INTRODUCTION

Who is this book for

Blender is an amazing program and there are many reasons why someone would want to learn how to use it, but what are the advantages of learning it from a book? When I was choosing the content and deciding on the structure I had to make the following assumptions about you, the reader:

- You are relatively new to Blender, at least in its current form, but you can find your way round the internet and your computer, install programs, and generally manage directories and do file house-keeping.
- You wish to accumulate knowledge and skills in a systematic way rather than wanting to jump straight in to find the answer to solve a specific problem.
- You want to read detailed instructions at your own speed, and you are occasionally frustrated by the, often excellent, online videos (both the ‘read’ and ‘speed’ aspects being quite variable).
- You value the ‘random access’ nature of books that allows sections in previous chapters to be quickly re-checked.

What this book tries to do

All practical skills can only be acquired by rolling up your sleeves and “doing it”. The books and videos on Blender that you find online (including mine) were not made by people who learned their knowledge by reading a book. It was gained gradually, over a long period, by using the software, getting stuck and discovering solutions.

At the end of the day that is what you will have to do too but this book should make your journey a little less frustrating.

My main aim has been to make learning fun and, in order to do that, I haven't stuck too rigidly to a learning hierarchy starting with simple but rather boring exercises, only progressing to truly amazing things right at the end of the book. Each chapter does have a main concept or feature of Blender at its core, but it often includes extra elements that are best learned in the context of a practical exercise, and could have been legitimately included in many different locations. Throughout the book the process of following the instructions and finding the relevant menu or variable is a very valuable exercise in its own right.

At the end of the book you will have a good concept of what Blender can do. You probably won't be able to remember many of the specific details of what you did but you will know how to set about finding out. Importantly you will be unafraid to try making things on your own, and much less confused by the wealth of help online.

What are the non-aims of this book

This isn't an exhaustive reference. Blender is a massive program and mastering it entirely would fill a book much larger than this.

In this book I intentionally try to avoid using the "expert" technique for doing things. When getting to grips with a new program it is much easier to think "I want to **Add** a new **Mesh** object in the form of a **Cylinder**" than, I need to press **SHIFT A** then **M** then **Y**. My objective is to be explicit and easy to remember in preference to fast and efficient. As the keyboard short-cuts are visible each time a menu sequence is followed, it is very natural for you to switch to using them when an operation has to be repeated many times.

I have also tried to explain how to do everything without resorting to specific add-ons. This might seem a perverse stance as add-ons are a really cool aspect of Blender that make so many things better. However it will be hard for you to judge the value of an add-on if you have never experienced the problem it sets out to solve. More importantly, when you are looking for help online, it will be more difficult to follow examples if you have not been using a "vanilla" Blender and have always bypassed several steps using an add-on. The exceptions to this rule are the use of rigify which has become a de facto standard for posing and animating characters, and unavoidably where I explain how to download, install and enable add-ons!

This book takes a subjective view of Blender. It represents my personal experience and, although I have tried to find out as much as possible about features I rarely use, there are some aspects that I have decided to skip over or even neglect completely.

How to use this book

I have written the chapters on the assumption that they would be read sequentially. Logically, the section on download, install and setup should happen before any usage of Blender. However in other areas I will describe something in detail the first time you encounter it, more briefly the next time, and in one or two words thereafter. For instance:

In Chapter 1 *"With the Cube still highlighted and the mouse somewhere over the 3D Viewport press the* ***S*** *key on your keyboard. Move the mouse backwards and forwards... then click anywhere in the 3D Viewport to fix the change…*

Scroll back out and rotate view so you can see the whole of the Cube then, using the object interaction mode drop-down (left, just below the Header line), select Edit mode. Switching between Object mode and Edit mode is such a common process that it is almost always done by pressing the ***Tab*** *key...*

Now, with the top face selected subdivide it into 121 squares. You can either use the menu option: ***Edge ▸ Subdivide*** *or the context sensitive right click menu where* ***Subdivide*** *is the first option. Open the Adjust Last Action panel and change subdivisions to 10 i.e. each original edge has become composed of 11 edges"*

Then in Chapter 2 *"Add a Plane to the scene, scale it up about 40 times then switch to Edit mode and subdivide it about 20 times"*

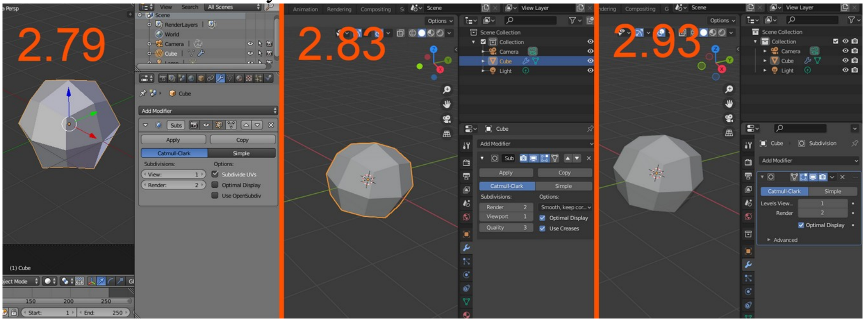
From time to time you will forget exactly what you did in a previous chapter, but because this is a book it's easy to flick back and check!

Writing this book was an iterative process: making a screen-capture video of each exercise, then writing up an explanation, then revising the video, then modifying the description etc. etc. So my recommended approach would be to read each exercise in the chapter at the same time as watching the video then return to do the exercise following the instructions, only resorting to watching the video when you get stuck. Unless you have a powerful computer with two screens it might be frustrating to try to watch videos at the same time as you follow instructions in Blender.

Different versions

The examples in this book use Blender version 4 but the appearance and functionality are very similar to version 2.8 onward, so if you can't upgrade to the latest and greatest, for whatever reason, everything should still make sense.

The change from 2.79 to 2.8 was a more significant one and, although most of the functionality covered by this book existed in version 2.79, the layout and styling are quite different. However not all graphics processors are capable of running versions after 2.79 so you might be stuck with that! It will be significantly harder to find the UI options, though you will quite quickly discern a pattern and, generally, the short-cut key combinations have remained the same.



Big changes from version 2.79

Videos and Blend files

You can find links to the videos along with a sequences of blend files accompanying each chapter on the web page *get-into-blender.com*. I have also included full sized color versions of all the figures in the book as the process of printing in black and white has made some of them less clear.

Where else you can find help

The online documents at *docs.blender.org* are comprehensive and easy to use.

There is a large and active community of Blender users so, obviously, an online search will often be your first resort for help. As well as the official docs, *blender.stackexchange.com*, *reddit.com/r/blender/* and *blenderartists.org* all have copious questions and answers, in the unlikely event that your question hasn’t already been answered there are many active users just waiting to help you. I would recommend creating accounts for all of these right from the start.

There are a few youtubers that I have really enjoyed watching over the years: Jan van den Hemel’s Blender Secrets, DECODED, Toni Mortero and Sophie Jantak are all worth searching out. You will find many more as you search for help or inspiration while you work your way through this book.

1

ORIENTATION

What the main elements of the User Interface are called and where they're hidden

In this chapter you will take a whistle-stop tour introducing you to the logic behind Blender's structure and readying you for the more intricate navigation later in the book.

As you acclimatize yourself to the Blender user interface you'll be building a realistic model of an emerald and diamond encrusted gold ring displayed on a velvet cushion. Exploring the different regions, editors, and menus will give you an inkling of how powerful Blender can be and how some relatively quick and simple steps can produce impressive results.

Download and Install Bender

The instructions on *https://docs.blender.org/manual* for installing Blender are comprehensive and clear; follow them carefully and everything should work without problems.

If you are using Windows, the download link on the Blender website will default to provide the standard Windows installer, which works perfectly well and is the most straightforward route for that operating system.

The alternative method, which is needed for macOS and Linux but also works on Windows, is to download a compressed folder containing everything you need to run Blender. For a long time Blender has followed a policy of packaging all dependencies into one *.zip*, *.dmg* or *.tar* file. The method of unzipping Blender into its own folder and running the executable file when you want to use it is simple with much to recommend it. It has allowed me to test the exercises used in this book on versions 2.79, 2.83, 2.93, 3.6 and so on.

Blender is always being actively improved so you might want to upgrade it in the medium term. For that reason, it’s not a good idea to install from the Microsoft Store or use the Debian package manager, as they sometimes lag several versions behind the current one. Blender doesn’t have an automatic update system built in so you need to check the blender.org website from time to time to see if there have been any important changes.

Blender Configuration and Setup

The configuration options within Blender are enormous but my policy throughout this book has been to stick to the default setup with zero modification. I would recommend that you also start that way so that my descriptions, screenshots, and videos match your own experience. The only exception is to enable emulation if you're using a keyboard without a numpad, but don't make the change now. We'll get to that later in this chapter, and the explanation of why and how to do it will make more sense once you have a little experience of using Blender.

Before you start any of the exercises in this book, make a folder for storing all the files you create as you work your way through. The name and location are up to you, but I recommend adding a subfolder for each chapter to keep everything from getting muddled.

Start with a Cube

Once everything is installed, start Blender. You will be presented with a splash screen with options to open previous work, but for now press the ESC key or click to the side of the splash. You should see the traditional starting cube (Figure 1-1) and many buttons and tabs, each one leading to yet more buttons and controls. There is logic behind all of this complexity.

