04 : The Basics of Blender Modeling, part 2

Contributed by Sphynx
Saturday, 10 June 2006
Last Updated Saturday, 10 June 2006

Blending for the faint hearted - Sphynx's guide to Blending

Tutorial 4: The Basics of Blender Modelling, part 2

As we have dealt with so much of the Editing menu (F9) in the previous tutorial, we may as well stay there and complete the two sections labeled [Mesh Tools] and [Mesh Tools 1]. We'll go through these sections quickly, as most of the functions in this area are things that you will use quickly, once, then move on in your modelling. They don't really need too much discussion, but repeat the exercises a few times so that you fully understand what they do, why, and what pre-requisits need to be in place for them to work effectively.

Exercise - Creating a primitiveMuch less hand holding now for simple things - we need a simple primitive to demonstrate on, so clear you scene and add a cube to layer 2, where we'll be working. If you can't remember how to do this, jump back a few tutorials and re-read them. Its a very simple process, and you should really get this in your head before running headlong into what comes next.

Once you have a cube in a 3D view, rotate the view so that it is pretty much 3 / 4 on, as I've been using in my examples images. We'll only do it this way as it is easier to see what some of these tools do when you can see a few different sides of the cube.

By the way, if you are still getting confused about which way is 'up' when you rotate your models here are two tips:

- First, if you can see the camera, it will always have an 'arrow' on top of it pointing up.
- Secondly, before starting to rotate, select a point or a face on your mesh - somewhere that you know where it is (e.g. The top of the cube). When you come to rotate, you'll know where you are based upon where that selected face is going. Ultimately, you won't need to do this - you'll quickly come to realise how the keys work.

The Mesh Tools buttonsThe Mesh Tools buttons cover various ways in which you can multiply the number of faces and vertices in your mesh. We've already taken a look at a big one - Extrude. The Mesh tools can nearly always by triggered by using one of three methods - either by clicking here in the Mesh Tools section of the editing menu; by pressing [Space] to get the main menu and then navigating down into the relevant sub-menu (usually under [Edit]); or by pressing the [w] key to display a context-menu that only deals with 'Specials'.

Gradually, you'll come to use this 'Specials' menu quite a bit. Once you learn the short cuts for these tools, you'll come to realise that there are only so many shortcuts that you can assign to unique keys on the keyboard. Instead, many of the shortcuts for Mesh Tools go through the specials menu - [Sub-Divide] for example, has the keyboard shortcut [w] followed by [1] (you'll see me write this as [w][1] in future). In other words, 'display the specials menu, then select the 1st menu item'.

Image: Mesh Tools
Subdivide and BeautyThe Sub-Divide button should not be confused with sub-surface subdivision, which we looked at in Tutorial 3. Sub-surface sub-division regarded 'virtual' division of a surface for the purposes of rendering. The Sub-Divide button however, physically sub-divides a face(s).

Normally, Blender will subdivide faces by 'quartering' them. The [Beauty] button simply toggles a feature by which the sub-division 'halves' the faces instead.
Exercise - Subdividing a faceJump into [Face select mode] and select one of the faces on the cube. Either select the [Subdivide] button or use [w][1] to sub-divide the face. You will notice that you get many more faces than you bargained for - the other faces split as well. Why? This is because you have added more vertices along the edges of where other faces are joined. Blender likes a nice, cleanly joined model, so splits up the faces that are joined to these vertices as well in order to keep the mesh hanging together properly.

Image: Subdiving a face

Press [u] to undo the sub-divide, then toggle the [Beauty] button and sub-divide again. This time, Blender should divide the face into halves instead of quarters, but don't be surprised if it doesn't - the sub-divide command is a little buggy and sometimes the [Beauty] button has no effect whatsoever. Press [u] again to get back to your cube.

Image: Subdivide smooth

One final form of Sub-division is [Sub-divide Smooth], which for some reason is not in the Mesh Tools section, but can be accessed via the main menu, and the 'Specials' menu by pressing [w] or just [w][3]. Try it, and Blender should ask for a percentage. Just leave it at 100% for now to see what it does - you'll see that when the sub-division takes place, Blender tries to 'smooth' the new faces into the rest of the model as best as possible. It's not perfect, but you can see that Blender has tried to make the end of the cube into a sort of 'dome'. Press [u] to get back to your cube.

