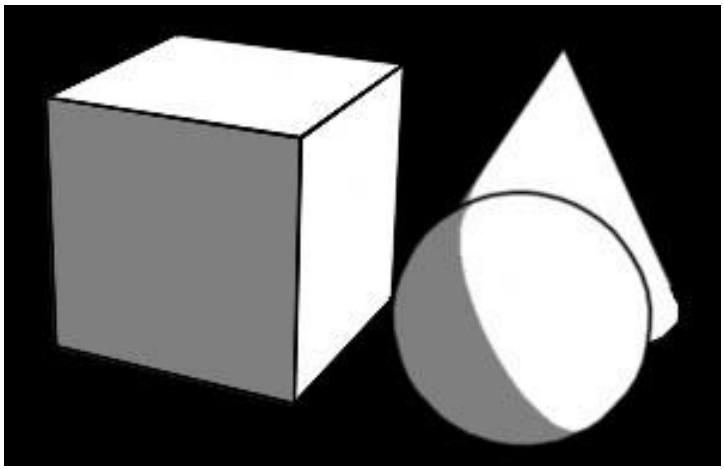
To build an outline around the shapes, select all the objects and select *Toon > Assign Outline > Add New Toon Outline*.







This will create a new paint effects shape around the outline of the shape and also against any sharp edges. In some cases (for example when rounded surfaces come to a point) you may get a more accurate result if you set *Profile Line*' to *Offset Mesh*.



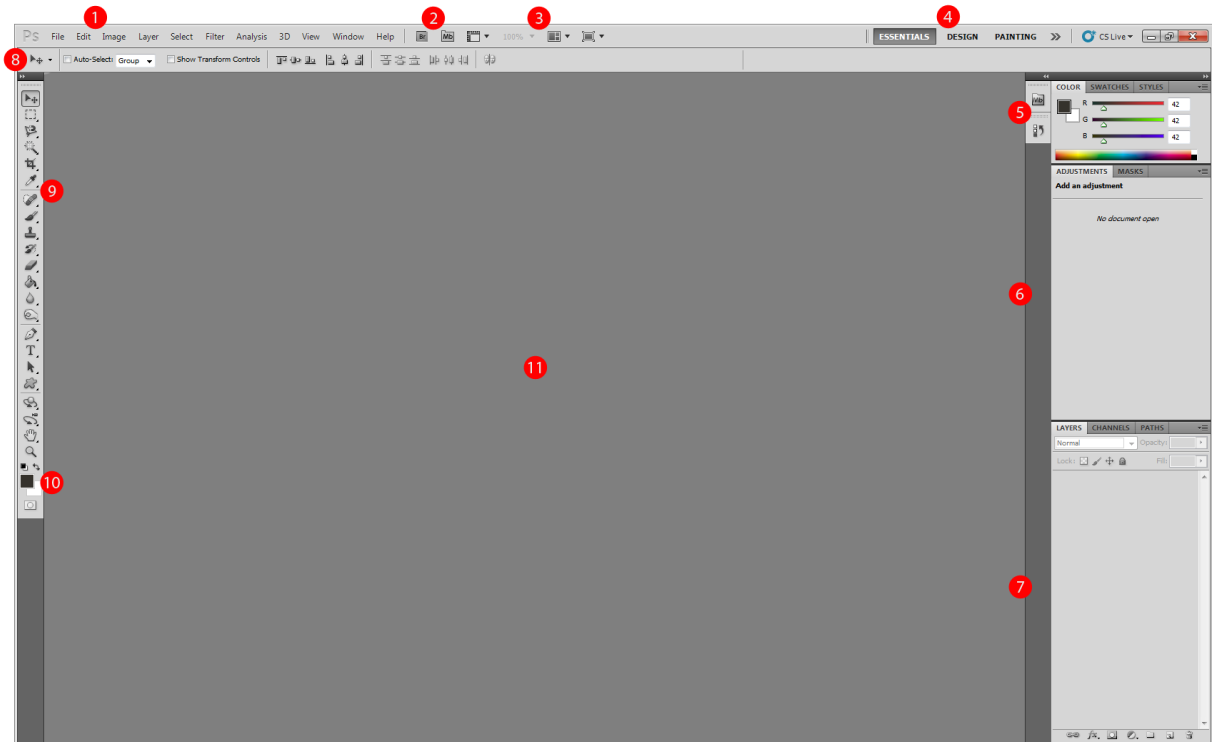
Experiment with the options and ramps in the *lightAngleShader* to get your desired effect.

Photoshop Basics

Introduction

Photoshop is a powerful image editing application that provides industry standard versatility and optimised tool sets allowing for the creation of professional and sophisticated images. Many animation books can often take for granted an understanding of Photoshop, but we know from experience many people do not always have this knowledge. So as oppose to you having to search for further tutorials we have dedicated this section to go over the basics of Photoshop and its uses within Maya.

Photoshop Layout



1. Main menu bar
2. Additional views and Bridging options
3. Zoom and image pane options
4. Pre-configured Menu-sets for appropriate design stages
5. Colours / Styles / Swatches options (all dependent on Menu-sets).
6. Adjustment masks / Fonts / Brushes (all dependent on Menu-sets).
7. Layers Menu, with appropriate tools
8. Currently selected tool and further options
9. Tool shelf
10. Colour selection pallet
11. Workspace

Photoshop Toolset

Move Tool (V) – Moves Selections and layers

Rectangular Marquee Selection Tool (M) – rectangular selections

Lasso Tool (L) – Freehand selection

Magic Wand Tool (W) - Selects similarly colours regions

Crop (C) – Reduced image size within selected region

Eyedropper Tool (I) – Samples the selected colour

Spot Healing Brush (J) – Automatically removes blemishes

Brush Tool (B) – Freehand paintbrush tool

Stamp (S) – Will clone a selected area to another

History Brush (Y)

Eraser (E) – Removes content

Paint Bucket Tool (G) – Flood fill

Blur – Spreads pixels within selection

Burn (O) – Freehand tool that darkens an area

Pen Tool (P) – Draws paths consisting of Bezier curves

Horizontal Type Tool (T) – Text tool

Path Selection Tool (A)–Selects & moves fragments of existing paths

Custom Shape Tool (U) – Allows the creation of custom shapes

Object Rotation Tool (K) – Rotates a 3D object

Camera Rotate Tool (N) – Rotates a camera in a 3D scene

Hand Tool (H) – Allows the navigation of a zoomed image

Zoom (Z) – Enlarges the canvas

Switch Foreground / Background Colours

Foreground and Background Colour

Edit in Quick Mask Mode – useful for selecting large areas quickly

All icons that have a small arrow in the bottom right hand corner indicate there are further tool options available within that specific tool set.

Chapter 3: Texturing

Mouse vs. Tablet

Can you do everything a tablet does on a mouse? Almost, the tablet however has the fundamental advantage of being able to use the pressure sensitivity attribute enabling you to taper off lines and blend colours more subtly. Similar effects can be obtained with a mouse combined with Photoshop features but as you all know a mouse click is either on or off, there is no in-between, you will have to put in a lot more work to attain similar effects to that of a tablet. In today's market you don't have to spend a fortune on a tablet but we would recommend getting one if you are interested in creating your own textures.

The Colour Palette

To Save a Colour to the Palette

- Open the swatches tab
- Select the paint bucket tool
- Left click within the swatches tab
- Save colour, naming it accordingly

Brushes

Editing a Brush

You may need to edit the appearance of a brush to get the desired effect needed for your texture. By default brushes are usually solid, uniformed and repetitive almost the opposite of what we want at this point. Therefore to break up the brush's continuity follow the below instructions.

- Choose an appropriate colour
- Select a default brush that will suit your requirements
- Go into the brush window *Window > Brushes (F5)*
- Edit all or some of the following characteristics at your discretion.
    1. Brush Tip Shape > Spacing: Use the slider to separate the brush strokes so there is a gap between each one.
    2. Shape Dynamics > Size Jitter: Slide to randomise the sizes and/or use the dropdown menu (if you're using a tablet) to link this attribute to pen pressure.
    3. Scattering > Scatter: Slide to randomise the positioning of the brush strokes.
    4. Size it to a reasonable scale.
    5. Plus any other adjustments you feel are necessary.

Creating a New Custom Brush

Under some circumstances you may require a brush to create a unique or repetitive shape for uses like adding blemishes. To create a custom brush:

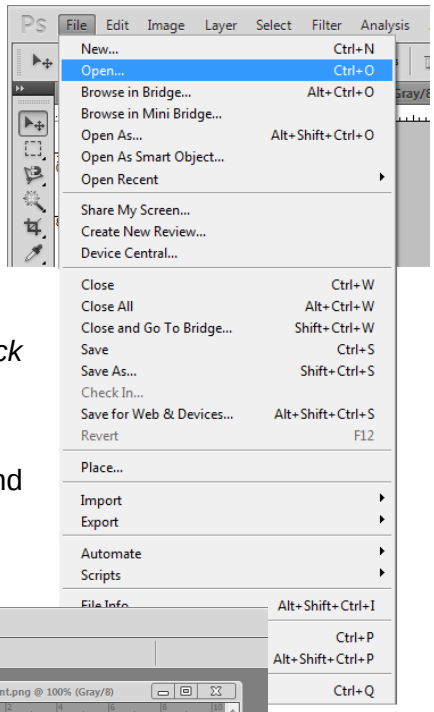
- *File > New*
- 50px by 50px workspace area, ensure you have a transparent background
- Locate a brush to give you a starting point (optional)
- Create your brush using the relevant tools.
- *Select > All*
- *Edit > Define brush Preset*
- Chose a name for your brush, once saved it will appear in the brushes pallet

Photoshop and Maya

Photoshop can save in a variety of image formats all of which Maya can recognise pushing these two programs to become widely recognised as industry standards. There are however a few issues when using PSD files as textures within Maya that you are wise to take note of. They become apparent when using standard methods of layer blending and layer opacity.

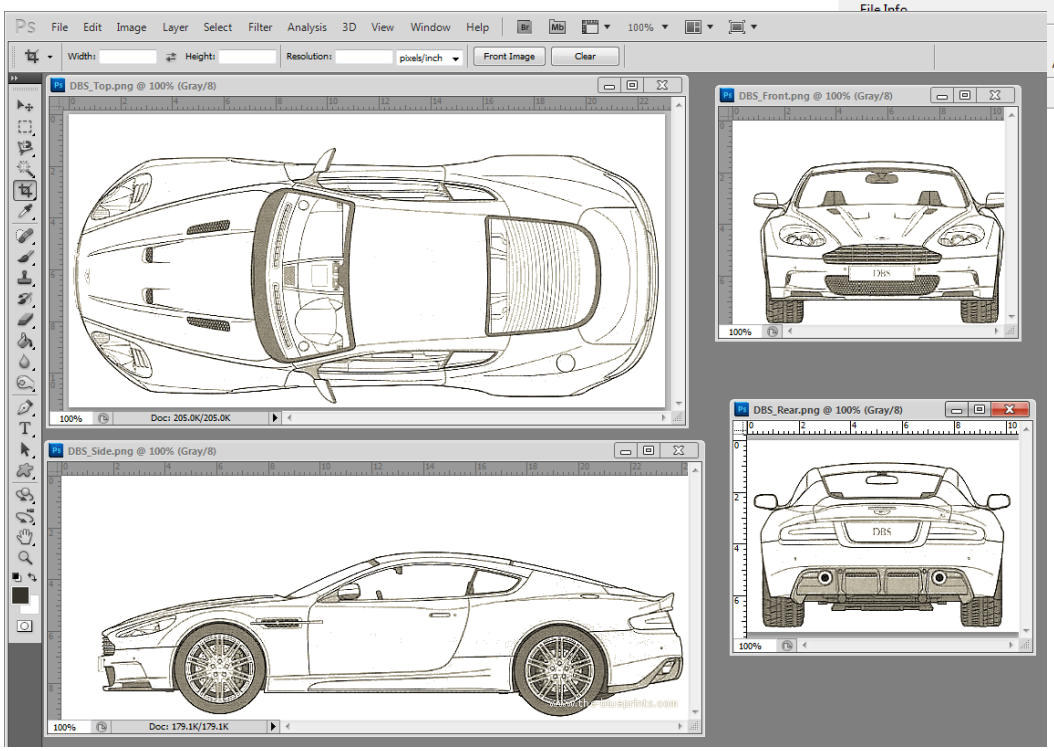
Preparing image planes for use in Maya

The first stage is to get all your images onto one canvas (if they are not already). To do this navigate to *File > Open* then locate your appropriate files and open them, once you have done this repeatedly (dependent on how many images you have) you will have tabs at the top of your screen each with a perspective shot of your model.



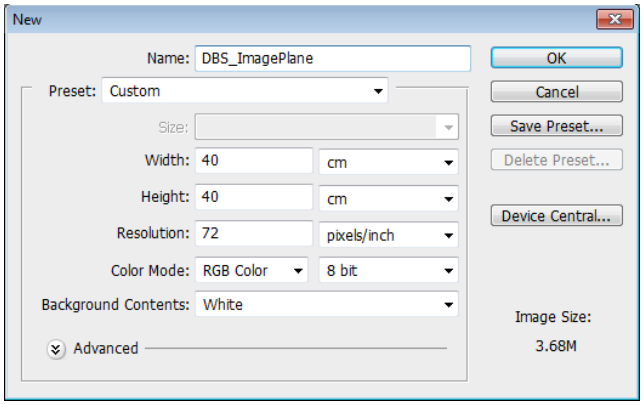
Note: *To bring a tab out into a separate menu, just LMB click and drag onto the workspace.*

For this example I will work in different panes, but you may find it easier to keep them in tab format.

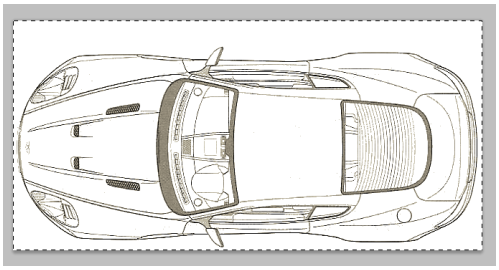


We now have our 4 separate images that we need to bring into one, but we need to ensure that the canvas we create will be of sufficient size to hold our complete image. Select a tab or a window, and navigate to *Image > Canvas Size* and click the option you will be presented with the size of the image, at his point make a mental note or write down the dimensions. The top of my car is approximately 24cm x 12cm (w x h), and the front of my car is approximately 11cm x 9cm. On a quick calculation we need a canvas at least 33cm x 35cm to house our images, so to give us some manoeuvrability we'll round it up to 40cm x 40cm.

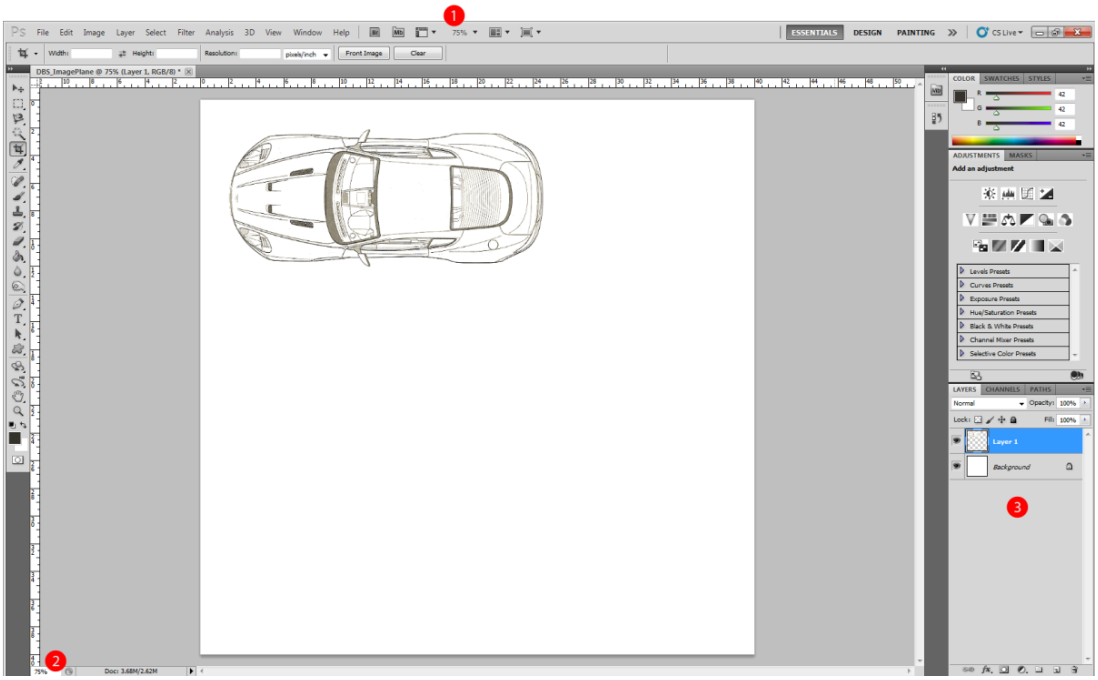
File > *New* and you will be presented with the following box. Fill in your required fields (40x40) ensuring the measurement in the right hand column is in centimetres. Give your file a name (at the top) then click OK.



Now open one of your existing tabs or windows that has one of your perspective views within it, and navigate to *Select* > *All* (*ctrl-a*).



A dotted line will have appeared around the edge at this point which means your image is selected. Navigate to *Edit* > *Copy* (*ctrl-c*) then open your newly created blank 40x40 canvas and select *Edit* > *Paste* (*ctrl-z*). You should then have something that looks like the below screen.



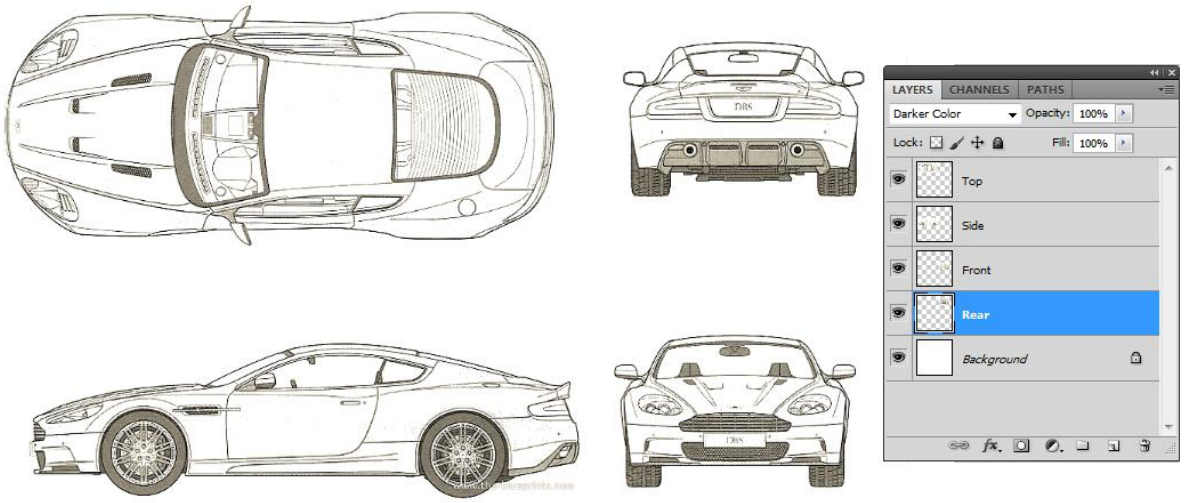
We've highlighted 3 points that might be of interest for you at this point:

1. Zoom menu – if your workspace looks unusually large or small, resize is using a percentage scale.
2. Also a zoom menu (as above).
3. Layers Menu – you will notice you now have two layers, one most probably stating background with a padlock and the other saying layer 1.

Photoshop Layers:

As you would expect; layers allow you to isolate items separately allowing you to manipulate or alter them in ways that will not affect the remainder of the canvas. Once you have a layer selected any alterations will only affect that layer and no other, if at this point you want to reposition your layer, ensure its selected, click the Move Tool (v), top icon LHS, then click and drag accordingly. You may have also noticed an eye on the LHS of each layer, by clicking this you will toggle the layers visibility on/off.

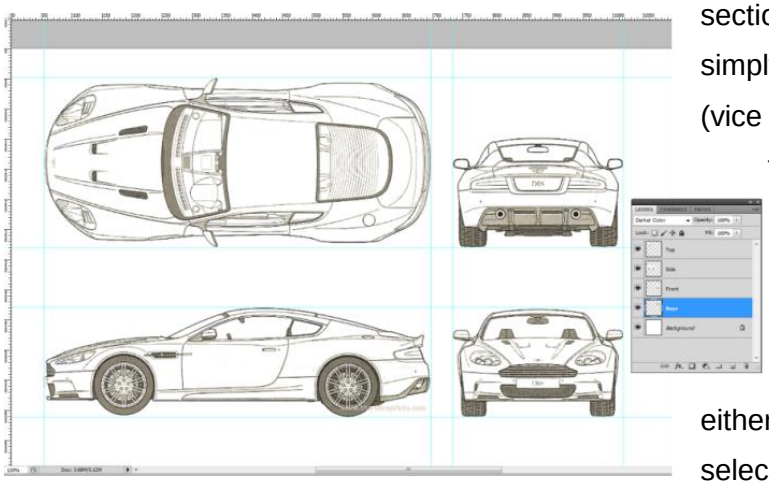
You now need to repeat the process three more times until all your perspective images are situated within the same canvas. You should then be presented with something that looks like this.



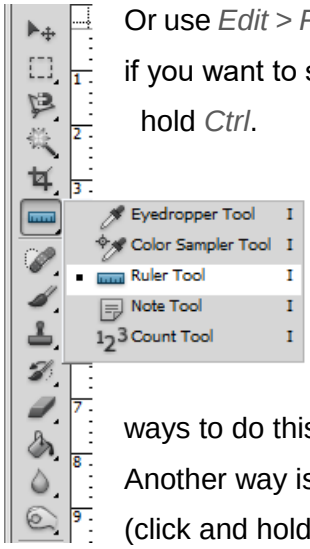
You may notice our layers now have names, so to add layer name firstly you can either select the layer *RMB click > Layer Properties* which will enable you to rename (and colour code, which we will not be using) or double click on the text itself where it says the layer name, being careful at this point to only click the layer name as double clicking the layer itself will open the *Layer Style Menu*, which at this point we will also not be using. We altered the dropdown option under where it says layers (default to *Normal*), to *Darker Colour*. The reason being it removes the white background from each image so if we move them closer together one of them will not disappear.

At this point I would recommend experimenting with the overlap options, highlight one of your images temporarily place it on top of another and filter through the dropdown menu to see the effect.

The next stage is to add in some guide lines that allow us to match up our image perspectives in order to provide us with the most accurate scale possible. To do this ensure the rulers are being displayed around the border of your work area, if they are not go to *View > Rulers* (*ctrl-r*). To create the light blue ruler lines you see on the image below, click on the white ruler section that surrounds the canvas and simply drag them onto your image, (vice versa to remove) and position them in such a way so you can determine if their scale matches or whether they need adjusting. If the image needs adjusting select the appropriate layer and either navigate to *Edit > Transform* and select the appropriate modification tool:



- Scale
- Rotate
- Skew
- Distort
- Perspective
- Warp



Or use *Edit > Free Transform* (*ctrl-t*) and you will be able to manipulate it accordingly, if you want to scale in proportion ensure you're holding shift and if you want to skew hold *Ctrl*.

Note: *If you make an error in Photoshop, to step back one process its ctrl-z, to continue stepping backward you need to press ctrl-alt-z.*

Once you have your images lined up as above its important to check that the width of the top correlates with that of the front/rear. A few ways to do this would be to rotate one of them 90° and see if it's within the margins. Another way is to navigate to *Window > Measurement Log* then select the *ruler tool* (click and hold on the eyedropper tool to be able to select).

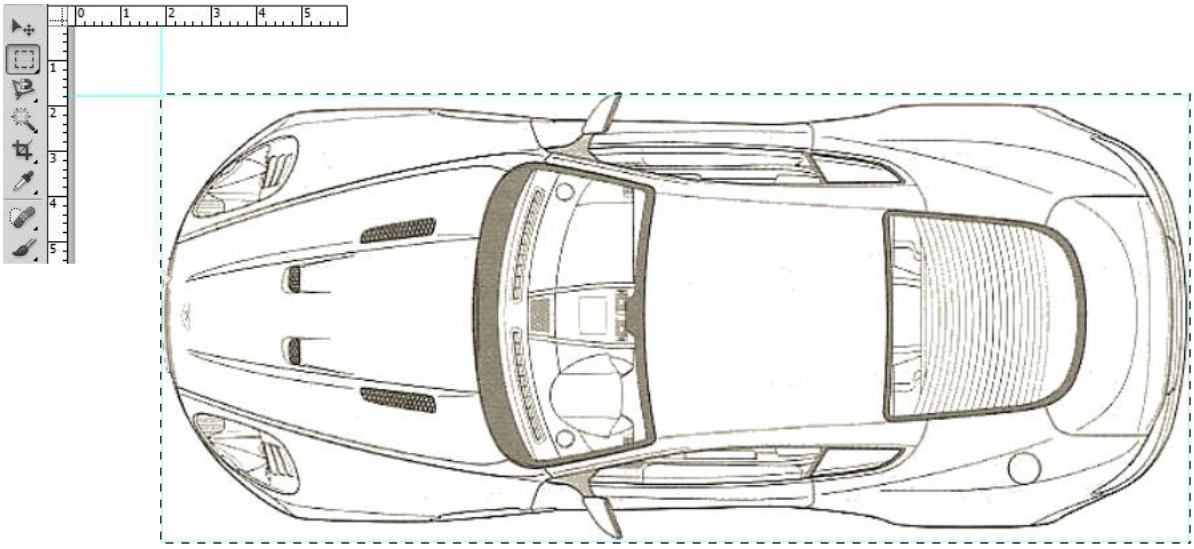
Then with this tool selected you can click on the guidelines one at a time and with the measurement log open click *Record Measurements*. Our measurements are 4pixels apart which we can live with.

ANIMATION (FRAMES) | MEASUREMENT LOG

Record Measurements

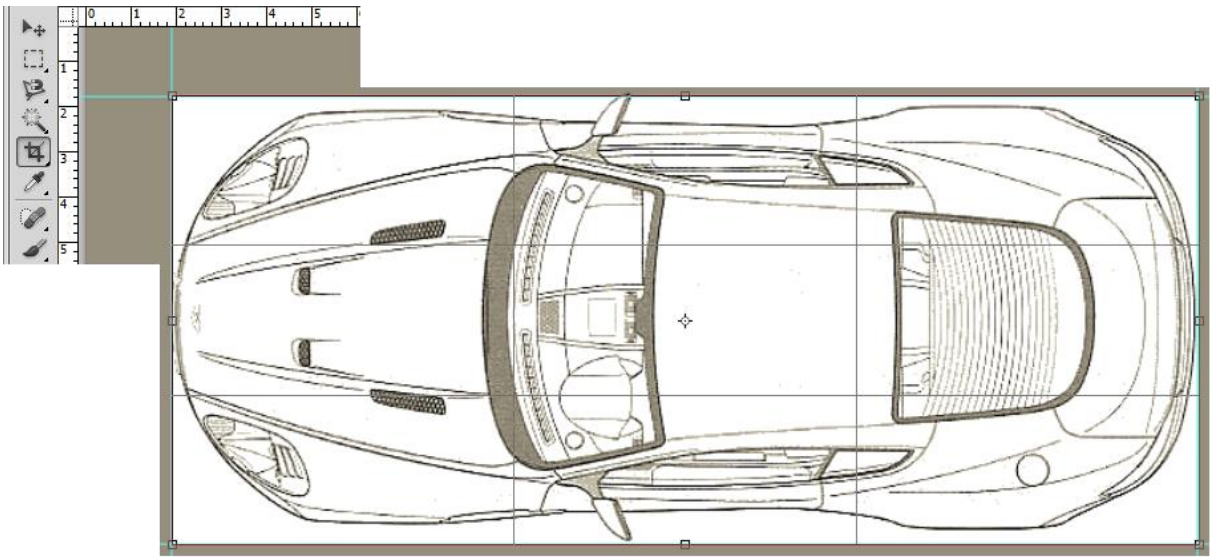
	Label	Date and Time	Document	Source	Scale	Scale Units	Scale Factor	Count	Length	Angle
0001	Ruler 1	06/11/2010 16:17:43	DBS_ImagePlane	Ruler Tool	1 pixels = 1.0000 pixels	pixels	1.000000	1	279.958854	-90.000000
0002	Ruler 2	06/11/2010 16:17:58	DBS_ImagePlane	Ruler Tool	1 pixels = 1.0000 pixels	pixels	1.000000	1	283.510036	0.000000

The final preparation stage is to save out the re-sized images back to singular files. There are a few ways to do this, the first of which is being to select the Rectangular Marquee Tool (by default it will snap to the ruled lines when in the vicinity) and highlight the image, copy the selection (*ctrl-c*) then going to *File > New* Photoshop's default sizing's in the box at this time will be the exact same as the copied image, so by clicking OK and then pasting (*ctrl-v*) the image it will fit perfectly. If you want to increase the canvas size to a whole number i.e. 8cm x 12cm for easy referencing navigate to *Image > Canvas Size* (this doesn't alter the image size whatsoever), save the file to complete.



When you get to the rear image (as it hasn't got guide all around) the car, to keep correlation with the front I hide/remove the front layer and fit the image between its guidelines this way you know it will be the exact same size.

The second option uses a similar selection technique but with the cropping tool instead. Once the image is highlighted double click within the selection and everything will disappear. Save the image then press *ctrl-alt-z* until you're presented with your full image again and repeat. Typical naming convention "Car_Top".

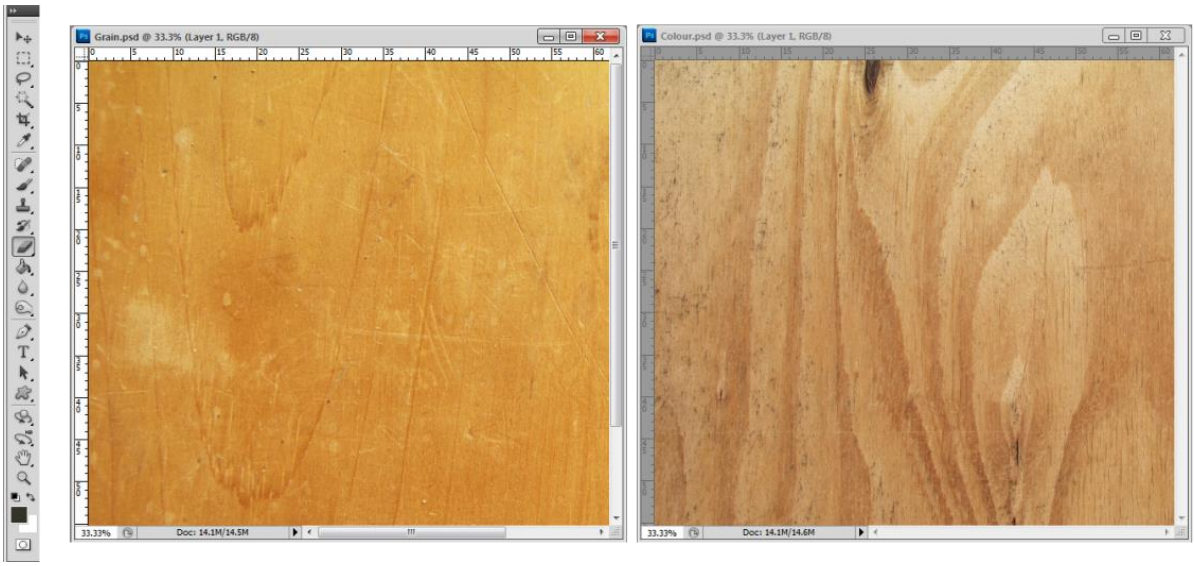


Colour Matching

There will be times when you like the design or print of one image but prefer the colour of another; in this section we will show you how to overcome this problem with various techniques.

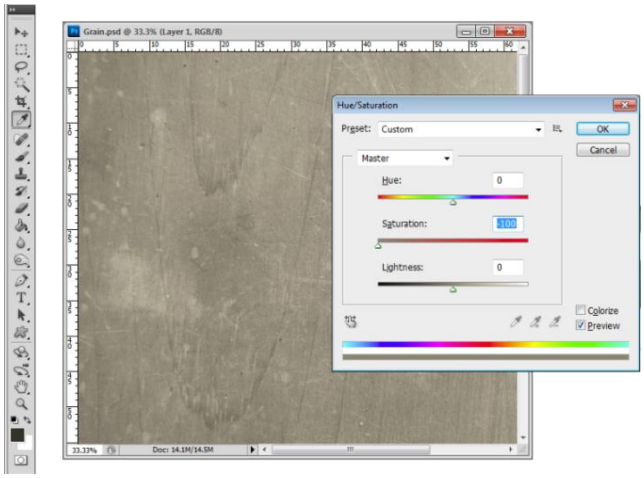
Match Colour

1) Open both your files in Photoshop. For all intents and purposes I'm going to say that I prefer the grain on the left image but the colour of the one on the right.



Note: for reference purposes you may wish to create a duplicate layer before continuing.

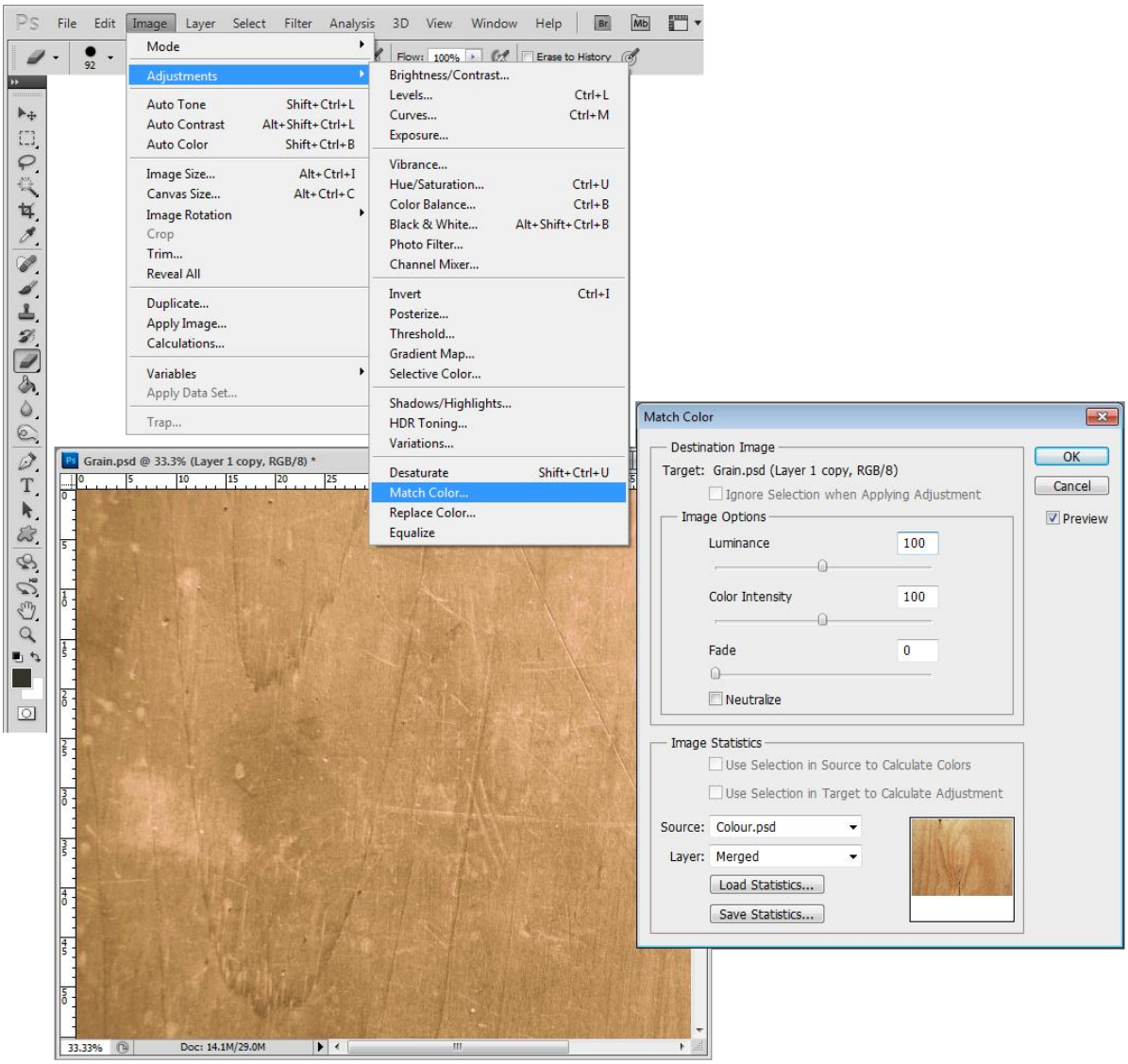
2) (Optional step – *completed in demonstration*) Click on the left image and desaturate (remove all colour) *Image > Adjustments > Hue/Saturation* (Ctrl+U), drag the saturation slider all the way to the left (-100). You will then be left with a greyscale image.



3) (Optional step – *not completed in demonstration*) Blur the colour you intend to match, select the file (in our case it would be the one on the right), navigate to *Filter > Blur > Gaussian Blur* (transition sliders accordingly).

4) The next stage is to navigate *to Image > Adjustments > Match Colour…* The 'Match Colour' dialogue box will appear and you will need to select the source at the bottom before you can adjust the values. The source will be the other file you opened at the beginning and you can specify a layer if you wish. Once the source has been selected (Colour.psd in our case) you can then use the sliders to create the image tone that you require, once finished click OK.

Occasionally on aspects such as wooden floors the divisions between them may require re-emphasising after the merge, to do this select *Filter > Sharpen > Smart Sharpen* and alter the sliders until the desired end effect is achieved.



As you can see it creates a nice colour merge between the two textures, resulting in a professional and realistic alternate wood effect.

Seamless Texture Creation

In this section we will take you through the process of creating a texture that can be repeated seamlessly, and reducing any repeating patterns that may arise. Bricks are a little more challenging than say a wooden floor but the processes involved are identical.

Step 1: Importing into Photoshop

Locate the file and import this into Photoshop, it makes for good practice that texture images are square and our preferred size is 1024 x 1024. When re-sizing an image try to make a little less work for yourself by not having half / quarter bricks visible on top and bottom, try to keep them whole, this isn't essential but a good practice.

Step 2: Offsetting the Image

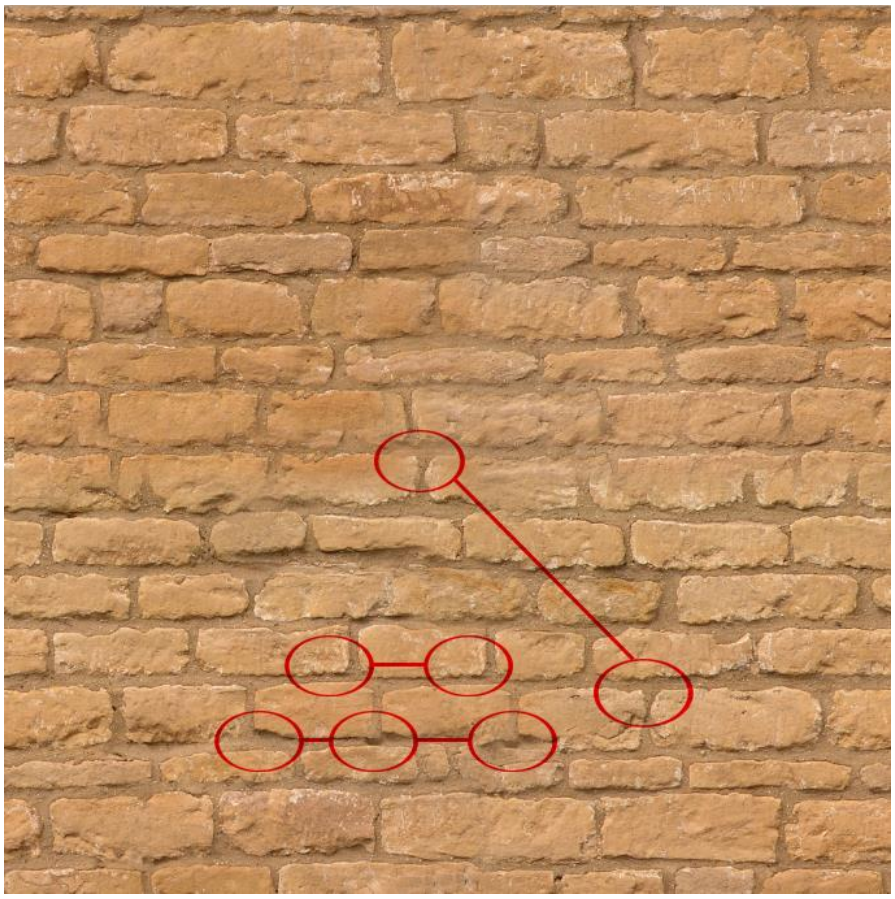
Once the file is within Photoshop (as sized accordingly) navigate to *Filter* > *Other* > *Offset*. Offset both approximately 50% (512 pixels) and generally at this stage you will have the appearance of a cross in the centre of your image

Step 3: Seam Removal

The next step is to use the clone stamp at 100% with feathered edges to remove the sharp visible seams both horizontally and vertically. To use this tool select an appropriate brush size (50-70px), hover over an area that you would like the clone to commence (usually something in the general vicinity is best suited as colours are often similar) hold *Alt-LMB*. This gives the starting point for the clone, then when you hover with your mouse curser you will see a preview of the stamp within the circular boundary of your curser. The tool moves in accordance to your mouse navigation so if this is your first time using the tool it may take some getting used to.