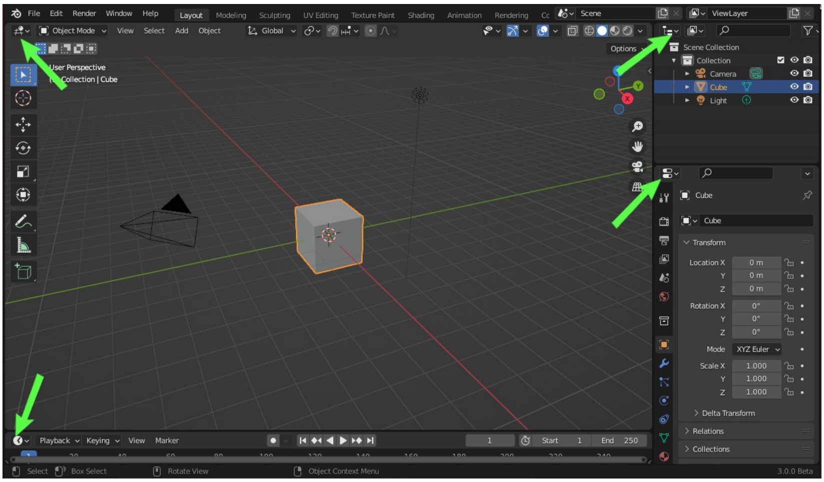


Figure 1-1 The starting scene

You'll be making many simple alterations to this starting cube to get a feel for the layout of the user interface, where to find the tools you need and how to make basic changes to objects.

Blender has a lot of help built in, so once you get used to some of the terminology and know how to look for information, your life will become much easier. However, there are a few features that will benefit from a little clarification before you start.

In Figure 1-1 there are actually four distinct areas, each containing a different *editor*, the main window has been divided vertically and each half has then been divided horizontally. The four arrows point towards the top left of each area where there is a drop-down menu that allows you to change the editor. Figure 1-2 shows the full list of available editors..

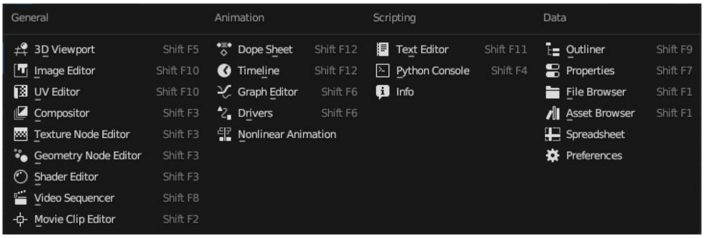


Figure 1-2 Full list of Editors

There are a lot of options, and some of those editors would require many pages to explain in detail, but the following are the main ones you will encounter in the book, roughly in order of appearance.

- *3D Viewport* is the place you will create and modify objects in your scene. This editor can be switched between different modes such as editing meshes, sculpting or painting textures onto objects.
- *Properties* allows you control most aspects of your scene. It is divided into many different tabs, some of which are listed in Figure 1-3, other tabs are only visible when selecting certain types of object or when using certain modes in the 3D Viewport. It isn’t too much of an exaggeration to say that the rest of this book is essentially a description of what all the tabs do.

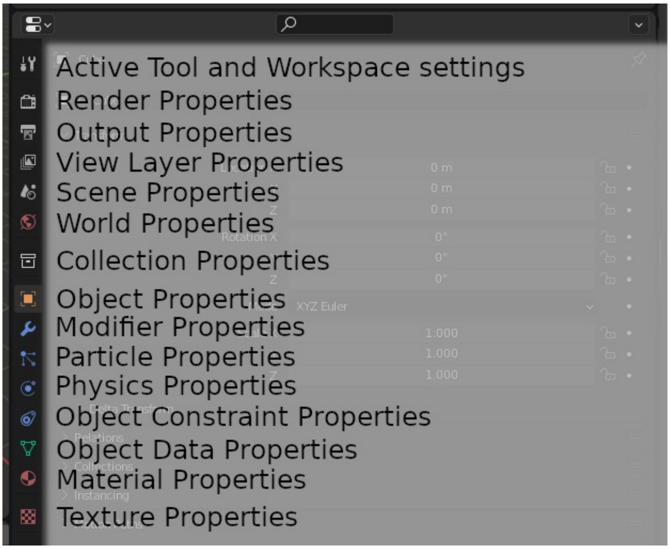


Figure 1-3 Properties Editor Tabs

- *Outliner* shows a hierarchical view of the data used by the blend file. This editor can be used to select, hide, organize components and so on, a little like Windows Navigation pane.
- *Timeline* is the last of the four editors in the default layout and is used to control animation. You will use it briefly in Chapter 2 and in detail in Chapter 12.
- *Shader Editor* allows materials to be edited using nodes, in a graphical, intuitive way. You will use this editor is every chapter of the book.
- *File Browser* accesses the computer file system and is used by Blender to save and load files when needed. However you can also use it in one of the screen areas which allows you to drag and drop media files.
- *Image Editor* lets you view or edit images and textures. You will use it in Chapter 5, Chapter 8 and Chapter 9.
- *UV Editor* is used to edit how 3D objects are unwrapped to allow images and textures to be applied to their surface. This editor will feature in Chapter 8.
- *Compositor* combines images, usually produced by rendering different parts of the 3D scene but also external images or movies.

The methods of composition are extremely sophisticated, comparable with fully fledged image editing software, but they use the visually intuitive nodes interface. You will use this editor in Chapter 13.

- *Dope Sheet*, *Graph Editor*, *Drivers* and *Nonlinear Animation* editors are all used for controlling different aspects of animation and you will encounter them in Chapter 12.

Mouse-over Tool Tips

Move your mouse over the tiny drop-down button just below the Blender logo in the top-left corner of the screen (in Figure 1-1 the icon looks like a ball resting on four crossed sticks). A tool tip should appear telling you that the current editor type for this area is the 3D Viewport. Now explore the screen using mouse-over, don't worry too much about remembering which elements are where but get used to how long you need to hover and whether to hover over the label or entry field. Verify that the four areas on this default layout workspace are 3D Viewport, Outliner, Properties and Timeline Editors.

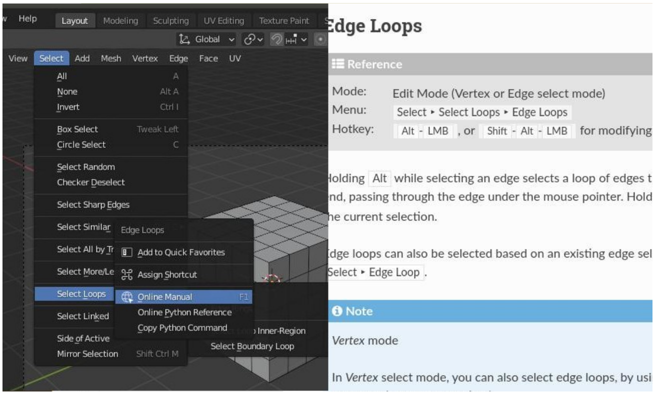
For the rest of the book, instructions will refer to the name visible by hovering the mouse, for instance "Modifier Properties" (hover over the blue wrench) or "Material Properties" (hover over the pink and black sphere).

Find the Shortcut from the Menu

A significant past criticism of Blender was that everything required shortcuts (CTRL, ALT, SHIFT) in combination with letters and numbers. For power users this setup provided a very efficient work flow, but for mere mortals, remembering all the key combinations was almost impossible. Now, however, Blender provides a menu route to each of these actions (as well as all the shortcuts) and in many cases, a graphical manipulator as well. In addition, the right-mouse-button context menu normally has options appropriate to whatever action you intend to take.

As you moved the mouse over different buttons on the 3D Viewport, you should have noticed the shortcut reference. The mouse-over clue is missing for one or two shortcuts (for instance, selecting edge loops). In those cases the right-click context menu on the tool or menu option allows you to open the online documentation at the specific reference.

Figure 1-4 shows the right-click menu that appears for **Select ▸ Select Loops ▸ Edge Loops** with relevant documentation



from the link to the online manual on the side.

Figure 1-4 Finding help and shortcut

As backup, if you can’t find a menu or button to give you a hint, press F3 and search. New Blender users typically start with menus and buttons, but after a while they start using shortcuts for more frequently performed actions.

Throughout this book I'll generally include the full menu path, which is more descriptive and provides a mechanism for finding the shortcut. In cases where the shortcut is obscure for some reason or is significantly easier to use than the menu option, I'll reference it when I first introduce the functionality.

Changing the Cube

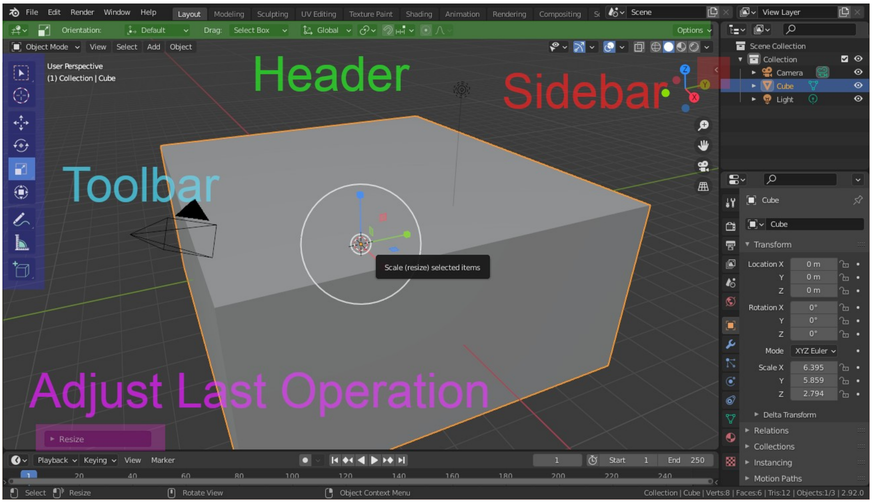
In this chapter you'll modify the default starting cube to become the mounting cushion for an engagement ring. After that you will add other objects and edit them to form different components of the scene, and finally you will render your composition to an image file. Links to the files and videos are on *get-into-blender.com 1.ORIENTATION*

Let's start with some of the basic methods available in Blender for modifying objects and their properties. This scene has three objects: Camera, Cube, and Light, and you can left-click each one to select them in the 3D Viewport or Outliner Editor areas.