Fract Subd (Fractal Subdivide)Fractal sub-divide does exactly the same as the normal sub-divide, but adds a random offset to the positioning of the vertices. This can be useful if you are trying to create a more 'organic' mesh, such as ground or a cliff-face.
Exercise - Fractal SubdivisionSelect one face of your cube and trigger a Fractal Subdivision. Take a look at what Blender has done in the division of the face, and study how this varies from the normal sub-division that you looked at earlier. Press [u] to return to your standard cube.
Noise, Hash and XsortWe'll leave the Noise, Hash and Xsort functions as we really need to set up a more complex mesh in order to see their effects. With Hash and Xsort, it is probably debatable that you will ever see their effects in the modeller at all - these two functions effectively change the order in which the vertices are stored within the data block. Why? Well, with some functions and material calculations, Blender will go through the vertextes in a nice, clean order so that certain calculations can be made. This is not always the required method however, so Hash causes the sequence data to be randomised, while Xsort will sort the data in the X direction. Don't worry if this is over your head for now - they have a very specific use anyway, and you won't use then that often.

The [Noise] function specifically uses the co-ordinates of the vertices in the mesh for any materials that you attach. We'll look at this further once we start adding materials and textures.
ToSphere[ToSphere] is often badly explained. The basic purpose of the [ToSphere] function is to morph a selected mesh into the shape of a sphere, but to do this it manipulates only the vertices that already exist in the mesh. No new vertices or faces are created - [ToSphere] works only with what already exists in the mesh.

The implications of this are simple: On irregular objects, all of the vertices are moved to the surface of a sphere usually based on the average distance between the closest vertex to the origin, and the farthest vertex from the origin. On some regular objects however, such as a cube, all of the points are already equi-distant from the origin - using [ToSphere] on a cube therefore, will see absolutely no physical change whatsoever.

Exercise - Spherising the MonkeyAs a cube will not demonstrate this feature, create a new Blend file ([File] > [New]) and add the Monkey primitive. Once added, the Monkey will be selected in Edit Mode, so all of its vertices will be selected by default.

Image: Pre-spherized

Image: Spherized

With the Monkey selected, choose [ToSphere]. You will still be able to roughly tell that we are looking at the Monkey, as there is a sufficient pattern created by the individual faces to maintain the overall appearance, but all of the vertices will now have been moved onto the surface of a sphere. Switch to Object Mode using [Tab] and press [z] to enter solid mode, and this will become clearer - though you will still see the rough outline of the monkey features.

Keep the Monkey mesh as it is - we'll use it again in the next exercise.

SmoothThe Smooth function is pretty obvious - it takes the vertices and faces selected and tries to physically 'smooth' the surface that they form. This is not a virtual smooth, like [Set Smooth] - this really does move the faces and vertices. Again, demonstrating this with a cube is pretty pointless - the [Smooth] function will work on a cube, but smooth all of the sides of a cube together and all you get is another, smaller cube. Keep smoothing the whole cube and all that you will see is a gradually shrinking cube (though it is useful to smooth parts of a cube individually...)
Exercise - Smoothing the MonkeyWith the monkey from the last exercise, select the [Smooth] (or the [w][0] shortcut) function. You should see the Monkey change it's shape ever so slightly - effectively, the angles between points are rounded off and vertices moved where appropriate to accomodate the smoothing process. Keep doing it, and while the Mokey will get smoother and smoother, it will also shrink in size.

Splitting and SeparatingThere are two functions with similiar purposes, but also with subtle differences. While [Split] is in this button area and [Separate] is not, it seems logical to deal with them together so that we can look at the differences between the two.

If we take a typical cube, it is constructed of eight vertices and six faces. Each face shares a vertex with other faces. We can also look at this from the opposite point of view - each vertex is shared by more than one face. As the vertex is physically the same vertex, Blender can look at all of the faces during, say a [Set Smooth] request, and continue its smooth curve across the vertex as if all of the faces were one and the same - in fact, this is exactly how it is done.

But what if we did not want this to happen? Afterall, we are not living in 'bubble' land, where everything is nice and smooth. Curves come to an end. Sometimes we need a sharpe crease between curves.