Step 4: Tidying up the texture

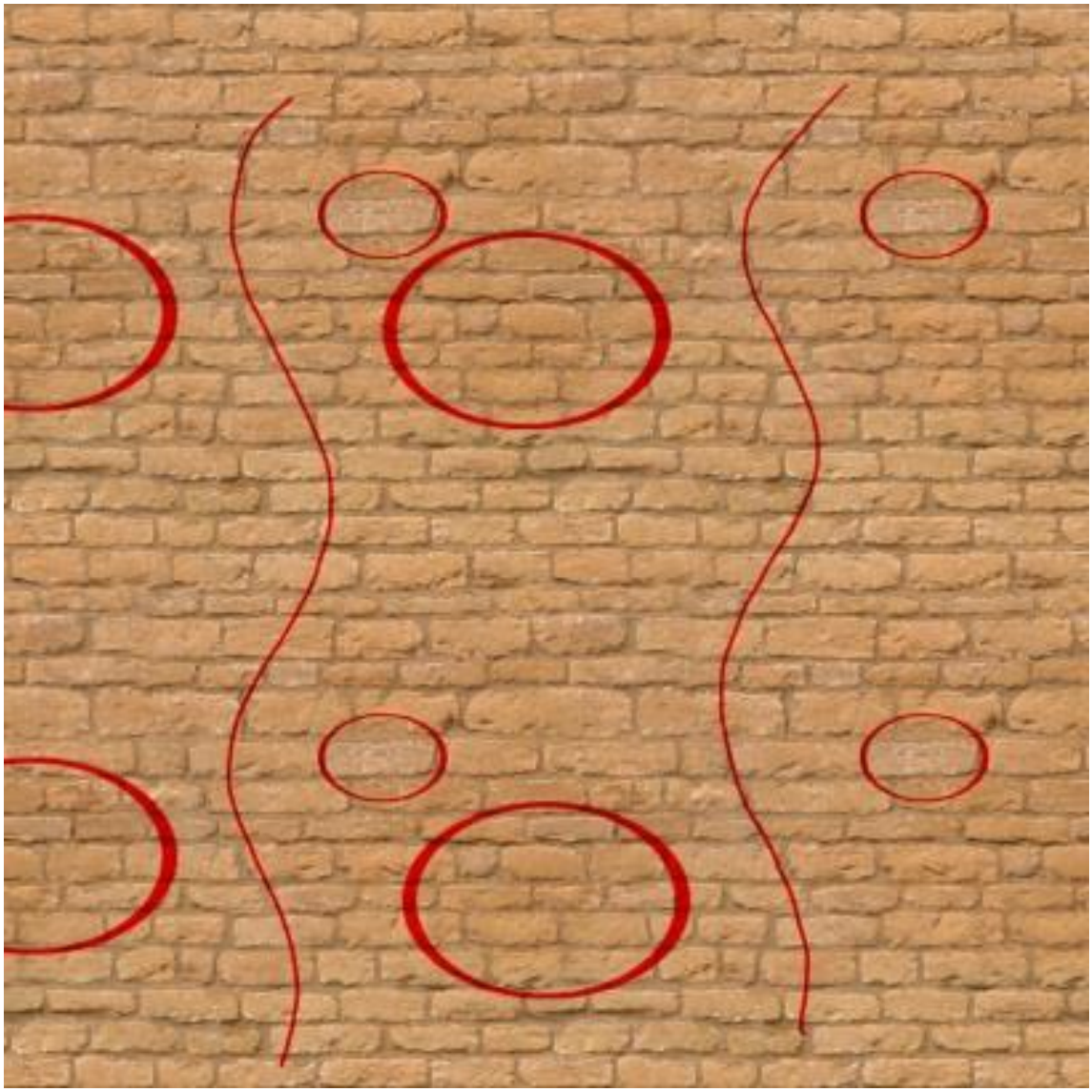
You will then most probably need to tidy things up a little, whilst also noticing that the horizontal line is not as visible as the vertical as we cropped this earlier at a suitable point. The majority of the initial work outlay it best carried out with the *clone stamp* with no opacity applied. Below we hoped to illustrate the ideology behind this process by paying particular attention to duplicating the corners and edges of similarly sized bricks we are able to help mask the seam down the centre.



Continue defining the corners and edges, and using random brick features to define and distinguish others. You may find at times sections become blurred, to counteract this once you have found the feature you want to replicate, click in the same place repeatedly to apply the stamp more visibly. If as well keep referring back to the same sections or can still see the seams etc., the easiest thing to do is jump back to **step 2**, (offsetting the image for a second time using different numbers), then continuing with the above process.

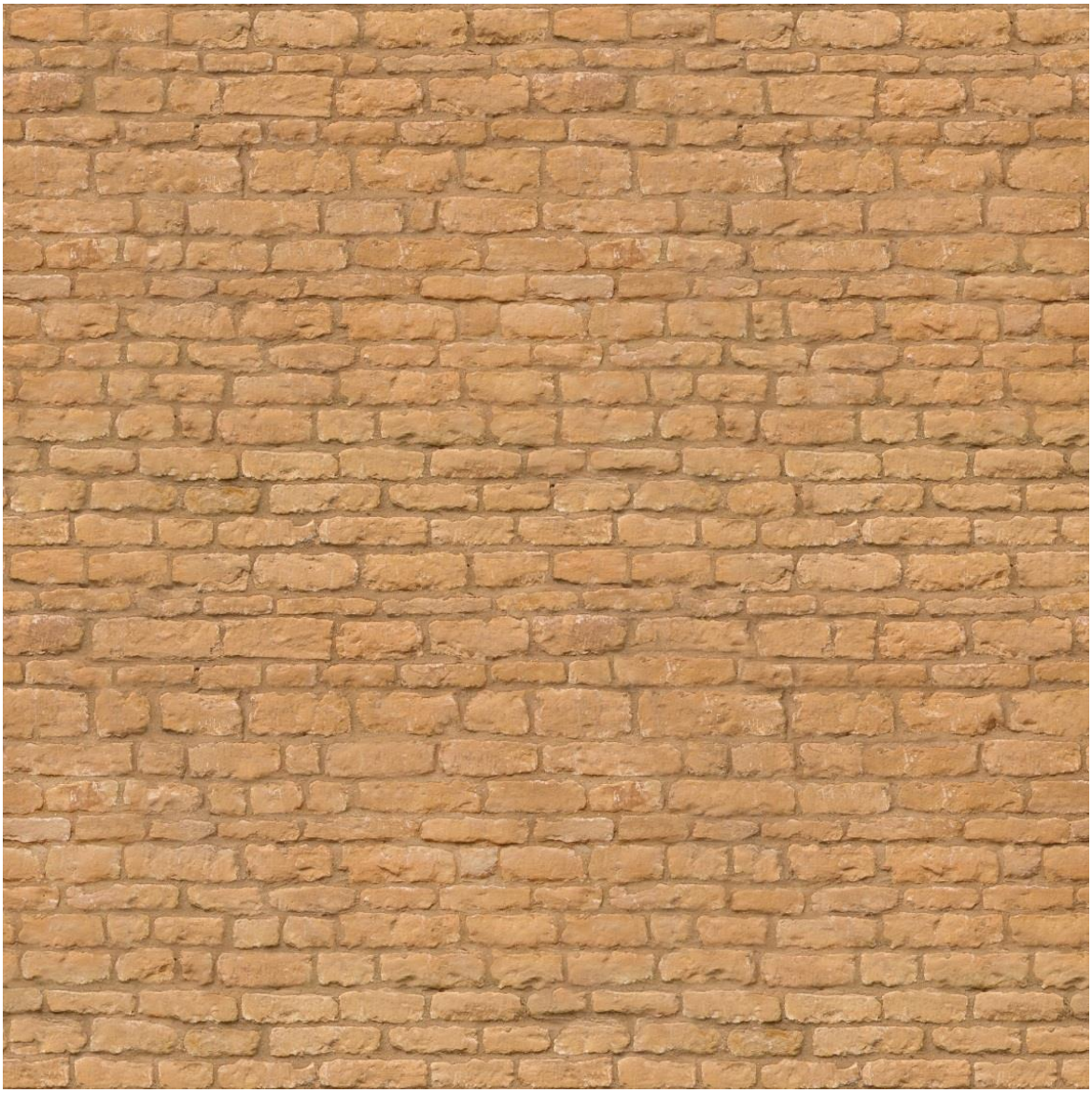
Step 5: Checking Seams / Enlarging the Image

The next step you can see if what you have done makes for a seamless texture. Create a blank canvas 2048 x 2048 and past in your image 4 times and position appropriately, from all accounts we have achieved a seamless texture but the only thing is the human brain is exceptionally good at recognising patterns, so if this is to be spread over a large area without additional props you may want to look into disguising these “repeating attributes”. We’ve highlighted a few obvious ones, which will always generally include a distinguishable brick pattern, some shading and a brink that simply stands out from the rest of them.



Whether you want to stick with the larger image or refer back to the previous one the general processes will be the same.

- To lighten your dark spots use the dodge tool on a low (5%) exposure until satisfactory.
- If using the large image take out the key repeating features, colour some of the bricks and alter their appearance using techniques from earlier steps to give a truly authentic feel.



The above image has had “arguably” the most distinguishable repeating features removed/disguised. The process can take a little bit of time in the early days but you’ll quickly adhere to the tools and techniques if done regularly.

You can of course advance on these textures as you see fit, however once you have finished tweaking or if you decide to perform tasks such as colour matching, just ensure you offset the UV’s to check no seams have been re- created, (if visible they should only be slight and at this stage it shouldn’t take long to rectify).

CHAPTER 4: LIGHTING

Chapter 4: Lighting

Introduction:

As most of you would already presume, lighting a scene in Maya is much like that of the real world, many elements from situations like studio or natural lighting can be adapted, then replicated for use within Maya. Ensuring the scene lighting is done effectively is pivotal when trying to make your character belong within a scene. The default lighting of Maya is only used when there are no other lights within the scene.

Getting Started

Many peoples' first experience with lighting is, let's just stick any light in and hope it looks alright. Normally followed by playing with additional lights, moving them around, or using ridiculous settings until you get bored.

We want to venture into various types of lighting, take you through a couple of techniques and teach you some workflows that can be used for any piece of work.

Lighting Types:

Ambient

This is used to replicate a scattered or non-directional light that you see all around you on a daily basis.

Area

This light source replicates the realistic distribution of a soft light source like that emitted by a florescent tube.

Directional

A singular large light source (like that of the sun), that creates very little effect on surface shading due to its magnitude.

Point

As the name suggests this light source originates at a point and has an omnidirectional luminosity like that created by a standard bulb or a candle.

Spot

As you may guess the spot light emits rays in a cone like fashion, and would be used to replicate things like car headlights or a lamp.

Volumetric

Lights objects within a desired area, good for replicating a closed environment like that inside a car.

Lighting Terminology

Intensity

Simply put is how bright the light source is.

Drop-off

This is where the light fades perpendicular to the axis rather than along it; creating a feathered effect around the edge.

Specular Highlight

This is the bright spot of light that a light has on an object, most noticeable on the rounded edges of an object.

Decay Rate

Is the effect distance has on a light source, the further away the light from the source the less intense are the effects. Decay regions are used to

Penumbra angle

This relates to the area of lessening strength around the outer edge of the beam of light:

Shadows

Shadows play in extremely important part in creating scene depth and it's important you put some thought into how you want your scene to look as a difference in light source could make all the difference.

Depth maps

Depth map shadows are a quick and efficient method of adding shadows to your scene. It consists of projecting a map of a certain resolution from the light source to calculate the shadow that should be cast.

Samples:

The greater the number of samples, the smoother the shadow will be (but the longer it will take to render).

Softness:

Defines how soft the shadow will be at the edges.

Depth map filter size

A similar result to that of the penumbra angle but on the edge of a shadow as opposed to a light source.

Motion-Blur Shadows

Even with motion blur enabled, Maya is unable to calculate it for shadows so you need to perform a workaround.

There are two ways to do this. The first would be to switch to Mental Ray® Shadow Maps, which do calculate the motion blur (but will also increase render time, and will change the look of your scene somewhat because you are using a different renderer).

The Second would be to create a separate Shadow Pass and use compositing software to blur it and put it together in post.

Volumetric lighting effects

The ability to cast shadows through fog can exert some beautiful results. This is key to atmospheric shots such as the shadow of a window through a dusty church.

Ray-traced Shadows

An increase in render time will be noticed but Ray Tracing provides key advantages to those of Depth maps.

Because it is a more accurate method of calculation, Maya can render transparency and colour through an object, a stained glass window through Ray tracing will show the colour and density difference of the glass in each pane.

Ray tracing also allows you to set a degree of attenuation, so that the shadow 'blurs' the farther it is from the base of the object. This is most applicable to object that need to look like they are being lit softly, by a light box for example.

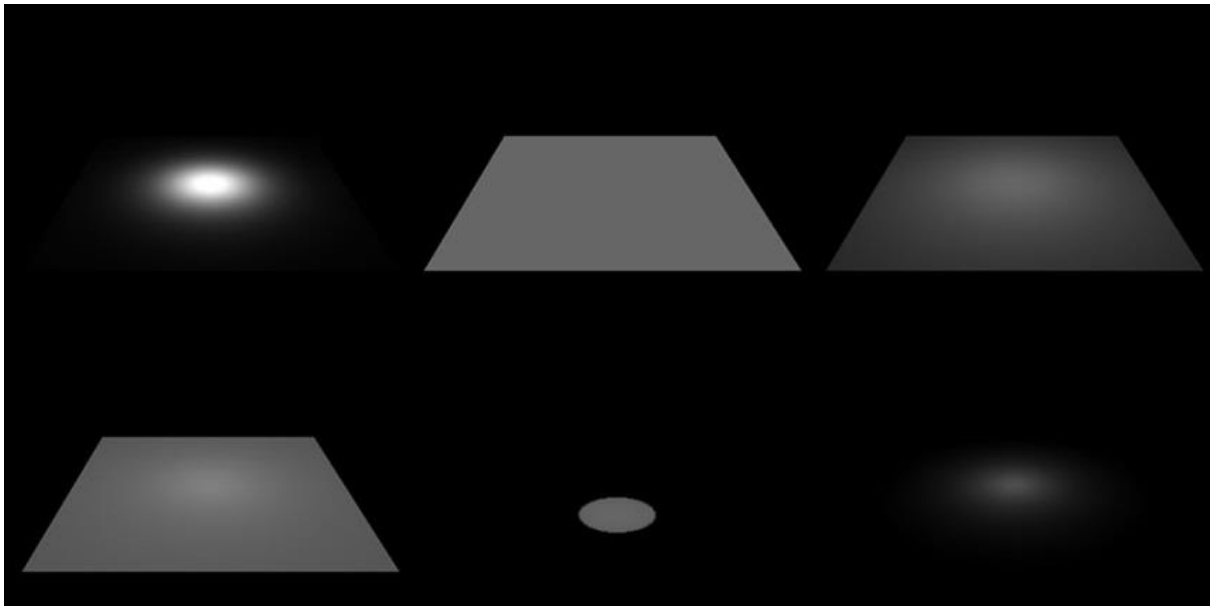
Mental Ray® Shadow Maps

Re-using shadow maps for static items can help decrease render times

Mental Ray > Render Settings > Shadows > Rebuild Mode > (set to) Reuse Existing Maps

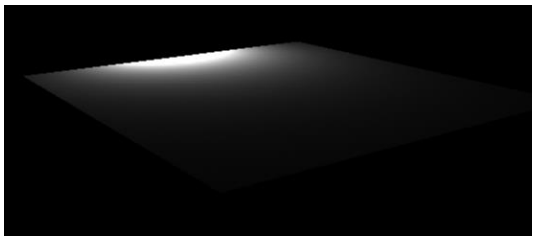
Light Source > Attribute Editor > Mental Ray > Shadows > (enable) Use Mental Ray Shadow Map Overrides

Lighting Examples (Rendered using Mental Ray)

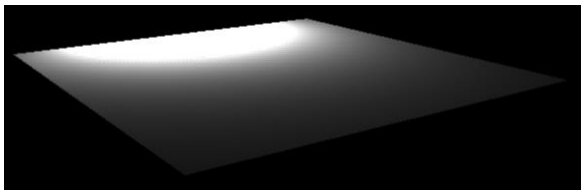


(Left - Right: Area, Directional, Point, Ambient, Spot, Volumetric)

We will opt to use the Area Light as it has natural decay so let's see what this looks like when it's put on the side of a scene.

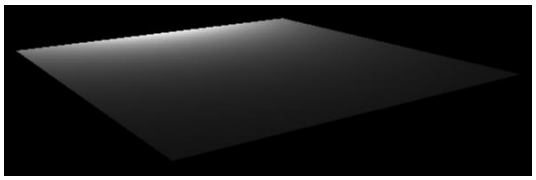


As you can see we're not getting an awful lot of light onto the image plane. At this point some people have a tendency to increase the intensity and/or move the source. So let's have a look at what those people would see if they popped the intensity up from 1 to 3.5.



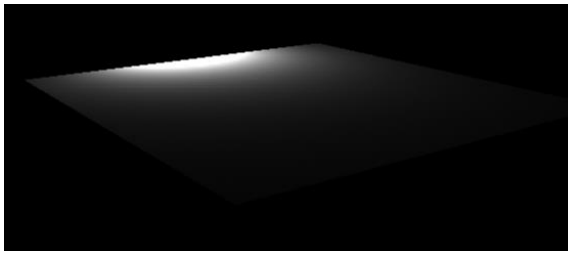
Looking at the image on the left we now have more light in the scene but have also acquired a large hotspot. Let's move back the source back a little bit and crank up the intensity to compensate for the move.

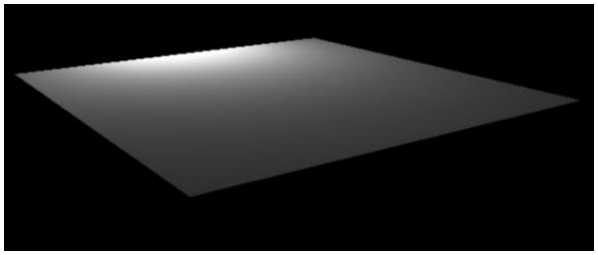
There is now more light and less of a hotspot, but in reality this is an awful way of lighting a scene and should be avoided at all costs.



Gamma Correction

Now we're going to backtrack slightly, what you're actually seeing in that first image is a raw format without any Gamma correction. Gamma correction is required to compensate for the properties of human vision specifically in relation to luminance. All images you interact with on a daily basis, whether it's through your TV, Magazines, or even a computer are subject to some form of gamma correction.





Same image, but the image on the right has a gamma corrective lens applied to the camera.

Setup a gamma correction lens

Firstly we would recommend creating a new camera (leaving the perspective camera with its default settings) *Create > Cameras > Camera.*

Next select your scene camera (not Perspective, easiest way is to navigate to *Windows > Outliner* select new camera), Open the *Attribute Editor > camera shape tab > Mental Ray > Lens Shader (press the chequered box)> Lenses > mia_exposure_simple*

It will jump to the lens shader and you will see some additional features that we will come back to a little later on:

- Pedestal – Blackness control
- Gain – Brightness
- Knee – Helps to lower hotspots in the scene
- Compression – Provides intensity for the Knee
- Gamma – Darkness

Physical Sun and Sky

We will be creating a very quick environment setup, that Maya has nicely provided for us *Render Settings (ensure you are in mental ray) > Indirect Lighting > Environment > Physical Sun and Sky > Create*

Now you can have a little play. Open the outliner and look for the “sunDirection” light, make it bigger using the scale attribute (r) so you can actually see it in you (to scale) scene and you will notice it’s rotated down at approximately 75 degrees, from that angle would make it around mid-afternoon.

Now here’s the cool part, it doesn’t matter where a directional light is positioned within your scene it’s purely based on its angle, so if you rotate the direction directional light down to around 10 degrees, all your shadows will change and the horizon should begin to appear more “orangey”. If you also drop your camera to somewhere near the horizon, and towards the source you should see the sun spot, and if you continue rotating the “sunDirection” you will see a sunset.

Now it doesn’t get any easier than that. But if you would like to delve a little further into the settings here’s what has been done. You will notice two new additions in the windows **> rendering editors > Hypershade > Utilities**

Physical Sky

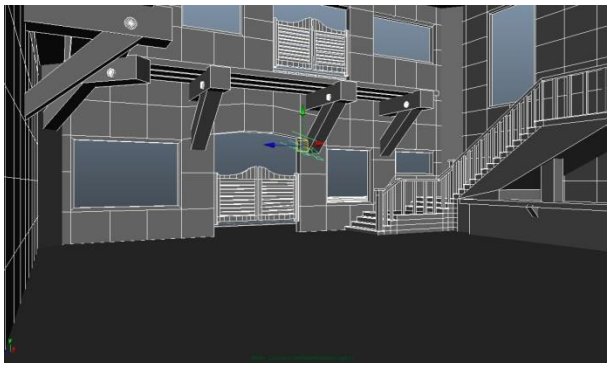
- Multiplier – Intensity of the sun
- R,G,B – Won’t be used
- Haze – Adds a “haze” to the horizon and creates a more yellow scene
- Red, Blue Shift – Intensify the feeling of scene temperature or compliment a sunset scene
- Saturation –
- Horizon Height – Moves the horizon up/down
- Horizon blur –
- Ground Colour – Changes the ground colour (drastically alter scene appearance – final gather etc.)
- Night Colour
- Sun Direction – Not really used as we will be using “sunDirection” object within the scene to move the sun around.
- Sun Disc Intensity - If you have a shot that incorporates the sun disc
- Sun Disc Scale – Size of the sun

Physical Sun

You will notice that the main settings are mapped to the physical sky node. For our purposes they will remain linked but you can always have a play around if you wish.

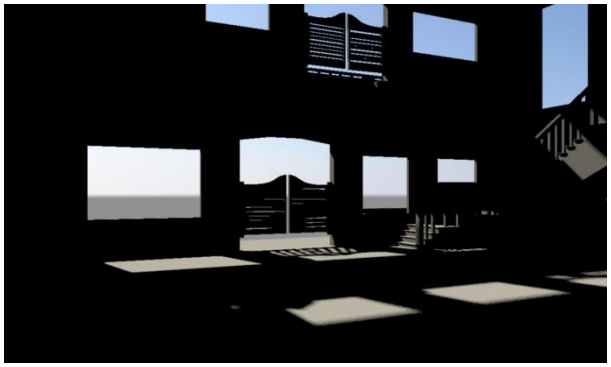
Examples

Scene



A very basic low geometry scene with a physical sun and sky node you can see in the centre.

Mental Ray Render: Day



Rendered with a gamma corrective lens and a physical sun and sky, with default settings only

Mental Ray Render: Dusk (Rotated the sunDiretion)

Same as above only with the sunDirection rotated to a lesser angle, creating longer shadows and a time that would imply dusk is approaching.

Render Settings

- ***Indirect Lighting > Final Gathering***
    - Point density : Reduce from 1 to 0.5
    - Interpolation: Increase 10 to 20

- ***Quality > Raytrace/Scanline Quality***
    - Max Sample Level: From 0 to 2
- ***Quality > Sample Options***
    - Jitter: Tick the box

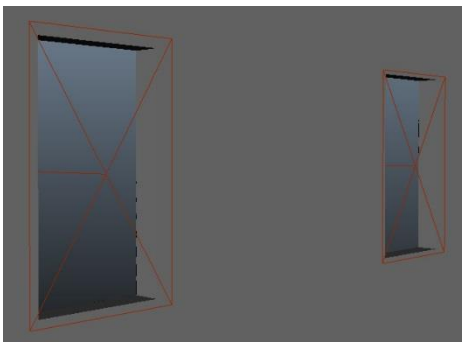
You will notice an increase in render times but with dramatic effect

Portal Lights

A portal light helps to brighten the scene by bleeding exterior light such as the sun and sky into the scene. Portal lights need to cover all “holes” in your scene where light could escape as their purpose is to bounce the final gather rays back into the room.

- Create an area light
- ***Attribute editor > Light shape > mental ray***
    - ***Area Light***
        - ***Tick Use light shape***
        - ***Tick Visible***
    - ***custom shaders > Light Shader (chequered box) > mental Ray > mia_portal_light***

Next step is to repeat the process covering all windows and doors. It is best practice to have one area light per hole, however if you have a few holes in close proximity then you may opt to cover them with one area light.

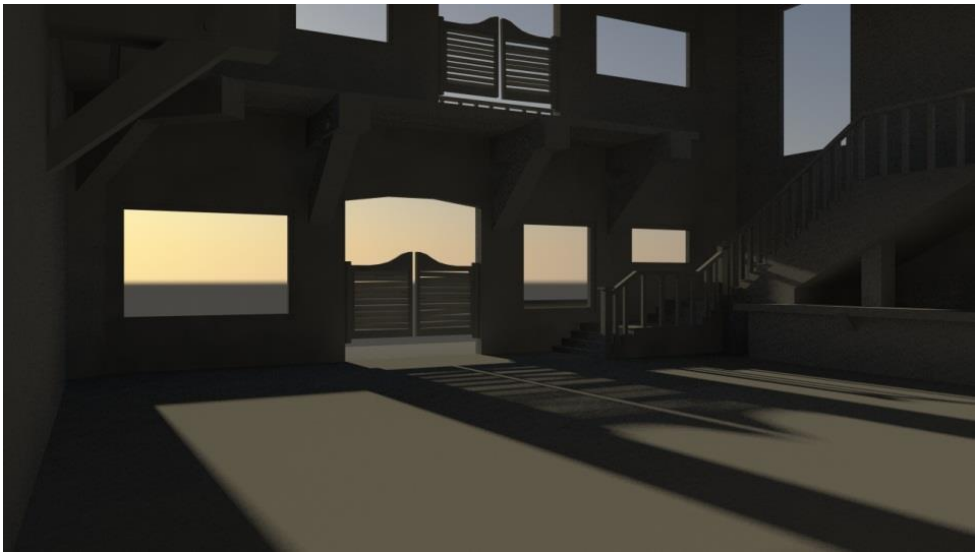


Create a separate portal light shader for each area light

A greater degree of control, meaning you can increase light intensity from a specific window however this level of control isn't always required.

Map the same portal light shader to many/all of the area lights

This is my preferred method and I usually map one portal light to every area light on that side of the building. This way you control all of the light on one side of the building via one setting and room lighting can be adjusted quickly. To do this simply open the ***Hypershade window > utilities*** and selecting each area light in turn and MMB click and drag mia_portal_light into the Light Shader.



Adding Textures

Depending on the availability of light within your scene you may find it still rendering a little dark. If you haven't already the next step would be to add in some textures as these can have a dramatic upon the final result.

Textures added:

- Wood – Floor, Banister, Beams, Balcony, and Window Frames
- Glass – Windows, Tumbler, Whisky Bottle, Chandelier
- Backing Plate & Wallpaper

Adding additional light into your scene

There is no right or wrong way to do this so we will address a few techniques. Take note that when we are referring to light bounces these can be drastically changed by adding textures into your scene, bouncing the default grey lambert is going to create a darker scene than a beige wallpaper.

Increase the portal light intensity

Not our preferred, method however by increasing the portal light intensity will subtly increase the light into your scene, however if you're having to use considerably high values (bearing in mind 1 is usually an accurate representation of natural light), I would possibly return the default value (1) and try another method.

Final Gather

The "secondary diffuse bounces" is the field you need to increase. This is the number of times the light source will bounce of a surface. The higher the number the more light is bouncing around your scene but you will notice a considerable increase in render times as this number increases

Global Illumination

When using global illumination to calculate secondary diffuse bounces its best to ensure the final gather secondary diffuse bounces are returned to "0". You will also need to emit photons into your scene for this to work.

Jump back into your Portal light(s), and go into the attribute editor

- ***Caustic and Global Illumination > Emit photons (tick)***
- ***Custom Shaders > Photon Emitter >*** Either
    - ***Create relationship to the existing portal light*** *(i.e. copy and paste content from* ***Custom Shaders > Light Shader*** then press enter*) Or*
    - For greater control you can create an additional Portal light (refer to above process) to control the photon emitter however for most circumstances mapping it to the same one will be sufficient.

Reducing render times

The easiest way to reduce your render times whilst maintaining the improvements we have just made is to take a final gather map. This is for convenience, and works best with a static camera. If you wish to make changes to your scene i.e. lighting, camera angle, textures etc. it's advisable to repeat the below process.

- Set a camera bookmark or key your camera in position
- ***Render Settings > Indirect Lighting > Final Gather***
    - ***Rebuild > ON***
    - ***Primary Final gather File -*** *Add file name (i.e fgMap)*
- Render your Scene (keep the image)
- ***Render Settings > Indirect Lighting > Final Gather > Rebuild > Freeze***
- You should notice an increase in render times of around 30-50%

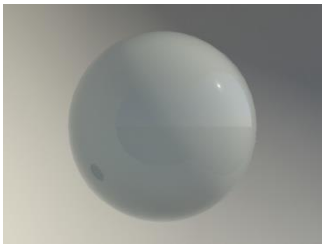
Interior Lighting

The sun position would imply it's fairly late in the day so our lighting setup needs to be warm but not too overpowering.

To achieve this effect you need two primary elements:

- The light itself
- The glass material where it originates

Glass Material



A very simple *mia_mater_x > presets* > FrostedGlass*.

Here is an example of the preset when it is applied and rendered in an empty scene.

Light

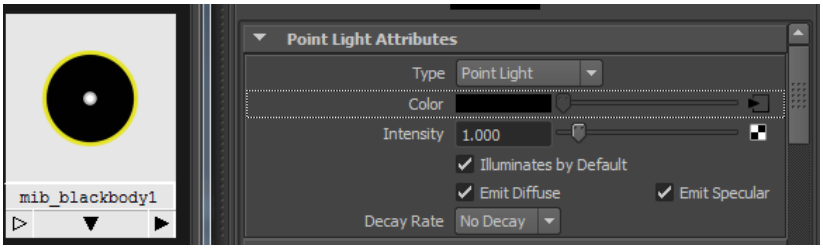
First off we need to:

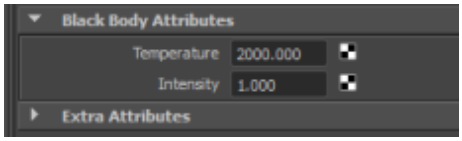
- *Create > Lights > Point Light*
- Point light *attributes > Decay Rate > Quadratic* (Light fades with distance)

Note: When applying decay in models that are to scale the intensity value needs to be very high. In our scene we will be applying values of 4,000 – 11,000.

- *Raytrace Shadow Attributes > Use Raytrace Shadows (tick)*

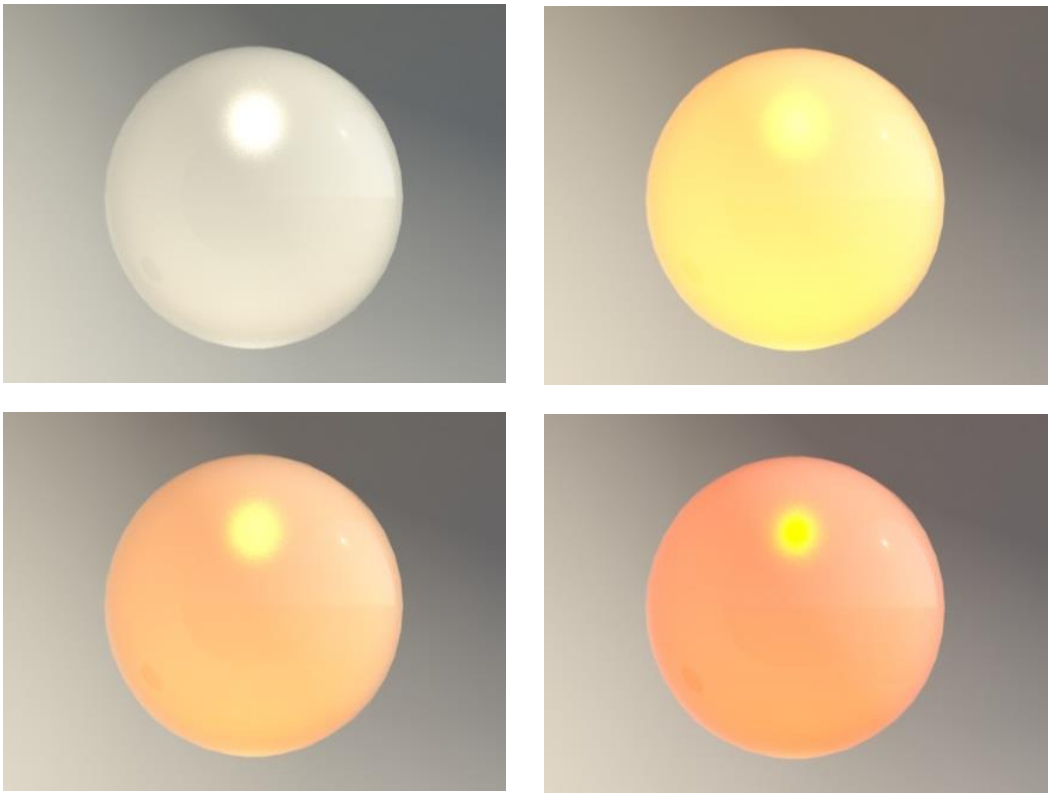
Now we are going to open up the *Window > Rendering Editors > Hypershade* and in the *MentalRay Lights* create an *mib_blackbody*. MMB click and drag the *mib_blackbody* onto the *colour* attribute of your *Point Light*.





Click the square with an arrow going into it after the colour and you will be taken to the Black Body Attributes.

Have a play with the Intensity value of the Point Light and the temperature of the mib_backbody to cycle through a range of lighting for your environment.

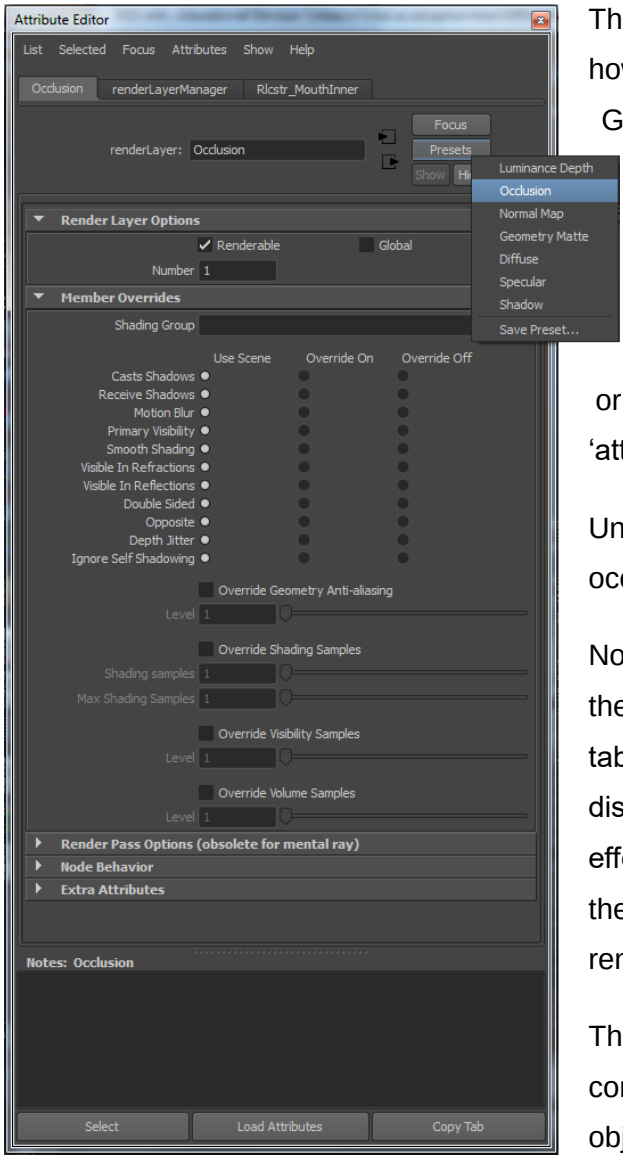


We have used a temperature value of 2000 and intensity values of between 4,000 -11,000 on different lights in the following render.

More advanced lighting techniques:

Ambient Occlusion:

Ambient Occlusion is a shading method based on the proximity of nearby objects. This results in a faster (albeit less realistic) Global Illumination effect, and is usually combined with a fast Maya Software render to give a very efficient compromise between image quality and speed.



That is not to say that Ambient Occlusion is fast however, despite taking much less time that full Global Illumination, large scene can still take 10mins or more to complete.

Setting up an Ambient Occlusion pass in Maya is very quick. In the renders tab, create a new layer and call it ‘ambientOcc’ or similar. Right click on the layer and select the ‘attributes’ option at the bottom of the menu.

Under the presets button, select ambient occlusion from the drop down list.

Now all we need to do is tweak the settings of the shader node. The key setting is on the 2nd tab and is named “samples”, spread and distance. The ‘samples’ is the quality of the effect. The lower the number, the more ‘grainy’ the image will be, but the faster it will be rendered (and vice-versa).

The spread affects how harsh or soft the shader comes across, a setting of 0.8 is good for most objects, but lower can be useful sometimes to emphasise cracks and creases in the geometry.

The max distance determines how far away a face has to be before its ‘occluding’ effect is calculated. This setting is entirely dependent on the size of your scene, too small and you will not get a strong enough effect, but too great and the scene will look very dark.

When you are happy with the result of your Occlusion Pass, we want to take both the original diffuse (colour) render and the occlusion pass into Photoshop (or After Effects if you have a sequence). Lay the occlusion pass over the top of the colour render and change the blending style to ‘multiply’.

There we have it. Our finished image:

Light-Linking:

You can use light-linking to control which lights affect specific objects/shaders. This can be useful to limit a certain light to a specific object, for example if you want to illuminate some of the environment, but not affect the lighting of your characters.

You can also use light-linking to stop Maya from trying to calculate shadows/reflections/etc. for objects that are out of range of certain lights, in a large scene this can go a long way to reduce render time.

You can access light-linking from *Window > Relationship Editors > Light Linking*. There are then two options, Light-Centric or Object-Centric, depending on whether you want to specify which objects are affected by specific lights, or which lights you want to affect specific objects.

Unorthodox Lighting techniques:

If you can get away with faking something, DO IT!

If there is a way to fake a complicated simulation in a fraction of the time, it's always worth looking into. The results will rarely be as good, but the small compromise of image quality can give you a huge gain in rendering times.

An example of this would be;

Use lights as shadows:

The intensity value of lights can be any value, including negative figures. By providing a strong decay, and placing the light at the base of an object (rounder objects work best, of course) you can emulate the soft shadow effect of global illumination.

Raytracing:

Raytracing is how Maya simulates light traveling through your scene. Light can be reflected or refracted dependant on the properties of a material.

To enable raytracing select *Window > Rendering Editors > Render Settings > Maya Software (tab) > Raytracing Quality > on (checkbox)*.

3-Point Lighting

The technique's simplistic ideology and versatile approach have enabled it to become possibly the most popular lighting technique used across todays various media forms as well as forming the base for more complex lighting structures. The technique uses three lights, but the first stage is to create a camera you will use to render the scene:

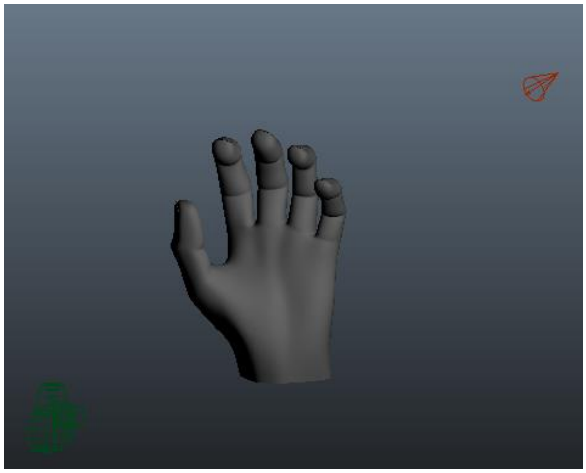
Create a new camera

- *Create > Camera > Camera* and position accordingly.