NOTE From this point on when I say "click" or "drag" without mentioning which button, I mean left. The more specific mouse button instructions may also be referred to as LMB, MMB, and RMB, as shown in Blender's online documentation in Figure 1-4.

The selected object is highlighted in the Outliner as well as outlined in orange in the 3D Viewport. Click the Camera and Light objects before selecting the cube again, and notice that the options in the Properties Editor change as you click from one object to the next. Sometimes when you are following instructions and your screen has different options from the ones you expect, check that you have the correct object selected.

To change the size of the cube, select it then click the **Scale** button in the Toolbar on the left (it's the one highlighted in Figure 1-5). I've indicated the four Regions in the 3D Viewport area that you will need



to use often and which ***don't*** have a mouse-over giving their name.

Figure 1-5 The names of four regions on the 3D Viewport

The Toolbar menu will contain different options depending on the type of editor (3D Viewport, Image Editor, UV Editor) and the mode selected (Object, Edit, Sculpt, and so on). If a Toolbar exists, press SHIFT-spacebar to access it as a pop-up, which also is a good way to save a bit of screen space if you don't have the luxury of a 4K monitor.

NOTE Before Blender 2.93 the default way to access the Toolbar was via the spacebar, but that has been changed to toggle playing or pausing the animation in line with most other applications. You can change shortcuts such as this one in Blender with **Edit ▸ Preferences ▸ Keymap**.

Scale the cube using the *gizmo* (yes, that's the technical term for the pink, yellow, and blue thing in a white circle, centered on the cube in Figure 1-5). You can drag the colored points to scale in X, Y, Z, X + Y, X + Z, or Y + Z directions. To scale uniformly, drag from anywhere in the white circle that isn't colored. Do several different types of scaling operations, and then click to expand the Adjust Last Operation where you will see the actual values of the last scaling. You can adjust these values specifically, so try changing to X to **2.0**, Y to **2.0**, and Z to **1.0**, and then in the Properties Editor, select the Object Properties tab (orange square), and in the Transform section, look at the Scale field, which is the open tab on the right of Figure 1-5. You should see different values because those are the total of all the scaling operations as opposed to just the very last one you did.

Now you will scale the cube using shortcut keys. With the cube still highlighted and the mouse somewhere over the 3D Viewport, press S. Move the mouse backward and forward, then press and release X, then try Y, Z, SHIFT-X, SHIFT-Y, and SHIFT-Z, moving the mouse to see the effect. Finally, type the number **3.14159** then click anywhere in the 3D Viewport to fix the change. When you type a number after a sequence of shortcut keys such as this, Blender will automatically interpret the value as a scale, rotation or displacement. The formula that will be executed when you finally click is visible at the top of the 3D Viewport. You should have seen that the axis or plane for scaling was constrained by following the shortcut with other keys to modify its behavior.

This cube scaling exercise might seem like flogging a dead horse, but there's one last option for tweaking properties: in the middle of the Scale X value of the Properties tab in the Properties area (where it shows 6.395 on the right in Figure 1-5), while holding down the LMB, move the mouse from left to right and back again. You should see the scale increase and decrease as you move the mouse. Finally, in the Scale value boxes, change X to **8.0**, Y to **8.0**, and Z to **1.0** which will explicitly set the scale to exact values.

Moving the Viewpoint

Another crucial aspect of navigating Blender is moving the point of view around the 3D Viewport. With the mouse over the scene, press the MMB and move the mouse around, then scroll the mouse wheel to rotate and zoom. In order to "pan" the view, press and hold SHIFT on the keyboard *before* pressing the MMB. The gizmos on the right of the 3D Viewport provide the same controls but also allow orthogonal views along each axis or from the Camera position.

DIFFERENT VIEWS

Cameras in Blender are actual objects in the scene that are used to "render" the final image or video. As well as being moved and animated like a real camera, they have properties such as focal length and aperture to control depth of field. The active camera's view is different from the working point of view you see on the screen but there is an option to align the Camera to the current view and vice versa.

Occasionally, moving the viewpoint around becomes awkward. As an example, try scrolling in with the mouse wheel or zoom gizmo until you are inside the cube, then keep going even when the view doesn't move. Quite quickly you will find that you are "stuck." You can rotate the view, but panning and zooming don't seem to do anything. If you now needed to move so you were viewing the inside of one of the corners from up close, you would have to scroll back out quite a long way (eventually it would start to have an effect), then rotate 90 degrees from the direction you wanted to go, pan sideways, rotate back toward the target corner, and finally zoom back in again! It's possible to do once you get used to using the controls, but often it's much easier to change view with Walk Navigation which uses the "game" controls (W, A, S, D, Q, and E) to move the viewpoint around. The shortcut for this (**View ▸ Navigation ▸ Walk Navigation**) is SHIFT-` often called backtick or AccentGrave in the Blender docs. On some laptops there is a conflict and this shortcut doesn't work. If that is the case then change it to CTRL-F using **Edit ▸ Preferences ▸ Keymap**. Figure 1-6 shows a cross-hairs in the center of the screen indicating the direction of travel.

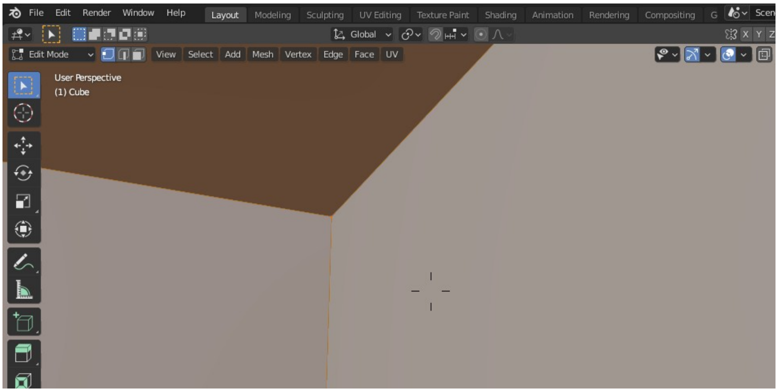


Figure 1-6 Moving the viewpoint in Edit mode using Walk Navigation

To end Walk Navigation, press LMB. To instantly move close to whatever is targeted by the cross-hairs, press MMB, and to jump back to where you started, press RMB.

Finally, if you first switch to Camera view by clicking the icon on the right, the location of the Camera will change as you “fly” around the scene using Walk Navigation. You can also use the arrow keys on the keyboard for Walk Navigation, but they lack the up and down modes provided by Q and E.

Editing Meshes: Vertices, Edges, and Faces

The fundamental object used for 3D modeling is the *mesh*, which is defined by a list of points called *vertices*, each with an X, Y and Z location. In addition *edges* are defined connecting pairs of vertices and *faces* are defined by three or more vertices. Editing a mesh enables vertices to be manipulated individually or in groups, as well as providing ways to add or delete the three mesh components.

Scroll back out and rotate the view so you can see the whole of the cube. Then, using the object interaction mode drop-down (third line down on the left in Figure 1-6), switch from Object mode to Edit mode. Switching to and from Edit mode is such a common process, you'll almost always press TAB to switch. To change modes without using a mouse, a pie menu is also available via CTRL-TAB, which is useful if you need to swap modes frequently.

In Edit mode, you can see the mesh that forms the object—in the case of a cube, 8 vertices, 12 edges and 6 faces. All the vertices are selected initially, but when you click one of them, all become deselected apart from that one. Holding SHIFT while clicking on a vertex toggles the selection of that vertex; it behaves like the CTRL button in most other applications! It might take a bit of getting used to so have a play around at selecting and deselecting vertices.

To see different parts of an object, it's possible to swivel, zoom, and pan the viewpoint; however, toggling the X-Ray view on and off is a very efficient way of fine-tuning vertex selection. The X-Ray button icon has two overlapping squares; in Figure 1-6 it is the furthest right button visible. Finally, keyboard shortcuts A and ALT-A are frequently used to select or deselect all the vertices.

Subdivide

To make the cube into a mount for a ring you will need to subdivide the top face so that it can be made convex and also, so that some of the subdivided faces in the center can be depressed to form a slot. Subdivision is one of the simplest function in Blender to create new vertices, edges and faces. If one face is selected then subdivided, a number of smaller faces will be created along with the appropriate new vertices and edges.

Being able to select edges or faces instead of vertices, and perform appropriate actions specific to them, is another crucial aspect of using Blender. Try switching between vertex, edge and face selection using the three buttons in Figure 1-6 just to the right of the drop-down showing Edit mode. Now, with the top face selected subdivide it into 121 squares using **Edge ▸ Subdivide**. (There is no shortcut, but Subdivide is the first option in the context-sensitive RMB menu). Open the Adjust Last Action panel and change subdivisions to **10** so each original edge becomes composed of 11 edges.

In general you move vertices in Blender by first selecting them, then **Mesh ▸ Transform ▸ Move** (or use the relevant gizmo in the Toolbar). However, moving, scaling and rotating vertices, edges, faces or whole objects are such common operations that you will almost invariably do this using the G, S, and R shortcuts often in conjunction with a subsequent X, Y, Z, SHIFT-X, SHIFT-Y, SHIFT-Z and sometimes followed by a number representing a distance, scale factor or rotation in degrees. The change becomes fixed when the LMB is clicked. These shortcuts in Edit mode behave the same way as when you were scaling the default cube in Object mode earlier. Select a few vertices and do some trial move, scale and rotate changes, undoing each with CTRL-Z before continuing.