To do this, we can split off faces. What implications does this have to the mesh? Well, the most obvious visual implication is that any curve created by a [Set Smooth] operation will stop at the edge that was split. It is also however, important to know how this affects the vertices and faces themselves.
Exercise - Splitting the coneCreate a new Blend file and add a new primitive - this time a Cone, with 24 segments. Enter Object mode using [Tab], select [Set Smooth] and press [z] to go into solid mode. You should now see a nice, smooth cone on the screen.

Image: Splitting a cone (1)

Image: Splitting a cone (2)

Image: Splitting a cone (3)

Hit [z] again to go back to wireframe, then [Tab] to go into edit mode. If anything is selected, press [a] to deselect everything, then press [b] to enter box-selection mode. Your cursor should now change to be the centre of a cross-hair extending to the edges of the window in which you are operating. Go the top corner of one half of the cone, and press the left-mouse button, then drag the mouse to the bottom-right hand corner of the same half of the cone - we want to include in our box all of the faces in this half of the cone. Release the mouse button, and all of the faces within the box should be selected.

Image: Splitting a cone (4)

Now select [Split] or the keyboard shortcut [y]. Press [Tab] to go back to object mode, and press [z] to go back to solid drawing. See the difference? You should now have a clean crease or seam down the two halves of the cone.

So what has actually happened? Before the Split, the cone was constructed of 26 vertices (24 around the base, 1 in the centre of the base, and one at the top) and 48 faces (24 in the base and 24 forming the cone). After the split, we still have 48 faces, but our vertex count has gone up by 4 points.

Why? This is because the vertices that formed the edges of the faces that we separated are no longer 'shared' between the two sets of faces. At the border between the two halves, the shared vertices have been duplicated, and one set of vertices are now owned by each of the two halves of the cone. We can't physically see the new points, as they both occupy the same points in space.

Exercise - Proving new vertices existGo into edit mode using [Tab] then into [Vertex Selection Mode]. If anything is selected, press [a] to deselect everything. Now take a look at the top left of the main Blender title bar. It will say something like 'www.Blender.org 236' and then a series of numbers (the 236 by the way, is the revision number of the Blender application that you are using). What we want to look at here however, is 'Ve:0-30'. This is the number of vertices currently selected (0) out of the total number of vertices currently displayed in this mesh (30).

Image: The point and face counts

Randomly choose one of the vertices on one half of the cone (on the base - deliberately choose a vertex that you know is in the middle of the faces and not one on the border between the two halves). Once it is selected, the title bar should change to 'Ve:1-30'. Press [a] to deselect everything again.

Now go into box-selection mode again by pressing [b] and box-select the point of the cone. You should see the title bar change to 'Ve:2-30', and not 'Ve:1-30'. Why? This is because the cone has been split into two halves, and each half has its own copy of the pinnacle point - they just happen to occupy the same point in space, so you can't visually see the division.

Just for completeness, lets cover some more things that lead on logically from what we've just done. We just used [b] to enter box-selection mode. As we mentioned in a previous tutorial, this is one of those shortcuts that has a multi-use. If you instead press [b] twice, Blender switches first into box-selection mode (displaying a cross-hair cursor), and then into brush-selection mode denoted by a circle around the mouse pointer. The brush-selection mode allows you to 'paint' the vertices or faces that you want to select. Just press [Escape] to turn it off.

Finally, up in the title bar you will also see a section displaying 'Fa:0-48'. Logically, this is the number of faces currently selected, and available for selection. Next to this is something like 'Mem:3.10M', denoting how much memory is being used by Blender for the current operation, data blocks etc. Typically this will grow as you increase the size of your file, but will also jump tremporarily when you are rendering. Finally, on the end, you will have the name of the object that is currently being addressed by all of this information.

One last feature to look at before taking a look at [Separate]. If you are not in edit mode, switch across using [Tab], go into [Face select mode] and deselect everything using [a]. Now select one face only (any face) in one side of the cone. It is possible to select 'more' or 'less' of the faces surrounding a currently selected face using the [Space] bar to bring up the main menu, then the menu items [Select] > [More] or [Select] > [Less]. It is of course much easier to simply use the keyboard shortcuts [ctrl+numpad +] and [ctrl+numpad -]. Hit [ctrl+numpad +] now and more of the faces around the one that you selected are added to the select.

Repeat the keystroke a few times. It selected quite a few faces, then stops. It does not select the whole cone. Why? Well, normally it would keep going to the entire mesh was selected, but the [More] and [Less] selection methods will only keep working within the current distinct group of faces. Because we split the cone in half, and we only selected a face in one half of the cone, Blender is forced to stop when it gets to the split edges.