Light 1 (Key Light)

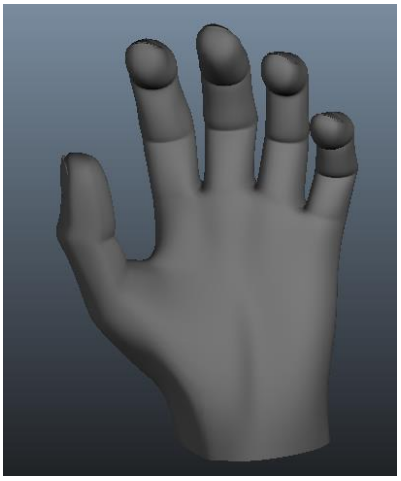
The key light will have the strongest light source and will have the most influence upon your scene.

- *Create > Lights > Spotlight*
- Set approximately 45° around from the camera, raise it above your object and angle it down accordingly.
    - To look through the light whilst you position it *select light > Panels > Look through selected*



Note: *to see how your scene is using lights (above), within your image view go to Lighting > Use All Lights.*

Light 2 (Fill light)



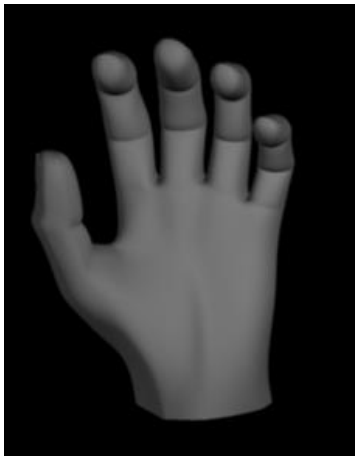
A secondary light which is positioned on the opposite side of the camera from the key light. Ideally this will cast little or no shadows.

- *Create > Lights > Directional Light*
- 50% light intensity and below is a recommended starting point as this helps to soften lights and shadows, but you can tweak these at your discretion.

Light 3 (Back Light)

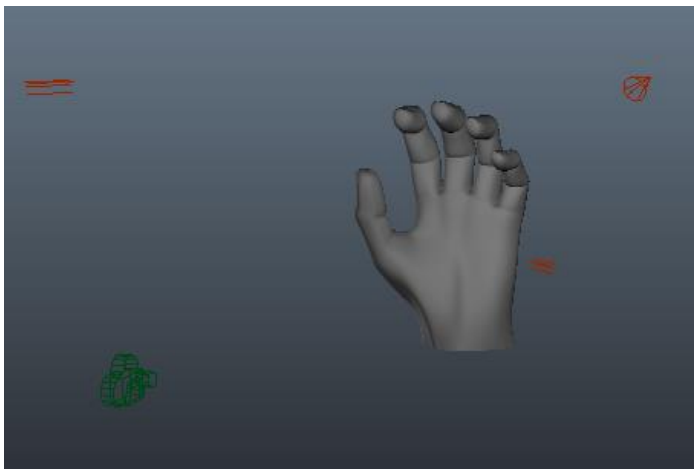
Positioned behind your object and can be situated either side for varying effects. The purpose is to separate the object from its foreground, to provide definition and add a 3D look.

- *Create > Lights > Directional.*
- Rotate so it creates a very thin light rim around the edge of your object



More often than not you will require shadows to be cast, however this may cause you to have multiple shadows being on the ground. To counteract this effect you will need to perform a light link.

Note: Alternate Lights can be added at your discretion and if you feel a light type we have used doesn't look correct for your scene you can replace it with a suitable alternative.



CHAPTER 5: RIGGING

Chapter 5: Rigging

Introduction to Rigging

Rigging can often be related to solving a puzzle. You are given a model and an explanation of how it needs to be able to move and you have to solve the problem in the most efficient way, both in the way that the computer understands the control (so that the playback is fluid) and the way that the animator interprets the control so that it is intuitive.

Joint Chain

Joint chains are the most common method of driving a mesh object. Joints represent the object's skeleton and the object's mesh is bound to the joint chain to allow it to deform and conform to the shape of the chain as the joints are rotated.

Forward Kinematics

To achieve a forward kinematic movement it requires the individual movement of each singular joint and for it to be keyed correctly. This would mean if you wanted to make your character scratch his head you would need to key the shoulder, elbow and wrist movement individually. Although this sounds time consuming, because this is how our bodies actually work, it gives us much more realistic movements and helps an animator maintain correct arcs of animation.

Inverse Kinematics

Inverse Kinematics attempts to simplify the process by reverse-engineering our movements. After the character has been fully rigged (using the same example as above), all you would need to key is select the wrist and drag it towards your characters head and the movements of the shoulder, elbow and wrist would be automated. This is particularly beneficial if the character has to hold on to something because however the rest of the character is animated, the location of the hand (and the resulting shoulder/elbow/wrist rotations) will be locked onto the desired position, a door knob for example.

IK/FK Switch

Obviously it is not therefore desirable to only have access to one of these solutions, so we would usually set up a custom attribute to switch (or blend) the character between IK and FK control.

Constraints

Constraints allow you to constrain certain attributes of an object to another object. The most common example is a parent constraint. This treats the object that has been constrained as if it is a child, so if the main object is moved or rotated (not scaled) then the constrained object will follow.

The advantage of using a constraint is that it allows you specify which attributes are being constrained (only the Translate Y for example) and also allows a constraint weight value between zero and one which controls how much influence the constraint has. This means that we can set up a constraint that will move an object by half the value that another object is moved.

Various types of constraint exist, but the three other commonly used are:

Orient – constrains the rotation of an object to another.

Point – constrains the absolute translation of an object to 'pin' it to another.

Aim – also controls the rotation, but instead of mirroring the target's rotation is locks a specific axis to always point towards it (for example setting up eyes).

There are many other types of constraint, two more worth noting are *scale constraints* which lock the scale down (this is often combined with the parent constraint) and *pole vector constraints* which act similar to an aim constraint pinning the pole vector information of an IK handle to a specific object.

Direct Connections

By means of the Connection Editor, a single attribute can be directly link to another using what's known as a dependency.

Human IK

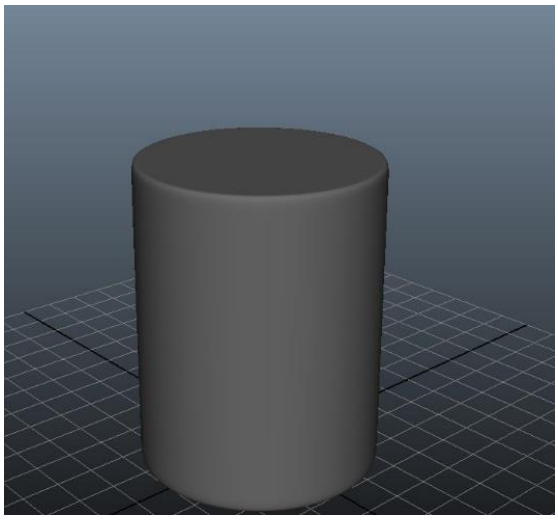
Something that we will not explore in this book, but may be worth making a note of is Maya's recent Human IK solution. This is Maya's automatic skeleton builder, which can very quickly build a powerful rig for your model ready for animation.

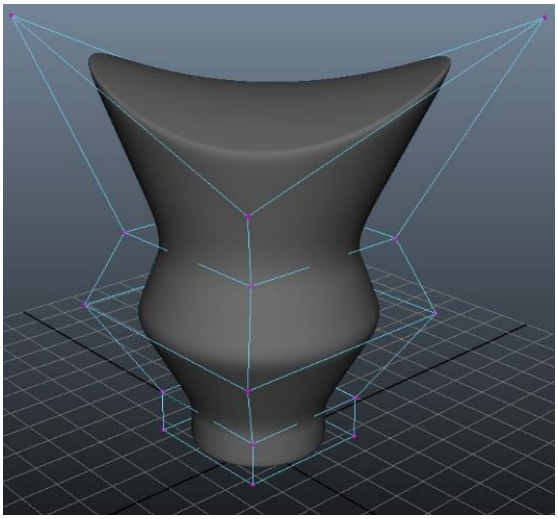
Deformers

Deformers (as their name would imply) deform the appearance of the object they surround or are attributed to. The most commonly used deformers are:

Lattices

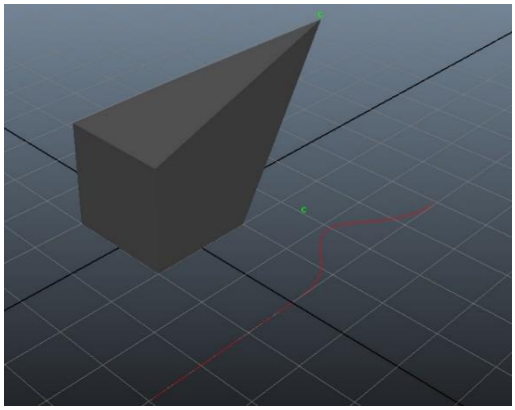
A lattice is a frame that encompasses your object and allows you to deform it by manipulating the lattice points. Lattices are often used as modelling aids, to pull parts of model around in a smooth way however because they have a live connection with the objects that are inside them they can also be used to build complicated powerful rigging systems.





Clusters

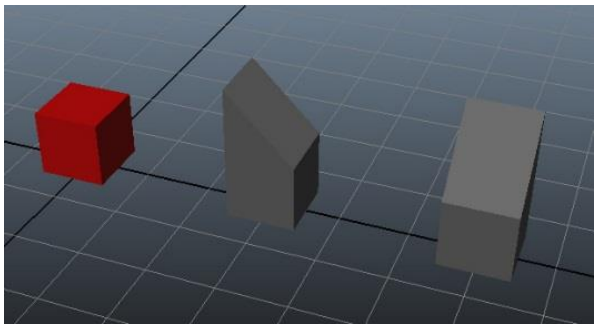
Clusters are used to group points together. These can be vertices on a mesh, cvs on a curve of even lattice points, and this makes them very versatile. Their primary role is within the rigging realm as you will see later. The cluster will be assigned a handle and you can use this to manipulate them accordingly.



Blend shapes

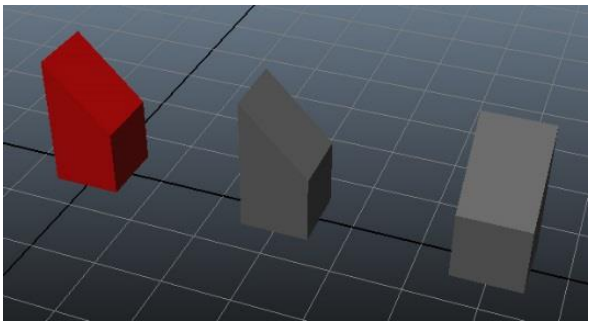
Blend Shapes are used to deform an object, relative to its original state. You are able to control the level of influence a blend shape has, as well as combine multiple blend shapes at the same time giving you complete control of deformations. Blend shape applications are vast but most often used within facial rigging, where the smile, mouth open and other shapes are separated in order to be combined during the animation process. This being so they can also

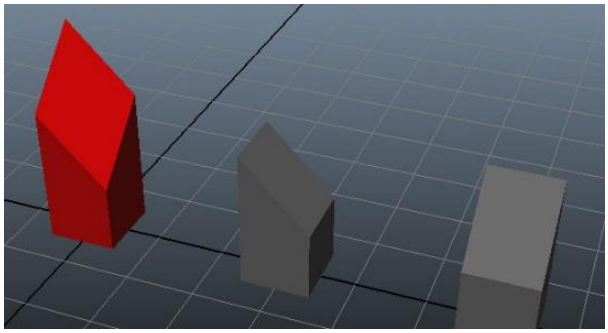
be used in less obvious ways, such as manipulating the shape of a lattice object to give an object a very specific uniformed deformation.



This is our original object (red) and two blendshapes, created by duplicating and modifying the original.

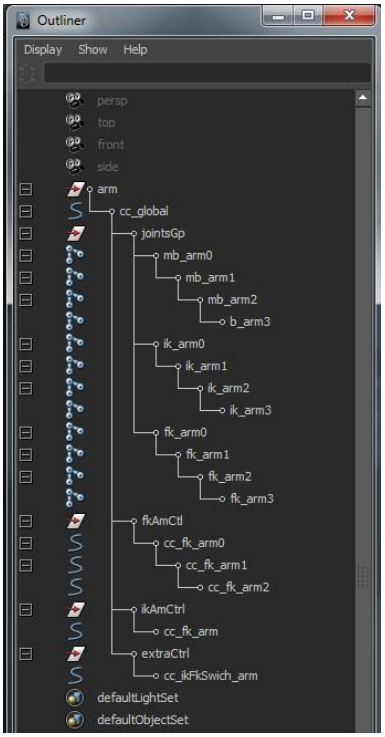
We can use our blendshape node to add the changes of the first blendshape to our original object.





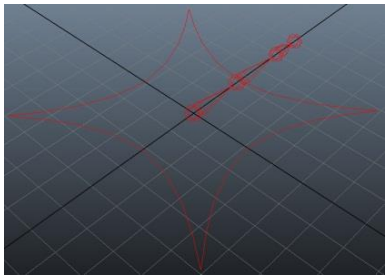
Now if we add the second blendshape information to our object you will see that both shapes are combined because it adds the relative change per vertex to your shape. This allows to huge flexibility and is especially valuable for facial animation.

IK/FK Switch

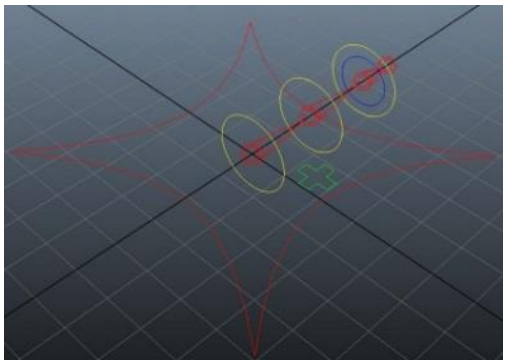


The ability to switch between IK and FK control can be very valuable in animation. One of the most common applications for this are character arms. For the most part arms want to be animated in FK because it allows a more natural movement however some actions, especially those where interaction with another object is concerned, are much more effective under IK Control. This means that we want to be able to use both on our character, and switch between the two during our animation.

For our IK/FK switch system, we will have three identical joint chains. One will have FK control (with its own control curves), one will have IK control (with its own control curve) and our third joint chain will be the driver for our character which will have a control curve to blend between the FK and IK.

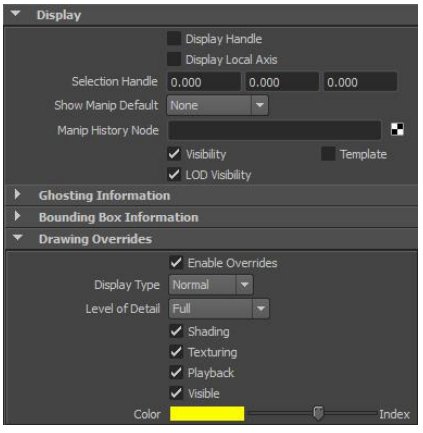


Create a joint chain. Duplicate this joint chain twice, adding the prefix '*ik*' to the joints in one duplicate, and the prefix of '*fk*' to the other. This helps us identify our joints. We want to create a few control curves ready for the system. First create a large global control around the system. Here I have started with a large circle and then in component mode (*F8*) adjusted the shape.

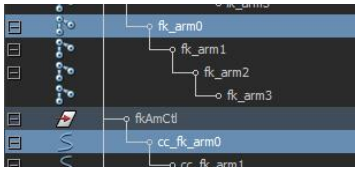


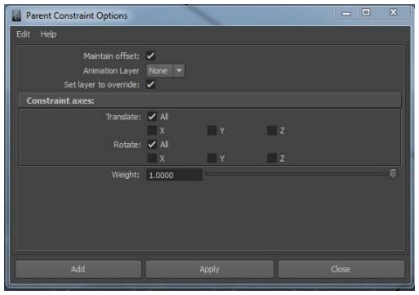
Create a control curve for each joint and label them with the prefix 'cc_fk_' these are going to be our FK controls. Parent each one below the one before.

Create a control now for the IK control (at the location of the wrist) and an extra control next to the joint chain that we will use to switch from IK to FK. I have used the display override attribute to colour the different control groups.

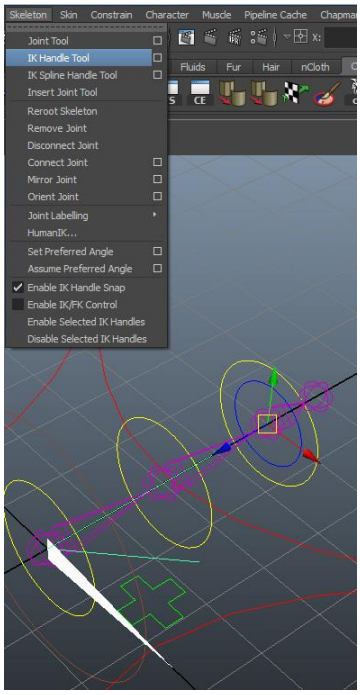


Your outliner should fit the following hierarchy:

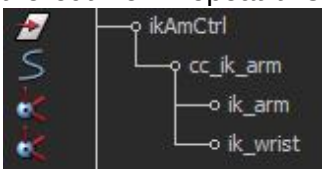




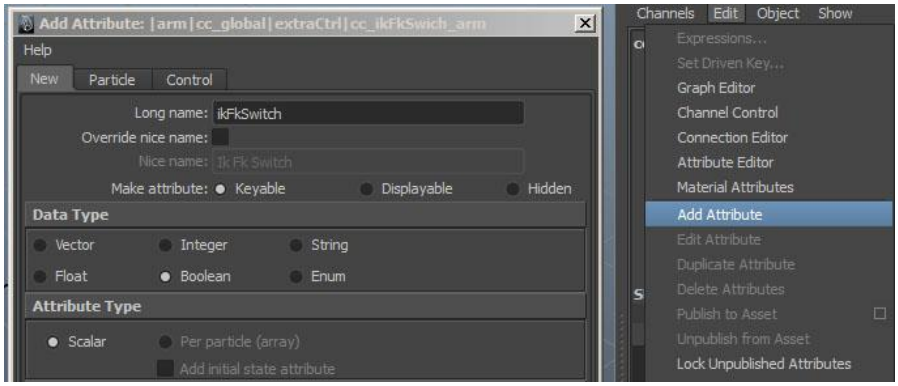
We are going to create parent constraints to connect the FK controls to the FK joints. Select the control, and then *ctrl-select* the corresponding joint in the outliner. In the animation menu set, click the *Constrain > Parent Constraint* tool to link the two objects, and then repeat this process with the other controls and joints in the chain.

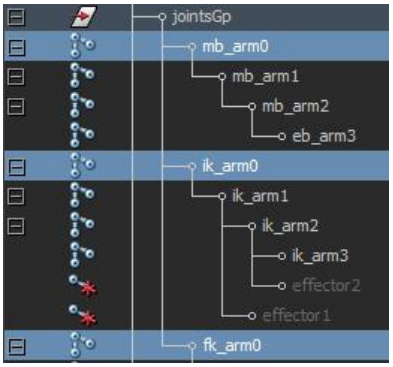


For the IK system, select the *ikHandle tool* control and *ctrl-select* the first and third joint of the chain in the outliner. Repeat this process with the last two joints. Parent the *ikHandles* to the *cc_ik_arm* control curve.

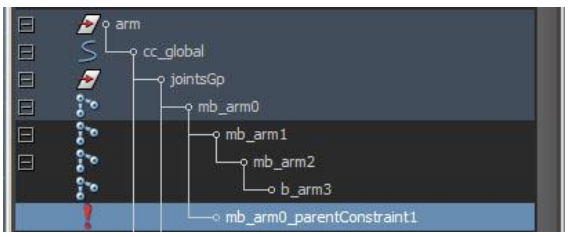


Now that our control systems are set up (you can test them if you wish) we need to feed both into our control chain and build the system to swap between each one. Select the *cc_ikFkSwitch_arm* control and add a new attribute called *ikFkSwitch* with the minimum value of 0 and a maximum value of 1. We can also lock and hide all of the other attributes on the control.

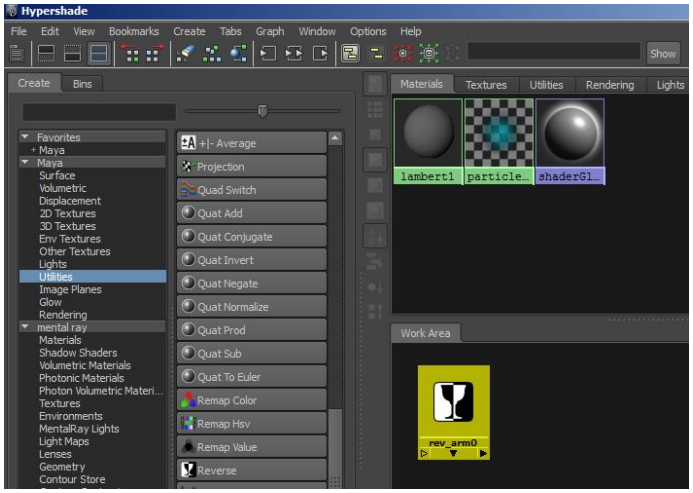




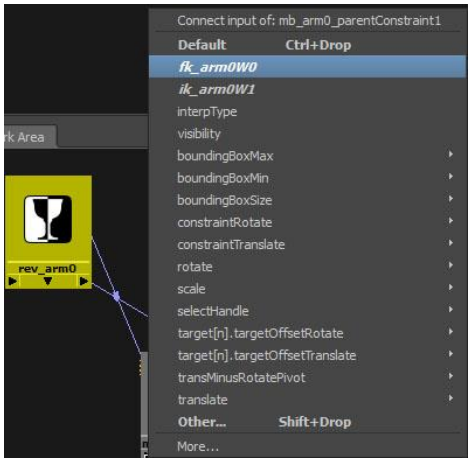
To feed the control chain, select the first *FK*, *IK* and control joint in that order. Now create a parent constraint.



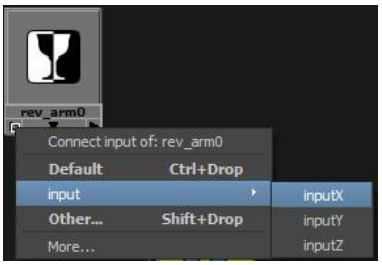
Open up your Hypershade. Although primarily for working with shaders, this is one of the most useful areas for us to build rigging systems by directly connecting specific attributes. On the left of the Hypershade is a menu, under *Maya > Utilities* create a reverse node and name it *rev_arm0*.

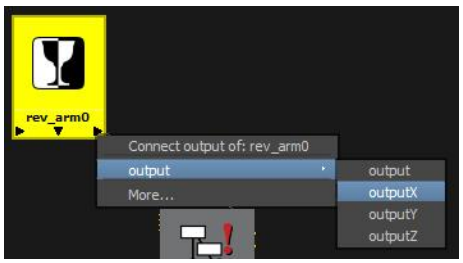


Middle click drag the *mb_arm0_parentConstraint* into the Hypershade.

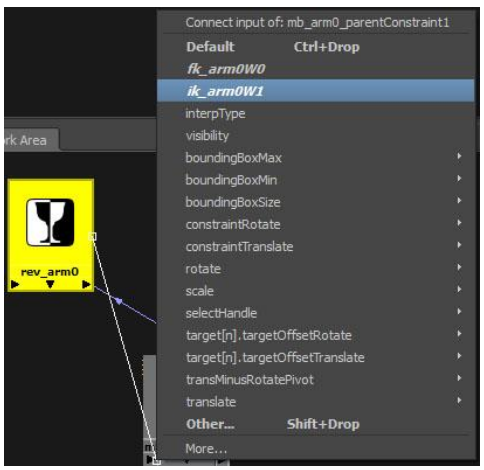


Right click hold and select *fk_arm0W0*, then *left click hold and select the inputX attribute on the reverse node to connect the two*.

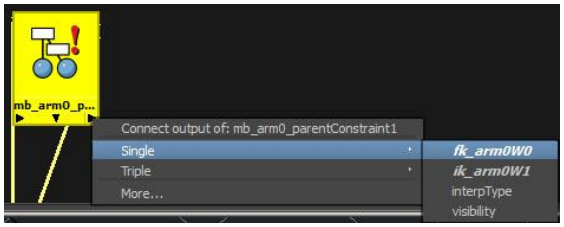




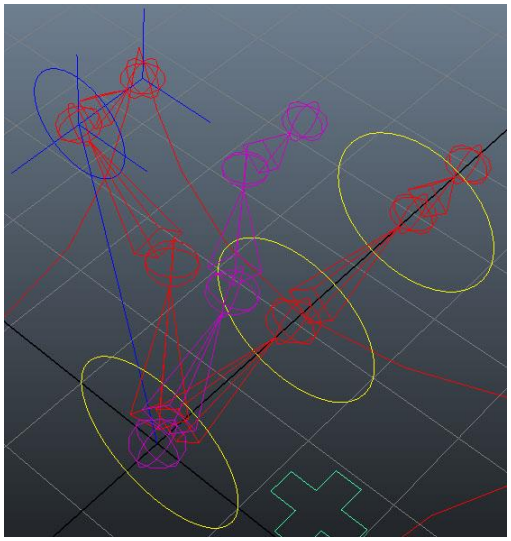
Then we want to take the outputX attribute from the reverse node and connect it to the *ik_arm0W1* attribute in the parent Constraint Node.



Middle click drag the *cc_ikFkSwitch_arm* curve into the Hypershade and connect the *ikFkSwitch attribute to the fk_arm0W0 attribute*.



Repeat this process for the other joints in our chain.



Now if we change the *ikFkSwitch attribute* on our control, it will swap the parent constraint influence between our FK and IK joints.

If we grab the IK handle and move it, the joint chain will follow, but if we change the value of the *cc_ikFkSwitch* control you will see the control chain blend between the two.

This concludes our tutorial however one important thing to note is that we can use the technique to switch between any two joint chains, regardless of how they are rigged. This means that we could swap between FK and a Dynamic link chain for example (this gives us control over the dynamics if we run into problems) or even an IK chain as above with an IK Spline type chain that we will be building soon.

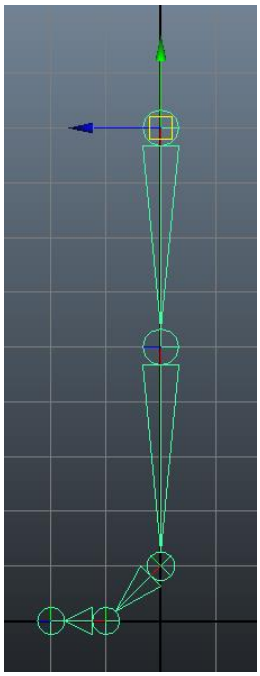
An important thing to consider is that with this specific method, a change in the scale of the joints will not be reflected onto the driving joint chain. If we want the scale information to be passed on (and also be switchable) then we need to repeat the parent constraint processes with scale constraints, including building new reverse nodes and linking everything together with the ikFkSwitch control curve (we can use the same switch attribute).

Chapter 5: Rigging

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.

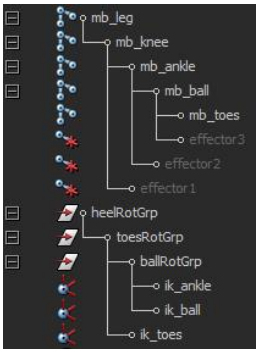


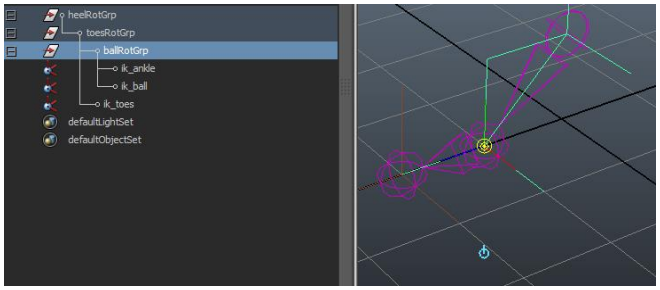
First we need to build our leg joints. Using the *Joint Tool*, create and name your leg joints in the side menu. We need a *leg*, *knee*, *ankle*, *ball* and *toes* joint.

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:



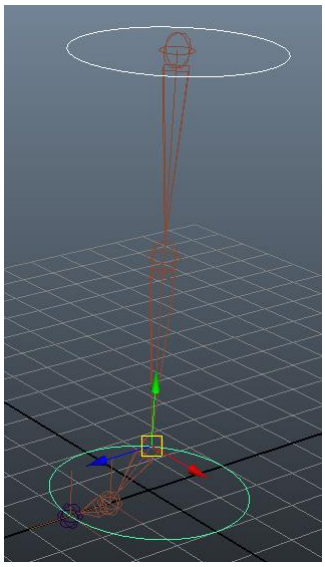


Now that our groups are built we need to pin the pivots to the right points. Using *insert* to toggle or by holding the '*d*' key combined with the '*v*' key, point snap the pivots of the groups to joints as follows:

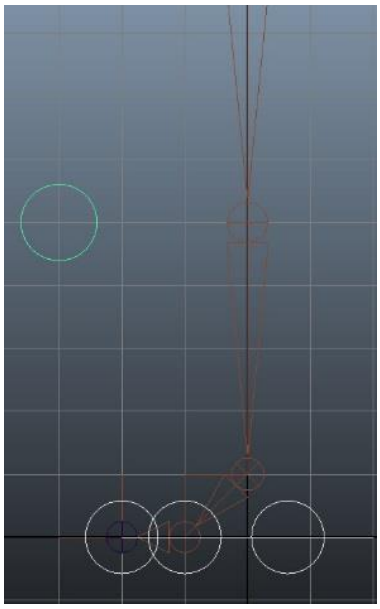
ballRotGrp - *ball* Joint

toesRotGrp - *toes* Joint

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



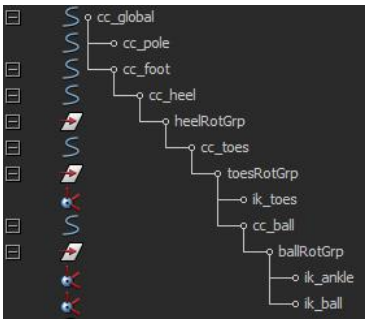
Using NURBS circles we are going to build a control system. First we need a global control to be built at the root of the leg up joint, then an overall foot control where the pivot is set to the ankle joint.



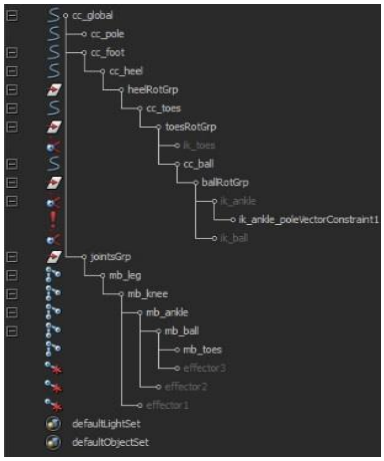
In the side view create three more circles and create them on the toes, ball and heel location.

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*).

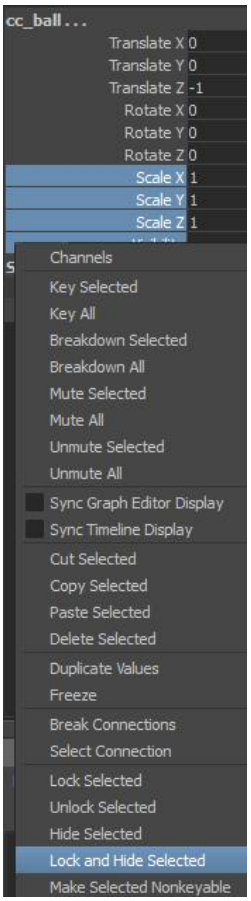
Re-arrange and group the controls to match the following Hierarchy: Remember you can use *middle click and drag* to move objects around the outliner.



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



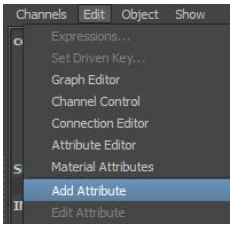
Our final step for this configuration is to clean our scene up a little and limit the control of our controls. First group the joints into a *jointGrp* and parent that group under the *cc_global*. We can also hide all of our IK Handles while we are in the outliner.

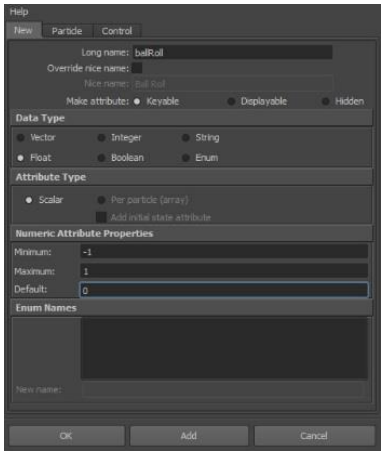


We can lock the scale and visibility attributes in all of our controls apart from the *cc_global*.

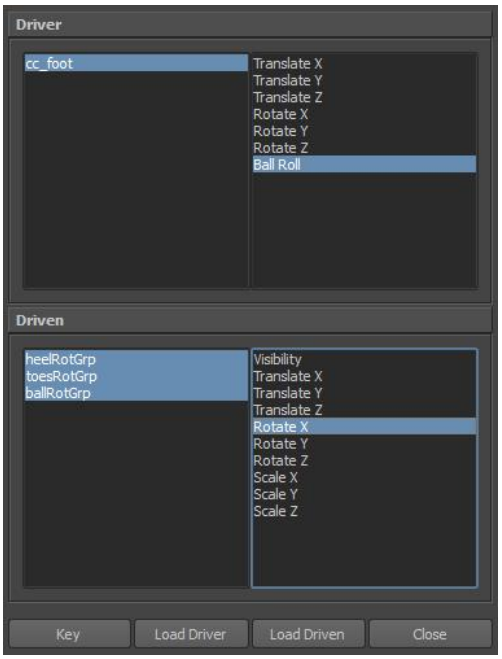
In the same way we can lock and hide the *translation* attributes in your *channel box* for the *cc_heel*, *cc_ball* and *cc_toes* controls and the *rotation* attributes on our *cc_pole* control. For the *cc_heel*, *cc_ball* and *cc_toes* controls you could also lock and hide the Rotate Y and Z to make 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.

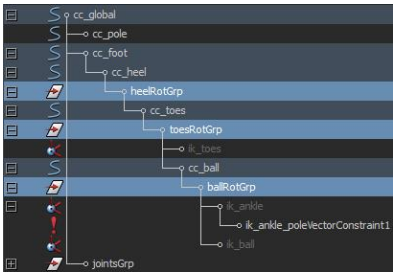


Select the *ballRoll* attribute in the *channel box* and open the *Set Driven Key* tool. We will use this to tool to set up the connection between the *rotate X* attributes for our reverse foot lock groups and the *ballRoll*.

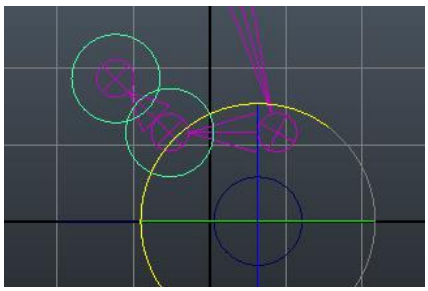
With the Set Driven Key tool open, click on *Load Driver* and make sure the *Ball Roll* attribute is selected. In the outliner select the *heelRotGrp, toesRotGrp* and *ballRotGrp* and then press *Load Driven* to set them as your driven objects. Select them again in the *Set Driven Key* tool and then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

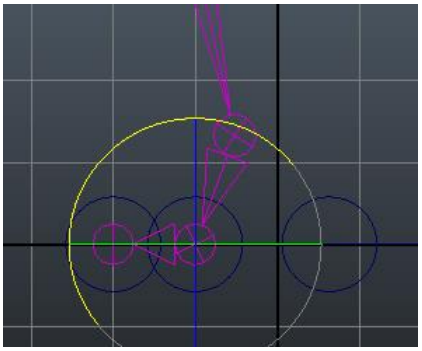


This will be the start position for our ball roll, which is when the heel is being planted. Rotate the *heelRotGrp* back to *-45°* in the *X-axis* and then press *Key* in the Set Driven Key toolbox.



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.

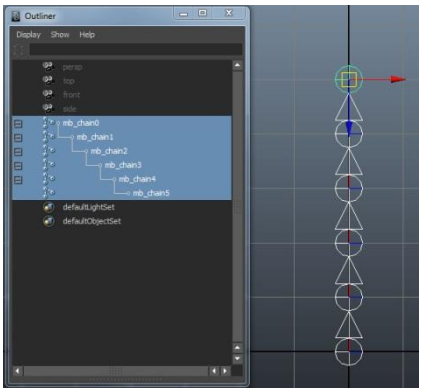
Our final step is to set the *Ball Roll* value to *one* and key the *toesRotGrp* with a *Rotate X* of *45°* and the *ballRotGrp* back to *zero*.



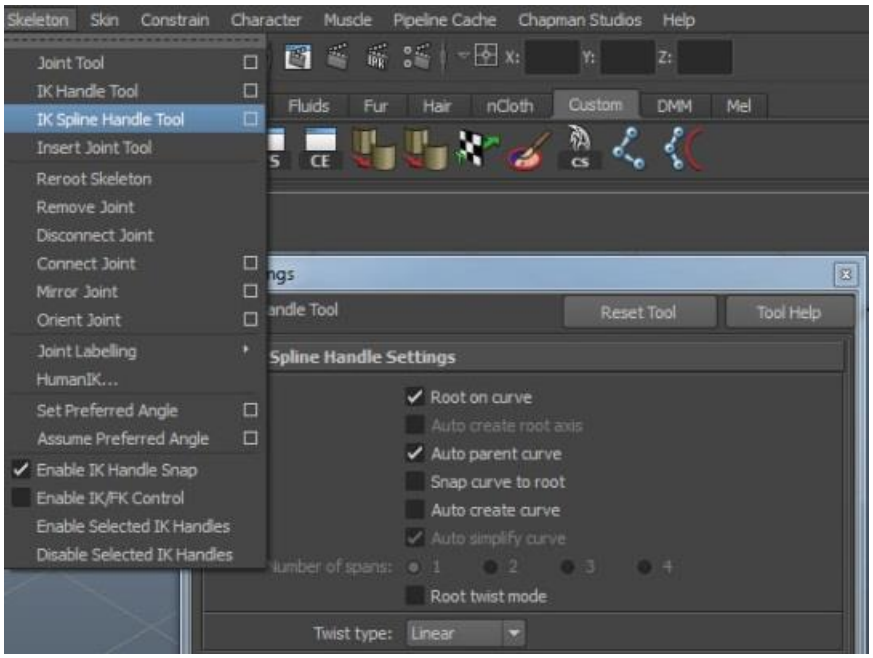
This concludes our reverse foot lock tutorial. You can test the foot roll by selecting the *Ball Roll* attribute and *MMB click dragging* the mouse in the view pane. As you scroll from *-1* to *1* you should see a nice rolling foot simulation that we can use to speed up animation, whilst also having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

IK Spline with Squash / Stretch

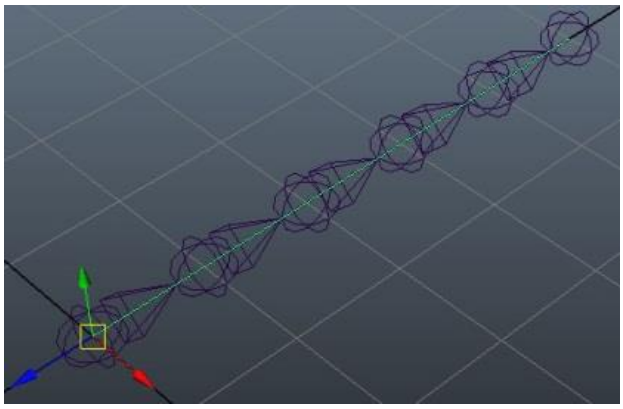
We are going to build a joint chain controlled by a different type of IK system than before. Previously we set the location of the base joint and built a control curve for just the tip. With a spline IK system we will also have control of the overall curve of the joint chain.

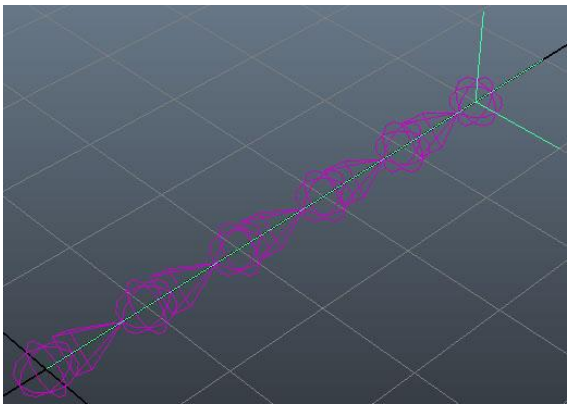


In the top view, start by creating your joints. I have held the '*X*' key down to snap the joints to the grid. Rename the joints to '*mb_chain*' or similar. Now using the *cv curve tool* and holding '*v*' to snap to each joint, create a *CV Curve*.