VARIATIONS OF THE MOVE, SCALE AND ROTATE SHORTCUTS

Several aspects of using the G, S, and R shortcuts are hard to find using mouse-over help or search, but they can often be very useful:

- Use **SHIFT** while dragging with the mouse for fine control. This also works when changing almost anything by moving the mouse such as value sliders, color pickers, even snapping to grid increments.

- Rotate by pressing R twice to enable track-ball rotation and simplify tasks such as rigging and posing. It's often a time-saver when posing bones or positioning objects in a scene.

- Press **CTRL** while moving, rotating, or scaling to enable snapping. You can press **CTRL** and **SHIFT** at the same time to snap at a finer scale.

- In Edit mode press G twice to constrain vertex movement along existing edges. You will probably find fewer circumstances when this option is needed, but on occasion, it can prove very useful.

One button in the Header that is much used when editing meshes is Proportional Editing (near the middle of the second line in Figure 1-6 under the word *Shading*). This allows graded movement of large numbers of vertices. The size of the field that Proportional Editing effects is shown as a circle, and you can adjust the diameter using the mouse wheel. You can revise the proportional editing values in Adjust Last Action as well as the actual amount of the change.

To make the top of the mount cushion convex you need displace vertices downwards by increasing amounts, the nearer each is to a corner, if Figure 1-9 you can see the curve that I achieved. Change back to Vertex select, if it is still set to face select, and turn on Proportional Editing. Now select all four corner vertices of the top face and move them down using shortcut G then Z then LMB, in a similar way to the scaling you applied to the cube earlier. The three transformations; scale, move and rotate are so ubiquitous while modeling that I will refer to them throughout the book by their shortcuts rather than their menu path. You are aiming to form a nice convex surface, but you may get a better shape if you do this in several steps with different sizes or profiles for the Proportional Editing. I found that the Sharp profile was nearest to what I had in mind.

When the shortcuts are followed by X, Y, or Z to constrain movement, scale, or rotation to one axis, a colored line will appear to indicate the constraint.

Extrude

Once the top of the mount cushion is rounded, select five adjacent faces in the middle to extrude downward to create a “slot” to hold the ring., To do this, change back to face select, press the LMB to select the first, and then press SHIFT-LMB to add four more to the selection.

Select **Face ▸ Extrude**, move the mouse downward a small distance, then press the LMB to fix it. Extruding is a common process, so you'll very quickly learn to use the shortcut E. The default behavior is extruding perpendicular to the surface (shown by a blue axis line), and pressing Z repeatedly will toggle between local perpendicular, free movement, and global Z axis. On the other hand, the Toolbar extrude gizmo has a handle to drag for perpendicular extrusion, and you can click and drag from anywhere else in the circle for “free” extrusion.

Creating the Ring

From this point on you’re going to add some more simple geometrical shapes and edit their meshes and material properties to make the ring, an emerald, a setting and some diamonds.

1. To create the band of the ring, switch back to Object mode and select **Add ▸ Mesh ▸ Torus**. This torus will become the vertical band in the cushion so select that object and rotate it through 90 degrees using R, Y, 90 then LMB and scale it to match the slot in the cushion.

2. Add two polyhedrons to represent an emerald and its mount. The one with fewer sides will become the mount and the higher-sided one the emerald. Select **Add ▸ Mesh ▸ Ico Sphere** with one division and do another with two divisions (expand the Adjust Last Action to change the number of divisions). Scale the two icospheres appropriate to the size of the ring,

3. Copy the emerald to form a basis for four diamonds. With the emerald selected copy it using CTRL-C and paste it with CTRL-V. So long as the cursor is over the 3D Viewport the copied object will appear in exactly the same location as the object it was copied from.

4. Scale down the copied icosphere then copy it three more times. These smaller icospheres will be diamonds set into the ring (look ahead to Figure 1-10 to see the sizes I used, but design it to your own tastes—after all, these carats cost nothing). Press G, Y, then LMB to move the copy to one side along the Y axis. Scale it down then repeat the copy, paste, and move to one side step three times.

To make the torus look more like a ring, you'll change its cross section from circular to oval. Your objective is to select a circle of vertices on the inside of the ring and scale them up, then select a circle of vertices on the outside and scale them down as shown in Figure 1-7.

Select the ring object then switch to Edit mode to allow you to move vertices. However before you attempt to do this you need to learn a little more about selecting vertices in Blender.

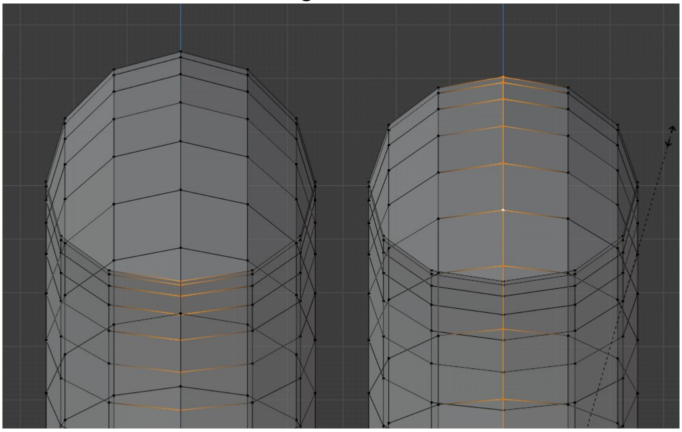


Figure 1-7 Scaling the inside and outside ring of vertices

More Ways to Select

There are several ways to select vertices, edges or faces beyond simply clicking on them. When you first start Blender, selection defaults to select box which allows you to draw a rectangle around the vertices you want to select. There are actually quite a few other methods for selection but, initially, you will find that much can be done using the default select box on the Toolbar and shortcut C to circle select for fine-tuning.

With circle selection the size of the circle is controlled with the mouse wheel and vertices are added by LMB and "painting", removed by MMB and the selection process ended by RMB or pressing the enter key.

You can add to, or remove from a selection by pressing SHIFT or CTRL at the same time as dragging the rectangle,

5. Your objective here is to select just the innermost vertices using circle selection. Toggle X-Ray on and switch to orthographic view so the ring can be seen from the side by clicking on the X in a red circle on the view rotation gizmo. You will find that it's quite difficult to control, but making the circle size near to the internal diameter of the ring does make it a little easier.

NOTE The shortcuts for changing to orthographic or camera view use the numpad. If you are using a laptop or a keyboard without a numpad it is still possible to emulate one by selecting from the main menu **Edit ▸ Preferences ▸ Input ▸ Keyboard ▸ Emulate Numpad**. In my opinion the numbers for changing the view are less intuitive to use when they are laid out in a line instead of a grid and the view rotation gizmo is quick and easy to use. However CTRL-PLUS and CTRL-MINUS are very useful shortcuts for the unwieldy **Select ▸ Select More/Less ▸ More** (or **Less**) so I strongly suggest you enable the Emulate Numpad option. From this point on when I refer to PLUS or MINUS I mean the keys on the numpad or emulated numpad.

---

6. Now try to get the inside curve of the ring more oval. Scale the innermost vertices up with proportional editing toggled on and the diameter adjusted to an appropriate amount. This is a good example of a time when it's useful to open the Adjust Last Operation pane and tweak the Proportional Size value while watching how the mesh changes.

7. To select the outermost ring of vertices you will use by far the most efficient method for selecting loops of vertices. Select a representative edge, then **Select ▸ Select Loops ▸ Edge Loops**.

NOTE Edge Loop selection is such a useful feature that you will quickly prefer the shortcut ALT-LMB rather than the menu route, which is much slower. This shortcut also happens to be one that doesn't get a hint in the menu see (Figure1-4) neither does the shortest path shortcut CTR-LMB. Experiment with these two, also with SHIFT.

---

8. Scale the outermost vertices down with proportional editing on. Remember to toggle proportional editing off again when you've finished.

9. Blender allows objects to be "smooth shaded" or "flat shaded" and for this exercise we want the ring to be smooth and the gem stones and mount to look faceted. Switch back to Object mode, select the ring then in the menu **Object ▸ Shade Smooth**.

10. The icosphere with fewer sides will become the mount so you will transform it into more of a conical shape. Select the icosphere, switch to Edit mode and move the top vertex downwards with G, Z then LMB, then select the five vertices nearer to the bottom and scale them inwards. You could constrain the scaling to the XY plane with S then SHIFT-Z, but as all the vertices are in the same plane it will have no effect so you can just use S.

11. In reality the ring and the mount will all be one piece of gold and you can join them in a similar way in Blender. Switch back to Object mode and using SHIFT-LMB, select the ring as well as the mount object then, with the ring and mount both selected, join them to form a single object using **Object ▸ Join**.

NOTE **Important:** This is a potentially confusing feature of Blender: When multiple objects are selected there is a redder shade of orange used for highlighting all objects prior to the last one and there is a similar difference in highlighting in the Outliner Editor area on the top right. The redder orange objects are termed "selected" and only the last selected object is termed "active". For this exercise the order of selection (and shade of orange) doesn't matter but for many processes it is crucially important so this is something you need to watch out for when you read the documentation or follow videos online.

Adding a material to an object

For the rest of this exercise I'm going to guide you through aspects of Blender that you will only cover in detail later in the book: materials, shaders, nodes, textures, particles and rendering are all things that you will definitely need to understand eventually, but not at this stage. Your objective while following these instructions is just to become more comfortable finding your way round the user interface.

1. To see what your gold and gem materials look like you need to select Material Preview from the four buttons that control the Viewport Shading at the top right of the 3D Viewport (to the right of the X-Ray button).

The options are: Wireframe, Solid, Material Preview and Rendered. They are all useful for different purposes but increase in computational requirements from left to right. Rendered can be slow if Blender has to simulate all the light paths being reflected, refracted and scattered by many objects in the scene. Material Preview uses some clever algorithms to give a good approximation to the final product.