OK, so what is [Separate], and how does it differ from [Split]? Its very subtle. Create a new blender file and add a new Cone primitive - in exactly the same way as we did above. Follow the exercise exactly until you come to the part where you pressed [y] to split the faces. Instead, press [p] and you should get a context-menu asking whether you want to separate what you have selected or all loose parts. Choose [Selected].

Right - the select turned black, but not much else happened, did it? Well, press [Tab] to go back into object mode and take a look at the you mesh that is selected. Now, the whole cone is not selected - only half of the cone. Click on the other half and it gets selected, but the other half-deselects. What is going on?

Image: Splitting a cone (5)

Well, what has happened is that while [y] splits off the faces within a mesh, [p] splits off the faces and then separates them into a totally different mesh. What we have on screen now are two meshes, with two separate origins. Both origins may be currently located in the same place, but that's only because that's where it was when we separated the faces.

Exercise - Proving we have a new meshSelect one of the meshes so that it goes pink, then press [g] to go into grab mode. The mesh will turn white. Move the mouse to another location and drop it by pressing the mouse button. You should now see two physically separated meshes, and two physically separated origins.

Flip NormalsThe [Flip Normals] button, also accessible using the shortcut [w][9] does exactly what it says - flipping the normal of a selected face(s) into the opposite direction and effectively reversing the way the polygon is facing.

Why do this? Well, imagine creating a 'stage' for scene. One way of doing it would be to create a box, maybe removing one or two sides, and then placing all of the 'actors' inside the box. The problem is that the stage may well not be visible as the walls of the box are all facing outwards. To correct this, we could select the walls and flip their normals - changing the walls to face inwards instead.

Two quick ways of ensuring the all of the normals of a particular object are facing in the same direct are using [Recalculate Outside] (shortcut [ctrl+n]) and [Recalculate Inside] (shortcut [ctrl+shift+n]). Blender will go through the entire mesh and order all of the normals to be pointing in the same direction. These last two should be used cautiously however, as what Blender regards as 'inside' and 'outside' can be debatable depending upon the logic being used and the shape of the mesh. You may have some tweaking to do with [Flip Normals] to get it perfect.

If you want to see exactly where the normal is facing, turn on [Draw Normals] in the [Mesh Tools 1] section.
Rem Double and Limit (Remove Doubles)[Remove Doubles] should be used cautiously as it can destroy hours of fine tweaking of a mesh - trust me, I've been there. In itself, [Remove Doubles] is a very important function that will go though

your mesh and find vertices and faces that share the same point in space (the Limit button sets the limiting distance that tells Blender what 'the same point in space' actually means). Once it finds vertices that 'share' the same point, it merges them into one vertex.

The use of this function is particularly important when meshes are imported from another application - some conversions of models can leave vertices and faces in strange mathematical quandries, so much so that when you start to use them the application can crash. Using [Remove Doubles] first, can remove a huge number of these problems.

The reason that it can destroy hours of work however, is when you are not totally aware of what it is doing. Take our Cone exercise for example, where we split the mesh in two. Use [Remove Doubles] on that mesh, and the two distinct copies of vertices that form our crease down the side would be merged back into one set. As a result, the two halves would reform and the crease would disappear. If you have a model with thousands of such detailed creases in place, putting them back again can be a nightmare. As a result, seriously consider saving your work before performing a [Remove Doubles], just in case...

A closely connected function to [Remove Doubles] however, is [Merge Vertex], which merges selected vertices into a single point. To use [Merge], select two or more points to be merged then select [Edit] > [Vertices] > [Merge] or just [alt+m]. Blender displays a context menu asking whether the points should be merged at the centre (i.e. the average position) of the selected vertxes, or at the current location of the cursor.
Exercise - Merging pointsCreate a new Blender file, and add a cube primitive. Deselect everything using [a], then select two points in the same face. Now press [alt+m] and select [At Centre]. Blender will tell you how many points were removed (nearly always 1 less than the number selected) and the screen display will updated.

If you had wanted to merge the two points into one at the current location of one of those points, the easiest method of doing this is to select the appropriate point then use [shift+s] to display the [Snap] menu, and select [Cursor -> Selection]. When you perform the merge, as described above, you would select [At cursor] to merge into the selected location.