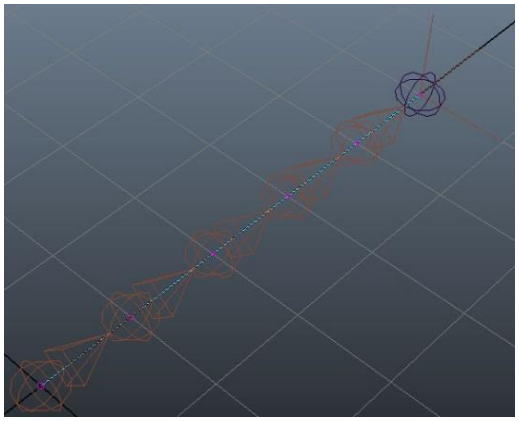


In the IK Spline Handle Tool settings, ensure that *Auto create curve* is deselected and with the tool still selected select and ctrl select in the outliner (or *shift-select* in the viewport) the first and last joints in the chain followed by the curve we just built.

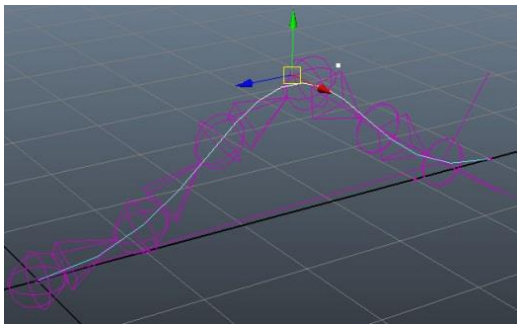




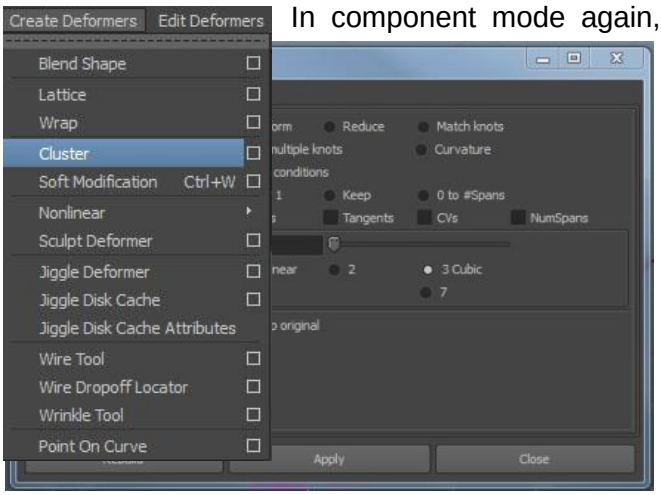
An *IK Handle* should be created (and be selected) and an effector should be created under the second to last joint.



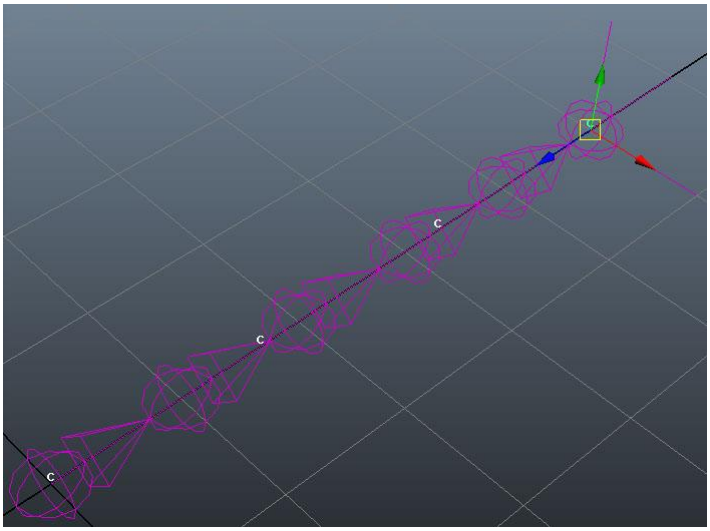
If we select the curve and go into component mode (*F8*) we can see that we have six control vertices defining the shape of our spline (you could also take this opportunity to move the vertices and see how the joint chain is affected by the shape of the curve). We want to reduce this number to four.

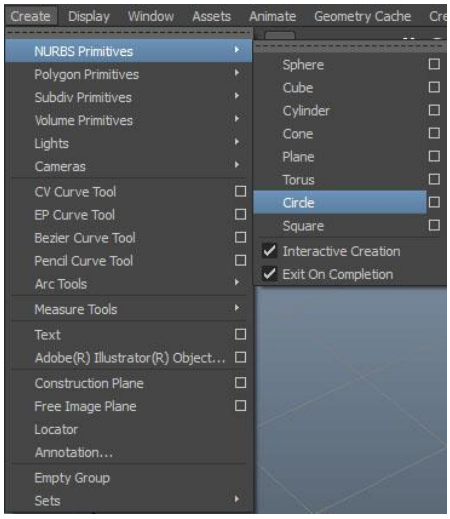


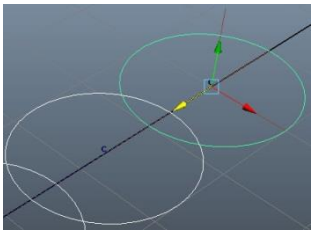
Select the curve and switching to the *Surfaces* menu set click the *Edit Curves > rebuild curve* □ The number of resulting vertices will be the number of spans plus the degree of the curve, which in this case is three. So for four control points, we need the number of spans to be four minus three which is one. Rebuild the curve.



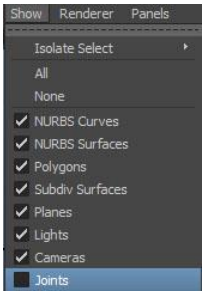
In component mode again, select the first vertex and click *Create Deformers > Cluster* □ Ensure that '*relative*' is selected and click apply to build a cluster to control the vertex. You will need to make sure you are in the *Animation* menu set. Repeat this process to create a cluster on each of our curve control vertices. Rename the clusters to '*cl_chain*' or similar.



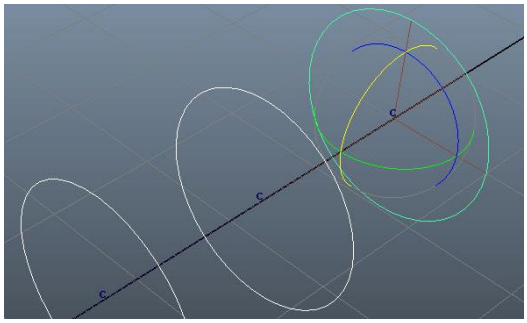




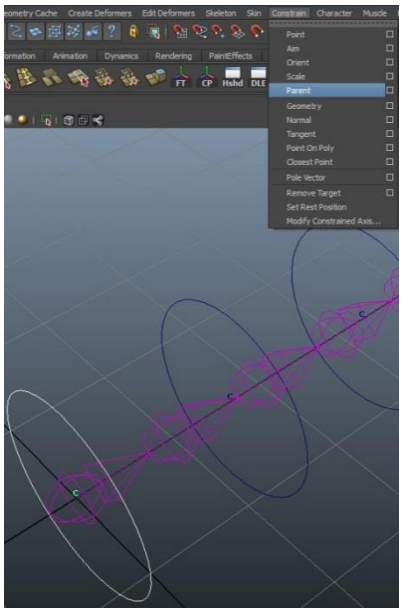
Create a circle (*Create > NURBS Primitives > Circle*) on each of the cluster points. You can use '*V*' to snap, and you may find it easier if you temporarily hide the joints visibility in the viewport Show menu.



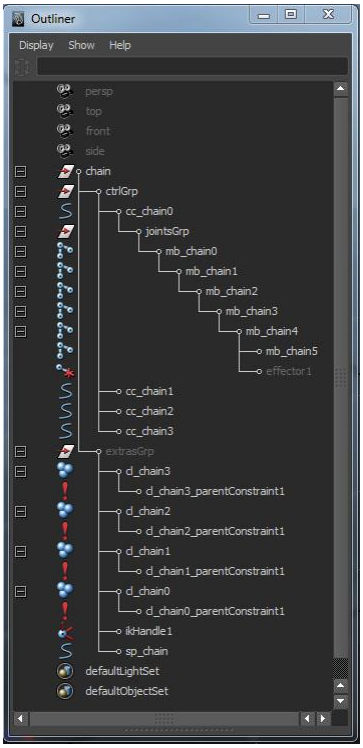
Rotate the control curves by 90° in the X axis and then delete history (*Edit > Delete by Type > History*) and freeze transformations (*Modify > Freeze Transformations*). We can bring our joint visibility back now, and rename our controls to *cc_chain* or similar too.

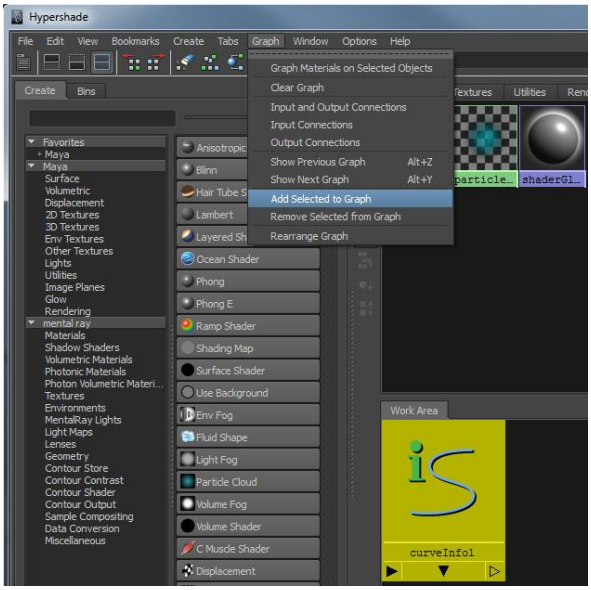


Select the first control curve followed by the first cluster and create a parent constraint (*Constrain > Parent*) this will drive the cluster from the control curve. Repeat this procedure on the other three curves and clusters. Remember that we can repeat the previous Maya action by pressing '*g*'.



Now we have a completed Spline IK joint chain. Clean up the scene by organising your *Hypershade* as seen below. Essentially we want to separate the control objects from the rig objects so that it is neat and we can hide the stuff that we don't need to see. When we bind an object (such as a tail, or the spine of a character) we will also hide the *jointsGrp*.



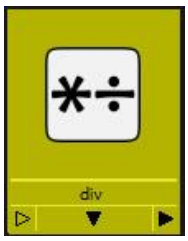


Although we can use this rig as it is, what we really want is the ability to squash and stretch the chain in between the first and last control. To do this we are going to use some MEL commands using the Command Line in the bottom left of our Maya window. First we need to create a node to tell us the length of our curve. Type in "*arclen –ch 1 sp_chain*" (where sp_chain is the name of your curve) and press enter.

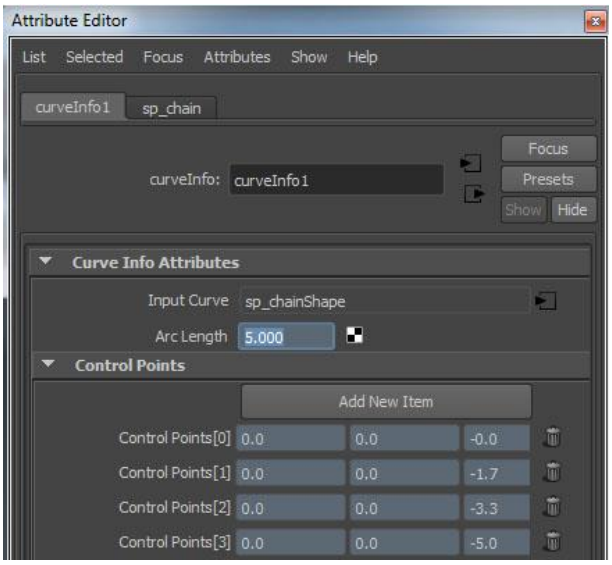
We should get the message "// Result: curveInfo1". We need to remember the name of our node because it is not visible in the outliner or viewport. Type in "*select curveInfo1*" and press enter to select our node.

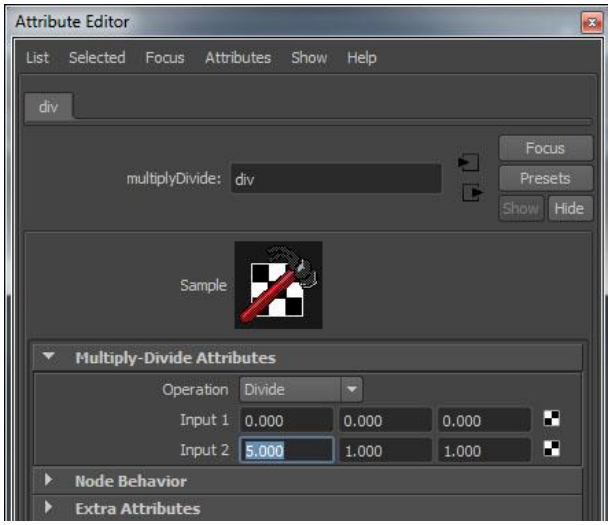
Open the Hypershade window up and click on *Graph > Add Selected to Graph*. The curve info node will appear in the work area. In the left hand menu, create a new Multiply Divide node (*Maya > Utilities > Multiply Divide*) and rename it '*div*'.

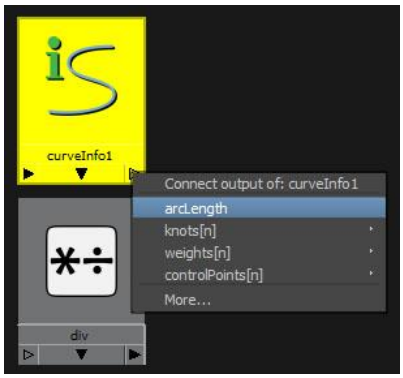
We want to use the multiply divide node to calculate the difference between the curve length at rest and the curve length as it is. To get the value at rest (which it is at the moment)



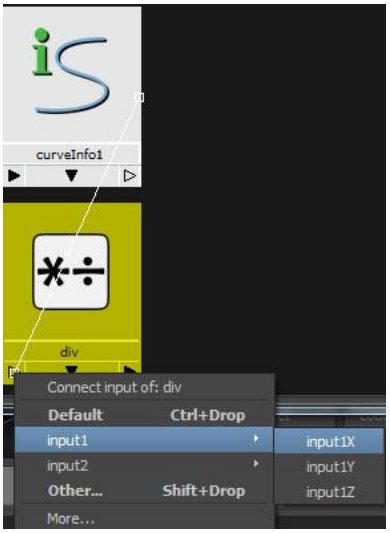
select the curveInfo1 node from the Hypershade and copy the value (in my case this is five). Select the multiply divide node and paste this value into *the Input 2 X* field. Also make sure that the operation is set to '*Divide*'.



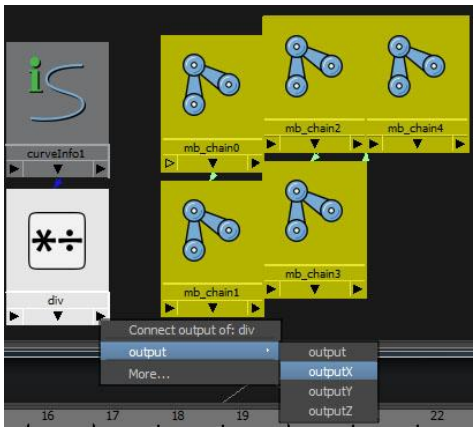


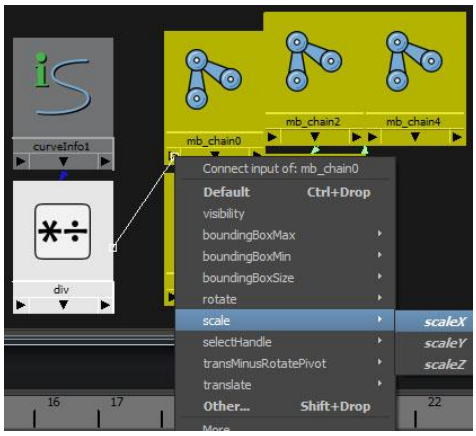


Right click and hold on the output of the curveInfo1 Node and select '*arcLength*' and then left click and hold on the inputs of the div node and select '*input1 > input1X'.* The div node will now be able to give us the value of how much our join chain is squashed or stretched by.

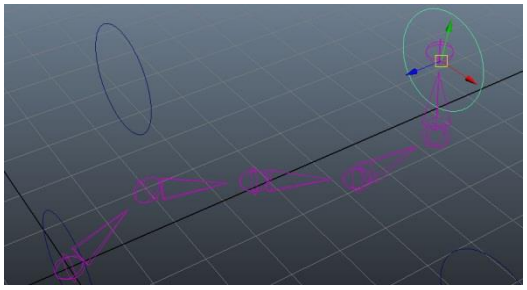


Select all the joints in our chain apart from the last one and *add selected to graph* as we did before. Now like we connected the curveInfo and div nodes, we want to connect *the div > output > outputX* to the *mb_chain0 > scale > scaleX* and then connect the div outputX value to each of the other joint's scaleX attributes.





Now when we move our control curves to change the shape of our joint chain, the joints will automatically stretch to the correct length.



Rigging a Flag:

The following tutorial will help familiarize yourself with how you can combine different deformers to create a simple flag rig. This will also go some way to help you get used to the accurate organization of your assets.

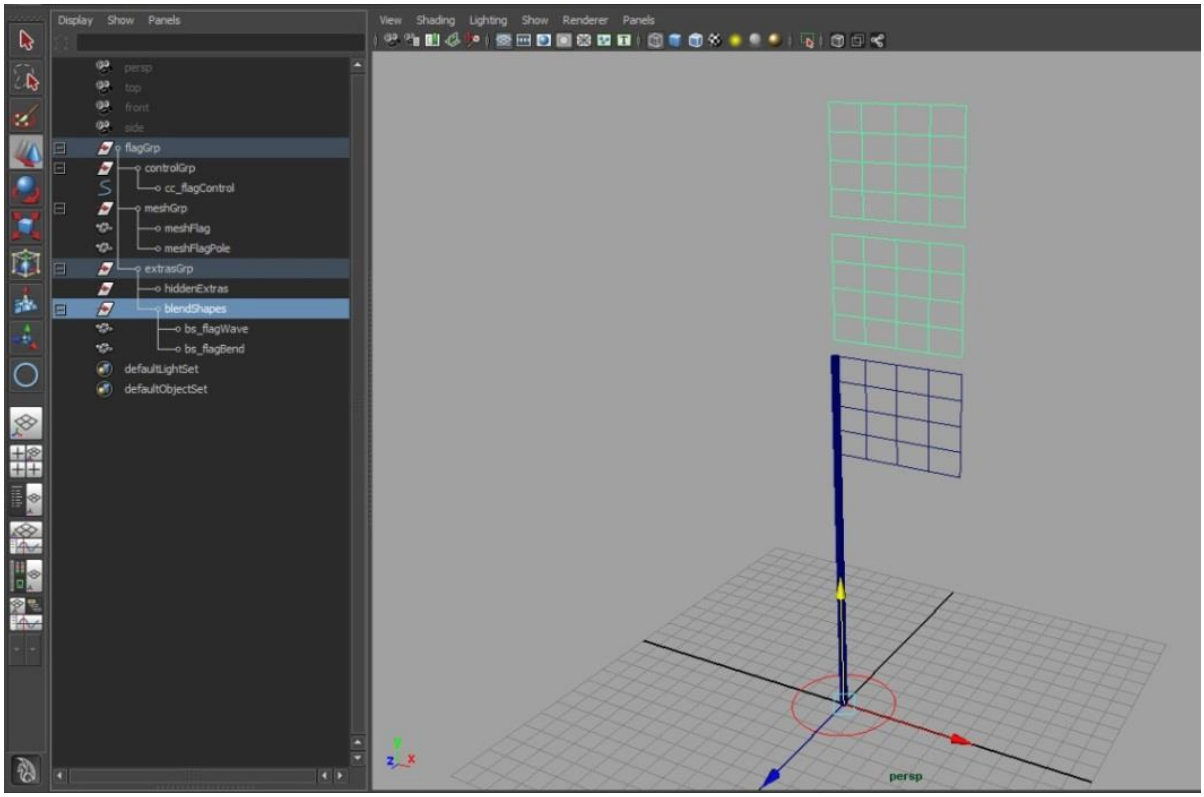
This may be viewed as a somewhat complicated approach to rigging a flag, and in some ways you would be correct but the aim here is to provide you with knowledge of techniques that have the ability to be applied to almost anything.

We created a very simple flag and pole model for you to start with.

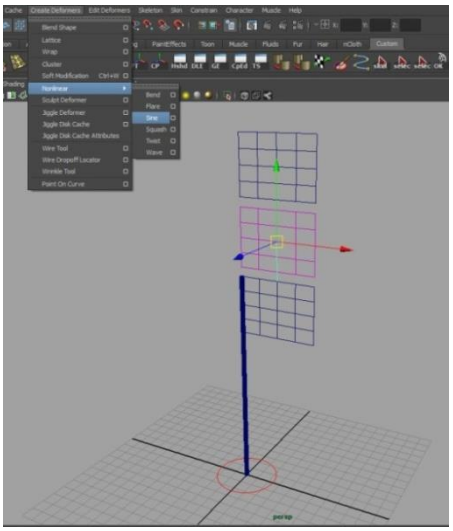
Because we want a few different types of flag deformations to be mutually exclusive, we are going to create blend shapes to drive the flag.

Step 1:

Duplicate the flag a couple times and move the duplicates out into their own space. We also want to group these into a *blendShapes group*, and name them appropriately (*Ctrl-d* to duplicate, *Ctrl-g* to group and *MMB click and drag* to move *the blendShapesGrp* to the *extrasGrp*).

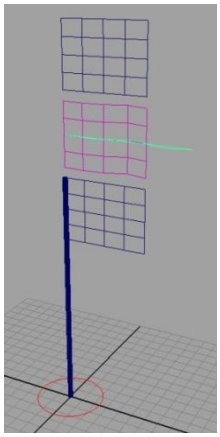


Step 2:



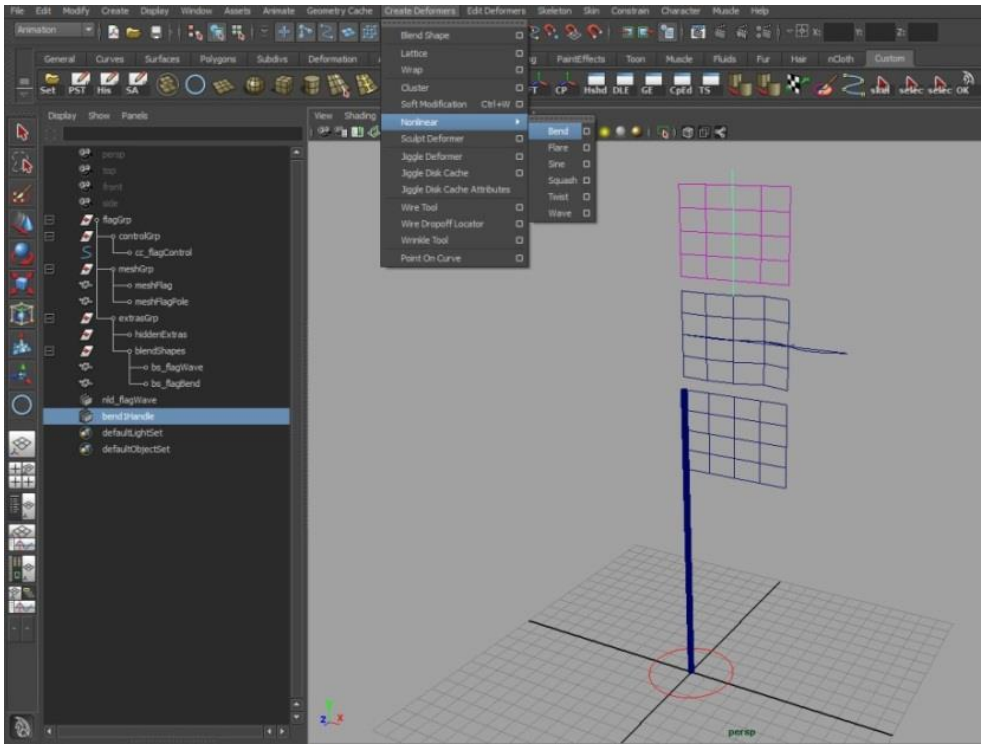
First we will add the wave effect for the flag. With the *Wave* shape selected, apply a Sine deformer to it.

Step 3:



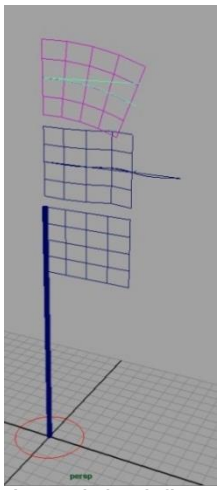
To give the correct kind of effect, first we need to move and rotate the deformer into place. Set the Rotate *Y* and *Z* to *90*, and then move the deformer out so that the centre of it is nearing the flag tip. We also need to extend the length of the sine curve, so increase the *high bound* to *2*. We want less movement at the pole end, so increase the *drop-off* to *1* and so that we can see the effect we will be getting, increase the *amplitude*.

Step 4:



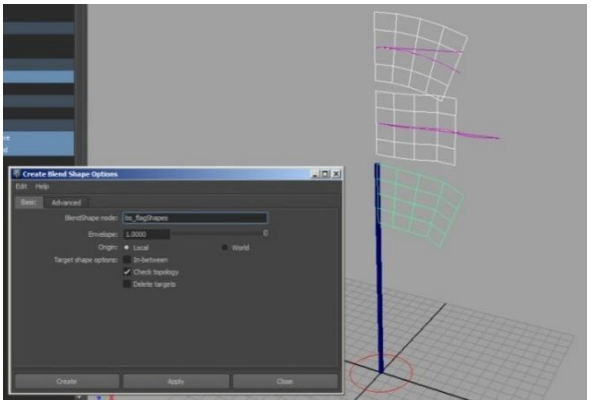
The second blend shape is going to give the flag a drooping effect. With the bend mesh selected, apply a *bend modifier*.

Step 5:



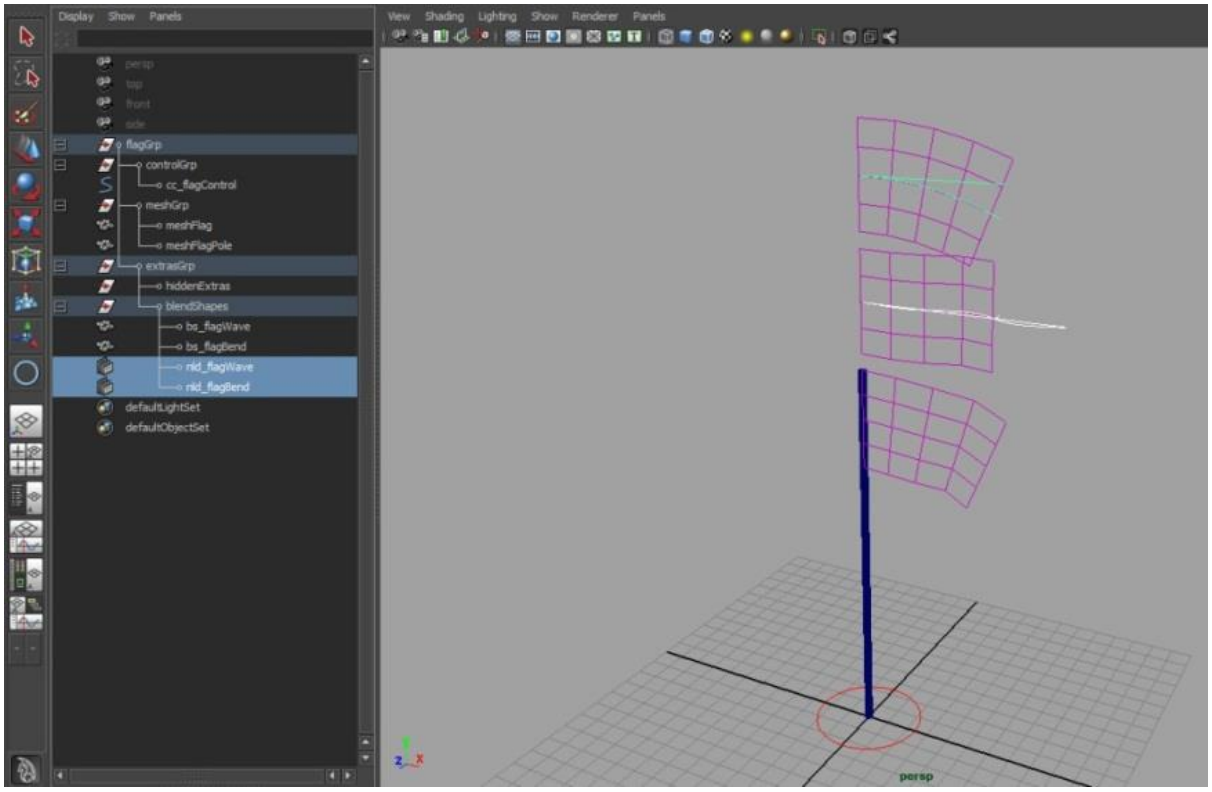
Again, we need to move and rotate the deformer into the correct position, and adjust the high and low bounds. We can also increase the curvature to give an impression of the effect we will be getting.

Step 6:



The next step is to create the blend shape node that will drive the main flag mesh. Select the two blend shapes and then shift select the original flag mesh. In the options box, be sure to give the node a sensible name. When you have created it, set the two blendshape nodes to '1' to activate them (you should see the flag mesh deform).

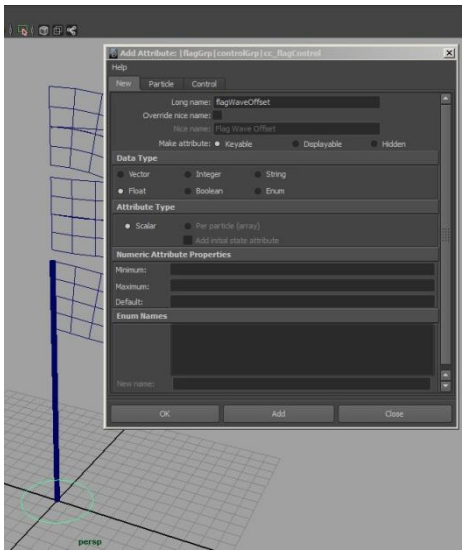
Step 7:



Now we can get around to creating the controls to move the flag, and a little bit of cleaning up. First, Move the two deformers into the blendShapes Group.

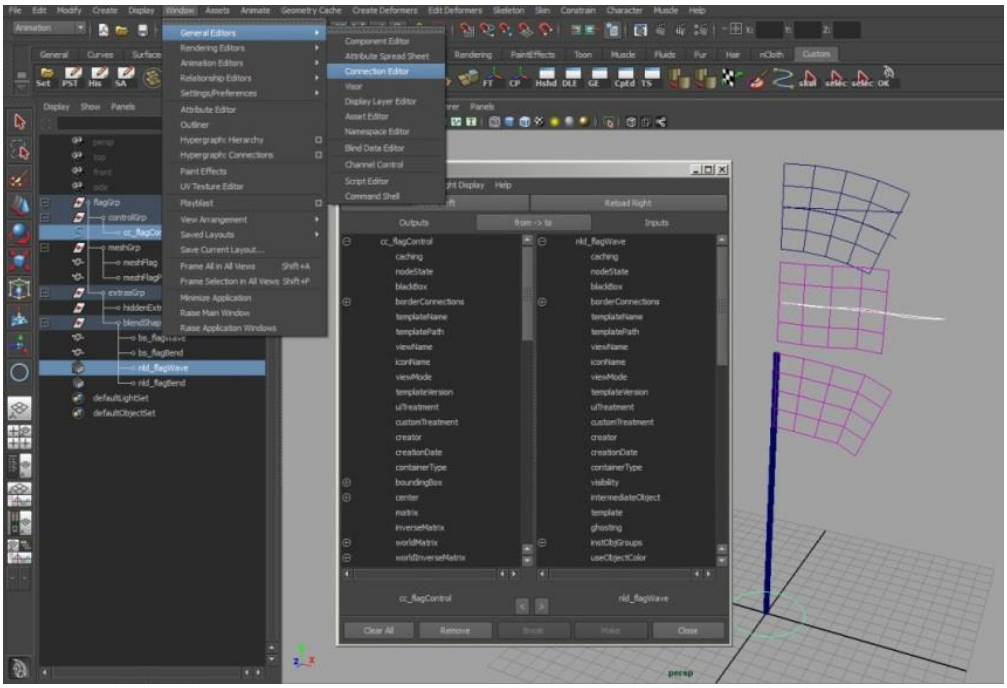
Step 8:

Select the flag control curve, and then in the Channel Box go to the add attribute options. We want to add four main controls, a way to change the value of the *droop* and the *wavelength*, *amplitude* and *offset* of the *wave deformer*. I usually add a *Boolean* (*on/off*) attribute to begin with, and then lock it (right click on the attribute, lock selected).



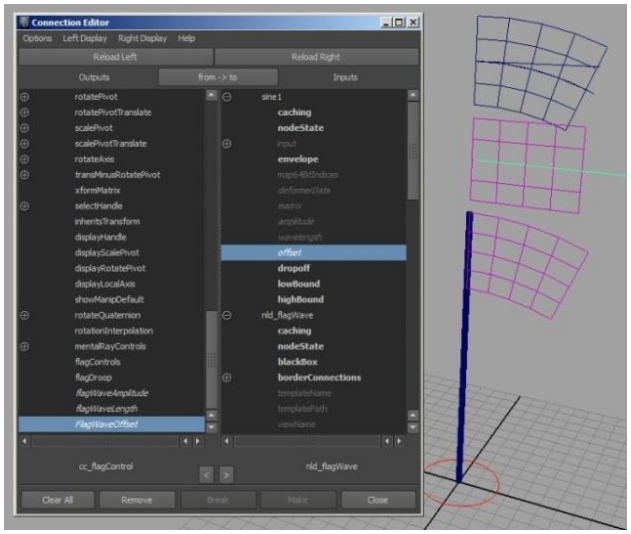
Step 9:

We will start with the wave deformer, select the control curve and *shift-select* the *wave deformer*, then bring up the *connection editor*.

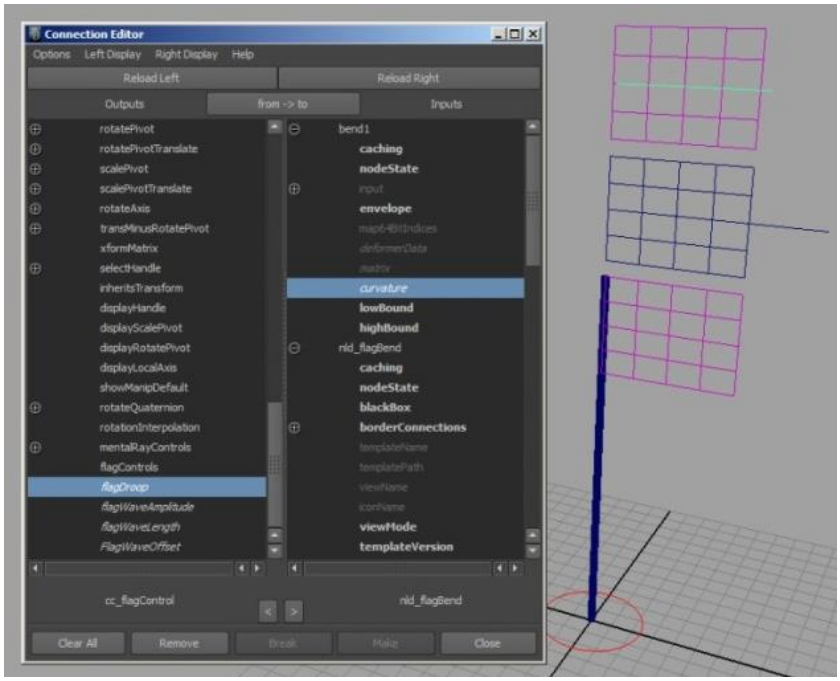


Step 10:

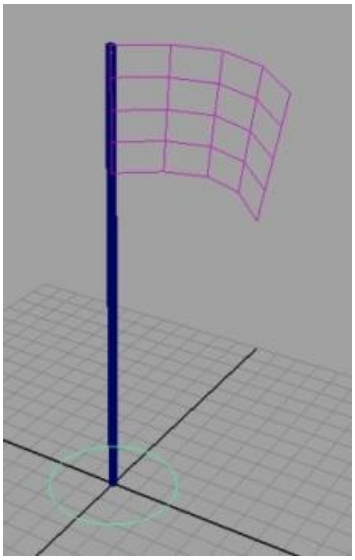
We want to connect our custom attributes to the corresponding deformer attributes. First we need to bring the deformer's shape node into the right side of the connection editor, select the deformer, and then the 'sine1' input node, and then click *'Reload Right*'. You will then be able to see the *amplitude/wavelength/offset attributes* and can link them to the global control.



Step 11:



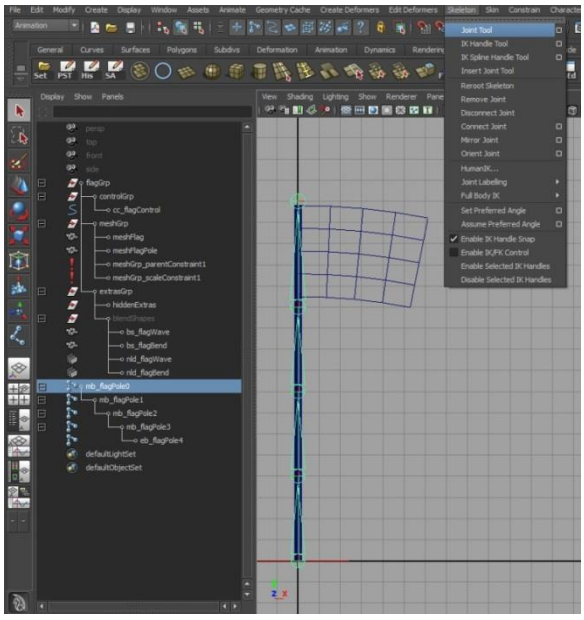
Now we can do the same with our bend deformer and the droop control.



Step 12:

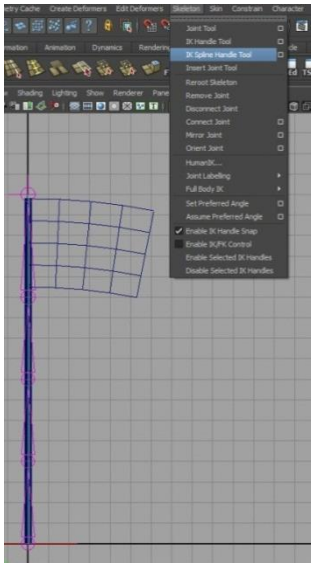
When everything is connected, test your new controls out. We can also go ahead and hide the blendShapes group.

Step 13:

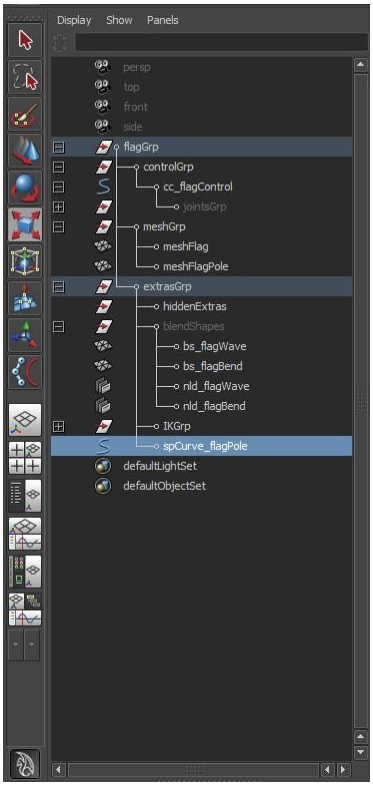


We would like to also bend the flag in any direction, and maintain the rest of the setup we have created. We are going to create a Spline IK driven by a control at the tip of the flag. In the front view, using the Joint tool, create about 5 joints along the spine of the flag. Name the joints appropriately.