2. First create a material to represent the gold. With the combined ring and mount still as the selected object, click on the Material Properties (pink & black sphere icon) in the lower right Properties Editor area and click on **+ New** to create a new material. Note there is another **+** button on the right that will add a new slot, you will learn a little more about material slots in the next chapter.

3. The new material will have Surface set to the default Principled BSDF Shader which is very flexible. To make it look like gold you just need to scroll down and change the Base Color to yellow, increase the Metallic component and decrease the Roughness.

Using Nodes

You should already have a pretty good looking gold, however to give an antique, hand-made look to your material I will quickly take you through the process of adding some nodes. Again, don't worry about the details of this for now, using nodes is an important feature of Blender so it will crop up again and again through the book.

4. To work with material nodes you need to set one of the areas to use the Shader Editor. You can do this by clicking on the Shading tab at the very top of the window which will change the Active workspace. You can modify the type and size of the different editor areas quite easily but the Active workspace tabs are a convenient way to switch between tried and tested default layouts (see Figure 1-8). The bottom of the screen will now contain a Shader Editor area and you can zoom in on this with the mouse wheel, pan the view with the MMB and select and drag the nodes with the LMB.

5. The first node to add is one that will produce a random, beaten pattern which can be used to change the reflection from the surface of the gold. Using the menu at the top of the Shader Editor area **Add ▸ Texture ▸ Noise Texture**, un-check the Normalize option, then connect its output (Fac) to the Base Color of the Principled Shader by dragging and dropping the dot to create a connector line. This is just a temporary setup to check what the texture looks like. Now increase the Scale to about 18 so the pattern looks proportionate to the size of the ring then drag the output connection to Roughness and disconnect it from Base Color. Older versions of Blender had a dedicated node for this called Musgrave texture.

6. To tone down the strength of the pattern you can feed the output through a node to multiply it by a small value. Use the menu **Add ▸ Converter ▸ Math** to create a new node and drop it onto the connection between the Noise Texture and the Principled Shader nodes, which should automatically insert the node into the connecting “pipe”. If you miss then you can reconnect the inputs and outputs so they match Figure 1-8.

7. To distort the surface of the metal further you can also use the noise pattern as a displacement. Change the type to Multiply and the value to 0.01 then drag a second connection from the output Value to the Displacement input of the Material Output node. You can adjust the multiplication value to vary the unevenness in the final render.

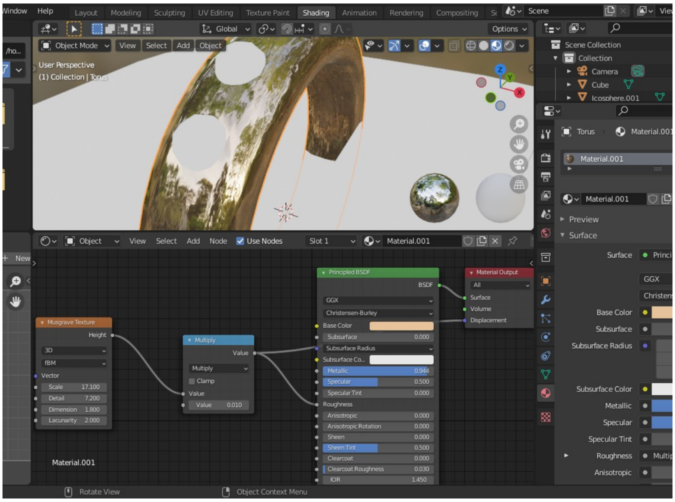


Figure 1-8 Shading workspace

8. Next you need to create an emerald material in the relevant object. In the 3D Viewport select the emerald object icosphere and add a material as you did for the ring.

9. To make the material transparent, in the Material Properties tab, click on the Surface shader and change it from Principled to Glass BSDF, change the Color to green, Roughness to zero and increase the IOR a bit (refractive index of emerald is about 1.58)

10. Select one of the diamonds and add a new material to that object changing to the Glass BSDF shader as well, but with Color white and IOR 2.4. For the other three diamonds select the same diamond material from the drop-down list in the Material Properties tab. You can see the drop-down half way up Figure 1-8, just above the Preview section.

11. Finally arrange all the objects so they look to be part of one ring. Move the ring down so it nestles in its slot then move the emerald and diamonds to their positions on the ring, this will be easier to do using the orthographic view.

Velvet with Quick Fur

Your next objective is to make a luxurious velvet material for the cushion. As a first attempt it would make sense to use the Velvet BSDF shader.

1. Select the cushion object which started life as the default cube. It will already have a white material called “Material”. In the Material Properties tab change the Surface shader from Principled BSDF to Velvet BSDF and the color to dark red.

2. With Viewport Shading as Material Preview it looks rather flat so change to Rendered, the rightmost of the four buttons. However it still isn’t very velvety.

3. “Quick Fur” sounds hopeful but you will find that it is applied unevenly unless you apply the scaling. First click **Object ▸ Apply ▸ All Transforms** then **Object ▸ Quick Effects ▸ Quick Fur** then In the Adjust Last Operation panel un-tick the three check-boxes; Apply Hair Guides, Noise and Frizz. Reduce the View Percentage to **0.1** and change the Density to **High**.

4. If you expand the Cube object in the Outliner Editor you will see that a new object has been added called Curves, as a child of Cube, as visible on the right of Figure 1-9. This has now become the selected object. Click on the Modifier Properties tab (blue wrench) and in the modifier called Generate Hair Curves, reduce the Control Points to **2**. In Set Hair Curve Profile, increase Radius to **0.005** and in Interpolate Hair Curves, increase the Density to **750**.

QUICK EFFECTS

Adding Fur, Explode, Smoke or Liquid at the press of a button sounds really amazing and I would certainly recommend playing around with these, once you've finished the exercises for this chapter. However the Quick Effects are just shortcuts to setting up Blender Physics, Particles and Modifiers with appropriate settings and many of them will be frustrating if you try to create a rendered image or animation before you have covered the relevant topic in the later chapters.

5. You need to set the fur fibers to a similar red to the base cushion but you can also add some some variety to that material using a different texture node. With the Curves object still selected, in the Material Properties tab a new Fur Material will have appeared, change its color to dark red. Select the Cube object (it should still have the original material called "Material") and in the Shader Editor area add three new Nodes **Add ▸ Texture ▸ Voronoi Texture** then **Add ▸ Input ▸ RGB** then **Add ▸ Color ▸ Mix Color** mix the Voronoi and RGB and feed the result into the Velvet shader, adjusting the controls until you get an interesting result, the node arrangement is shown in Figure 1-9.

In Blender version 3.5 the method for generating hair changed from using particles to use geometry nodes. If you are using an older version then you should probably skip this enhancement until you cover particles later in the book.

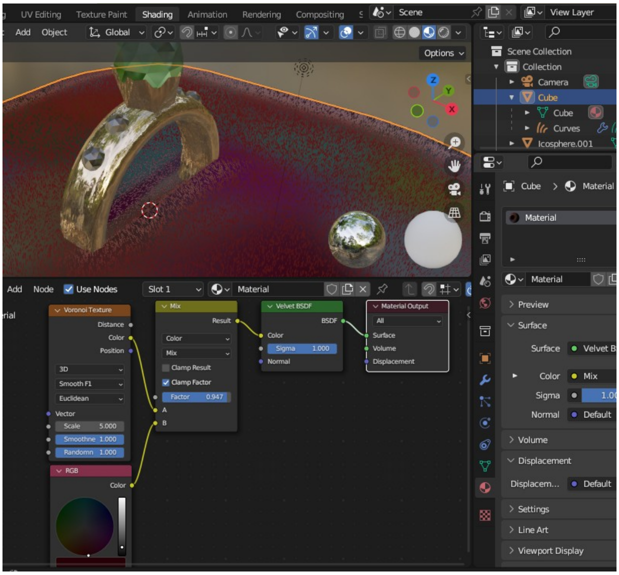


Figure 1-9 Velvet cushion material

Don’t spend too long fine-tuning the appearance of the velvet material as this will undoubtedly change when you position the camera and set the render engine in the next section.

Moving the camera

To compose the final image you need to position the camera and run a few test renders to finalize all the settings.

1. For the exercise in this chapter you will use the cycles render engine. In the Render Properties tab of the Properties editor (the icon like the back of a camera) change the Render Engine drop-down from **Eevee** to **Cycles**. You will learn more about the differences later but, at a simplistic level, eevee is quick but cycles is more realistic.

2. Position the Camera by clicking on the camera button on the right of the 3D Viewport. If you scroll out with the mouse wheel you will see a rectangle surrounded by a dotted line to indicate the image frame. Use SHIFT-` and WASD game style controls to “fly” with Walk Navigation so the scene is well composed within the camera’s field of view.

3. To start the render, select from the menu right at the top of the window **Render ▸ Render Image.** While fine-tuning the material settings you can stop the render early by pressing ESC, the first time will stop the render and pressing ESC again will close the window. In subsequent chapters you will see how to set a small render region so you can quickly check whether the settings are correct.

A synthetic window

You will have probably noticed that the reflections from the gold ring are much more interesting and produce a more realistic result when the Viewport Shading is set to Material Preview compared with the render. We will see later how to set this up properly but for the time being you can improve the scene by adding a big square "light" behind the camera with a vertical and horizontal bar in front of it. This will add highlights and illumination as if from a window.

1. First of all create a rectangle that emits light. Create a plane using **Add ▸ Mesh ▸ Plane** then scale it up, move and rotate it vertically behind the camera. Toggle the camera view off by clicking again on the button in the 3D Viewport or simply move the view with MMB and SHIFT-MMB. Add a new material to the plane and change the Surface to use the Emission shader.

2. Now put a vertical and horizontal bar in front of the light. Add an additional plane then scale, rotate and move it to form a bar in front of the window. Finally copy, paste and rotate the bar to make a cross.