Image: Merging points

Notice in this example however, that we can now see one of the bad things about quads (four sided faces). Two of the faces are now 'bent'. They are made up of four vertices, but they are now no longer in the same plane. If we really wanted to keep our cube like this, we should really convert those 'bent' (or more accurately 'non-planar') faces into triangles.

The easiest way to do this, is to jump into [Face select mode] and select the two offending faces. Next, press [ctrl+t] or choose [Edit] > [Faces] > [Convert to triangles] from the main menu under [Space]. You will see how the faces get converted into triangle shapes. The reverse is to select two (or more, Blender can intelligently make the most appropriate decisions on this) and press [alt-j] to convert triangular faces into quads.
Spinning and ScrewingThe Spinning and Screwing tools are all connected with a rotation of some sort, coupled with a duplication of selected meshes, faces or edges as the rotation progresses. You may hear these tools being called other things in other applications, such as 'lathe' in Lightwave, but essentially they all rely on three important values to do their work.

Degrees

The [Degrees] button sets the number of degrees around which the operation will turn. For some operations, the [Degrees] setting is used in opposition to the [Turns] setting and vice-versa.

Steps

While an operation is performed around a certain number of degrees or turns, Blender still needs to know how many individual steps to break that rotation into. An operation around 90 degress for example, specifying 9 steps, would result in clean breaks every 10 degrees. For a quick idea of what this means before we look at the operations themselves, think of a 36 sided cylinder. You could think of this as being a rotation of 360 degrees, with 36 steps - each face occupying 10 degrees.

Turns

Some operations only operate on a full 360 degree rotation, so specifying a number of degrees is pointless. We can often however, specify how many rotations are to be made. This is what we do with the [Turns] button.

Keep Original and Clockwise

The [Keep Original] toggle button simply specifies whether the mesh, face or edge upon which we are operating should be considered part of the final result, or whether it should remain where it is and the final result is a new, separate mesh based on a copy of the original.

The [Clockwise] button simply specifies the direction of rotation when an operation is being performed. This is also a toggle button and is selected by default - so any operation will rotate clockwise around the pivot point. Logically, if this button is turned off the rotation occurs counter- or anti-clockwise.
ScrewOK, onto the tools themselves. The [Screw] operation performs a repetative spin on a mesh or set or edges, usually moving downwards as the rotation progresses. Effectively, we can use this tool to create a 'screw' - hence the tools name.

There are three things to remember. First, the [Screw] function only operates on complete cycles of 360 degrees - we therefore use the [Steps] and [Turns] values. Next, the centre of the screw revolution is centred on the position of the cursor, so make sure that you place it in the correct location for the centre of the screw. Finally, the cross-section of the shape that you want to use in building the screw is important. The top-most vertex of the cross-section is always going to joined to the bottom-most vertex after one rotation - as long as you keep this in mind, you can create some really great shapes and control the descent and shrinkage of the screw very precisely.

Exercise - Making a screwThere are several ways of making a cross-section to use for this exercise - but we'll just use a simple square plane and delete a few edges. First, we need to be in the front 3D view, so start a new Blender file and if you only have one 3D view in use, press [numpad 1] to get into this view.

Next, add a new mesh primitive - in this case a simple plane. You'll just get a square shape in the centre of the 3D view. Zoom in so that you can see it better. Now, lets use a selection method that we've not used before - [Edge Select mode]. Get into this mode by pressing the appropriate toolbar icon, then select the top and right most edges (remember that you'll need to use [shift] to select them both. Press [Delete] and select [Edges] from the context-menu that is displayed. We should now have an 'L' shaped edge on screen.

Image: Making a screw (1)

Image: Making a screw (2)

Image: Making a screw (3)

Switch across to [Vertex Select Mode] and select each vertex in turn, press [g] to go into grab mode, and drag them out to roughly the shape shown in the image.

Finally, jump back into [Edge select mode] and press [a] to select the entire shape - this is important, as the [Screw] tool needs to operate on a curve (a series of interlinked edges in Blender). Make sure that you have 9 [Steps] and increase the [Turns] value to something like 4, then select [Screw].

Image: Making a screw (4)

If you've selected the button marked [Screw] instead of a keyboard shortcut, the cursor may change to a question mark and ask you to select the 3D view that we are operating in - this will only happen if you more than one 3D view on screen. Select the correct one, and Voila - the Blender screw tool.