Step 14:



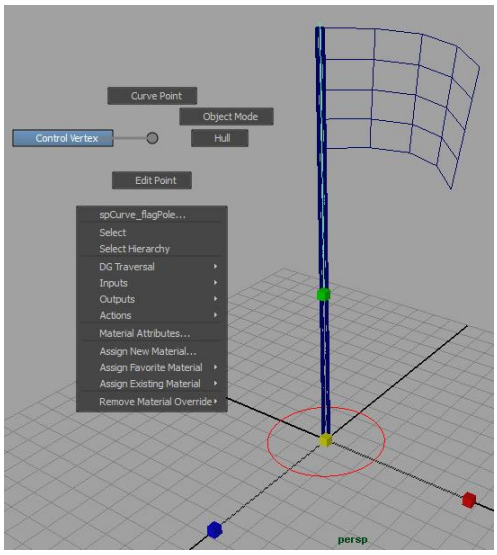
Using the IK Spline Tool, select the first and last joints in the sequence.



Step 15:

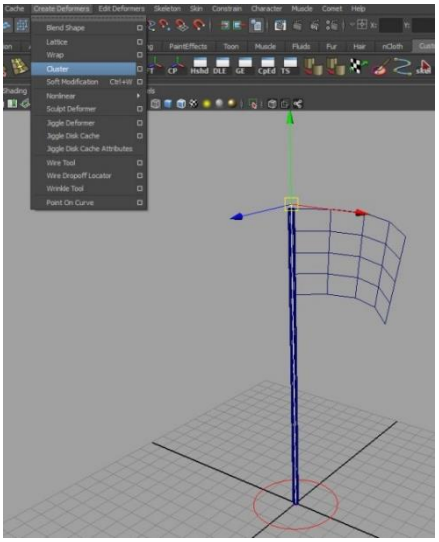
You can also group and hide the IK node and the joints and organize them into the hierarchy. The spline curve should also be moved into the extras group.

Step 16:

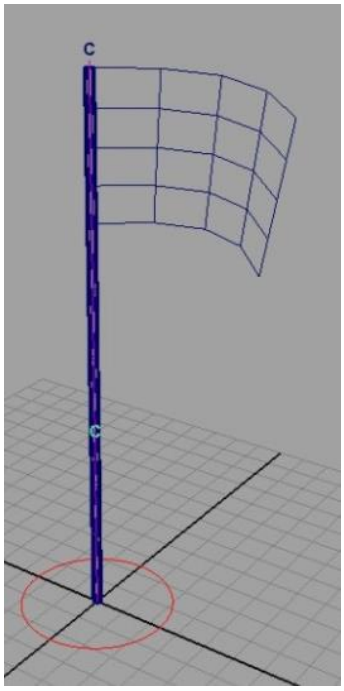


Select the spline curve, and *marquee-select* the top control vertex, with this selected we're going to assign a cluster to this point.

Step 17:

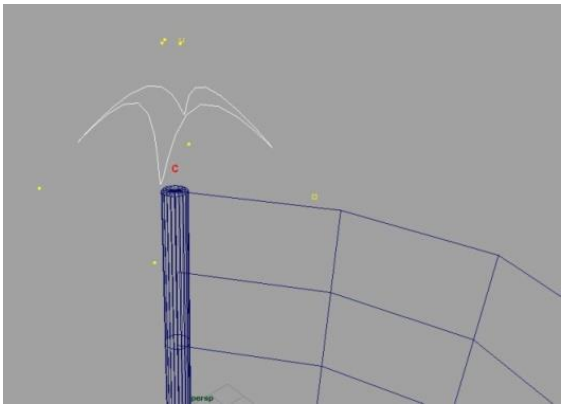


We want to do the same again with the bottom 3 vertex points of the curve.

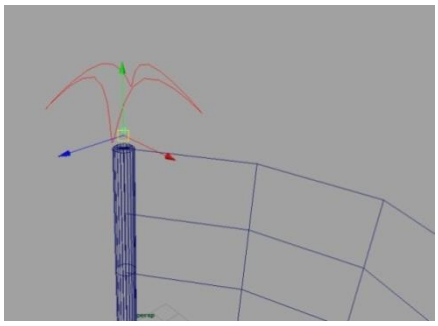


Step 18:

Group the clusters and parent them under the global control. We need to create a curve to control the bend of the flag, so using a NURBS circle, create a shape and move it to the top of the flag. Ideally, the centre pivot of the curve wants to be the same as the cluster.



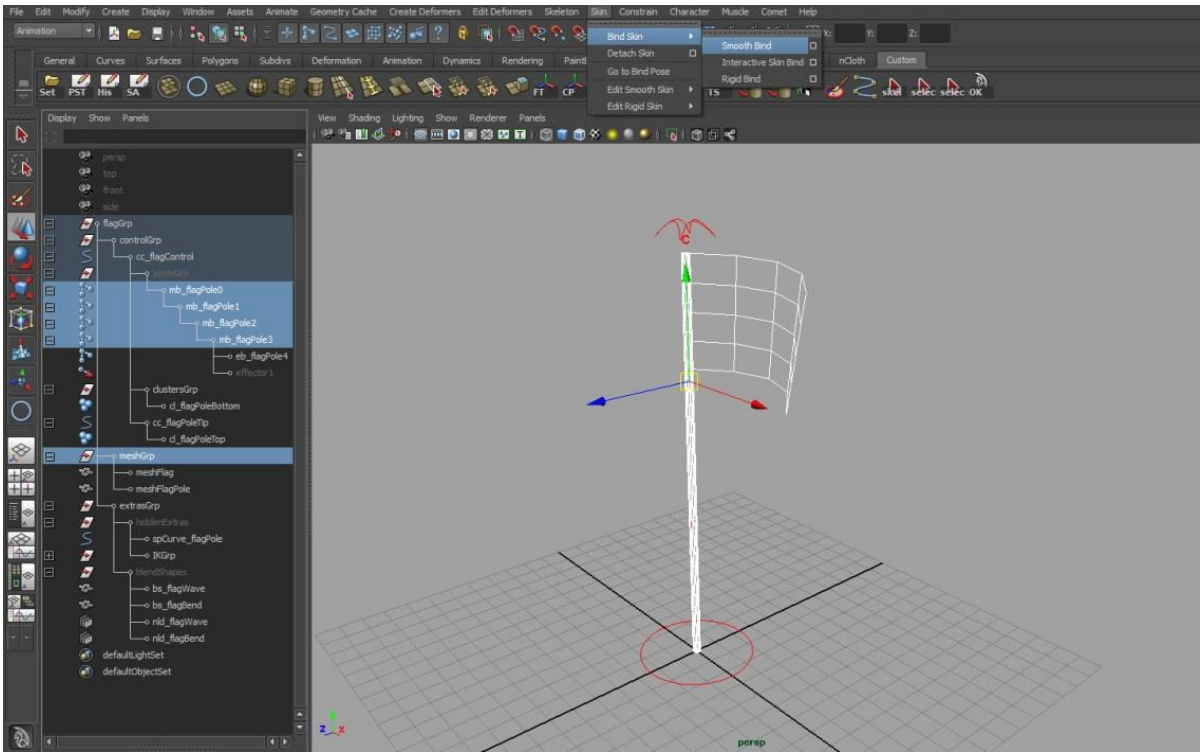
Step 19:



You can then parent the top cluster to the new curve (which in turn wants to be parented under the *global control*).

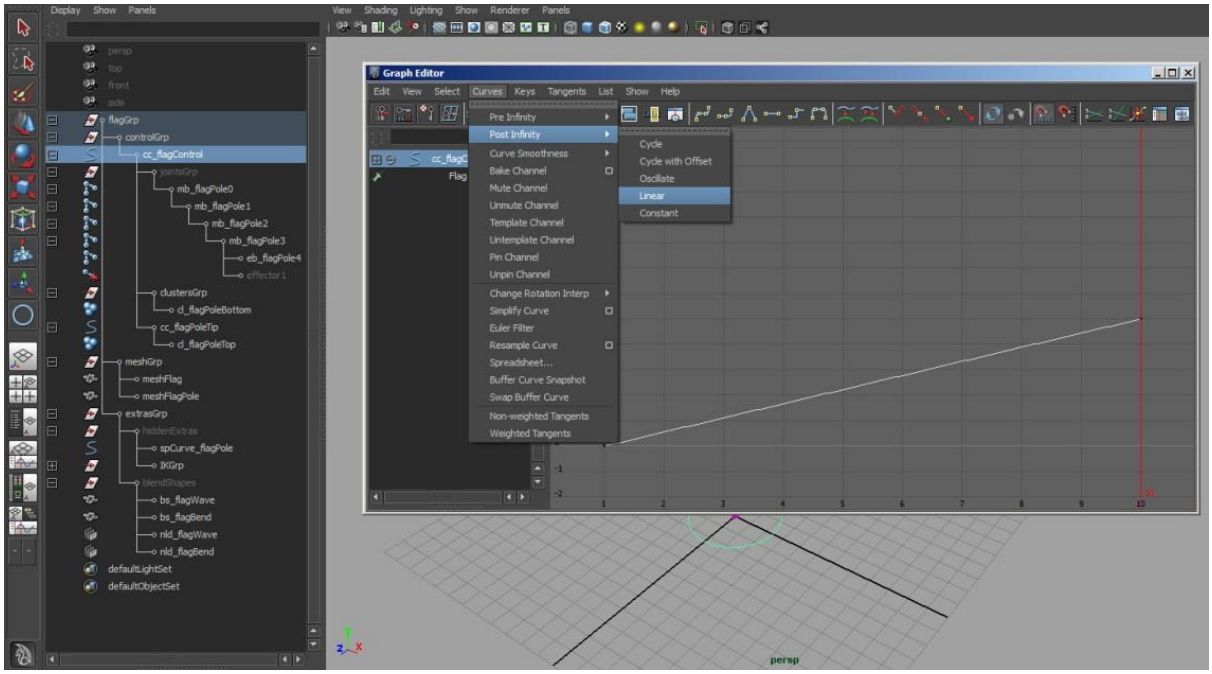
Step 20:

In order for the flag mesh to be driven by the spline IK, we are going to bind it to the joints. Select the meshGrp and then Ctrl-select the first 4 joints and create a smooth bind.



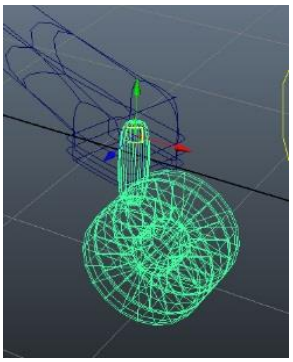
Step 21:

That concludes the rig of our flag. We can now easily create a continuous flag wave by keying the flag Offset Attribute with a post-infinity curve set to linear, and at the same time, dynamically change the amplitude, wave length and flag droop during your animation.



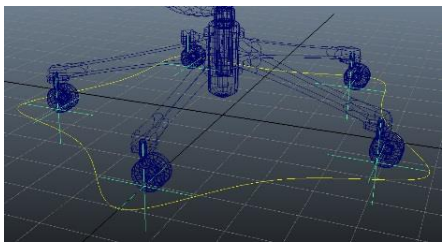
Office Chair Wheels

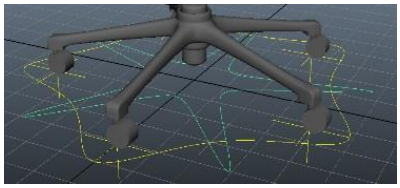
Open up your office chair scene. Here is one I created earlier but all you really need is a basic chair with wheels. First let's create a chair global control and group it within a *ctrlGrp*. Now make sure all your mesh components are grouped into a *meshGrp* node. We need to make sure that each of our wheels has its own group and also that the pivot of the group is in the correct location for the wheel to spin, especially if they are off centre like in our case.



We are going to make the rotate Y of our wheel groups point towards the rest points on our global control and build a sub-control to define the location of the chair itself.

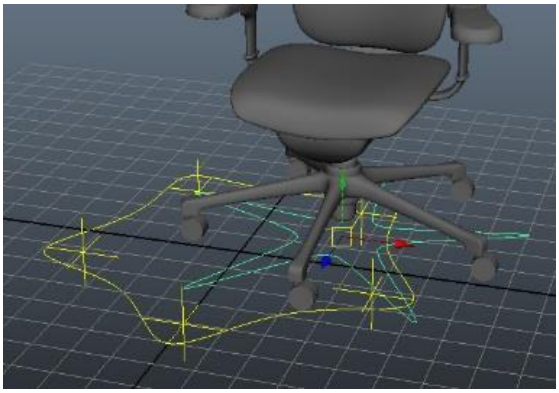
Build a locator *Create > Locator* on each of your wheel pivots and group them as *locatorGrp* or similar. Group this under your global control. The Translate Y position is not too important, I have placed them on the floor but at the same level as your wheel pivot would also be logical.

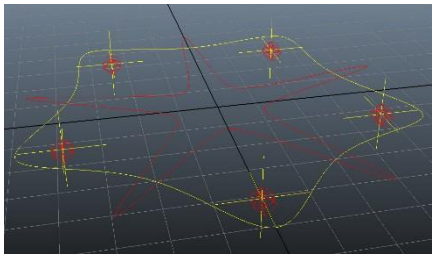




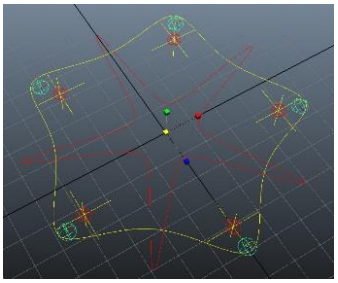
Now let's build our sub-control from a simple NURBS circle. Modify the shape if you desire, but make sure it is easily discernable from the main global group. Delete history and group this under your global.

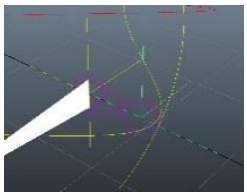
Now constrain your meshGrp to your controls. You want to use a *Parent Constraint* between the meshGrp and the sub-control but make sure your *Scale Constraint* is linked to the overall global control. So if we move the global control everything follows and if we move the sub control the chair follows but the global control and the locators stay where they are. It is also worth locking and hiding the *Scale*, *Visibility* and *Translate Y* attributes from the sub-control at this stage.



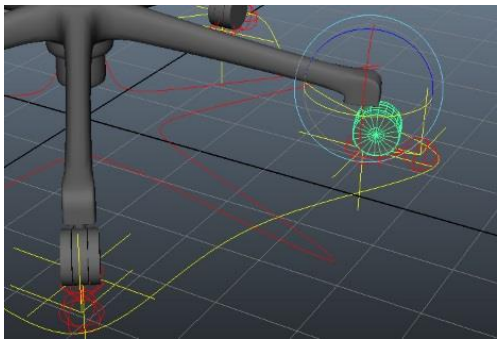


Hide your geometry for a moment (either hide the group, use display layers or use your *View pane > Show* menu) create a joint and with the move tool active use '*V*' to snap the joint to your first locator, then *duplicate* this joint so that there is one joint on each of your locators. *Duplicate* all of your joints and group them, then scale the group up a little. This will spread the new joints out towards the tips of our wheels. Now *freeze transformations* on that group parent the new joints under each respective joint (so you should have a two-joint chain for each of your wheels) and delete the scaled group. Group all of your joints and name the group *jointsGrp* or similar and then parent that under your sub-control.

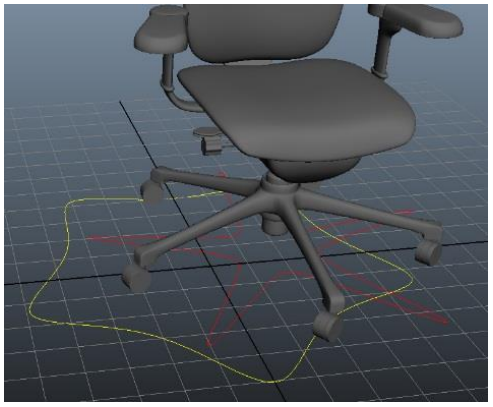




Now use the *ikHandle* tool and ctrl-select the first and last of your joint chains within the outliner. Remember we can use the '*G*' key to repeat the last used operation. Parent the IK Handles into a group call *ikGrp* or similar and parent that to the global control.



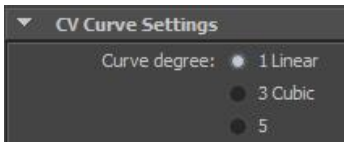
Bring back your geometry. We need to bind the wheel rotation to the joints. First make sure that your wheel groups are rotated in the correct direction (all should be facing outwards). Now select the first joint of each chain and *Ctrl-select* the corresponding wheel group and open the *Parent Constraint* □ options. Make sure that Maintain offset is ticked and apply it to each of your wheels. Now if you move or rotate your sub-control you will see that the wheels point towards the original location. Select your *jointsGrp*, *ikGrp* and *locatorGrp* and hide them to keep your scene clean and tidy.



You now have an office chair with wheels that will point relative to the chairs movement.

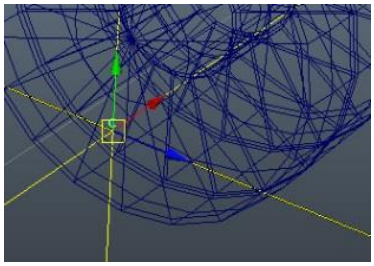
Auto Rotation

As an advanced tutorial, I am also going to take you through how we can add an automatic rotation to a wheel, using the office chair as an example. First we will need to bring back the locatorGrp so make that visible.



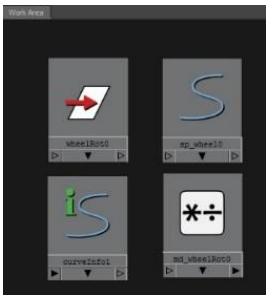
Open your *Create > CV Curve Tool* □ Option and select the *Curve degree* as *1 Linear*. Create a curve with two vertices in your scene and call this *sp_wheel0* or similar. In vertex mode, click *Create Deformers > Cluster* □ options and ensuring that relative is not selected build a cluster on each point. Name the clusters *cl_wheel0_a* and *cl_wheel0_b* or similar.

Now in the Mel box in your command line type the command *arclen –ch 1 sp_wheel0* to create a curve info node on your curve. This will tell us the distance between the two points so that later we can calculate the rotation.



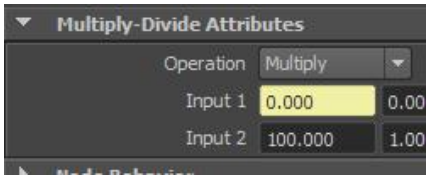
Snap the clusters to the wheel locator and then parent *cluster a* to the global control and *cluster b* to the sub-control.

Open up your *Hypershade* and with the wheel rotate group / geometry and the *sp_wheel0* curve selected click on *Graph > Add Selected to Graph*. Select the *sp_wheel0* curve again, and scroll across the attribute editor until you see the *curveInfo* node. Press *select* and then add this node to the Hypershade also.



Create a *multiplyDivide* node from the *Maya > Utilities* group and rename it *md_wheelRot0* or similar.

Now we need to make our connections. Either within the Hypershade by right-clicking and left-clicking the output and output arrows or by using the connection editor connect the *curveInfo1.arcLength* attribute into the *md_wheelRot0.rotateX* attribute and the *md_wheelRot0.outputX* into the *wheelRot0.RotateZ* attribute (depending on your chair model your correct wheel rotate axis may be different). Now we can control the rate at which the wheel rotates by changing the *Input 2 X* value for our *md_wheelRot0* node. In my particular scene I found a value of 100 works pretty well.



This concludes the rigging of our office chair wheels. You can use a similar technique for car wheels, and especially if you have animated your car along a spline curve you can use the length of the curve and your cars position to work out the correct rotational values for your wheels.

Limiting End User Control

An animator's job is too animate, and therefore the only thing the animator wants to manipulate are the control objects. You can make this easier for them by limiting what the animator can select.

The first thing to do this is to ensure that anything the user should not be changing, such as deformers and joint chains, are in hidden groups, but you cannot 'hide' geometry. You can however stop the geometry from being selectable.

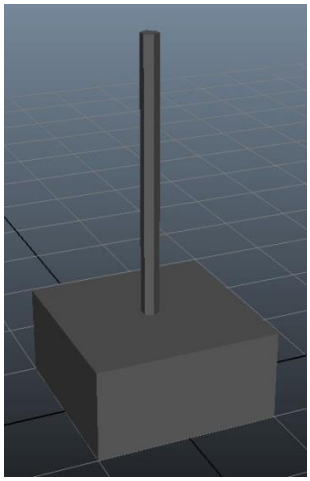
There are two ways to do this, you can put the geometry group into its own layer and set that layer to be referenced (*see display layers on page 43)* or you can also override each piece of geometry's display attribute to be treated as referenced.

In the attribute editor, go to the transform node and tick the Display | Drawing Overrides | Enable overrides box. You can then set the Display Type to reference.

This is a very time consuming process however, so I have written a short script, available on the disk to automate this process with all objects selected.

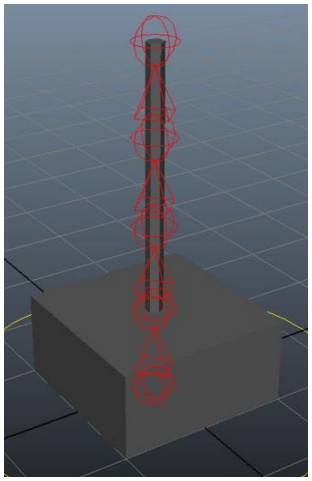
CHAPTER 6: DYNAMICS

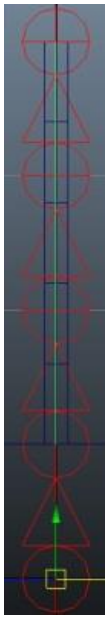
Basic Jiggle / Dynamic Chain



This tutorial will teach you how to setup a simple dynamic chain, which you can also use for basic muscle/fat jiggle deformers.

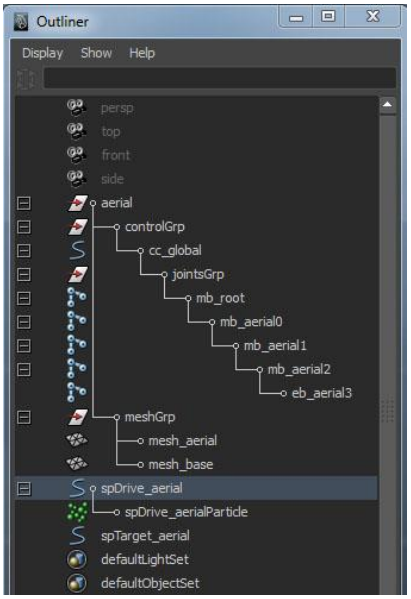
Ideally we would suggest you build your own aerial on your own model using the following guidelines, however we have also provided our own basic model if you run into problems, which you can find on the disk.

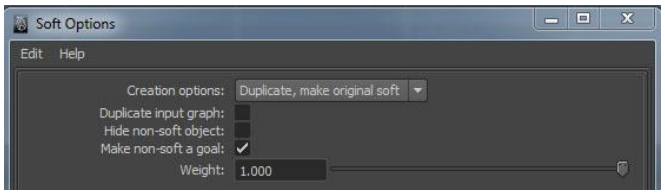




You need an aerial that is skinned to a joint chain where the joint chain has a parent control curve. In our case we also have a 'root' joint.

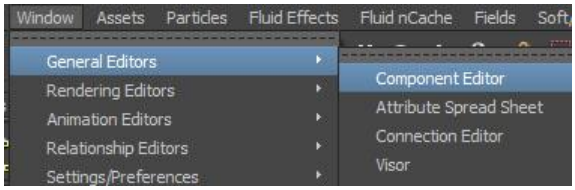
In the options box for the CV Curve Tool, select 1 Linear and, in the side view using point snap, create a curve along the joint chain with one cv per joint. Name this '*spDrive_aerial*'. To allow the curve to act dynamically, make sure you are in the Dynamics menu set and with the curve selected go to the *Soft/Rigid Bodies | Create Soft Body* options. Set the mode to Duplicate, make original soft and enable *Make non-soft a goal* with a weight of 1.





This will create a system whereby our original curve is now being driven by a particle cloud (parented to the original curve) that is trying to align with the duplicated curve, which we will rename to '*spTarget_aerial*'.

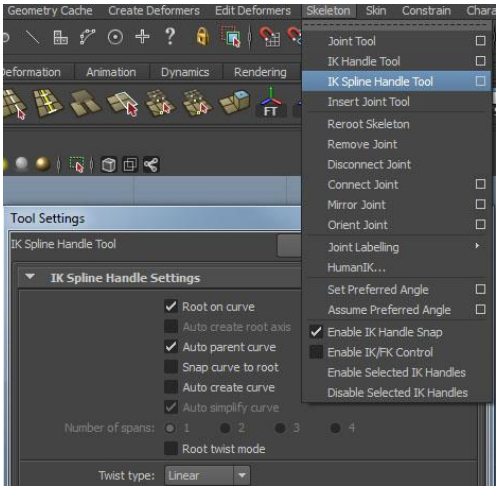
To control the bendiness of our aerial, select our particles from the outliner, press F8 to enter component mode and marque select around the aerial to select each of our particles.

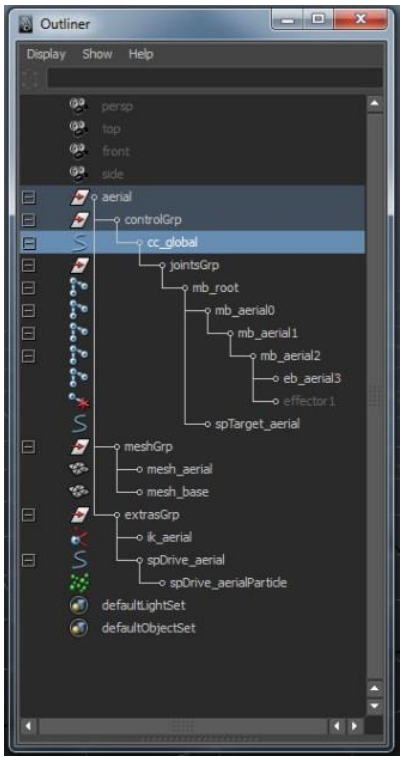


In the component editor; *Window > General Editors > Component Editor* go to the Particles tab and change the goalPP value accordingly.

Experiment with different value to get your desired flex, but I find that these values make a good start point. The other main attribute that governs the flexibility of the dynamic chain is the particles' Goal Smoothness.

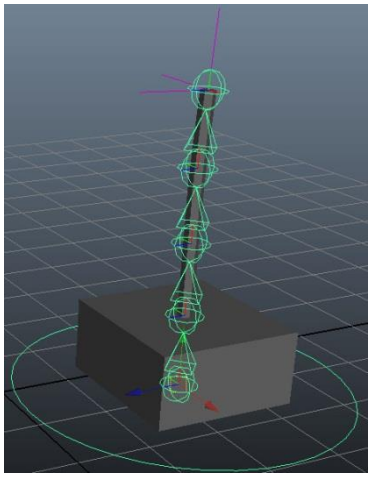
We are going to drive our joints using a spline IK with our original curve as the driver. Move to the animation menu set and go to the IK Spline Tool Options disabling the 'Auto create curve' checkbox. With the tool selected, select the first joint in the chain in the outliner and then ctrl-select the last joint in the chain followed by our spDrive_aerial curve. Rename the new ikHandle to ik_aerial.

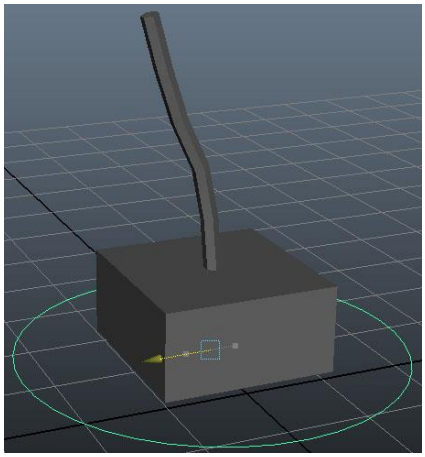




To clean the scene up, group the spDrive_aerial and the ik_aerial and parent the group under the overall aerial group. Name it extrasGrp. Parent the spTarget curve to the group / object above the aerial, which in this case is the mb_root joint. We can also hide the jointsGrp and extrasGrp. Your outliner should resemble something similar to ours:

You must bear in mind that because of the way we have set up our dynamics, they will only be calculated during animation playback, which means that when we now move our global, the aerial will appear to stay in its last location until we hit the play button. In the same vain, it is often worth rendering a few frames before the animation starts to allow the aerial to settle (or for those users who are a little more advanced, we can set the solver's initial state to a more suitable location.





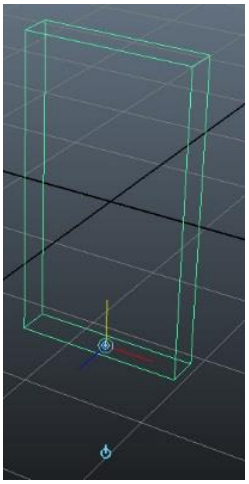
To test our new dynamic, in the Dynamics menu set again, extend our playback range to about 1000 and click on *Solvers* | *Interactive Playback*. Now with the global selected, move the platform around and watch how the aerial on top reacts.

Creating Dominoes

Introduction

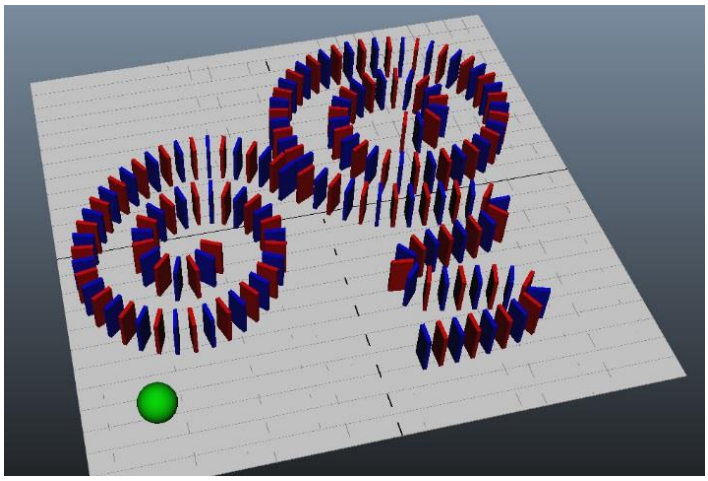
In this tutorial we will get to grips with building dynamic simulations and create a dominoes scene.

Building your scene



We are going to build a very simple scene for our dominoes simulation, first of all create a plane *Create > Polygon Primitives > Plane* and make sure it covers our grid. This is going to be the floor. Now comes the laborious process of building our dominoes. Create a polygon cube and scale it (*R*) to shape it into a domino and then move it to the place you wish the dominoes to start falling. If you centre the pivot of the domino (press *Insert* of hold *D* to enter move pivot mode, and hold *V* and *middle-click* on the *Translate Y* manipulator handle at one of the base vertices of your domino) you can easily snap the object to the grid. It is important that we do not have any intersection with the ground.

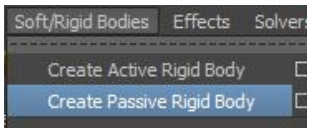
Now we want to distribute the rest of our dominoes by duplicating this piece *Ctrl-D* and moving them into a pattern. As a rule of thumb, I find it is usually best to leave a space of between a quarter and a third of the height of the domino. This provides enough space for the domino to knock the other over, and is close enough to still cater for sharp curves in our patterns. Finally,



we want to create a ball object that we are going to roll into the first domino to get things moving. You should be left with a scene that resembles the image on the left, albeit with your own domino pattern. I have coloured the objects in my scene with simple lambert shaders created in the Hypershade.

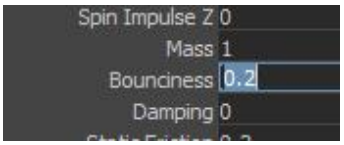
Select all of the objects in your scene and make sure you have frozen all transformations *Modify > Freeze Transformations* and deleted any History *Shift-Alt-D*. We can also group our dominoes to clean the scene up a little.

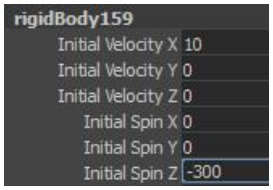
Setting up your Dynamics



Select the floor and in the dynamics menu set (*F5*) set it as a passive rigid body *Soft/Rigid Bodies > Create Passive Rigid Body*. This will set the floor as a collision object for the dominoes (and the ball).

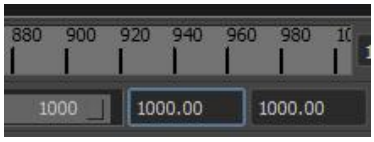
Now select all of your dominoes and the ball and set them as active rigid bodies *Soft/Rigid Bodies > Create Active Rigid Body* This may take a little while if you have got a lot of dominoes in your scene. The default values are not going to make the animation very realistic, so without going into too much detail we are going to make a few adjustments. Select all of your dominoes and in the *channel box* under the let's set the bounciness down to *0.2*.



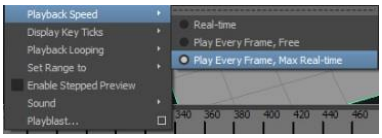


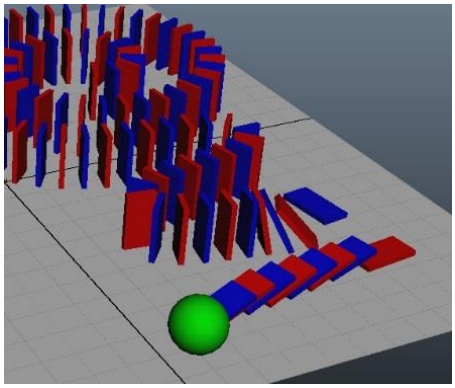
Now in order to make our dominoes fall, we need the ball to roll into the dominoes and knock them over. Select the ball and in the *channel box* again scroll to your *rigidBody* node and change the *Initial Velocity X* to *10* and the *Initial Spin Z* to about *-300* (your value and axis may be different if your sequences starts in a different direction).

Now when we play the ball spins in the direction of the domino but stays in its position. This is because there is no gravity so the ball is actually just spinning in the air. Select all of the dominoes and the ball and select *Fields > Gravity*.

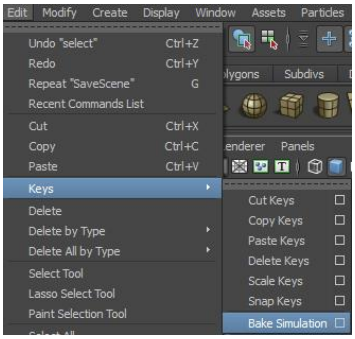


Make sure you have enough frames for the animation (I have picked 1000 frames just to be sure) and also make sure that your timeline is set to play back in real time *Right-click on the timeline > Playback Speed > Play Every Frame, Max Real-time*. This ensures that Maya calculates the dynamics accurately.



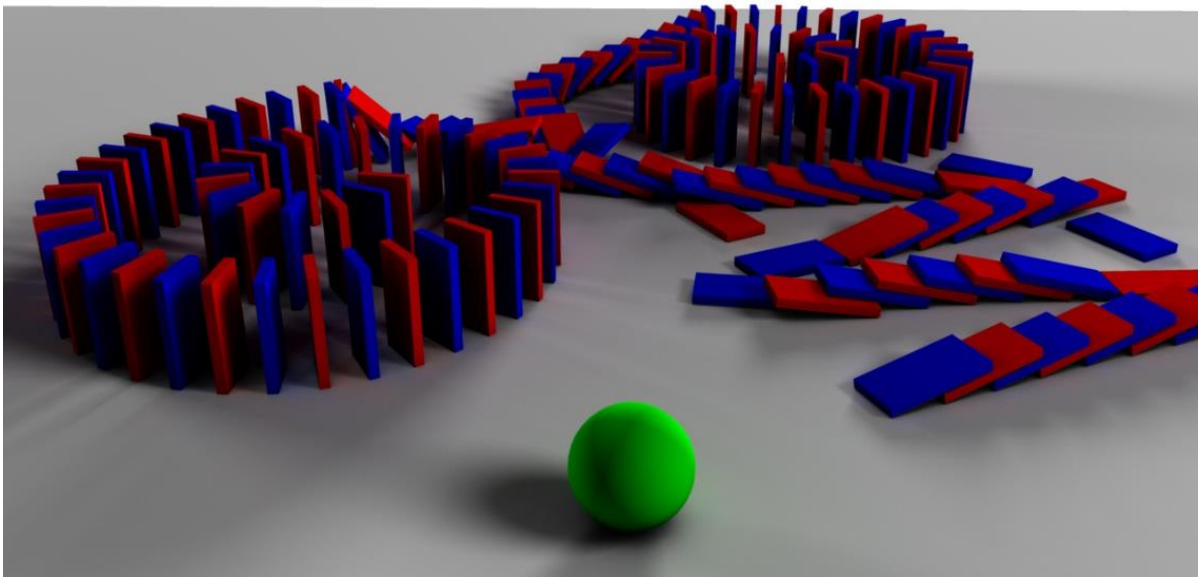


Now when we play our scene the ball hit's the first domino causing the rest to start to fall. If we continue you with the simulation, you will probably find that at after a short while the simulation will run very slowly. To get a better idea of how it is going to look, create a Playblast by *right-clicking* the *timeline* and selecting *Playblast* □ option box. This may take a while but when it is complete you will have a real-time example of the complete dominoes simulation.



You can tweak the rigid body settings if you wish (maybe you want the ball to be less bouncy or have more mass) and then when you are happy with the result let's bake the simulation so that we don't have to wait for it to re-calculate the dynamics each time. Select your dominoes and ball and click *Edit > Keys > Bake Simulation* □ and make sure *Shapes* is deselected. Now press *Bake* and watch Maya run through your simulation again. Again this may take some time. Now we can delete the rigid bodies (Select all the dominoes and ball, press the *down arrow* to pick your shape nodes, the *right arrow* to select your *rigid bodies* and *delete*) This removes the dynamics and uses the key frames on our objects so that it is easy for Maya to scrub through the animation.

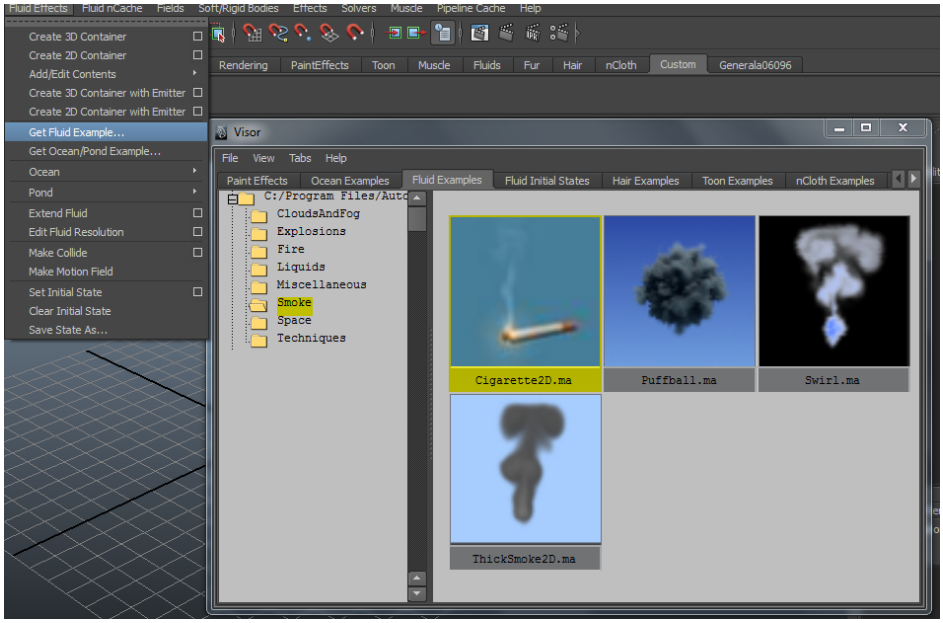
That concludes our Dominoes tutorial. Experiment with other fields and objects in your own simulations and when have you have finished, use the techniques described in our rendering section to render your sequence.



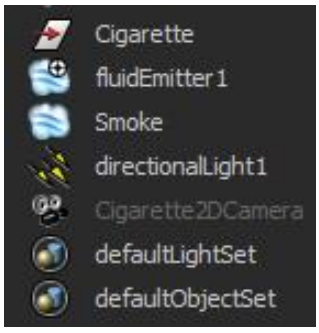
Fluid Effects

Cigarette Smoke

Fluids can become highly complex and very resource intensive. We are just going to introduce you to a couple of the presets so you have an understanding of where to find them and what they are capable of reproducing.