The final render

The finished render (Figure 1-10) will probably take a few minutes but when it's finished you can save the image to your working folder with **Image ▸ Save As**

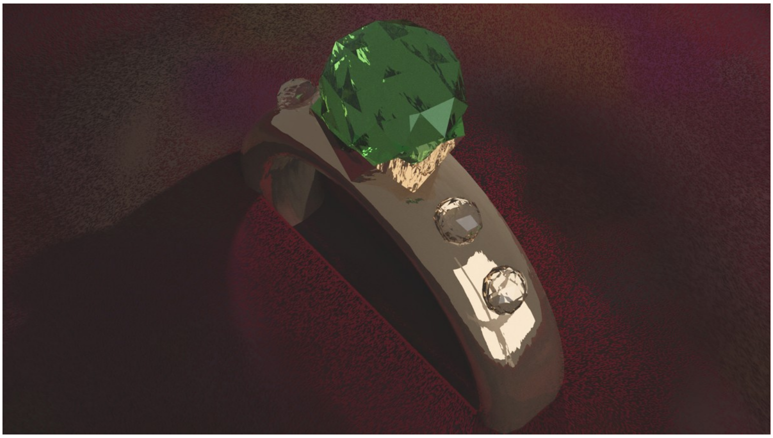


Figure 1-10 Final render

Conclusion

The main purpose of this chapter was for you to get used to navigating the user interface. You saw that the default layout workspace is divided into four editor areas each with its own special features and functionality. As well as moving, rotating and scaling objects and manipulating vertices you used two other methods for creating new vertices in a mesh: subdivide and extrude. As the book unfolds you will learn how to use the many other mesh editing tools from the Toolbar and lots of ways to generate mesh real estate.

Selecting objects is straightforward at one level but it is also something you will need to be careful of later when you learn about parenting objects. Selecting vertices can also be simple, but being able to do it efficiently when working with complex meshes is an important skill and you will build an armory of different approaches as you work your way through the rest of this book.

The next chapter introduces the subject of modifiers which will provide you with a treasure chest full of methods to build complicated geometry very quickly. As you complete the exercises you will need to use the lessons you learned in this chapter and, at times, you may find yourself racking your brains to remember what to do. But take your time, refer back when you get stuck, watch the online videos and check the documentation. It's all part of the learning process and very soon you will become familiar with all the intricacies and quirks of the Blender user interface.

2

MODIFIERS

A powerful and flexible mechanism for making changes to objects

Having learned the basics of selecting objects and vertices, rotating the view point, and switching between regions and editors, it's time to get to grips with the next core concept: *modifiers*. Once again, this particular subject could fill several books; rather than try to memorize everything, the goal is to explore and gain an appreciation for the kinds of things you can do with modifiers, bearing in mind that this chapter presents only a tiny sample of that capability.

Modifiers (also called *non-destructive modifiers*) allow you to apply a process or function to an object to produce the appearance of a new object, but they leave the original object intact. In addition, you can apply successive modifiers to the same object, each taking the output of the previous one and modifying it further. In this chapter, we'll start using modifiers to create three increasingly complex projects so you can grasp their potential power. Files and videos for this chapter are on *get-into-blender.com 2.MODIFIERS*

Booleans: Adding and Applying a Modifier

Boolean modifiers provide a way to construct complex shapes from simpler ones by subtracting or adding one object to another. As an example of working with Boolean modifiers, let's make a coffee cup:

1. Start Blender, delete the default starting cube, and select **Add ▸ Mesh ▸ Cone**. In Adjust Last Operation, change Radius 2 to **0.6**, and then rotate the object by 180 degrees about the X axis.

2. From **Object ▸ Duplicate Objects**, press **Z**, and move the duplicate slightly upward then scale it down, but ensure the smaller cone protrudes above the top surface of the larger cone, as shown in Figure 2-1.

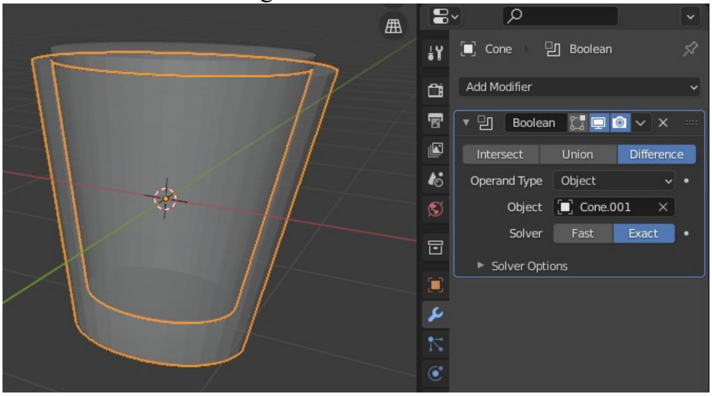


Figure 2-1 Boolean difference modifier using two truncated cones.

3. Select the larger cone, click the Modifier Properties tab on the Properties Editor area (blue spanner icon) and **Add Modifier ▸ Generate ▸ Boolean** (Figure 2-1). Keep the combination type as **Difference**, select the smaller cone (Cone.001) as the object to operate on.

4. To see what the resultant shape looks like, click the eye icon in the Outliner Editor to toggle the visibility of the smaller cone.

5. For the cup handle, go to **Add ▸ Mesh ▸ Torus**, rotate it 90 degrees about the X axis, scale it down, and move it to an appropriate position on one side of the cup (easiest done in orthographic view along the Y axis).

6. To make the handle more elegant, carve some of the ceramic material from the outside of the torus using a cylinder: **Add ▸ Mesh ▸ Cylinder**, rotate it 90 degrees about the X axis, scale it down, and move it so it contains most of the torus as shown in Figure 2-2. Scale it slightly more in the Z direction so the handle is thinner at the top and bottom.

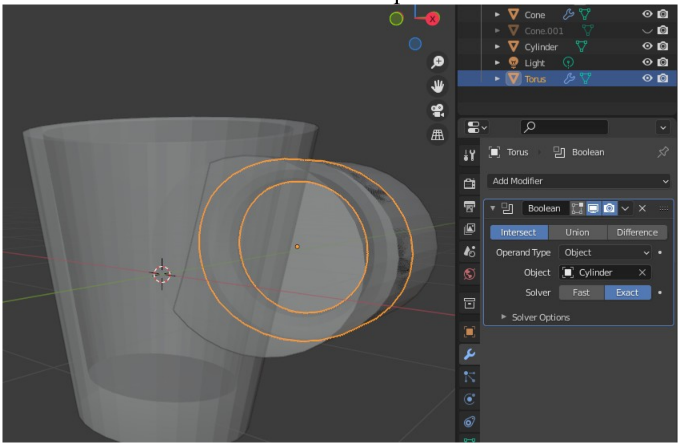


Figure 2-2 Handle made from a torus intersecting a squashed cylinder

7. With the torus selected, add a boolean modifier and change to **Intersect** with the cylinder object. Toggle the visibility of the cylinder in the Outliner Editor, and adjust the scale and position values of the cylinder until the handle looks just right.

8. Select the larger cone object and add another boolean modifier, but change the type to **Union** with the torus object as the target and toggle off visibility of the torus in the outliner. You should be able to see that something is wrong here: because the torus is

being added after the cone has been hollowed out, it protrudes inside the cup. To fix it, drag the boolean difference modifier down below the union modifier using the area to the right of the cross (it contains eight white dots; see Figure 2-3). An option in the drop-down menu also allows you to change the order of modifiers.

9. At this point, moving any of the objects would break the assembly. You can fix this by “applying” the modifiers, which effectively bakes them as a permanent change to the mesh (in Chapter 3, you’ll learn a different method to solve this problem). Select the torus, and in the modifier, choose **Apply** from the little drop-down between the camera and cross on the top line (Figure 2-3). Select the larger cone and apply first the union, then the difference modifier. Delete the smaller cone, the torus, and the cylinder objects.

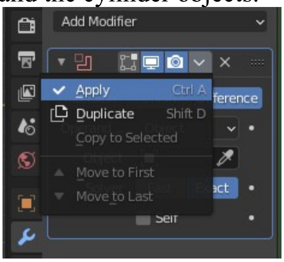


Figure 2-3 Apply is hidden in a drop-down menu.

NOTE Watch out for the little downward pointing chevron indicating a drop-down menu. In later chapters, when working with vertex groups and shape keys, you will find crucial functions hidden beneath a tiny button like this.

A significant difference between the boolean union modifier and the join method we used in the last chapter to merge the emerald setting to the ring (select multiple objects then **Object ▸ Join**) is that the modifier removes all the overlapping internal vertices (Figure 2-4).

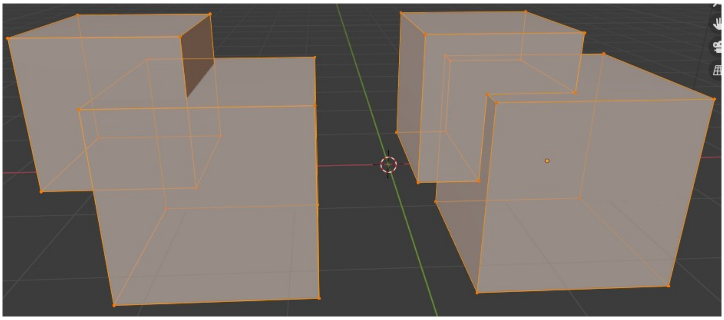


Figure 2-4 Join on the left, union modifier on the right

Join also doesn’t integrate the two meshes. In Figure 2-4 the faces of the two cubes on the left pass through each other without forming edges and vertices at the intersection.