SpinThe [Spin] tool is what other applications sometimes call 'lathe'. It effectively takes a cross section and 'spins' it around a central pivot-point, joining each segment as it rotates. It is this tool therefore, that can be used to create everything from tyres to wine-glasses, and rocket-bodies to gas-cylinders.

Exercise - Spinning a wine-glassCreate a new Blender file, and get into front view using [numpad 1]. Add a new primitive - we want lots of vertices to play with, so create a circle with 12 vertices then zoom in so that you can see them. As before, select each vertex, press [g] and drag them into roughly the shape that you can see below. If you can't remember how to get those points perfectly aligned in the centre of the view, remember the [n] key to bring up the 'Transform Properties' dialog and that x=0 is the perfect centre.

Image: Spinning a wine glass (1)

Image: Spinning a wine glass (2)

Once you have the shape, switch to top view using [numpad 7] - remember that when we are spinning, we are spinning around the pivot point so we want to be able to place our cursor down the centre of the wine-class. You probably have not moved it however, so leave it in the centre of the 3D view. We still however, need to be in the top-view for this operation.

Change the degrees to 360 (the maximum that you can), and increase the steps to something like 10. Finally, make sure that you are in Edit mode and press [a] to select the entire mesh ([Spin] needs to know what vertices you want to spin - it won't just work on the whole mesh), then select the [Spin] button. As with the [Screw] tool, if you have more than one 3D view on screen, the cursor will change to a question mark - just click in the top-view to tell [Spin] where to operate. The cross-section should now be spun into a wine-glass, but were not finished yet - there are still one or two things to do.

Image: Spinning a wine glass (3)

Image: spinning a wine glass (4)

There are two things that we really need to do. Normally we'd do them automatically and immediately, but for this first time lets modify what we are looking at so we can see where the problems lie. Use your rotation keys to rotate the 3D view into a 3 / 4 view so that you can see the entire glass. Switch to object mode and make sure that the glass is selected, then select [Set Smooth]. Next, press [z] to display solid, then turn on [SubSurf] and take the [SubDiv] value up to 2 or 3. You may (not guaranteed) see two things - first, there seems to be a sharp edge down one side and secondly, there seems to be some strange ridges around the glass.

Image: Spinning a wine glass (5)

Image: Spinning a wine glass (6)

First, while the [Spin] tool spins the cross section around, it does not join up the two ends. If we were to render this wine-glass therefore, we'd get a nice glass but a clearly obvious seam down the side. The start and end vertices should be so close however, that we should be able to correct this by just selecting [Remove Doubles]. Do that now, and that sharp edge should disappear.

Secondly - those ridges. If you have them, its probably because some of the faces have their normals pointing in the wrong direction. This is typical in this type of operation and pretty simple to cure - just jump into edit mode and choose [Recalculate Outside] from the [Edit] > [Normals] menu (or just press [ctrl+n]).

And that's that - one wine glass.

SpinDup (Spin Duplication)[SpinDup] performs a similar action to [Spin], but instead of taking a cross section made of edges and lathing around a centre pivot point, [SpinDup] takes entire sections of a mesh and duplicates them at the centre points of the steps selected. One important thing to remember here however, is that we are not duplicating (i.e. copying) objects. [SpinDup] is designed to duplicate parts of a mesh but still within the same mesh - in other words, all of the duplicated parts still only have one origin.

Let's try an example to explain that, shall we?

Exercise - Spin duplicating a cube-meshStart a new Blender file, go into top view ([numpad 7]) and create a simple cube primitive. We should be in edit mode already with all of the vertices selected, so just hit [g] and drag the cube off to the side and drop it. Notice that the origin has not changed - we've moved the position of the vertices, and not the object position itself.

Image: Spin-duplicating a cube (1)

Image: Spin-duplicating a cube (2)

Next, make sure that the [Degrees] is set to 360, and select say, 9 [Steps]. Next, just select [SpinDup] - if the question-mark cursor appears, just select top-view. You should see eight new copies of the cube appear, but all within the same mesh. If you press [Tab] to go into object mode, all of the cubes are formed around the same origin and can be moved and operated on as a single object.