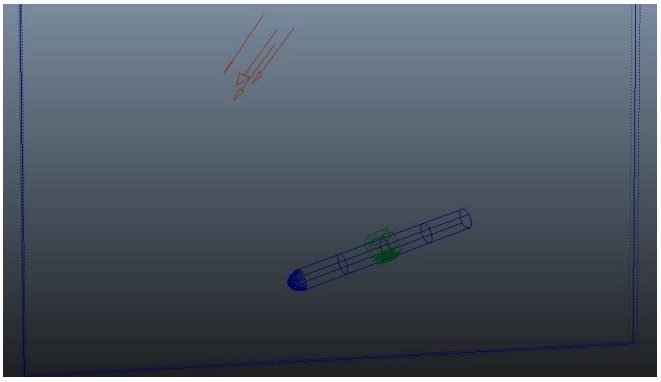


Start off my going to the *Dynamics Menu > Fluid Effects > Get Fluid Example* you will see the below screen. Right click on an image and choose to import it.



After you select you select the cigarette you will have the following imported into your scene

Increase the number of frames on your time slider and press play, and stop it at any point.

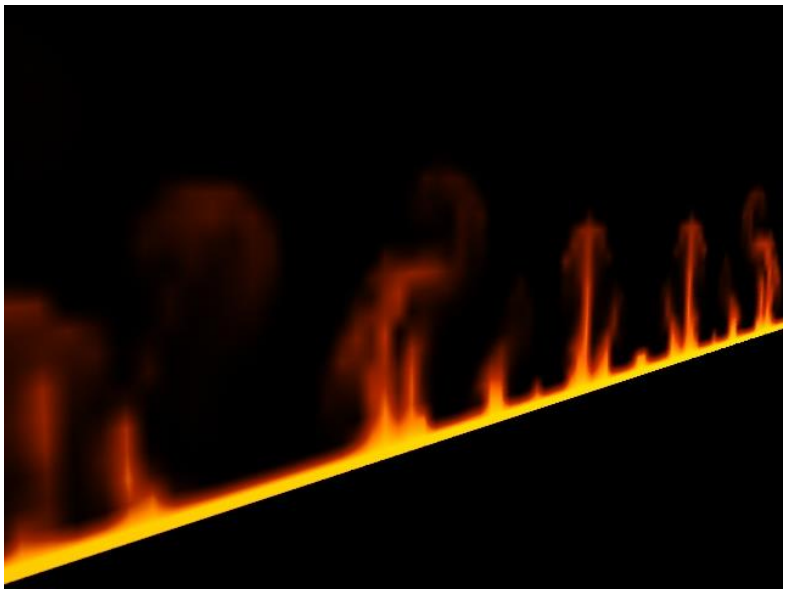


A quick render with default settings using Maya Software provides you with a lovely prop to be used in your scene.

In most instances this will be enough, however you can always go into the settings and do a little reverse engineering to see what changing some of the settings will do

Fire

Using the same approach as previously discussed however this time we'll be choosing to import some fire presets into our scene.



As you can see you can get some lovely presets from Maya with very little work, but recreating these from the beginning is no easy task. If you are intrigued by these effects try playing around with the cigarette preset and see how the settings can be manipulated and improved. Why not have a go at re-texturing it as well, see if you can't add that extra bit of realism.

CHAPTER 7: ANIMATION

Chapter 7: Animation

Introduction to Animation

Animation is the most difficult and most time consuming application of Maya. This is where an animator will take our rigged characters and bring them to life, moving them around and interacting with the environment. Although there still rules and regulations concerning animation it is much less regimented and structured than other areas of the workflow such as modelling or rigging.

A good animator is akin to a good artist; skills earned over time and styles that are unique to the individual. When working in a team of animators, you can often pinpoint who has worked on which character just by the way they move – their digital fingerprint is as unique as their minds.

Principles of Animation

Timing

Timing is the most important aspect of animation and the best way to get better at animation is to study the world around you. The time placed between key frames determines everything, too long and your animation will be boring but too short and your animation will lose fluidity. You should also aim to keep motion smooth and fluid.

Weight

Accurate representation of weight is another of the most influential factors on the believability of an animation. This can be as simple as adding a delay on picking up a heavy object and a further delay on your walk cycle as the character carries it.

Arcs

Characters should, as with everything in life, move in smooth arcs. This helps maintain fluid, attractive motion. This is why we recommend animating with FK controls where possible because by keying rotation rather than translation, a character's limbs will move more gracefully.

One key thing to remember is that the speed has an influence on the arc's angle. The faster an object is travelling, the straighter this curve will be. Keeping your animation curves smooth in your graph editor will give you much nicer results and allow you to be much more efficient in your animation.

It is also worth mentioning that arcs are just as important in your subtle animation as big arm swings. Even a character's head moving from side to side, if you can make it a nice arc instead of a linear transition the result will be much more engaging.

Exaggeration

Simply replicating realistic animation can look uninteresting in an animated setting so it is encouraged to exaggerate your characters movements. This can be as simple as pushing a character's body language a little further but you must be careful with overdoing it – exaggeration is not the same as pushing a character too far to the extreme.

Anticipation

This technique is used to draw the audience in and prepare them for an action, for example pausing before throwing a punch, or a gymnast bending their knees before leaping into the air.

Squash and Stretch

First of all, this is not the same as our squash and stretch rigs that allow us to wildly distort our cartoon characters for cartoon effects. The *Squash*, such as the character crouching down in preparation for a jump, and the *Stretch* at the point where the character is fully extending just before / as it leaves the ground should be given the correct emphasis.

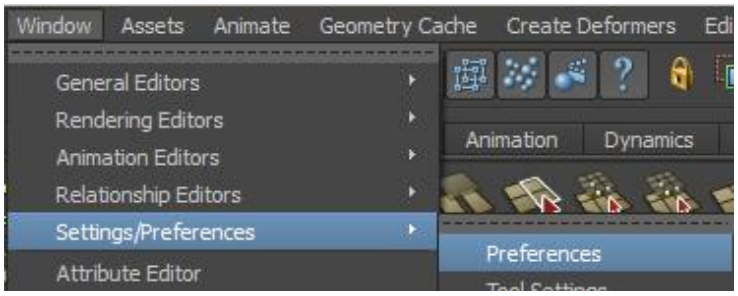
Personality

Characters must be appealing to the viewer so that they can relate to what is happening on the screen. The most important thing is for you character, hero or villain, is to invoke interest from the audience.

Secondary Animation

Adding secondary animation gives the scene support. It should never be so bold as to overpower the main action, but rather should provide backing to the overall scene, occasionally going by virtually unnoticed. This does not always have to be hand keyed, more and more these secondary effects are being controlled by dynamic systems such as hair and cloth.

Pose to Pose



When you are animating a scene we recommend you first plot your animation using key frames and then fill in the details later. This literally means posing your character at all of the most extreme key points of your animation as a basic timing guide. It is useful during this stage to set the default key tangents from the *Settings / Preferences > Preferences* window. Set *Settings > Animation > Default In Tangents* to *Linear* and *Default Out Tangents* to *Stepped*.

Straight Ahead Action

This technique is the alternative to pose to pose (and the most common for most people starting animation) essential you start from frame one and animate fluidly from the get-go. This tends to invoke more spontaneous animation at the cost of timing accuracy.

It is worth noting then, that these two animation techniques work in tandem and we recommend you use a combination of these methods in your own work. First block out your animation using pose to pose to get a general sense of timing and your key poses sorted, then you can use straight ahead action to blend in between these poses and add detail to your animation.

Staging

Staging is essentially how the animation is read by the audience. This can be done by strong camera work, lining up each of your shots to best explain the action to the viewer.

Animating in Maya

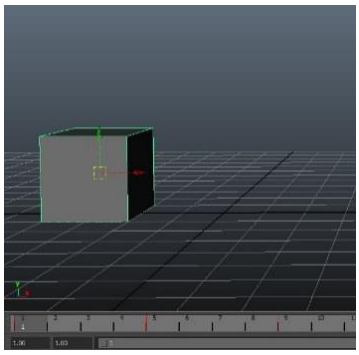
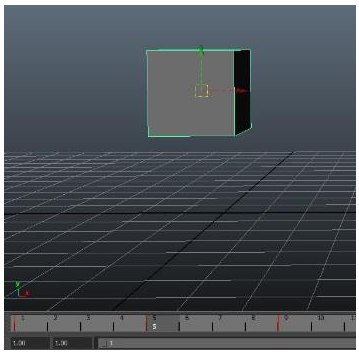
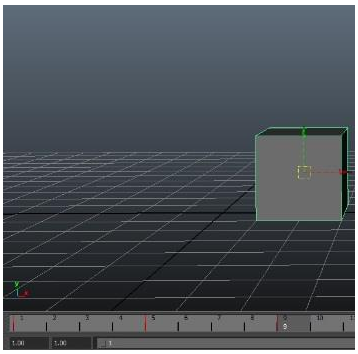
Animation in Maya is primarily key frame animation. The timeline splits time up into '*frames*'; usually (for UK TV) 25 frames per second. For each frame then, you can 'key' any of an objects attributes (translation for example) to hold that value at that frame.

Key Frame

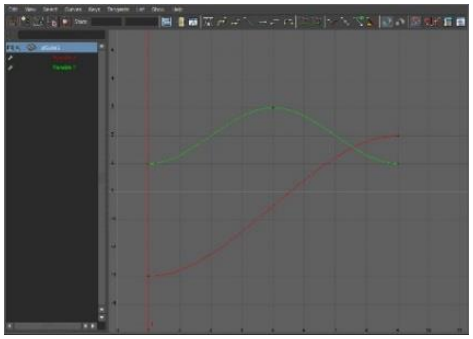
Probably the most commonly renowned styles of animation used within todays' industry is key frame animation. This involves positioning your character at a specific frame on the timeline to determine its appearance at that specific moment, *key* the relevant attributes and then move onto the point in time. When we scrub through the timeline we will see our character moving and deforming between each of our set frames.

Maya will then connect these keys together into a curve, working out in-between values.

This means you can very quickly get nice fluid animation, using only a few key frames.

Graph Editor



The graph editor (shown with the above cube animation) illustrates the keyed elements along the translate axis. As you can see from the diagram the attributes in question are colour coded, and each tell us information about how the animation will playback. We will have a more in depth look into the graph editor during our animation tutorials later in this section.

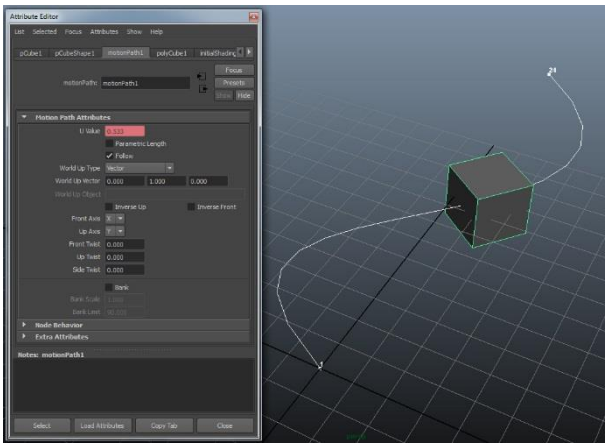
Translate X:

The cube moves from a value of minus three to three over the course of eight frames. We could have a straight line here, but by using flat tangents on the start and end points we ensure that the cube eases in and out of the animation.

Translate Y:

The cube moves from the floor, into the air and back onto the floor. This is shown by the green curve in the shape of a hump. Again we have flattened out our tangents to ease in and out of the animation.

Path Animation:



You can make an object follow the path of a CV curve using *motion paths*. The motion path start and end time can then be manipulated to control the amount of time that the object travels along the curve for and other attributes can be set for twisting and banking relative to the curve bend. These make it easier when an object needs to be rotated 90° to face forwards, for example.

Motion paths are most often used to move on object (such as a car) or a character with an animated walk cycle along a specific route. In both of these cases, to avoid wheel or foot 'ghosting' you will need to match the timing of the wheel rotation or walk cycle to the start and end frame of the motion path relative to the distance of the path. This is easy to calculate however, in the command bar, type *arclen <object name>* which will return the length of the curve. You can then use this number to divide the distance travelled per walk cycle by, giving you the number of cycles during the motion path. Multiply the number of cycles by the time it takes to do one cycle, and this will give you the total length of time for the motion path.

You can also use motion paths for advanced rigging techniques, such as tank tracks or making an object follow the curvature of an animated creature, but these are more advanced techniques that we will cover in future resources.

Expressions

Expressions are small snippets of code that are evaluated by Maya for each frame. These are used primarily for dynamics and particle systems, for example emitting particles when two rigid objects collide.

Set Driven Key

Used to set up interactions between attributes whereby the driver, drives the driven. The unique feature of this tool is that the relationship is keyed between the driver and driven nodes, which means you can view and edit the relationship in the graph editor and take advantage of your curve editing tools. By driving multiple nodes from a single driver you can create some nice fluid animation that is keyable within your scene (for example our reverse foot lock that we have set up on page 152).

Animation Layers

Animation layers give you multiple levels of animation that add together to give you a final result. For example, during the walk cycle we have created, we could create a new animation layer called head turn, and animate our head turning to face the camera a few steps into our animation and then back to default again.

Because our two animation layers are separate to each other, we can easily edit one without worrying about affecting the other.

Useful Animation Techniques

Silhouettes

When you are animating, especially while you are creating key poses it is valuable to check the silhouette of your character. This is a key instrument in keeping your gestures powerful and interesting for your audience.

In Maya a simple way to view your scene silhouette is to uncheck *Use All Lights* in your *Viewport > Lighting* menu and then also check *Use Default Material* in your *Viewport > Shading* menu. This stops any of your lights affecting the viewport (your character will turn flat) and then forces the viewport to use the default Maya material *lambert1*. Now all we need to do is change the *lambert1* diffuse colour to black in our *Hypershade* window.

This gives us a nice silhouette of our character, without changing the scene itself (when we hit render, the scene will render out as before).

Smart Blocking

We can apply our principles of animation such as follow through during our pose to pose stage and include them in our key poses. This gives us a much more powerful bases for when we come in and fill the detail between poses.

We can also go in and tweak the finer details such as fingers and other secondary details at this stage. Check against your silhouette in your camera view while you are creating your poses to check the emotion and exaggeration is correct. You want to make each pose as interesting as possible.

Moving Holds

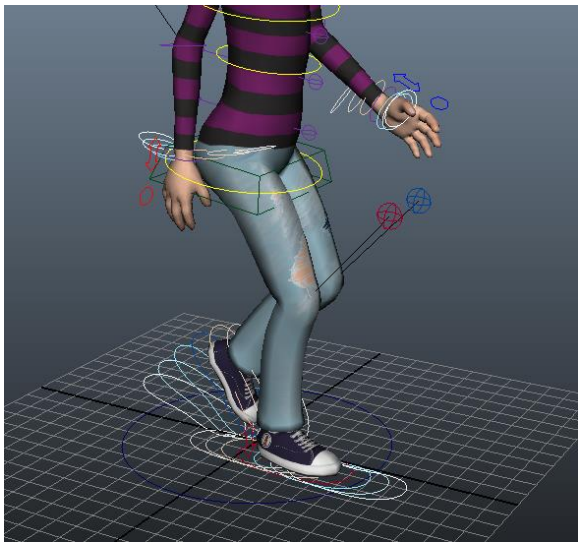
Moving Holds are an important device to let the audience read what the character is doing and second guess what the character may be thinking. The easiest way to create a moving hold is to go to the key you wish to extend the hold until and then *middle-click* the key where the hold starts. You will notice that the animation does not change, so now if you key your attributes (press '*S*' to key everything) it will key your original key frame with the value of the one your middle-clicked.

Motion Trails / Ghosting

An easy and visual way to check our animation arcs is to use *Motion Trails* which can be accessed in our Animation menu set (*F2*) and then *Animate > Create Editable Motion Trail* □. This will display a curve along the motion path of the selected object, including the location of any keys on that curve. New to recent revisions of Maya is the ability to move these keys in physical space, or even change the frame number that they fall under giving us near graph editor control in 3D space.

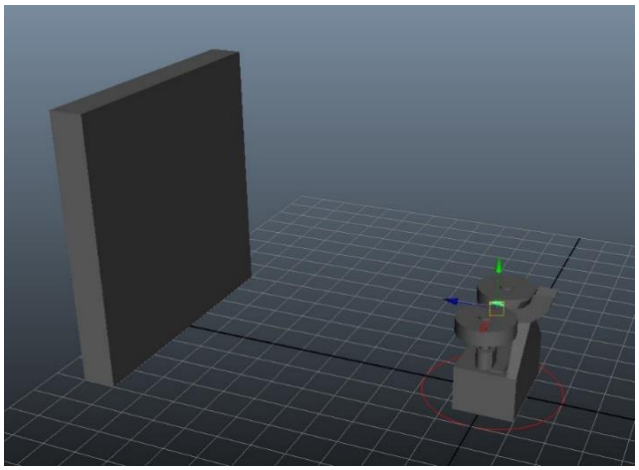
Ghosting is more suitable to checking our ease in and ease out keys. Still in your Animation menu set, *Animate > Ghost Selected* will show the selected objects previous and future locations by about three steps. Correct ease in and out is displayed by the frames getting closer together or further away respectively.

The feet and hands in this imaged are Ghosted:



Animating a Tennis Ball

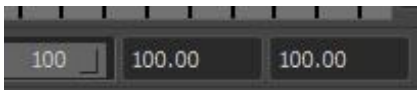
This tutorial will teach you the main principles of animation, specifically how to animate a tennis ball being fired from a cannon and bouncing off a wall to a stop.



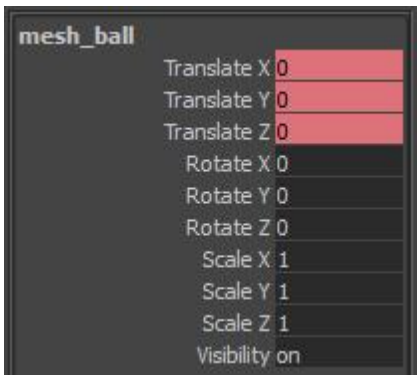
We would recommend you use your own scene for this as all you really need is a poly sphere to represent the tennis ball and a poly cube to represent the wall. If you are feeling extra adventurous you could build a tennis ball launcher like we have built in our scene. Make sure that you freeze your transformations *Modify > Freeze Transformations* and delete any history *Shift-Alt-D* before you start animating.

First let's plan our animation. We want the ball to launch towards the wall at a slight upwards angle and then bounce off the wall, fly over the tennis launcher and bounce on the floor a few times before it comes to a stop. The ball will drop slightly as it is flung, but will lose speed and drop at a steeper angle after each bounce.

Most likely your time slider is set to 48 frames, we are going to increase this to 100 to give us more time to play with.

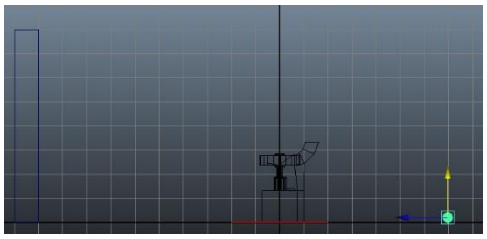


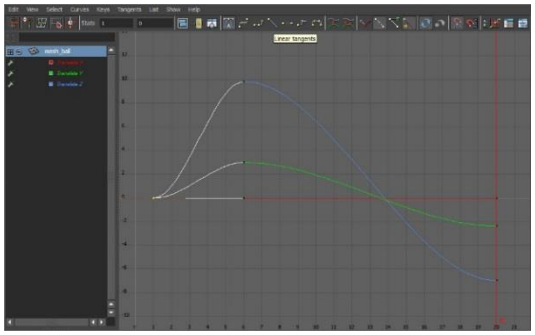
Ensuring we are at the frame one, select the ball and press *Shift-W* to key the translation in its starting position. We only want to create key frames for the attributes that we are animating; this keeps the *time slider* and *graph editor* clean and simple if we want to go in and change things. It also allows us to control other attributes such as rotation and scale without worrying about previous key frames getting in the way.



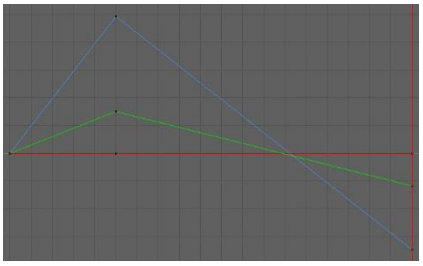
Move the time slider to *frame six* and in the side view move the ball to the wall in the *x axis* and up in the *y axis* by about one fifth of the distance and set another key frame.

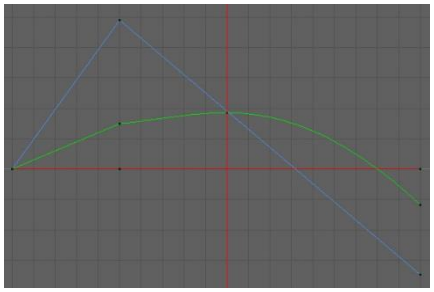
Now move the time slider further on to frame 20 and move the ball back behind the launcher and on the ground. If you play back the animation now, you will see that it is missing something – the 'bounce' off the wall is too soft and the ball leaves the wall in the wrong direction – it should bounce upwards first and then drop.



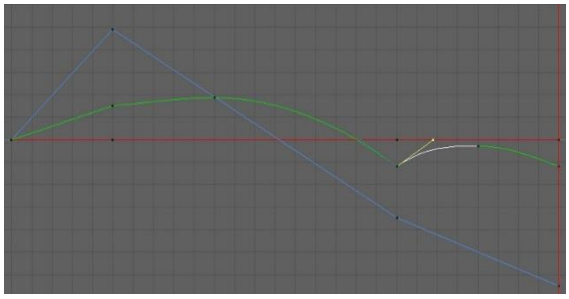


Open your graph editor *Window > Animation Editors > Graph Editor* and press *F* to frame the key frames. Select the first key frame, at the moment it has *Flat tangents* which means the ball will ease into the animation – because we are firing the ball straight from a cannon we want to change the tangents to *linear tangents*. This will give us an instant trajectory. In similar vein we want to change our other keys to linear because they are currently all impact points where the ball will bounce.

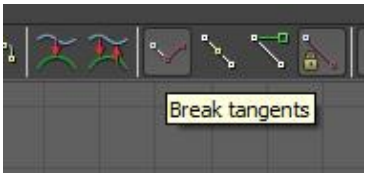


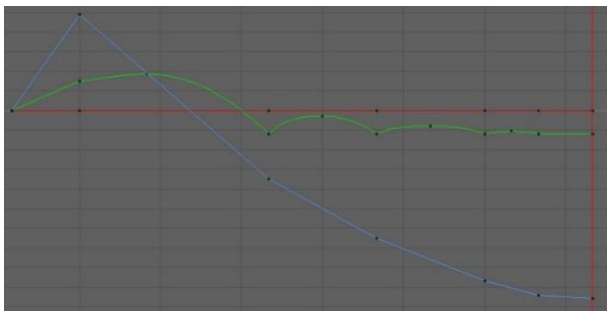


The bounce speed is more accurate, but we still need to make the ball continue upwards a little after the bounce. Move to *frame 11* and move the ball up in the *Y Axis*. *Right-click hold* the *Translate Y* attribute and select *Key Selected*. Keying the attribute individually ensures that the other translate values are not inhibited.



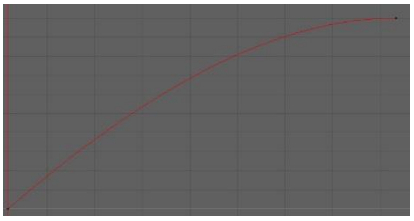
Now we want to use this method of animation to make the bounce again on the floor. When you have set your keys and are in the graph editor, one extra too we will need to use is the *Break Tangents* button. This allows us to move the in and out tangents independently. To change our tangents (the yellow lines coming from the point) make sure you are in move mode (*W* key) select the line and *middle-click drag* to change its direction.





Using the same techniques, add another couple of bounces, reducing the length and height each time. Your graph editor should resemble something similar to the one on the left.

Before we finish, we are going to add some back spin to the ball. At *frame one* key the ball's *Rotate X* attribute at *zero* and then on our last frame set the value to something like *1000*. The initial key should be made linear to resemble the curve on the right.



Play with the in and out curves to get the correct bounce speed, and you should be left with a simple ball being fired into a wall and bouncing back.

Creating a Walk Cycle

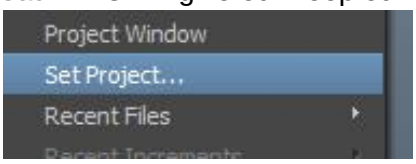
A walk cycle is one of the most important and often most overlooked part of animation. The walk cycle is vital to making your character believable – in the same way that you can discern you friends from their body language alone, it is the subtleties in the weight and balance distribution that makes a good walk cycle.

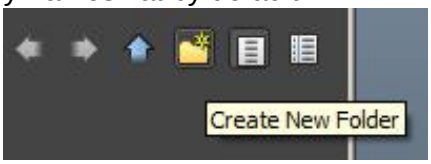
Character Rigs

For the most part we have always recommended that you use your own models, your own rigs and apply our techniques to your own scene. If you already have a suitable rigged character that is animation ready then load it up, but if you are more interested in getting stuck into animation that modelling and rigging then we recommend that you use one of many freely available rigs. We recommend the popular '*Andy Rig*' created by ***John Doublestein***. Designed specifically for students it has all the control systems we could want including IK/FK switches and squash stretch.

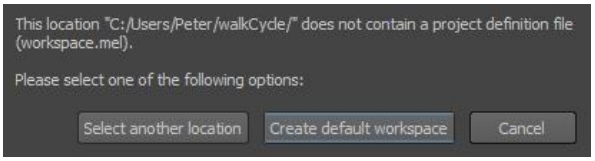
Animating the Walk Cycle

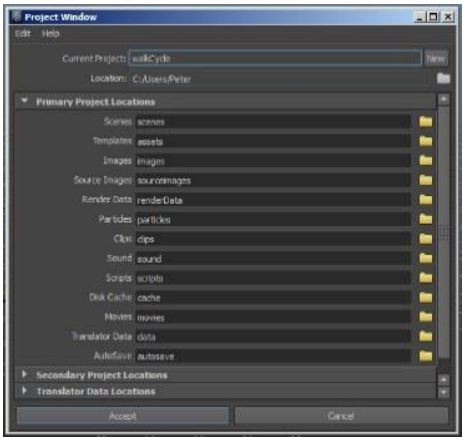
First we need to double check our animation preferences. Open up your *Window > Settings Preferences > Preferences* window and under the *Settings* tab make sure that our Time is set to *25 fps* (standard UK PAL). Under the *Animation* tab set both the *Default In tangent* and *Default out tangent* to *plateau*. This will give our keep our key frames flat by default.

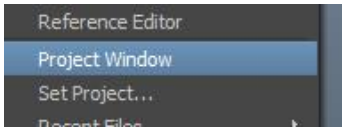


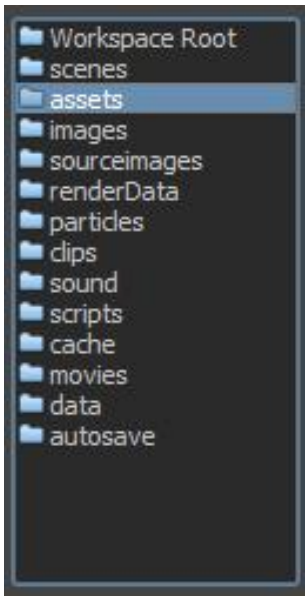


Now let's set up our project. Under the *File* menu click *Set Project* and then create a new folder with your project name, let's call it *walkCycle*. If a dialogue pops up, select *Create default workspace*. Now to finalise our project and create our subfolders, simply click on *Project Window* in the *File* menu and then *Accept* the default folder names.

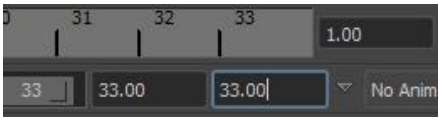




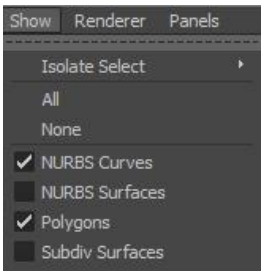




Copy your character rig into the assets folder, then back Maya click *Create Reference* from your File menu. Click on Assets from the left hand menu and select the rig your wish to animate with.

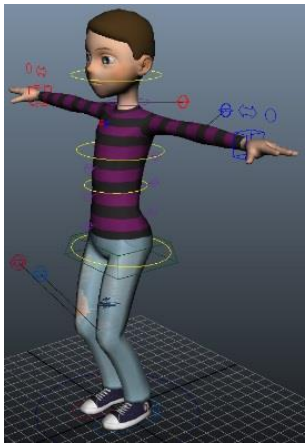


While we are still setting the scene up, change the playback range to 33. This is going to be the length of our walk cycle.

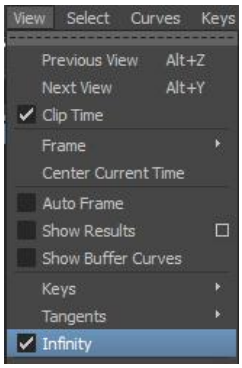


In your viewport *Show* menu, select *None* to hide all objects and then tick *NURBS Curves*, *NURBS Surfaces* and *Polygons* to make them visible. This will keep any joints and other visible objects hidden while we are animating.

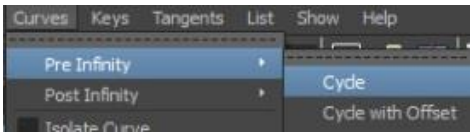
Select all of your controls (if you are working with your own rig then make sure that only controls are selected, not your mesh or any other nodes that could break if keys are applied). We can also turn on our *Auto Keyframe toggle* on in the bottom right.

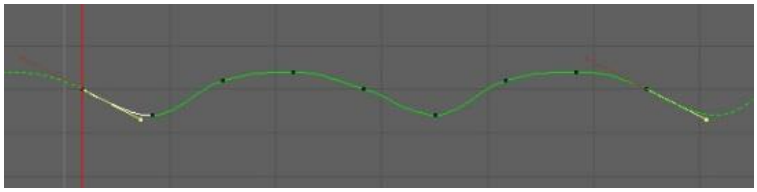


We have four key points during our walk cycle, causing our characters centre of gravity to create a wave pattern. Frames 5, and 21 are when the hips will be at their lowest, frames nine and 25 the highest. Let's plug some values in to give us this bob to work with. Make sure you are at frame one and key the hips *Translate Y* at zero, then key the value on frames 17 and 33. Move to frame five and bring the hips down, auto key should have created a new key for us. To set this same value you for frame 21, middle click on frame 21. You will see that the animation does not update, but if we key the *Translate Y* attribute it will create a new key with the previous value. This technique is also great for keying hold poses. Now on frames nine and 13 and then 13 and 29 key the hips in a more elevated position to give us a nice smooth bobbing effect.



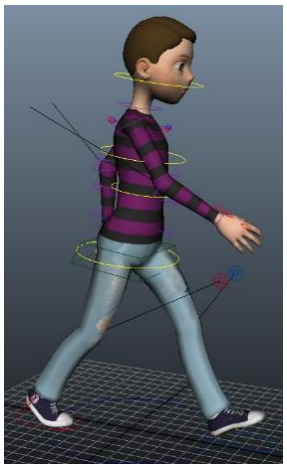
Open up your *Graph Editor* and press '*F*' to frame up your translate curve. In the Curves menu select *Pre-Infinity > Cycle* and *Post-Infinity > Cycle*. This will force the curves into a loop and to show us this loop we can enable the Infinity box in the view menu.



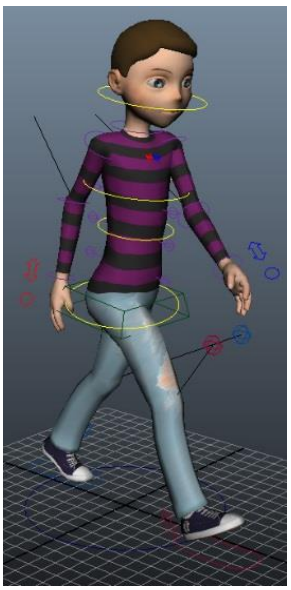


This will give us a dotted line either side of our key frames to show us how the value will continue. At the moment our loop has this ‘step’ or kink at the beginning. Select the first key frame and then shift select the last and press ‘flat tangents’ to make sure they are the same. Now select and shift select the tangents for the key frames and in ensuring you are in move mode (W) move the tangents down so that the curve is smooth.

Go back to frame one and move the feet into position (starting with your left foot leading) watching for any hyper-extension. We never want any of our limbs to be completely locked out because this will cause ‘popping’ to occur during our animation. We want to bring the rest of our character into a rough start pose. Rotate the body forwards a little and also rotate our hips to follow along with our legs. Your feet may be different if you are using your own character, but we are going to use our *Foot Roll* attribute on our foot controls to bring the toes up and heel forwards for our respective feet. We also want to ensure that auto stretch is turned off.

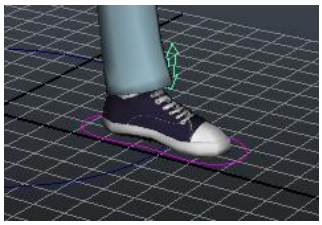


In our upper body, we want to bring our arms down and out (do not rotate them forward too far) and rotate our upper-body / chest control and shoulders out to follow our arms. For a female character it is often nice to rotate the arms out away from the body slightly, whereas with a male character we want to turn them in a little. I have also taken this opportunity to relax his fingers. With the first frame basic pose complete, select all of our controls and press *S* to key them in place. We also want to key everything at frame 33.

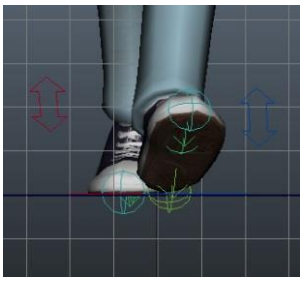


Now we need to build our right foot forward pose on frame 17. Starting with the feet copy the left foot values at frame one for your *Translate* and *Foot Roll* attributes and at frame 17 paste them onto our right foot values, then repeat the process for our other foot. Do the same process for our arms. The hips, chest and shoulders are easier – we simply want to put the negative (or positive) version of the current *Rotate Y* value to flip it.

Ok, so if we play back our cycle it is already starting to resemble a nice walk.

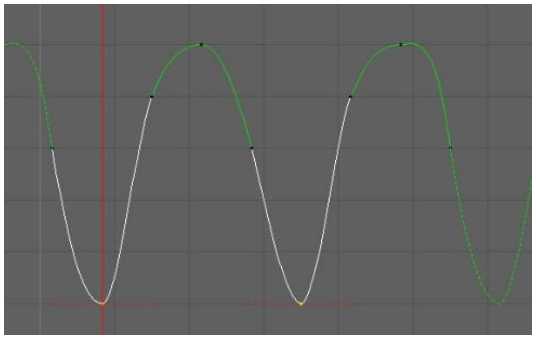


The feet are the most noticeable problems. By frame five (and 21) his foot should be completely planted, so let's go in and fix this by zeroing out the Ball Roll attribute on these keys.



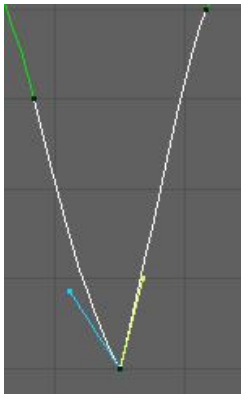
Switch to the front view. We can see that his weight is way off, so let's centre our feet a little. At frames one, 17 and 33 we want to bring both feet closer to (but not on) the centre line, at frame 25 we want to move our left feet out to the side and at frame nine we want to move our right foot out to the side. While we have our feet selected, I have also used the *Toe Pivot* attribute to point our feet out a little.

The main thing missing now is our feet actually lifting off the floor. Select your feet and make sure that your *Translate Y* and *Rotate X* attributes are keyed at zero. Now at frames nine and 21 for your right and left feet respectively, rotate the foot down and lift the foot up. This should create a nice arc for your feet to travel in.

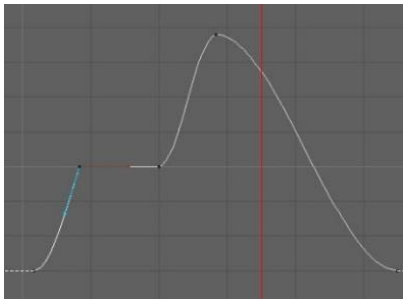


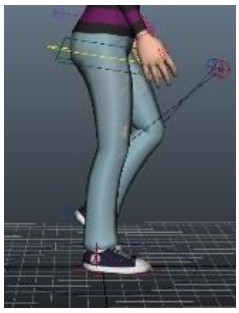
We need to add some weight to our animation, at the moment it is looking a little too floaty. With your hips selected open up your graph editor, select your *Translate Y* curve and press the *F* key to frame it. To increase the illusion of weight, select

your bottom keys (frames five and 21) and click on *Break Tangents* to allow us to move the in and out tangents separately.

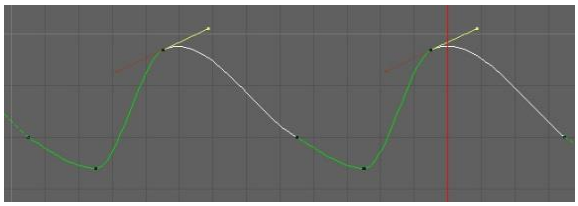


We want to increase the angle in and out of these points to make a point. Notice the weightier bounce when we play back our animation. We want to use a similar technique to give the feet more weight as the hit the ground. The left foot should resemble something like on the right, with a sharp in tangent at frame five. Our right foot will be similar, with the sharp tangent at frame 21.

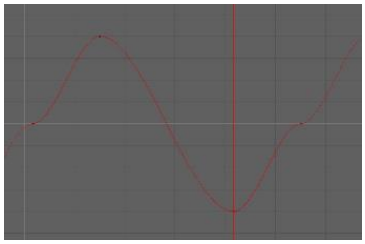




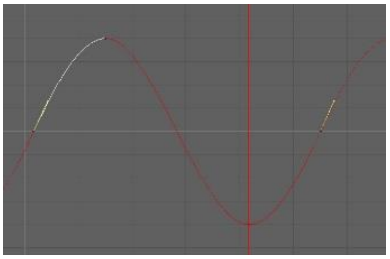
Now when we play back our loop, we can see that the weighting is much more realistic; our footsteps much snappier. To finalise our feet, let's check our leg extension on the plant and lift, making sure that we don't have any over extension (popping). On frame nine, let's move our body up to straighten the leg. To mirror this on frame 25, middle click frame 25 and key the *Translate Y* attribute.



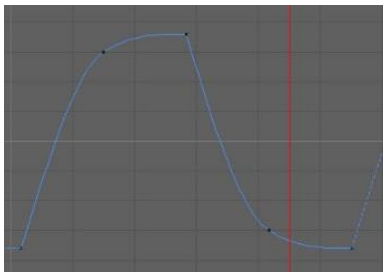
In your graph editor, grab the two top points and move the tangents to give the curve more of a hold on the higher position and lead into the next frame better.

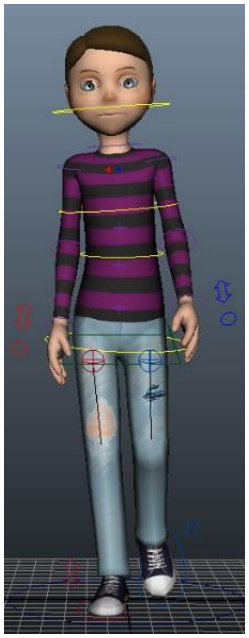


This is still looking much nicer, a much more dynamic walk. Moving on to our balance and weight distribution, move our body left on frame nine and right on frame 25 to match up with where our characters weight is being distributed. If we check our graph editor, we will find that we have a step again on the first and last frame, so let's select both of these points, flatten the tangents and then move both tangents up together.



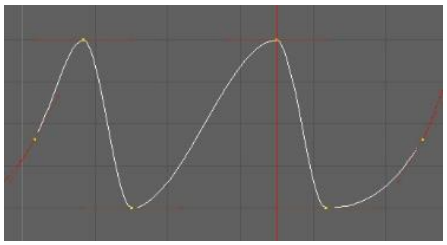
This will give us a nice side to side movement which we can enhance by rotating his hips in line with the movement. It is also nice to add weight to this rotation, to really emphasise the feet planting on the ground. Were nearly there but I'm still not quite happy with my foot planting – there is too long a delay between the heel hitting the ground and the toes. Go into your graph editor and bring the Foot Roll one frame sooner depending on your animation.





Now we can leave the bottom half for the moment and concentrate on the upper body.

Select your chest control and rotate it side to side on frames nine and 25 in the opposite direction of the hips. In the graph editor be sure to smooth the in and out tangents to remove the step from our loop. Playing the animation back you will see that instantly we have much more attitude going on in the walk cycle, especially from a more front facing angle.