Subdivision, Bevel, Mirror, and More

It’s hard to quantify the relative importance of Blender’s modifiers, but you are likely to use Boolean, Subdivision, Bevel, Mirror, and Displace with high frequency. You will utilize all of these while completing the main exercise for this chapter, but first let's look at some details of these modifiers and review a few other interesting ones. As an example, you will construct a simple ornament such as might have been cast by a primitive Bronze-age craftsman.

Subdivision

Subdivision is the process of creating a smoother, higher polygon mesh from a coarser one. When modeling with Blender, you should always try to create simple meshes with as few vertices as possible that generate smooth, detailed objects when a Subdivision modifier is added. Easier said than done!

1. Clear the scene and start a new one: **File ‣ New ‣ General** (don’t bother saving the coffee cup).

2. Add a second cube, scale it down slightly, then use the Boolean modifier to add the smaller cuboid shape to the initial cube three times, applying the modifier each time. Add the cubes on the side facing the negative Y direction (so the description below is easier to follow when I refer to X and Y; see Figure 2-5). Delete the smaller cube when you have finished adding it.

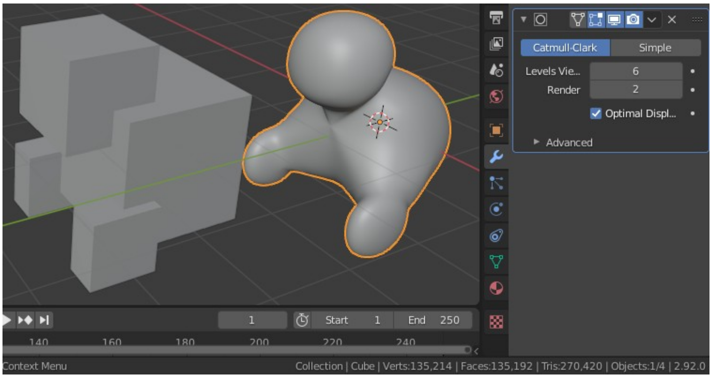


Figure 2-5 Don’t subdivide too much.

3. Select **Add Modifier ▸ Generate ▸ Subdivision Surface** and increase the Levels Viewport a few times. Don’t increase it more than five or six unless you have a powerful computer though, as each level increases the number of faces by a factor of four. To witness the geometric increase, check the bottom right of the Blender window to see Scene Statistics.

Bevel

Bevel adds additional faces along edges to make them chamfered or rounded. Obviously this is useful when an object's design requires beveled edges, however, the modifier is often used to add definition to edges that separate fairly flat surfaces, which turns out to be difficult to do well in 3D modeling.

4. With the cube object selected reduce subdivisions to 2 then **Add Modifier ▸ Generate ▸ Bevel**, but notice it has no visible effect because it is acting *after* the subdivision and all the sharp edges were removed by that modifier. The default setting is to bevel edges where the difference from one surface to the next is greater than 30 degrees, but there are various ways to specify which edges to bevel (we'll cover those different ways in Chapter 11).

5. Move the Bevel modifier up to above the Subdivision with the **Move to First** option from the drop-down or drag it using the top-right corner of the modifier pane. The subdivision surface follows the cuboids more closely because of the extra edges that beveling introduced.

Mirror

The *Mirror* modifier is another way to create more vertices quickly by duplicating a mesh across planes of symmetry. If you are making an object that will be symmetrical, it is much better to make only a half or a quarter of it and then mirror it.

6. Click the cross next to the little drop-down (top-right of Figure 2-5) to remove the Bevel modifier, then switch to Edit mode and select the four vertices on the unaltered side of the original cube. Select **View ▸ Sidebar**, click the Item tab, and **Transform** the values, setting the **Y** value of the vertices to **0**. Figure 2-6 shows the process.

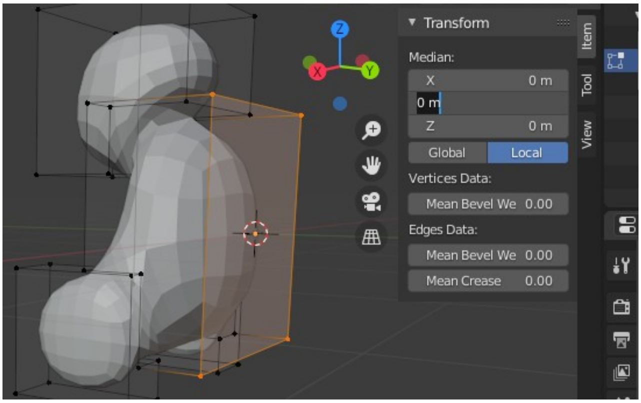


Figure 2-6 Alter the position of a selection of vertices.

7. Select then delete the face with these four vertices at its corners. When you press DELETE, you will get an additional menu to clarify which combination of faces, edges, and vertices you want to delete (in this case delete only the face, leaving the vertices and edges intact). If you use the RMB menu delete option, the choice will be more specific depending whether you are working with vertex, edge, or face select.

8. Select **Add Modifier ▸ Mirror** and set the axis to **Y**.

NOTE In general, the order of the modifiers is important and when applying them, you need to do it from the top down. However, for the Mirror and Subdivision surface, the result will be the same if the mesh is mirrored then subdivided or subdivided then mirrored.

Experiment with Other Modifiers

Try adding a few other modifiers to the stack. Most will have no visible effect because they need other components for them to work; however, the following exercise should yield interesting results. For each modifier, move it up the stack to see how the final result changes.

9. *Decimate* is a way to remove vertices but preserve, as far as possible, the original shape. Try **Add Modifier ▸ Generate ▸ Decimate**. A clearer display of the scene statistics is also available using the Viewport Overlays options (the drop-down next to the overlapping circles icon as shown in Figure 2-7) as well as other features to customize the 3D Viewport. With smooth shading, you can often reduce the number of faces quite significantly before the object becomes too misshapen, but notice that the faces become rather randomly shaped, which might be a problem for certain applications (as you'll see later).

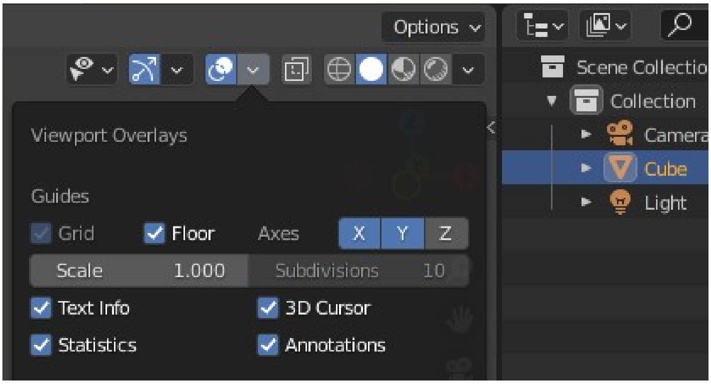


Figure 2-7 Viewport Overlays Statistics

10. In the Decimate modifier settings, switch to **Un-Subdivide**, change the **Iterations**, then switch to **Planar** and change the **Angle Limit**.

11. Delete the Decimate modifier then **Add Modifier ▸ Generate ▸ Remesh**. *Remesh* replaces the original mesh with one made from quadrilateral faces, the size of which is controlled by the Voxel Size. Be careful when reducing the Voxel Size, as you can easily overload your computer! (I always type a value rather than use the slider.)

12. Delete the Remesh modifier, then select **Add Modifier ▸ Generate ▸ Build**. Build is an animation modifier, so the object will disappear after you add it! Use the buttons in the Timeline Editor at the bottom of the window (Figure 2-8) to play the animation, which will create the object one face at a time.

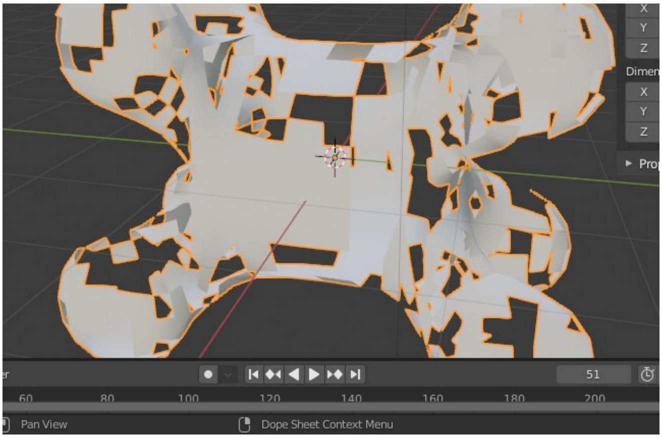


Figure 2-8 Build modifier showing the Timeline play button

13. Select **Add Modifier ▸ Generate ▸ Solidify** and increase the **Thickness** value to 0.1m. When you run the animation, Solidify will build the surface as solid blocks rather than an infinitely thin shell.

14. In the menu at the top of the 3D Viewport, select **Object ▸ Shade Smooth** to make the curved surfaces of the object look better, but at the expense of the sharp edges created by the Solidify modifier.

15. Try the other option **Object ▸ Shade Auto Smooth** which will preserve the sharpness of edges that are sufficiently acute. For many purposes this option will do the trick, but it leaves perfectly sharp edges that never exist in the real world. In later chapters, you'll use alternative methods that lead to greater realism.

Constructing a Complex Scene Using Modifiers

The rest of this chapter will round off your introduction to modifiers with a more elaborate work - a Martian habitat. The completed scene should look something like Figure 2-9.