Extrude and Extrude DupWe've already seen what [Extrude] does in a previous tutorial - the only thing to add is that if the button is used to trigger an extrude rather than the shortcut [e] within a 3D view, Blender will need to know which 3D view to operate on. In this case, a question-mark cursor may appear. Just choose the correct 3D view that you are working with.

As [SpinDup] is to [Spin], [Extrude Dup] is to [Extrude]. It is however, a little more restrictive in its operations - it will only extrude 'away' into whichever 3D view is being used at that moment in time, using the button marked [Offset] to control the spacing of each duplicate, and [Steps] to control how many duplicates are created. Again, this is probably better explained using an exercise.

Exercise - Extrude duplicating a cube-meshCreate a new Blender file and switch to a top view using [numpad 7]. Create a new mesh primitive - a simple cube. Blender should put us into edit mode with all of the vertices selected, so this is fine - it's what we want.

Change the [Steps] to 3, and change the [Offset] to 3. The cube that was created has co-ordinates all based on the 1 unit grid, so we want an offset suitable to actually see our handywork. Select [Extrude Dup], and if required click in the top-view window.

OK, so nothing changed. Hmm. Actually it did - remember I said that the Extrude Dup only extrudes 'away' from the current 3D view? What that means is that all of our handywork is hidden by the same object. Switch to front-view using [numpad 1] or use the rotation keys to rotate around the object to see the results of the [Extrude Dup] function.

The Mesh Tools 1 buttonsThe [Mesh Tools 1] section provides a few more functions for editing our meshes. We've dealt with some of them already - such as the toggle buttons that allow us to turn on and off various visual guides such as the face normals. Let's just tidy up some of the ones that we have not yet looked at.

Image: Mesh Tools 1

Centre

The [Centre] button operates on the entire mesh when in edit mode - it is basically used to return the origin of the mesh (and therefore the entire mesh itself) back to the centre of our 3D world. The [Centre] button should not be confused with the [Centre] button in the [Meshes] section of this menu screen - that button operates on the origin within the mesh, while this one moves the entire mesh, origin and all.

Hide and Reveal

[Hide] and [Reveal] begin to become very useful once you start to get a more complex model growing in Blender. [Hide] is used to simple visually hide from view any vertices or faces that have been selected. Logically, [Reveal] simply makes any hidden components of your mesh visible again.

The best use of this is when you begin to get a complex mesh on screen and it becomes difficult to make out some sections of your mesh to work on them. Selecting and hiding the parts not needed for your current detailing makes the job much easier.

Select Swap

[Select Swap] simply inverts your current selection - selecting one face of a cube then hit [Select Swap] will result in the rest of the mesh selecting and the original face deselecting - its as simple as that.
A final word...We are running out of space in this tutorial, so we'll put off our basic rendering till Tutorial 5. Seeing as we have been dealing with the basics of modelling however, there is one shortcut key that we can't really miss out - that shortcut is [numpad /].

This shortcut is a toggle-key which switches back and forth between 'global' and 'local' view, and will only work if you have an object selected in one of your 3D views. What does it do? Well, put simply it greatly simplifies the modelling process by removing a lot of the extraneous details that you don't need when working on a particular mesh - namely, all of the other meshes in the scene.

Hitting [numpad /] (or selecting 'Local view' in the [View] menu), centres the specified object and temporarily at least makes it the centre of the entire 3D world. You can rotate the view, zoom in and perform any operations that you want to perform without it affecting the rest of your 3D scene at all. When you are finished, just hit [numpad /] again, and you will be returned to the global 3D view.

Exercise - Visiting the local viewStart an empty Blender scene and create a simple primitive - any primitive will do. Press [Tab] to enter object mode, [g] to start grab mode, and just move it off to one side. Repeat the process three of four times with different primitives - moving them off in different directions, with maybe one or two overlapping.

Now think of this as a really complex scene - perhaps with far more meshes that what you have here. Click on one of them - one only - and hit [numpad /]. The 3D view should change to just show that single mesh - all of the other meshes will disappear. Use your rotation buttons, zoom in and out and see that it works in exactly the same way as the normal 3D view, then press [numpad /] again. Everything returns to normal.

What's in Tutorial 5?Tutorial 5 starts the next obvious step in modelling - adding materials to a simple mesh. At the same time, we'll want to take a better look at those textured meshes, so we'll also build a simple scene and render it using Blender's internal renderer.