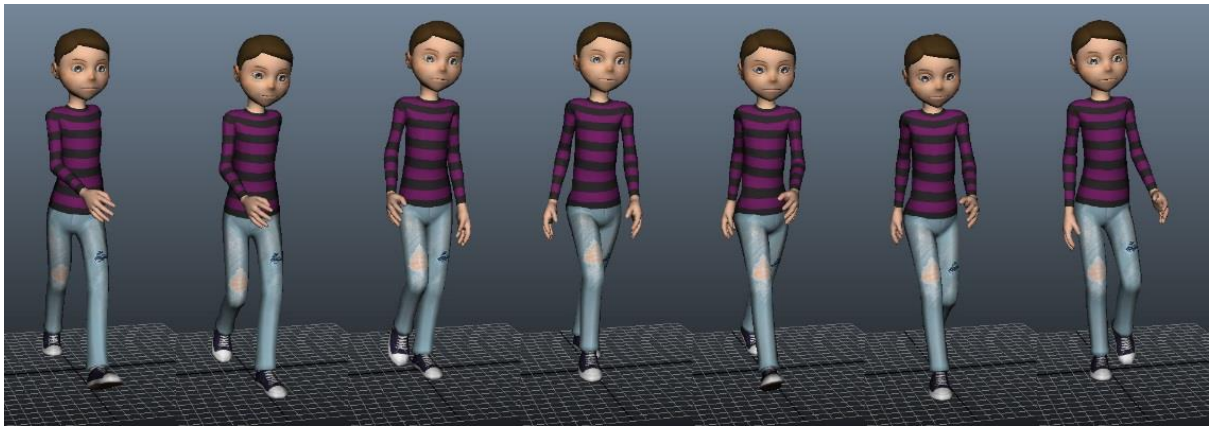


The front to back rotation is ever so slightly more complicated because it is reacting to the hips so it will be slightly delayed. Key the chest forward at frame five and 21 and straighter at frame nine and 25. Now after we have fixed the curve in the graph editor, we want to select all the curve points and *Shift-Middle-Click* to the right to move all the frames right by one frames.

Use this same technique to animate our head nodding, but move the head *Rotate X* keys forwards by two frames.

Our arms still look a little robotic. For our hands, repeat the steps for the chest and head but move the rotation keys way forward so that the start key is on frame ten. This will make our hands drag behind the arms to give us more fluid animation. You could also move our elbow key frames on by one or two frames.

Check all of your key frames in the graph editor, especially for kinks or steps in the curves and also for areas where you wish to add weight. Also keep an eye on your legs for popping. Finally hide your curves and preview your animation using a *Playblast* by right clicking on the



timeline and selecting *Playblast*. Although playback in Maya is a good start, sometimes the system can be choppy – a Playblast is much more realistic to your final outcome.

This concludes our walk cycle tutorial. You should now be well equiped to transfer these techniques onto your own characters and use the same techniques to create run, jump and other cycles within your animation.

Animating a Arthropod (Scorpion)

Animating a scorpion is no simple task, even the fundamentals of a walk cycle takes some thinking about when you are dealing with eight legs. The key thing to remember is that research is your best friend so the first step is to work out how an arthropod walks. Watching spiders and scorpions, we can see that generally the legs work in pairs, in a logical pattern – so using that knowledge we can draw up a spreadsheet with frame numbers and write a plan for which frame each foot lands on and lifts during the cycle.

This takes planning and patience, but you will end up with something that looks a little like this:

Frames	1	3	5	7	9	11	13	15	17	19	21
R1	**D**	D	D	D	D	**D**	U	U	U	U	U
R2	U	U	U	U	U	U	**D**	D	D	D	**D**
R3	U	U	U	U	**D**	D	D	D	D	**D**	U
R4	**D**	D	D	D	**D**	U	U	U	U	U	U
L1	**D**	U	U	U	U	U	U	**D**	D	D	D
L2	U	**D**	D	D	D	D	**D**	U	U	U	U
L3	D	D	D	**D**	U	U	U	U	U	U	**D**
L4	U	U	U	U	U	**D**	D	D	D	D	**D**

Following these directions, you can start to make the laborious process of keying the feet. First animate your body control to get the correct timing and placement for your feet, then animate the feet to follow. You will find that as you go on, you will be able to animate quicker and quicker.

As with any walk cycle, be sure to add weight to your feet and body movements by manipulating (and breaking) your tangent weights. This will go a long way to make your scorpion look more dynamic.

Bear in mind that this should be used as a guide, obviously if the scorpion is in an action sequence then his movement will be different, and there are other times you might need to take some creative licence with the plan.

Dynamics

This form of animation is based upon physical factors such as friction, gravity, collisions and wind. Once the variables have been set, the results can provide you with incredibly realistic movements, which would otherwise be quite difficult to ascertain. A large incremental domino rally for instance would be difficult to keyframe realistically, however depending on your computers' hardware using dynamics may prove easier but the drawback is that if the calculations are complex it can be relatively intensive on computer resources.

Motion Capture

Motion capture allows you to drive your character's using recorded data from an actor to give you absolute realism in your animation.

There are a few main methods for motion capture, which all have their pro's and con's with regards to location constraints, cost, and ease of implementation.

The most common method is optical motion capture. This involves an array of camera's surrounding a person (or multiple people) who have reflective balls on all of their joints. The cameras can then track each of these points and map the 3d location, capturing the motion of each ball. The benefit of this system is that is relatively cheap to set up, and it is also well known so there is a lot of support available. You can also be sure that the world location will always be accurate and the same, i.e. if the motion actor walks 10 metres and then returns to the same spot, the 3d character will also return to the exact same spot. One of the main drawbacks is that it is not portable; you can only capture data within the area dictated by the cameras. You can also get problems with occlusion (i.e. if something is blocking the line of sight of the cameras) depending on the number of cameras you are working with.

A relatively new method that is growing fast in popularity is gyroscopic motion capture. This involves strapping into a Velcro suit with magnetic gyroscopes on each joint, which feed into a wireless transmitter. Instead of tracking the physical location of points on the body, the rotation of each gyroscope is recorded. This main advantage then, is that it is completely portable. As long as the transmitter and receiver are in range (generally about 20 feet, but this distance is improving all the time) you can record the actor's movements. This means that you can have a mobile base station in the back of a car for example. There are a few disadvantages to this however. Because only rotation data is recorded (unless you opt for an accompanied GPS unit, which I will not get into because of complications this brings to the equation) the software has to evaluate the translation data, so if the actor walks away and returns to the same point as described in the optical capture example, they will not be in the same spot (although they will be nearby). This system also struggles to accurately record jump's, although I have personally used this method of motion capture to record various gymnastics with a lot of jumping around at it is perfectly doable – you just have to take into account that some extra time needs to be allocated to clean up this kind of data.

A very similar method to the gyroscopic motion capture (and developed by the same company) is mechanical motion capture. This records data in the same way (rotational only) but instead of magnetic gyroscopes, the actor wears a kind of exo-skeleton. Although this eliminates any magnetic interference, it is also much less comfortable to wear and more restrictive for the actor.

A very new method that although worth mentioning has not yet been refined for professional use, is by using Microsoft Kinect. There are a few companies currently developing this, IPiSoft being the quickest to a commercial product. Essentially it is very low cost optical motion capture, and certainly worth a look for those who do not have access to any other motion capture studio.

Once you have recorded your data however, there is still ‘cleanup’ that needs to be done before you can use it on your characters. For gyroscopic capture you will need to check and fix and foot slips (where the software thinks the opposite foot is planted) and then for all kinds of motion capture you will need to run through the recorded data to remove any anomalies, usually easy to spot as spikes in the animation curves. Most of the time you can apply a filter which will solve most problems, but in my experience there is usually something that has gone awry that needs some extra intervention.

Water:

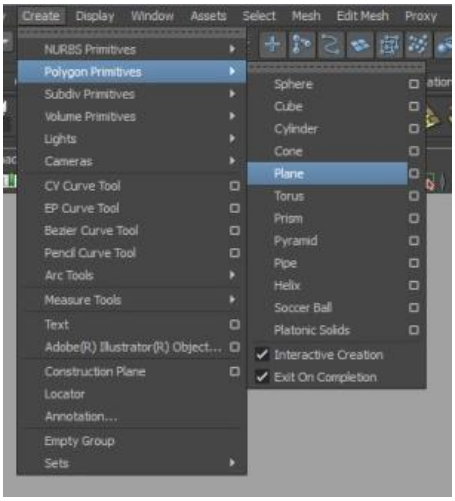
Animating water:

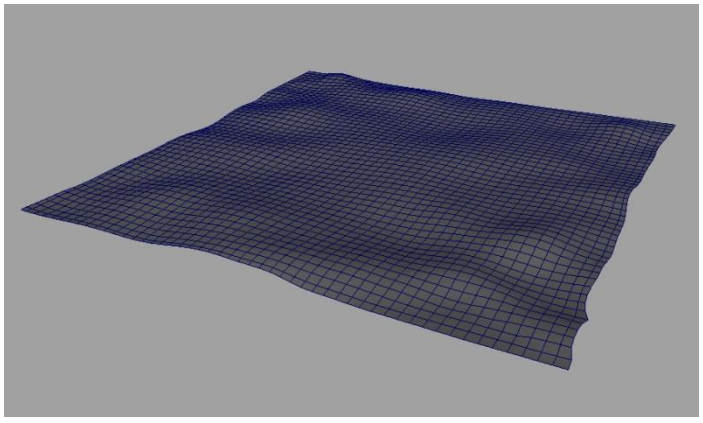
Water is something that people often struggle with, so we have put together a very fast and effective way, which is something picked up from one of our mentors. This is predominantly for non-realistic water, however, with some tweaking and a little work on a more realistic shader, we've found you can get some surprisingly realistic results.

The purpose of this tutorial is primarily to demonstrate that you do not need to use complicated real-flow set ups and huge particle simulations to get good looking fluids.

Step 1: Geometry Creation and Manipulation

To start off with, we will create a simple plane for the basis of the water's surface. In channel box, increase the subdivision of the *width* and *height* set to about *24*. Now we want to sculpt the water surface to give it a more uneven, natural water effect. You can do this using either the sculpt geometry tool, or you could just manipulate the vertices using the soft-mod setting with the move tool. Whichever way you are more comfortable with.

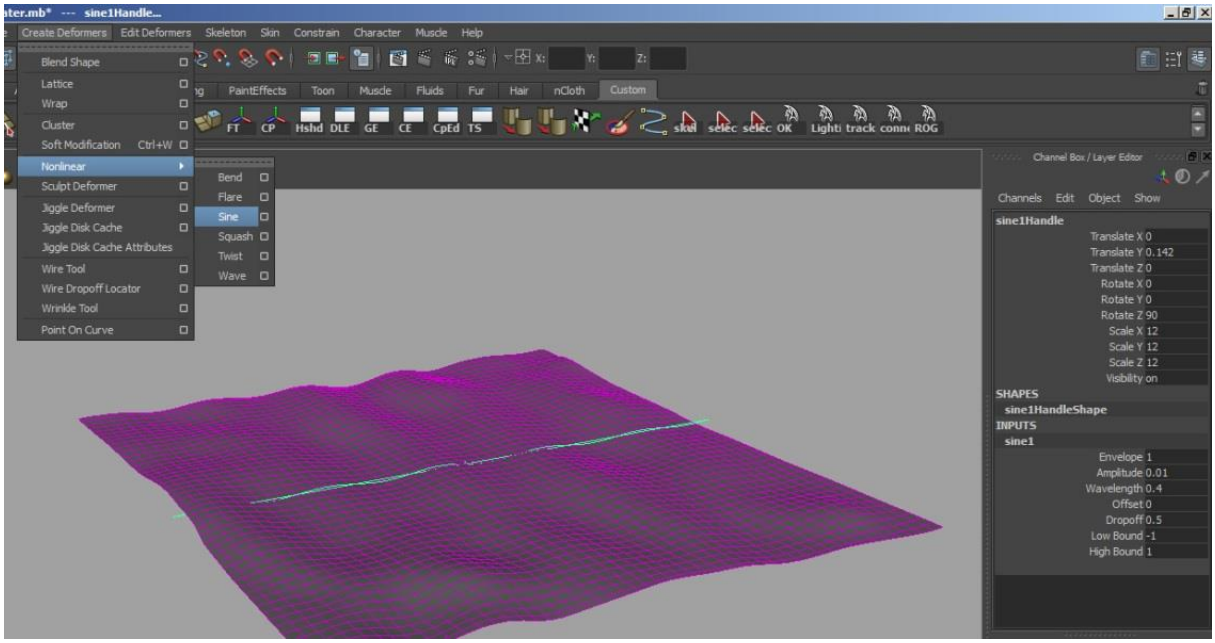




You should end up with something that looks like the following, feel free to experiment with different variants of choppy and still water depending on your application.

Step 2: Assigning a Modifier

To get a general wave going, were going to apply a sine modifier too it. After you have rotated the *Z axis* by *90 degrees*, set the *amplitude* to about *0.01* and the *wavelength* to about *0.4*. We have also chosen to set the *drop-off* to *0.5.*

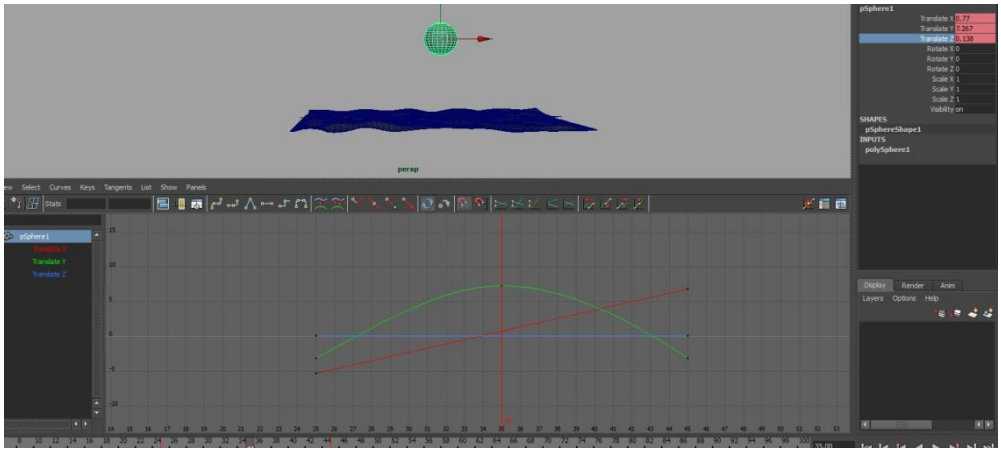


Step 3: The Ball

Now we're going to create a ball, and make it jump out of the water and back in again. Create a *sphere primitive* and place it under the water.

- Extend the *timeline* to something like *100* and key the poly spheres *translate* attribute at frame *25* on the *timeline*.
- Now you want to key the sphere's translate to the other side of the water at frame *45*.
- Go back to frame *35* in the middle, and just key the *translate Y* value to go above the water.

This should give us a nice trajectory curve for the ball to follow. You can tweak the curve further by bringing up the curve editor; you should get something that looks like this:



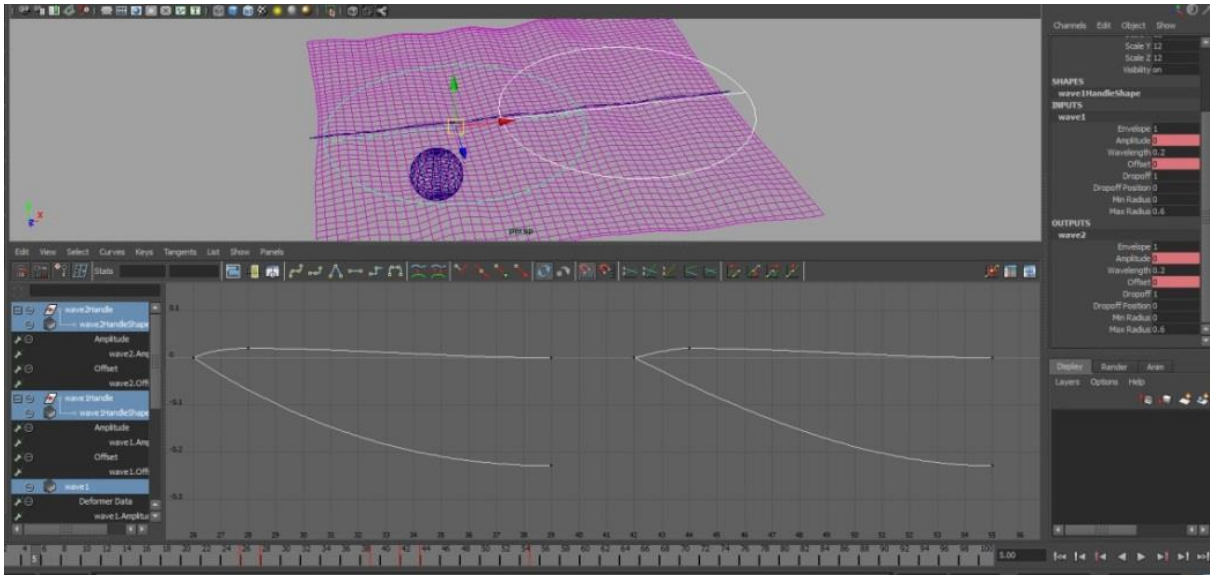
Step 4: Water Ripple

We want to create a ripple effect from the centre of the ball, so select the water plane and then create another non-linear deformer, this time the wave deformer. Scrub through the timeline to find the point where the ball intersects leaving the water, and move the wave deformer into position.

Now we want to animate the wave as the ball leaps out of the water, so bring up the wave input in the channel box and graph editor. The specific settings you use will depend on the type of look you would like, but the key is for the water animation to be seamless. Try to get similar curve shapes to the images below.

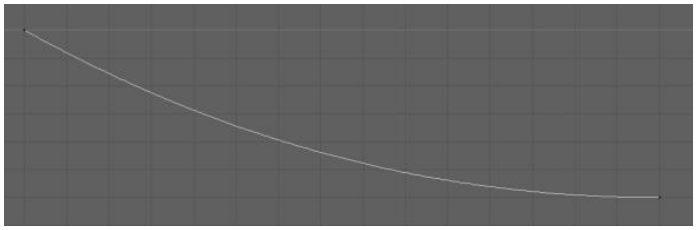
- The *amplitude* is the height of the wave, you want a dramatic increase from off to on, and then fade over time back down to zero.

Note: *Use the flat tangent button*

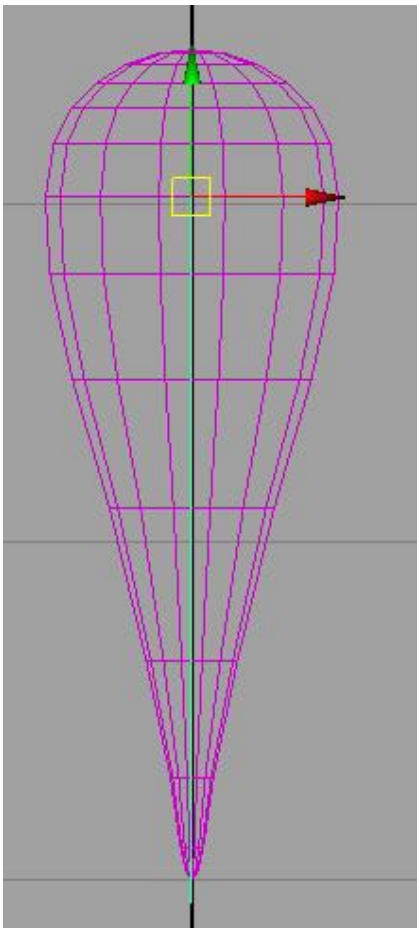


- The offset controls the animation of the wave, you want an instant start and a smooth finish.
- To give the illusion of waves spreading across the water, we also want to animate the max radius to increase over the duration of the splash.

Repeat this process on the second splash, when the ball hit's the water. Referring back to the previous images

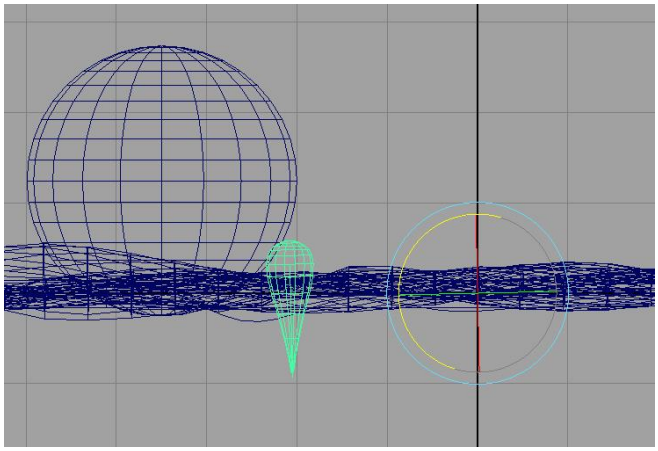


Step 5: The Splash



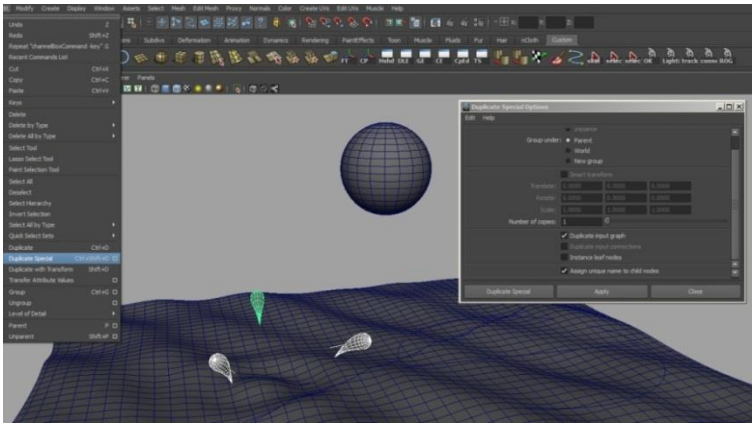
The next step is to create some 'splash' water-droplets. Starting with a sphere, model a simple water droplet shape. Apply a *bend modifier* to it, we are going to animate this to make the droplet more fluid.

Put the mesh and the deformer into a group and centre the pivot, we are going to use the *scale Y* on this group to make it squash when it returns to the water. To give us a nice clean arc, we are going to group this group again, this time moving the pivot over to the side of our water droplet. This means we just need to rotate this group to animate the water droplet in a nice arc. Name both of these groups so they are easy to select.

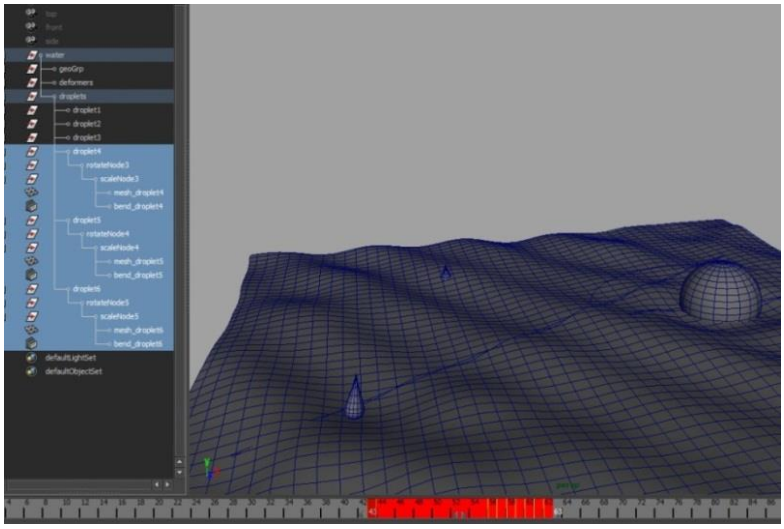


Select the rotate group from the outliner and animate it over the course of about *15* frames. If you have placed the pivot correctly, this should result in a nice curve out and back into the water. You can also animate the bend modifier, you want to adjust the curvature from *0* to about *-0.4* and back to *0* again during the water droplet's time in the air.

To duplicate the animated water droplet, whilst keeping the current bend and animation settings we want to use the *Duplicate Special* tool and ensure that *Duplicate Input Graph* is selected.



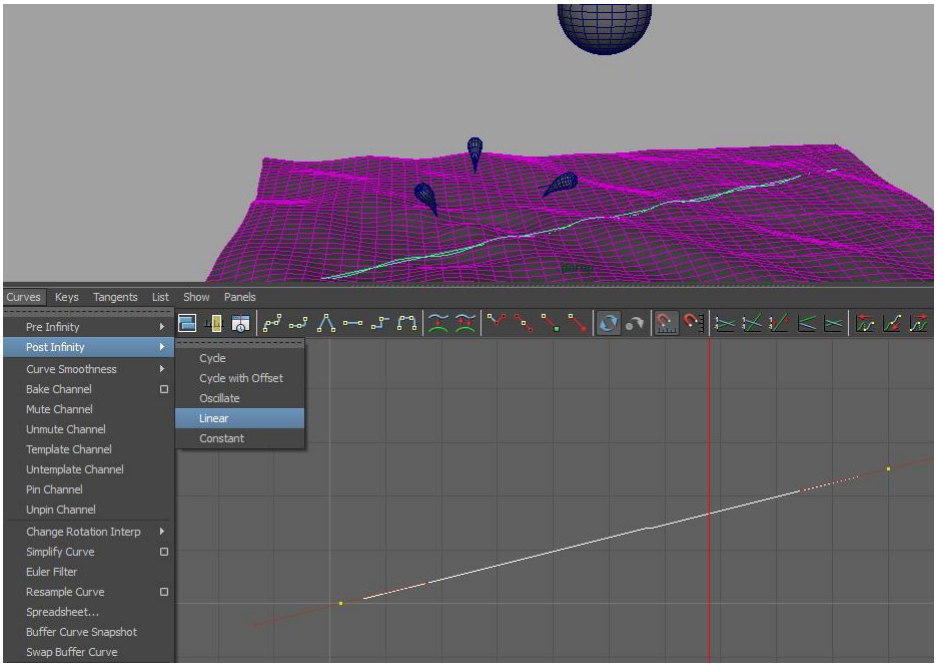
If you so wish, you can change the timing of the duplicates



We want to duplicate the droplets again and move them over to the splash as the ball later hits the water. Duplicate Special again and with all of the nodes selected, move the key frames along the timeline.

Step 6: Final Touches

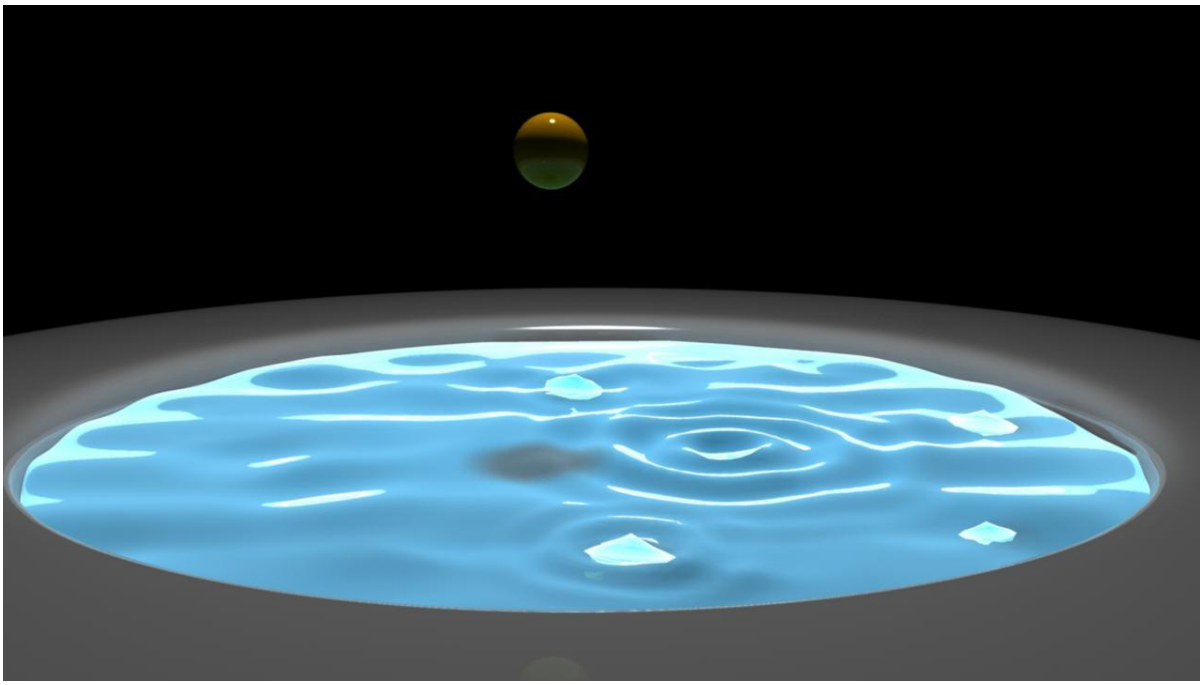
Now we are very nearly done. Our next step is to make the waves across the water animate indefinitely. Select the sine deformer and set the *offset* from about *0* to *0.5* over the space of about *50* frames. Now we want to visit the graph editor again, similar to how we manipulated the flag on page 170. Select the curve and go to *Curves > Post Infinity > Linear* so that the sine wave remains animated indefinitely.



This concludes our water tutorial, to further improve the look of the animation, you could also add secondary small wave deformers to be set off when the water droplets hit the surface.

All you need to do to complete the project, is to assign a water material, such as the mia_material_x preset ‘water’ on page 96 in the materials section of the book.

Once complete you should get something that looks like this:



CHAPTER 8: RENDERING

Chapter 8: Rendering

Introduction

Rendering is a huge part of your project. This is where Maya calculates the lighting, texturing and geometry of the scene and processes it into a final image. The most difficult part of this process is usually finding the right balance between image quality and render time.

Camera Basics

Focal Length/Angle of View

The focal length is the same as the equivalent of a 35mm camera lens. It is also directly proportional to and angle of view, which describes the angle range that the camera can see.

Film Backs

The film controls the size and distance from the lens that the film back is calculated. It is very unlikely that you will need to adjust this.

Film / Resolution Gates

This allows you to preview the framing and composition of your scene.

To enable your resolution gate, navigate to the desired view panel select *View > Camera Settings > Resolution Gate*. A rectangular shape will appear in your view; this illustrates what the camera will capture within your scene.

Also within the camera settings menu, is the *Safe Action* and *Safe Title* toggle boxes. These can be used to make sure that the action of your animation remains within the safe action box, and that any text falls within the safe text box. These rules were original put in place because some CRT TV's can cut the extremities off. As these are replaced with digital TV's, the problem is slowing disappearing, however, it is still good practice.

Chapter 8: Rendering

Positioning of a camera

Alt + LMB – Rotates the camera.

Alt + RMB – Dolly In and out (not zoom as it physically moves the camera).

Alt + MMB –Panning, horizontal or vertical depending on mouse movement.

Dolly vs. Zoom

As the simplified explanation above implies; the difference between the two is dolly physically moves the camera whilst zoom only changes the cameras focal length.

How each of them should be used

Dolly – During modelling phase and the general positioning of cameras. This is the effect of viewing an item through a human eye.

Zoom – When you want perspective to remain constant but get closer to an object

Together – These two effects can be used in conjunction with one another to create what is known in the industry as the vertigo effect. You can make some objects appear to fade into the distance whilst others appear to be getting closer to the camera. This effect is achieved by dollying out whilst zooming in or vice versa.

Creating a New Camera

Navigate to *Create > Camera* or whilst in any of the viewpoints select *Panels >Perspective> New*

Image Formats

Deciding on what format to choose for your renders can be confusing especially if you are unaware of the differences or even the advantages or disadvantages of some of these well knows industry standards.

Underneath each varied format we will provide details of a relevant 720 x 576 file from an identically rendered scene within Maya to help demonstrate one possible deciding factor.

AI (.ai)

Adobe Illustrator file format requires it to be output from Maya's Vector renderer as it will convert all aspects within a scene to editable spline paths to be later incorporated into flash editing programs.

AVI (.avi)

Windows Media movie format. This option may initially seem convenient, however it will often not fit the requirements for many final renders but can serve as a quick test to get the general feel of a specific scene. They do not give your very much manoeuvrability in the way of the final composite due to the fact its composited automatically thus not allowing for any type of render layers. They can also be problematic on occasions causing you to lose batch renders due to file corruption should the render become interrupted or seizes for any reason during this pivotal stage. In summary good for tests renders, but do not use for final piece.

IFF (.iff)

A photo format created and developed Alias. The format is not widely recognised within the industry but cam ultimately be used within Maya (using FCheck) and in packages such as Adobe After Effects which we will use later for compositing.

JPEG (.tif)

Joint Photographic Experts Group. Possibly the most renowned image format used in today's media due to its file sizes and usability. Having small file sizes however doesn’t come without its downsides, JPEG's can appear quite lossy and by default within Maya are set at 75% quality.

PSD / PSD Layered (.psd)

As most of us are aware this is Adobe Photoshop's standard image formatting. If a PSD Layered option is required this places the scenes background on an un-editable layer whilst all other objects are on their individual layers with a transparent background.

SWF (.swf)

As with the Illustrator format, this also requires Maya Vector to be the rendering output. SWF files have the ability to contain multiple images within one file.

TARGA (.tga)

This image format was developed by a company known as Truevision at some point during the 1980's. They are a widely renowned imaging format with the key advantage of being able to hold an alpha channel. Unfortunately due to its lossless nature and its ability to hold additional information file sizes can be considerably large.

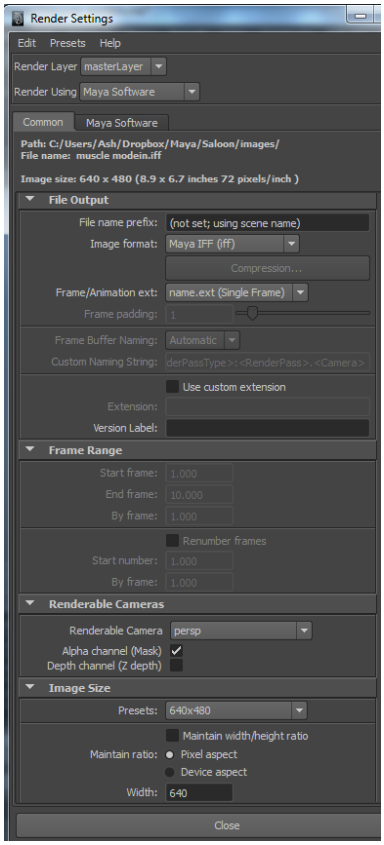
TIFF (.tif)

Tagged Image File Format. This file format is not dissimilar to the above TARGA's as they are also able to hold an alpha channel. The key difference is their ability to be compressed by various schemes but can cause problems with program capability if a certain scheme is not accessible on another machine.

Render Settings

Common Settings

The following diagram explains the Common pane in the Render Settings window:



The name of each image in your sequence. The image format you save your file sequence as. This controls whether you are rendering a single frame or a sequence and the naming conventions that go alongside it.

This sets the frame range that is going to be rendered. You can re-number the files for example if you are rendering negative frames.

Specify the camera you want to render from and whether to render from the alpha/depth channel.

The larger the image the longer it will take but the results will often be better. For sequences we advise sticking to conventional formats such as HD 720.

Playblast

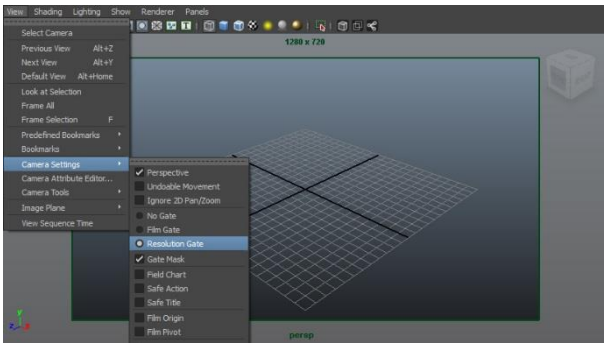
This is a render of your scene but without considering contributing factors such as lights and textures, it simply renders the scene as you see it within the workspace and is used to quickly ascertain a general feel for your animation.

- Click the RMB on the time slider and select *Playblast* □
- Within the option box set *Edit > Reset Settings*
- Leave settings as they are unless you require the file to be saved or you wish to alter screen size etc. (self-explanatory options).
- Then click *Playblast* to preview animation.

Rendering

Film Gate and Resolution gate

The two items are extremely valuable when we are framing our animation (Staging). Essentially they put a frame in your view that shows you how the camera is going to crop your scene when you render your image, regardless of the shape of the actual view pane.



Command line Rendering

Why render from the command line? We asked ourselves the same question the first time we came across this, and wasn't until our final year that we found some uses for it. Before we proceed it's important to know that the command line renderer will use the last saved render settings for that file so be sure they are all set up correctly before commencing this method.

- Windows: Open up the *Command Prompt* (*Start > All Programs > Accessories > Command Prompt*)
- Apple Macs: Open up the Terminal Window (Located within Macs OS X Utilities Folder)
- Navigate to the relevant directory i.e. cd c:\documents\maya\scenes
- Commence the rendering process enter

  render file_scene1.ma

  render file_scene1.ma -s 10 -e 40 (to render frames 10-40)
- Multiple scenes
- To cancel the render simply press Ctrl+c within the relevant window

Chapter 8: Rendering

Maya Software

Alias' own renderer is a very versatile, fast multi-threaded renderer. Because is it built directly into Maya it supports connection to and from every other feature in Maya. Many people will automatically disregard it as being inferior to the likes of Mental Ray (which is now bundled with Maya) and RenderMan, but this is not true.

In fact, when used correctly, Maya software renderer can be used to give feature-film quality results; by Shane Acker's Oscar-nominated film 9 was entirely renderer using Maya Software.

It does have its limitations of course, photorealism is near impossible, but there a few tricks that can be implemented to avoid the overheads of global illumination and final gather but give a similar result, which we will get into a little later.

Maya Hardware

Alias' Hardware renderer uses the power of your graphics card to render frames. Not all specific hardware and software combinations are able to support this, but when it is supported, in particular with the latest generation of Quadro and FireGL workstation graphics cards are used.

When supported, hardware rendering can yield a huge increase in time, but more complicated calculations are not always supported.

One particular use for using Maya's Hardware renderer is when rendering particles. Unless a specific

Mental Ray

Mental Ray, developed by mental images, is very versatile, production quality rendering software that has now been incorporated into Maya Software. Mental Ray's strength lies in its ability to calculate complicated effects using ray-tracing, photon maps and global illumination using multi-processor and multi-computer set-ups.

All of this still takes time however, and especially gaining a photo-realistic effect requires huge amounts of processing power and time. It is important to keep a balance between the image quality and the time it will take to render.

Because mental ray is so versatile, and completely programmable even, we will but skim over the surface, giving you a few simple techniques to take advantage of Mental Ray and incorporate it into your work flow.

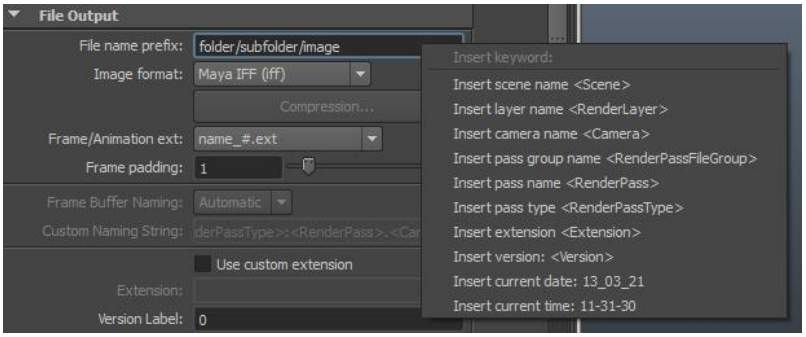
Vector Rendering

Rendering a vector image allows you to increase the scale of the final image without the normal distortion you get from a pixel based image.

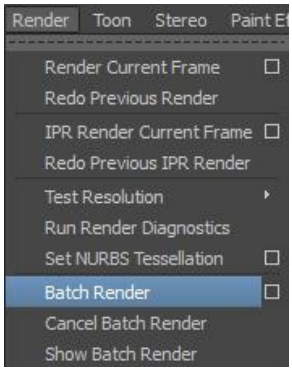
I must admit that in all our time in the industry we have both yet to find a practical application for vector rendering.

Batch Rendering

When you have finished your animation you will need to render it out and you don't want to be rendering it one frame at a time. Maya's batch renderer will use your render settings and project details to output each file in your sequence.



The most important thing is to make sure that you have set your project correctly, by default Maya will render your frames out into the *<project>\images* folder. You can organise multiple scenes easier by also choosing a subfolder. In your *Render Settings* you can add folders to the *File name prefix* by using the forward slash and you can also use wildcards to use the name of the camera or render layer. You can get a list of these keywords by right clicking the *File name prefix* field.



You can access the batch render option from the *Rendering menu set*; *Render > Batch Render*. You can also find the options to cancel the batch render and to show the current image. When the batch render has begun, you will see the progress frame by frame in the command line. It is worth noting that at this stage you can close Maya if you wish, this is especially useful if you have limited memory on your computer.

Before you initiate your batch render, it is worth double checking that the project is set to the correct folder. *Render Settings > Image File Output > Renderable Cameras* (select appropriate camera)

Introduction to Backburner

Backburner is a render manager by Alias. It has the ability to manage and distribute the rendering of a scene across multiple computers. The biggest advantage of this, over setting multiple machines to do specific frame sequences, is the time saved in setting the files rendering. One click will let the software assign tasks to each computer and assign new tasks

when those are completed. A task will also be assigned to a different machine and re-started if the software detects that the render has crashed, or timed out on a specific task.

Caustics

Light patterns produced by focused light, resulting in specular reflections.

Enable Caustics

To enable render caustics select *Render settings > Render Using > Mental Ray (activate)*. Then scroll to the *Mental Ray(tab) > Quality(tab) > Quality Presets > Preview Caustics*.

Note: *Before Caustics will take effect within your scene you have to enable Photon Emissions.*

Photon Emission

Photons must be emitted from at least one of your scenes' light sources in order for any caustic effects to take place. *Hypershade(menu) > Lights > (select desired light) > (open) Attribute Editor > Mental Ray > Caustic and Global Illumination > Emit Photons (On)*.

Global Illumination

These settings control the overall way that photons are treated by Maya. The greater the accuracy, the better quality lighting but it will also take longer for the render to complete.

Raytracing

The reflections and refractions value needs to be equal to (preferably slightly higher) than the number of times light needs to pass through your geometry. By default your tumbler (or bottle) will have two sides therefore a minimum value of 2 is needed, if you are going to fake the liquid contents with geometry (like above), you will need at least 4.

- *Render Settings > Quality > Raytracing (tick)*
    - Reflections = 5
    - Refractions = 5

  Max Trace Depth = 10 (Combination of above values)

Final Gather

Final Gather interprets information from the environment, to recreate both direct and indirect light forms. The indirect lighting effects are the niche of this rendering tool as it takes into account the bouncing effect that light has upon various surfaces.

Scene Setup

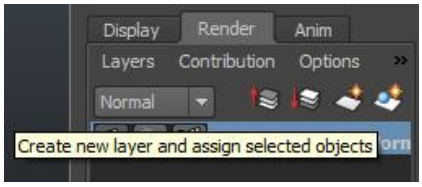
We need to create an environment within the scene to prepare for the final gather rendering.

- *Create >NURBS> Sphere* and stretch over the entire scene.
- *Create > polygon > Plane* this will act as the floor (ensure this is large enough to penetrate the environment sphere.
    - Assign either
        - An Environmental image
        - A lambert
    - Decide if you want you environment showing or not within the render, if not ensure your environment is selected and navigate to *Attribute Editor > Render Stats > Primary Visibility* (off)
    - Next we need to lighten the ambient colour *Attribute Editor > Ambient colour* (Light grey / white)
- *Create > Camera > Camera* and position it accordingly.
- Perform a render and store it for comparison purposes.

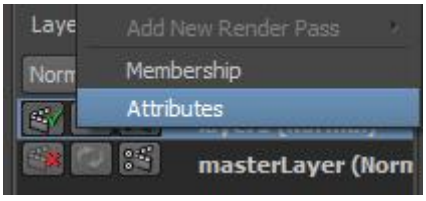
Ambient Occlusion

Ambient Occlusion is a method of easily rendering soft shadows to give a very realistic clay render of your scene. This particularly useful for compositing on top of a separate colour or beauty pass to add greater depth than shadows from the scene's lights alone. The way it works is to take the proximity of a point in space to another and the closer they are together, the darker they both will be.

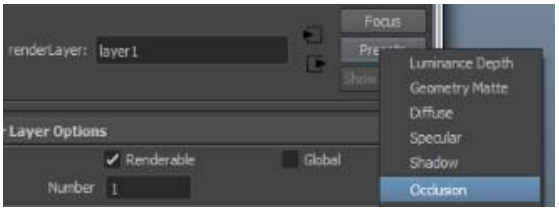
This is a typical result of an ambient occlusion pass:

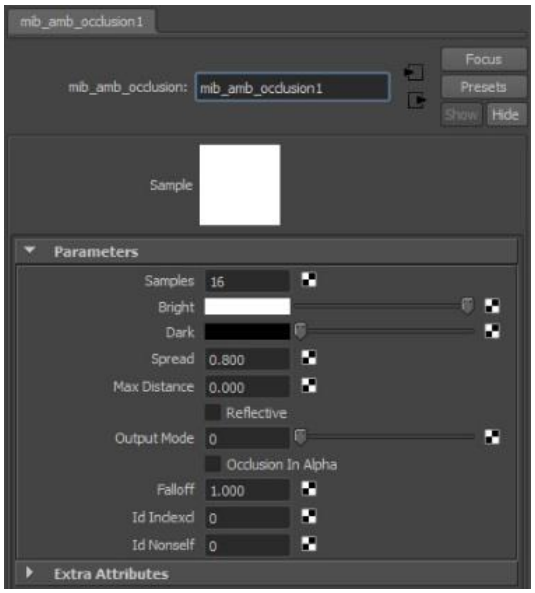


To set this up is simple. In your render layers panel on the bottom right select all of your geometry and click on *Create new layer and assign selected objects*.



A new render layer will be created. *Right-click* this and select *Attributes*, which will load the render node into your attributes editor. Here, if we hold the *Presets* button and select *Occlusion* this will load everything we need into the render node. The Attribute editor will now have our new surface shader loaded, with a node plugged into the *Out Color* channel. Select this node to reach our occlusion settings.





There are two main attributes to note here, the first is the samples, this is simply how detailed the calculations will be, the more samples the cleaner the final image will look but at a cost to render time. The second attribute I the Max Distance. This is how far away an object has to be for it not to be affected by another surface. I usually work with a value about half the total width of my scene.

And there we have it. Now if you render your scene with that layer selected, you will see a nice ambient occlusion pass.

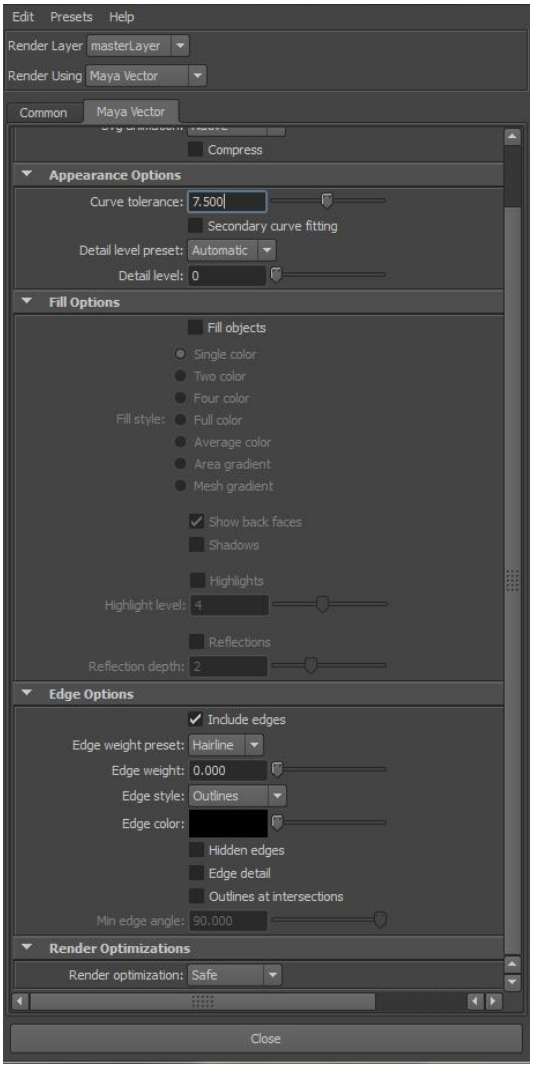
Using the occlusion layer

Render a standard textured image as well as an occlusion layer and import them both into Photoshop.

- Textured Image on bottom layer.
- Place the occlusion image on the above layer.
- Set the blend option of the occlusion to multiply.
- Adjust opacity to suite final effect.

Render a wireframe image

First of all you will need to load the *Vector Render Plugin*, this is located *Window > Settings/Preferences > Plug-in Manager > VectorRender.mll* to load click in the left hand box, if you want it to load on all subsequent start-ups (to skip out this step in the future) tick the right hand box also.

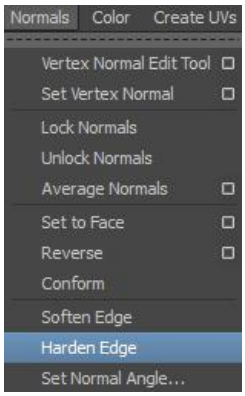


The next step is to adjust the vector render settings to output the wireframe,

Window > Rendering Editors > Render Settings select *Maya Vector* from the drop down list of renderers then navigate to the new *Maya Vector* tab. Scroll down to *Fill Options* and un-tick *Fill Objects*, then down to *Edge Options* and tick *Include Edges*, whilst un-ticking *Outlines at intersections*, the other option between these two can be altered at your discretion, and will alter aspects such as the colour (defaulted to black) and line thickness (defaults to hairline).

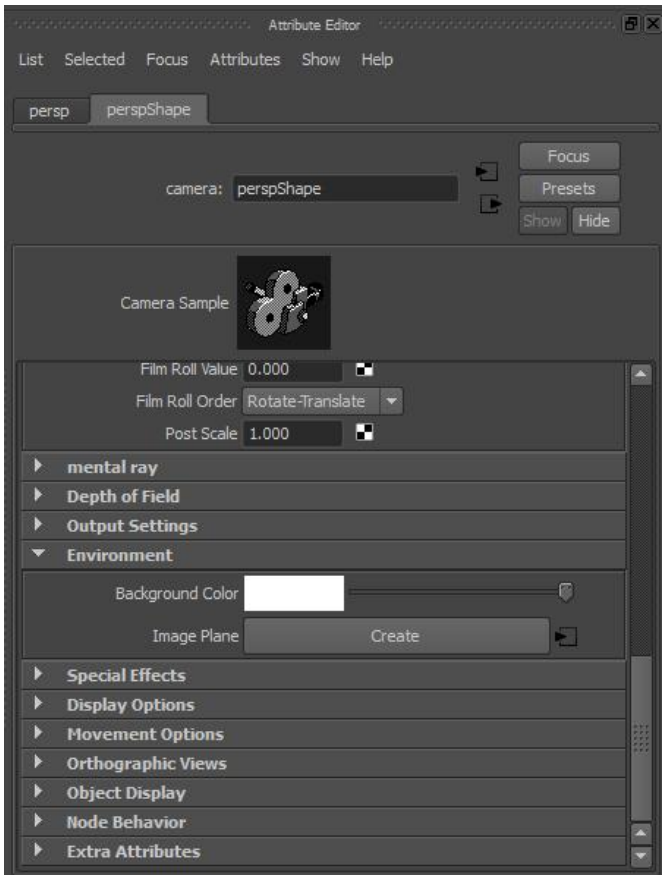
If you were to render at this point you would notice that:

1. Depending on your output colour you might only see a black screen
2. If you could see you would only get the mesh outline.



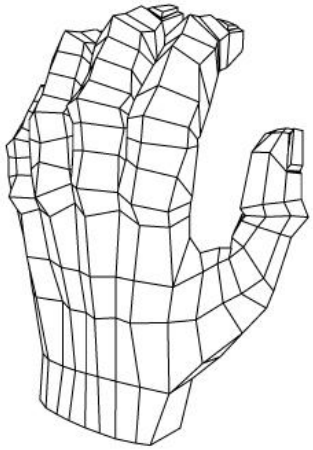
Therefore in order for it to include all of your mesh, you have to select your model and from within the polygons menu set select *Normals > Harden Edge*.

Secondly to alter the colour of your render background go to *View > Select Camera > Attributes Editor > Environment > Background Colour*, an appropriate alteration would be to select white.



If you wish for your image to be rendered as a vector (meaning it can be scaled with no loss of quality) you will have to set up the Render Settings and select *SVG* as your image format, alter your *start /end frame* accordingly then, open the rendering menu set and commence a batch render (*Render > Batch Render*), by default your image will be stored within your scene files under *images*.

Note: SVG files are not useable in Photoshop and will need a graphics program such as Adobe Illustrator to open.



To the left you will see a base mesh wireframe render of the hand previously made within the modelling section of the book.

Note: The best practices for overlaying these types of images to showcase your work is to place the wireframe over your final render and use the *Multiply* blending method in your favourite editing software.

Compositing

Compositing is the process of combining multiple layers into one. This can comprise multiple render layers (for example if we render separately our colour, reflection, occlusion, shadow etc.) or it could be combining our rendered 3d image with a background image such as a photo or video, or often both of these.

The more layers we render out, the more control we have over tweaking the final look of our animation without the need to re-render anything.

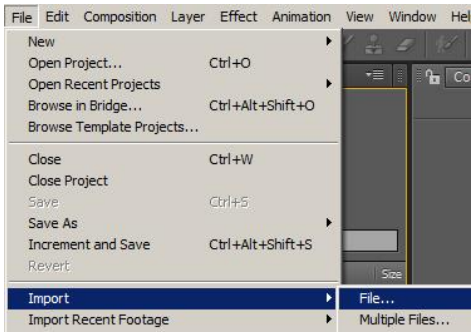
Compositing Software

There are several main compositing programmes around, each with their own advantages but our favourite package is *Adobe After Effects*. Its integration with the Adobe suite in particular *Photoshop* and *Premier* and the ability to read Maya's native **.iff* format make it easy to fit into our workflow.

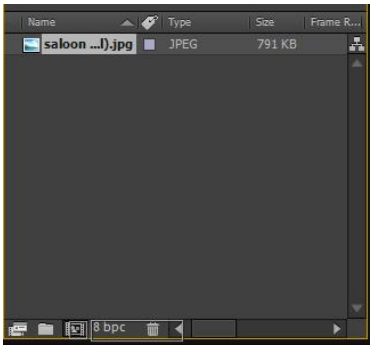
Chapter 8: Rendering

Introduction into Adobe After Effects:

Before you do any work in AE, make sure that everything is set up to the correct standard to the project you are working on – i.e. the default resolution and aspect ratio and frame rate (e.g. 720*586 @ 25fps for widescreen PAL).



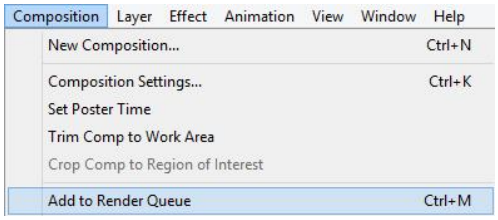
To import footage into AE, go to *File > Import* or double click on the project pane. Make sure that 'image sequence' is ticked, select the first image of the sequence and then click done.



If appropriate, you can change the interpretation of the footage at a later date (*right click > interpret footage*)

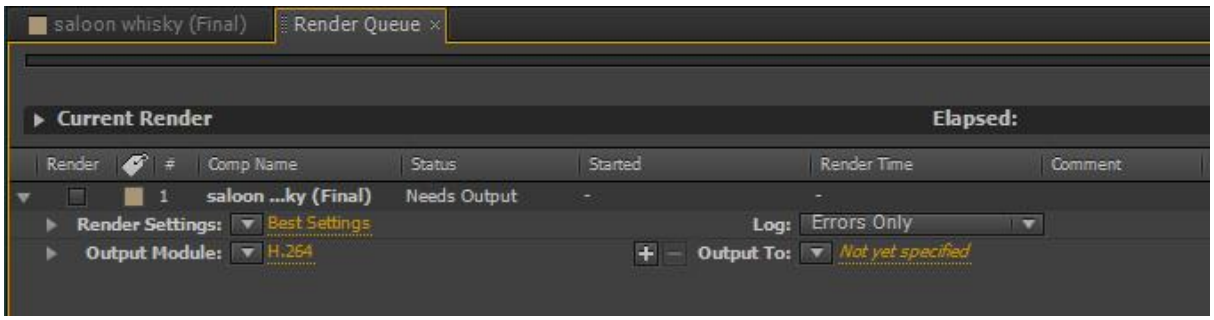
Drag the footage down onto the new composition button to create a new composition taking on the attributes of the footage. If you have other layers to add then you can then drag those into the timeline window.

For an occlusion pass for example, you will want to change the layer style to *multiply* and then adjust the *opacity* according to personal preference.



Finally we need to render our composition into a video file. With the composition you want rendered selected, click *Composition > Add to Render Queue*.

In the render Queue you can choose the format and the file name that you wish to export as.



CHAPTER 9: COMMON ISSUES

AH/Common Issues

Q) My scene is lagging, what can I do?

A) There is often no easy answer to why a scene is lagging as there are a multitude of reasons, to begin with the most commonly checked causes are:

- An issue which is now few and far between relates to computer hardware, thus ensuring your specifications are sufficient.
- Deleting object/scene history (*Edit > Delete by Type > History*).
- Reducing down from smooth to base preview (from **3** to **1**) and Textured to shaded mode (from **6** to **5**).

If none of the above rectifies your issue or if you tried them prior to looking here, take a look at the below:

- Hiding unused real estate by placing them in layers.
- If your scene is becoming increasingly large it may be worthwhile considering referencing (page 44)

Q) I can't undo.

A) Sometime Maya turns off the undo queue if it struggles with an undo task, to enable go to *Window>Settings>Preferences>Preferences>Undo* and turn it back on again.

Q) How can I link the distance tool to an object / camera?

A) Parent a locator with the relevant object / camera.

Q) When I zoom out of my scene I can't see it all unless I pan across

A) This is most probably due to the clipping plane value, *Window outliner > persp > attribute editor > Far clipping plane (increase value)*

Note: the value will have to be increase in all cameras individually if you are having these issues.

Q) Mirroring Geometry – multiple vertices become joined

A) After the merge navigate to *Channels Box > Mirror > Threshold* and alter the value

Q) Why whenever I select a portion of geometry i.e. a face, the corresponding face on the opposite side is selected?

A) Within the *Move Tool* options, *Reflections* is ticked, simply un-tick to resolve.

Chapter 9: Common Issues

Q) What is the purpose of Vertex Face?

A) Not a widely used feature but can come in handy when painting weights.

Q) The faces aren't together when I extrude?

A) The "Keep faces together" Attribute is set to "Off"

Q) I've downloaded a nice texture but now I need to repeat it how do I make it look seamless?

A) Creating a seamless texture requires some editing within a program such as Photoshop. Go to page 118 where we will discuss this topic

Q) How do I create the wheels on a car (or similar) to follow the direction of travel?

A) Please turn to Page 173 where we talk about wheels on a computer chair and how these can be transitioned to alternative models.

Q) The renderer I'm Looking for isn't visible within my render settings?

A) You need to tick the "*loaded*" option in the plug-in manager, navigate to *Window > Settings/Preferences > Plug-in Manager*… (I.e. Mental Ray = Mayatomr.mll, Vector = VectorRender.mll). To avoid this step in future ensure "*Auto Load*" is ticked.

Q) Crashing issues when rendering?

A) Crashing/Locking up when rendering is no easy task to rectify as it can be dependent on many factors, it will often be a case of trial and error but the best way to avoid the issues completely is to adopt best practice methodology which should limit these occurrences. Below are some things to take a look at.

- If using your own textures this can be caused by inadequate UV layouts and/or overlapping. Apply base shader to test and the outcome will determine your next course of action. (Turn to page 81 where we talk about UV layouts)
- If you are using Sub-divisional surfaces try converting to polygons and see if that resolves your issue (sub-divisional surfaces can be very resource intensive)

Q) My glass texture renders black

A) Most commonly cured by increasing the number of Reflections/Refractions in your *Render Settings*. The number should be higher than the amount of sides the light has to pass through.

Q) When I render glass (or any Dielectric Material), sections of them (sometimes random) appear black?

A) If you were to save your render as a .PSD file the black area(s) will most probably be transparent because the renderer has perceived nothing there (illustrated by the empty areas within the alpha). This is attributed to a Trace Depth Issue and can be rectified by increasing the *Refraction Depth* in your *Raytrace Render Settings*.

Q) How can I change the default background colour when rendering?

A) Go to *View > Select Camera > Attributes Editor > Environment > Background Colour*.

Q) How can I render out in wireframe?

A) It involves using the Vector Rendering function within Maya, for full details Turn to page 226 where you will be taken through step by step.

Q) How can I create a wireframe overlay?

A) First you will need to render out a wireframe version of your model 226, then you will need third party software such as Photoshop or After Effects.

Q) How should I go about creating a show reel?

A) After Effects is probably a good start. Spend some time researching other show reels and try to have something that makes yours stand out.

Q) What do employers look for in a show reel?

A) Show reels are specific to job role therefore ensure you have done sufficient research into the role, thoroughly read the job description and what it entails and construct something accordingly. Employers like to see something different from the “Norm” if your applying for an animation role don’t solely include walk cycles, and push and pull actions, take it to the next level.

AH/Using Maya in collaboration with other 3D packages

Migrating over to Maya

Q) I’ve been using Blender and I’m used to the Z axis relating to height and not Y, can this be altered.

A) Yes (*Window > Setting/Preferences > Preferences > Settings > World Co-ordinate System*) however before 3D was around 2D applications used 2 axis X (base) and Y (height), it therefore makes sense that Z would relate to depth. We would recommend making the

transition over as in the industry it could cause issues within models being imported in on the wrong axis.

Importing Issues

Q) Why do my models not import in the same axis

A) See “Migrating over to Maya” (above) in relation to the x,y,z axis

Q) I’ve imported a file from another program (i.e. CAD, 3DS Max) it’s visible in the outliner however I cannot view it within the scene.

A) If it’s in the outliner for all intents and purposes it’s in your scene… somewhere. Select the object in the outliner and press the “f” key to frame the object and it should now be visible. The most common causes of this are that the objects co-ordinates are not *0,0,0,* hence the reason it’s not in the centre of the grid.

Q) I’ve imported a character from another program (i.e. 3DS Max, Poser or Zbrush), when I render my character appears wrinkled, shrivelled or distorted, is this caused by a bump issue, if so are these handled differently within Maya?

A) Maya’s Bump value is set in scene units and defaults to 1, which is almost always too high. Open your bump map within the 2D texture file (page 91), and use the bump depth slider closer to zero until results reflect desired outcome.

CHAPTER X: A STEP FURTHER

MEL Basics:

Mel script is at the heart of everything you do in Maya, in fact, every button in Maya, every key press, is essentially just a shortcut to pre-defined Mel scripts.

Mel is a subject that will be covered in more detail in later books, however, for those of you that are comfortable with a basic level of Mel, or have basic understanding of C++ this should go a long way to help you understand the syntax and start writing your own scripts.

Repeat Along a Curve:

This script will allow you to duplicate any object along a curve, with the ability to edit both the original path, and the angle of the object after creation. The full script is available in the download's section of the DVD.

First we need to plan our script:

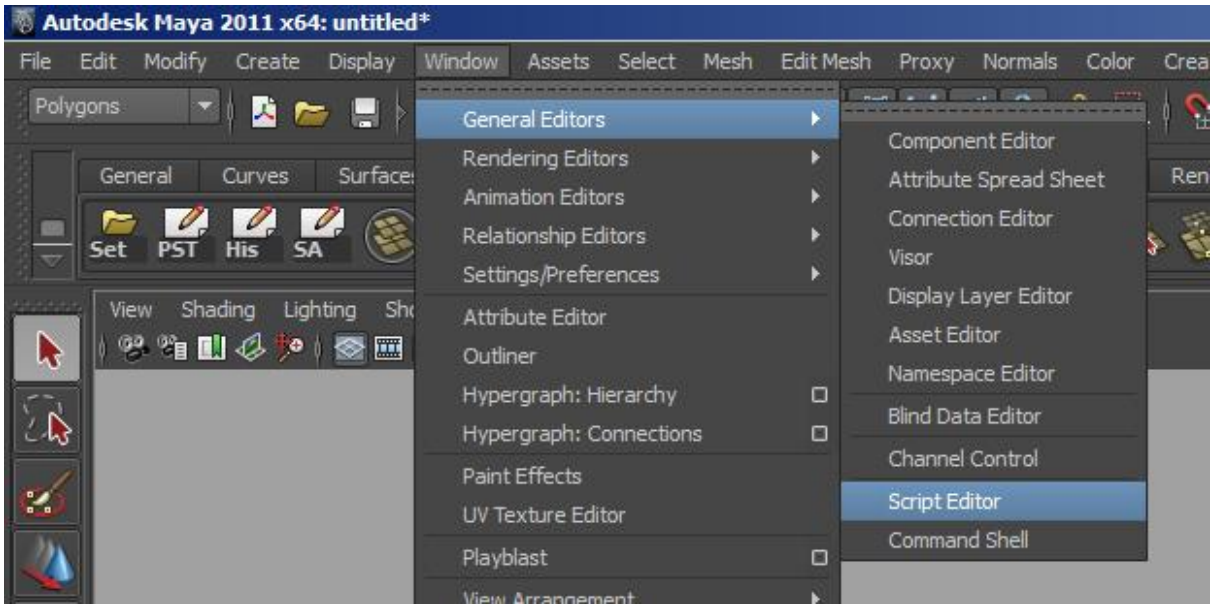
- Define the procedure, taking a value of how many duplicates we want
    - Get a list of the selected objects
    - Assign the curve and original object piece as variables
    - Create a control node and assign any global attributes we will need
        - Global scale
        - Global Bank value
    - Create separate nodes for the curves, locators and duplicate pieces
    - Create an offset curve from the original target curve
    - Create the locators that will act as the up direction targets for our pieces
        - Space equally along curve by disconnecting the uValue and assigning it as one divided by the number of duplicates, multiplied by the duplicate number
        - Assign to an array so that we can easily add them to the locator Group
    - Duplicate pieces along curve equally, using the corresponding locator as the up direction for the motion path.
        - Duplicate as instances to allow global changes to each piece
        - For each motion curve connect the bank value to that on the control node
        - For each duplicate piece set the scale to be the same as that on the control node
        - add each piece to the pieces group
    - parent the locator, curve and pieces groups under the control node
- Create a procedure for the menu

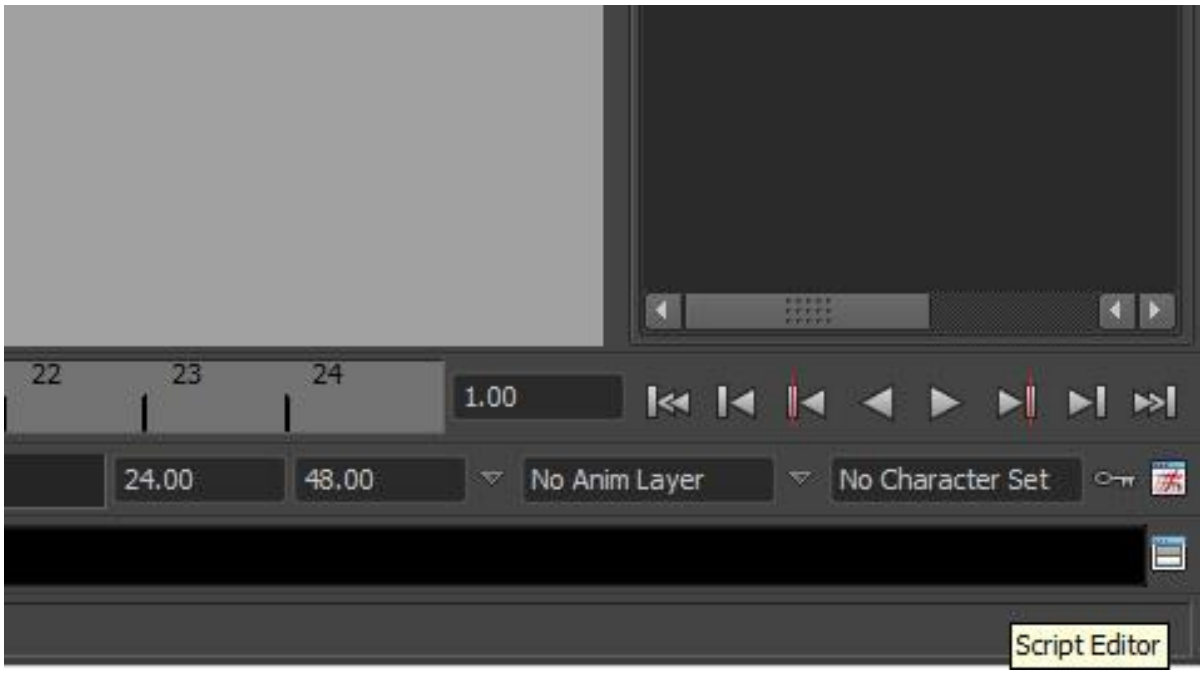
- Must have an editable field that can be used to input the number of duplicate required
- Create a button that launches the main procedure, sending the value that has been inputted by the user

- Launch the menu

Now that we have our plan, we can start writing our script. I won't list and explain every line of code in full, as there are a lot of instances where very similar, if not the same lines have been used, however, I will explain the first instances of codes as we get to them, and where appropriate, the reason for writing it in that way.

Bring up the script editor, go to *Window > General Editors > Script Editor* or click the button on the bottom right of Maya's UI.





```
global proc dupAlongCurve(int $n){
```

- This line tells Maya that you are going to write a procedure, and that it will be global, i.e. you can call the procedure without first sourcing it.
- After the procedure name `dupAlongCurve` there are two brackets. This tells Maya how to interpret any data you are sending to the procedure, in this case it is defining `$n` as an integer, which we will use to control how many duplicated objects we would like.
- Finally, there is an open bracket – this ties up with the closed bracket at the end of the procedure and defines the procedures contents.

```
//Get Data
```

- The `//` is used for annotating your scripts, anything written after this is treated by Maya as a comment.

```
string $sel[] = `ls -sl`;
```

- This defines the variable `$sel` as type string (that is to say, text)
- The `[]` is used to define it as an array (i.e. a list)
- Putting code between two ` symbols (the key to the left of the '1' on a keyboard) tells Maya that it needs to be processed to return a value. In this case for example, we must first get the object names of what has been selected before we can assign it to our variable.
- The `ls` command returns a string array of objects in the scene, and the `-sl` flag restricts that to the objects that are currently selected.
- You will also see a `;` (semi-colon) at the end of the line, this is how Maya knows that it is the end of the line (as you will see later, we can choose to layout the same instruction into multiple lines if we wish).

```
string $targetCurve = `rename $sel[0] tCurve0`;
```

- This code defines the variable `$targetCurve` as a string.
- First the object that is located at `$sel[0]`, that is to say the first object in the list (so the first object you selected) is renamed to tCurve0.
- Then the resulting name of the object (probably tCurve0 is assigned to our new variable.

- The reason we do it this way and do not simply rename the object, and remember what it is called for example, is because sometimes Maya will automatically change it for us, for example, if we have already created a curve called tCurve0 then our rename command will automatically adjust to tCurve1. By telling it to assign the result of the rename process to our variable, we don't need to worry about this.

```
string $piece = `rename $sel[1] piece0`;
```

- Very similar to the above command, this code renames our second selected object and stores it as the variable `$piece`.
- For the purpose of our script the following couple lines of code are not actually required, but I have included them here because they can be very useful in creating a squash-stretch rig. We could also use them to calculate a number of duplicates based on the length of our target curve.

```
string $cInfo = `arclen -ch on $targetCurve`;
```

- This code takes the object stored as the variable `$targetCurve` and creates a Node that works out the length of the curve.

```
float $cLen = `getAttr ($cInfo + ".arcLength")`;
```

- Now that we have created the arclen Node, we need to actually retrieve the value of the length of our curve.
- The `getAttr objectName.attributeName` command retrieves the value of the attribute you are after.
- Anything you put in between speech marks is treated my Maya as text. So by using brackets and then adding text to a string variable we get curveInfoNode.arcLengh.
- The value is stored in the variable `$cLen` as type `float`, which is a number that supports decimal places.

```
string $cNode = `createNode -n controlNode transform`;
```

- This line creates a transform node (essentially an empty group) with the name (indicated by the `-n` flag) `controlNode`.

```
setAttr ($locNode + ".v") 0;
```

- The `setAttr` command defines an attribute with a value, in this case it is setting the Locator Node's visibility to off.
- Note that for a Boolean attribute, you can use either 'on' and 'off' or '1' and '0'.

```
addAttr -ln "rogControls"  -at bool  $cNode;
```

- This line adds a Boolean attribute named `"rogControls"` to the object defined by `$cNode`.

```
string $t□ = `offsetCurve
    -ch on
    -rn false
    -cb 2
    -st true
    -cl false
    -cr 0
    -d 0
    -tol 0.01
    -sd 0
    -ugn false
    $targetCurve`;
```

- This is an example of how you can use more than one line to lay out your code. By laying it out in a more attractive manor, it is easier to follow if you need to return to it and change details. It is also easier for someone else to understand if you are passing it on to an animator.
- The key thing to remember is not to put a semi-colon at the end of the lines; these are used to complete that specific task.
- The `offsetCurve` command duplicates our curve with an offset, we are using the offset command rather than a normal duplication for a few reasons.
    - We can separately modify the offset curve, which we will use to drive the angle of our duplicated pieces
    - By default it will follow the curvature of our target curve so that we can easily manipulate the original curve whilst preserving any modifications that we have made

```
setAttr ($objUpCurve + ".ty") 1;
```

- Because we want our pieces to be upright by default, this line of code sets the translate Y value of the offset curve to be one. This will keep it one unit above the target curve at all times.

```
for ($i = 0; $i < $n; ++$i){
```

- This code is initiating a for loop, that is to say that for as long as the expression within the parentheses is true, keep looping the code within the curly brackets.
- There are three steps to the brackets of a for loop, first we have declared a variable, `$i` as zero. Then we have declared the rule that has to stay true for the loop to run; as long as `$i` is less than `$n` (i.e. `$i` is less than the number of duplicates we want) the code will be executed. Thirdly, we have told Maya to increase `$i` by one on each loop, so every time the loop is run, the number will increase by one until the value defined by `$n` is reached.

```
        if ($i != 0){
```

- Because we cannot divide by zero, this little bit of code ensures that the correct location for the current duplicate is only calculated when `$i` does not equal zero.
- The if statement works by setting a rule within the parenthesis, and if it is true then the code within the curly brackets is run.

```
float $one = 1.000;
```

- This creates a float variable with the value of one at three decimal places.
- There is a bug in Maya that means that if an integer is used within an expression (i.e. one) then the return value will also be an integer, which in the case of a fraction will always be zero. We can get around this by dividing by a variable instead.

```
$pieceLoc = ($i * ($one / $n));
```

- This code calculates the point on the curve that the current duplicate needs to be placed.

```
}
```

- A closed curly brackets closes the if statement.

```
string $t□ = `spaceLocator -p 0 0 0`;
```

- We often use `$t□` as a temporary string array when creating objects that return an array rather than just a string.
- Assigned to this variable is the new space locator created at the centre of our scene.

```
$cLoc = `parent $t[0] $locNode`;
```

- We can then use this temporary variable to return a value for our `$cLoc` variable as we parent it to the `$locNode`.

```
select $cLoc $objUpCurve;
```

- We want to select the current locator we are working with, and then the offset curve so that we can assign it to a motion path.

```
string $motionPath = `pathAnimation
    -fractionMode true
    -follow true
    -followAxis z
    -upAxis y
    -worldUpType "object"
    -worldUpObject $cLoc
    -inverseUp false
    -inverseFront false
    -bank true`;
```

- This attaches the current locator to the offset curve using the `pathAnimation` command

```
disconnectAttr ($motionPath +"_uValue.output") ($motionPath + ".uValue");
```

- Before we can set the location of the locator, we first need to disconnect the motion path's current uValue. This value between zero and one gives the fraction of the curve length that the object will be placed.
- By having a relative location based on the curve length, rather than an absolute distance, we can dramatically change the shape and size of the curves and everything will still be evenly distributed across the entirety of the curve.

```
setAttr ($motionPath + ".uValue") $pieceLoc;
```

- With the value disconnected, we can now set the value to be equal to the value we calculated earlier.

```
string $t[] = `instance $piece`;
```

- We are using the `instance` command rather than `duplicate` so that if we want to change the detail of the geometry at a later stage, all of the instances will be updated live.
- Also note that I have re-declared my temporary array `$t[]`.

```
string $motionPath = `pathAnimation
    -fractionMode true
    -follow true
    -followAxis z
    -upAxis y
    -worldUpType "object"
    -worldUpObject $cLoc
    -inverseUp false
    -inverseFront false
    -bank true`;
```

- This attaches our newly instanced piece to the target curve using the `pathAnimation` command
- The key flags to notice are the `-followAxis` and `-upAxis` which ensure that our instances face the correct direction and the `-worldUpObject` flag, to which we have assigned the `$cLoc` variable (i.e. the corresponding locator that is sat above the instance on the offset curve)

```
connectAttr -f ($cNode + ".objScale") ($cPiece + ".sx");
```

- Similar to the `setAttr` command that assigns a specific value to an attribute, the `connectAttr` command creates a direct link between two attributes.
- In this case, the custom attribute `.objScale` that we created on the control node earlier is going to be linked to (and therefore 'drive') the scale X value for our instanced piece.
- We will of course then connect the same attribute to the scale Y and Z values so that a global scale can be performed across every instanced object from one single control.
- Note that I have used the `-f` flag, this forces the disconnection of any prior connection that has been made to the attribute.

```
}
```

- This curly bracket indicates the end of our for loop.

```
parent $targetCurve $objUpCurve $curvesNode;

parent $piecesNode $curvesNode $locNode $cNode;
```

- These simple lines of code simply clean up our nodes so that they are under a single control node, it is always important to maintain a clean scene, especially when you could be dealing with any number of duplicates.

```
print "Complete";
```

- The `print` command will output whatever text is enclosed within the speech marks.

```
}
```

- This is the final curly bracket to close the procedure.
- We can actually launch the procedure now, select a curve, then an object and in the command line bar type:
    - `dupAlongCurve(n)`
    - where `(n)` is the number of duplicates you would like.

- We are not quite finished yet, however. We still need to build a simple user interface with an input box for the number of duplicates we would like and a button to facilitate launching our procedure.

```
global proc rogMenu(){
```

- Define a procedure for the menu.

```
 source generateChannelMenu.mel;
```

- This code tells Maya to `source` the `generateChannelMenu.mel` file so that we can use the menu creation commands that are located within.

```
 If (`window -q -ex ROGMenu`){
```

- The `-q` flag on the `window` command indicates that you want to query something, and the `-ex` flag asks if something exists.
- Literally, in the eyes of Maya you are asking whether there is a window that already exists with the name `ROGMenu.`

```
 deleteUI ROGMenu;
```

- If the window does exist, then this command will delete it.

```
}
```

```
Window
  -w 100
  -h 100
  -title "Control Panel" ROGMenu;
```

- This command creates a window called `ROGMenu` with a width of 100 and a height of 100.

```
text -l "no. of duplicates:";
```

- We can add text to the window in order to label objects in our user interface, or to give instruction into how the control should be used.
- For example, it is usually worth explaining what needs selecting and in what order before you run the script.

```
floatField -v 10 nFloatField;
```

- A `floatField` is an editable box where numbers (including decimal points) can be entered. The `-v` (value) flag is set to ten by default and has been given the name `nFloatField`.

```
button
  -l "Duplicate Along Curve"
  -ann "Select: curve, then geo, then NURBS."
  -c "dupAlongCurve(`floatField -q -v nFloatField`)";
```

- The create button command.
- The flags are label, annotation and command respectively.
- The `-c` (command) flag controls what code is executed when the button is pressed. In this case, Maya is told to execute the `dupAlongCurve(n)` procedure where `(n)` is the value of the `nFloatField` box.

```
    showWindow ROGMenu;
```

- The last part of the menu procedure is to show the menu window of course

```
}
```

```
rogMenu()
```

- Finally, once our procedures have been declared, we need to execute the menu procedure to launch our menu and give us access to the script.

Maya scripts can be called and executed in a few different ways. My preferred method is to save the script as a .mel file to a safe location, and then save the script to your shelf to give you instant access to it whenever you need it. You can find both the *Save Script* and *Save Script to Shelf* buttons in the File Menu of the Script Editor.

This concludes our scripting section. It is fair to say that this is at a level beyond the rest of the book, but if you have a programming background it should go a long way for you to acquaint yourself with Mel script and its syntax.

The capacity for Mel is virtually limitless and you will find that if you do choose to embrace Mel you can vastly increase your productivity in whichever area of Maya you are working in.

Index

Index

J

K

L

M

N

O

P

R

S

T

U

V

W

REFERENCES

Books:

Kundert, J.G. Larkins, M. Derakhshani, D. Kunzendorf, E. Keller, E. Livny, B. Sousa, M.E.A. Siomacco, E. (2005) Mastering Maya 8.5: Autodesk Maya Press: Sybex

Maestri, G. (2006) Maya at a Glance: Maya Press: Sybex

DVD:

Gnomon Workshop DVD Series (2005-2008) [DVD]

Digital Tutors DVD Series (2006-2010) [DVD]