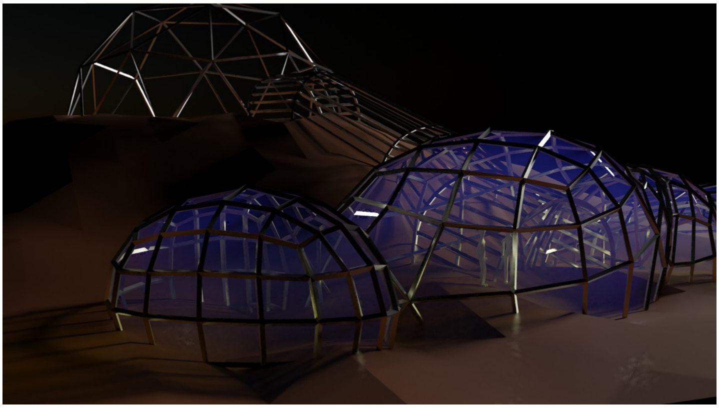


Figure 2-9 Partly completed Martian habitat

You will make the terrain and some habitat domes, but your first job will be to create simple manikins to populate them.

The Mirror, Skin, and Subdivision Modifiers

Starting with half of a stick figure, you'll use a mirror modifier to make it symmetrical, then a skin modifier to make it into a solid, but rather “boxy” figure. Finally you will add a subdivision modifier to generate a smoother result; it should look something like Figure 2-10 when you're finished.

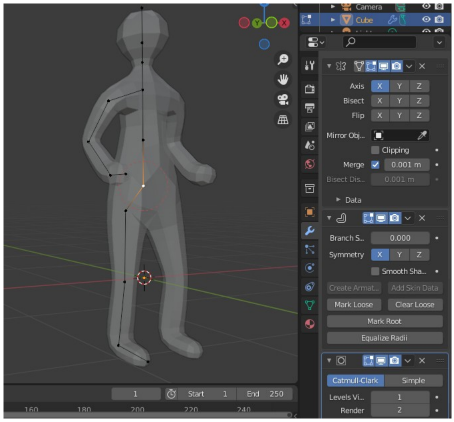


Figure 2-10 The completed manikin

1. Clear the scene and start a new one. Select the default cube, and then in Edit mode, delete seven of the vertices.

2. Move the remaining vertex to a location above the origin (X=0, Y=0, Z=1 for instance, but the exact height doesn't matter). Rather than use the Sidebar Transform to set the positions to zero, as you did in the previous exercise, toggle the **Snap** magnet at the top and set Snap Targetto **Grid** as shown in Figure 2-11. Use the X and Y orthographic views to position the vertex above the origin.

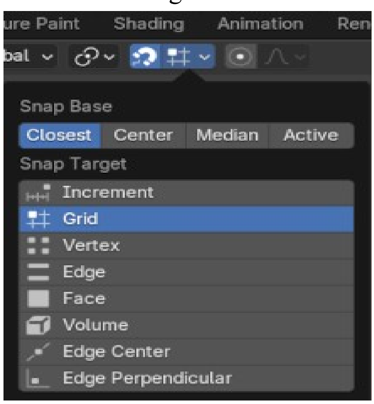


Figure 2-11 Using Snap to position vertices

3. Select the single vertex, extrude it foure times in the Z direction for the lower spine, upper spine, neck, and head. Then, using the orthographic views, extrude sideways and downward from the shoulders to create an arm and from the lower spine to form

a hip and leg. You'll apply a mirror modifier, so you need to do only one side. Although Blender can mirror in all three directions, the default direction is –X to +X, so extruding sideways in the X direction will make your life easier later on.

Switch back to Object mode and select **Add Modifier ▸ Generate ▸ Mirror** (setting the correct axis to match the orientation of the stick figure). Next, select **Add Modifier ▸ Generate ▸ Skin** and then **Add Modifier ▸ Generate ▸ Subdivision Surface**.

4. Return to Edit mode and select all the vertices (toggle **XRay** to be able to see them through the skin), then select **Mesh ▸ Transform ▸ Skin Resize** to make it a reasonable diameter. Selecting a few vertices at a time, adjust with **Skin Resize** to form head, shoulders, knees, and toes (you will find it much more convenient to use the shortcut CTRL-A at this tage). When the manikin looks reasonably proportioned, move it out of the way for later.

The Displacement Modifier

The surface of Mars where the research station is being built is a small crater: a gentle hill with a depression at its top.

1. Add a plane to the scene, scale it up about 40 times, then switch to Edit mode and subdivide it about 20 times. In **Adjust Last Action** for subdivide, you can type in a value, but the slider is limited to a maximum value of 10, so if you use the slider you will have to subdivide the plane twice.

2. The Displace modifier can move vertices by a fixed amount or an amount controlled by a Texture, so you can use that to make your terrain uneven. Back in Object mode, select **Add Modifier ▸ Displace ▸ Displace**, and in the modifier settings, click **New** to add a texture.

3. Select the Texture Properties tab (pink checkerboard icon). Select the texture just created in the modifier (probably named “Texture”), and choose **Clouds** for the **Type of Texture**. This will produce a very bumpy plane, so go back to the **Modifiers** tab, reduce the **Strength** to around 0.1 and apply the modifier.

4. Switch back to Edit mode and toggle proportional editing to on, then select one vertex and move it up so it creates a “hill” over

half the area of the plane. Reduce the radius of the proportional editing influence and move the pixel down to create the crater. Remember to turn proportional editing off again.

Extruding a Cylinder

The habitat is made from geometric shapes “sunk” into the terrain to form domes with an arched corridor linking them. The lower domes have been glazed and pressurized with air but the top dome and corridor are still under construction.

1. In Object mode, add an icosphere, then scale and position it in the crater so it forms a dome. Setting the viewpoint along each axis while positioning objects or vertices is the easiest way to do this step. The advantages of this method are that the view is orthographic (no perspective), so you can easily see what lines up with what, and any movement or rotation (but not scale) are constrained to a plane parallel with the screen.

2. Add a cylinder and scale, rotate, and position it so it's “submerged” half into the terrain with one end inside the dome. This will be the start of the corridor linking the top dome with the lower domes. The cylinder's axis should be parallel with the surface.

3. Switch to Edit mode with **face select** enabled, select the end of the cylinder, and extrude it into a series of short sections following the slope of the hill (Figure 2-12).

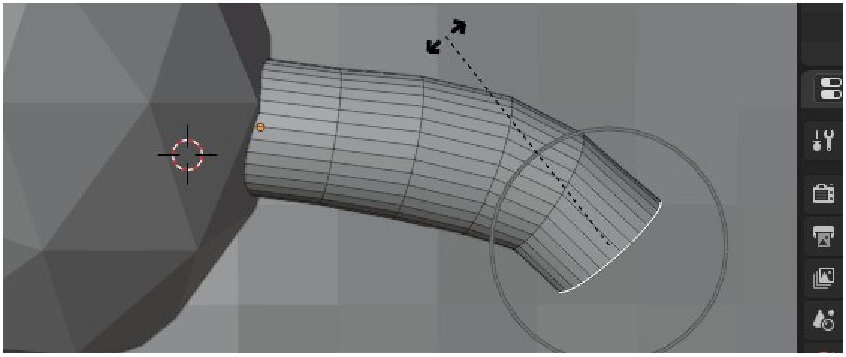


Figure 2-12 Extrude and steer the corridor.

4. “Steer” the corridor by rotating and moving the end face when viewing from above and follow the contours of the hill by viewing from the side.

Combining a Collection of Objects

For the lower domes, you'll start by using a Boolean modifier to join together four cubes, but rather than re-using a single extra cube, you'll move the four cubes into a Collection and combine them in one step. Sometimes assembling more than two objects at once is useful, and this is a flexible way of doing that.

1. Add four cubes near the lower end of the corridor, scaling and moving each one so they overlap each other corner to corner. Select all four cubes then select **Object ▸ Collection ▸ Move to Collection ▸ New Collection**. Select just one of them (to become the combined object), then **Add Modifier ▸ Generate ▸ Boolean** (Union), and finally switch **Operand Type** to **Collection** and select the one you just created (Figure 2-13).

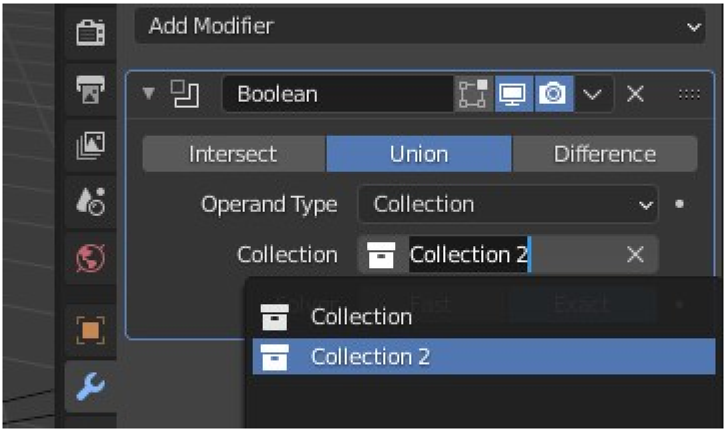


Figure 2-13 Boolean modifier acting on a Collection

2. After applying the modifier, delete the new Collection to tidy everything up in the Outliner Editor. That step won't delete any of its contents, in this instance the four cubes, but it will leave them outside the original Collection. Delete each of the three un-merged cubes, and drag and drop the combined object back into the original Collection.

3. Select **Add Modifier ▸ Subdivision ▸ Generate ▸ Surface** then scale and move the intersecting domes so they overlap the bottom of the corridor and are half submerged in the terrain so they look like domes.

The Wireframe Modifier

The *Wireframe* modifier works in a similar way to the skin modifier you used for the manikin, but it's more suitable for surface meshes. It converts the edges of objects into 3D “wires” and removes the faces. The resulting object is a cage resembling the framework of a glass-house prior to any glass being fitted..

1. Use a Boolean union modifier to add the corridor to the top dome apply the modifier then delete the corridor.

2. Select **Add Modifier** ▸ **Generate** ▸ **Wireframe** to the combined object, and add the same modifier to the lower domes.

3. Select the upper domes, then the Material Properties tab, then **+ New** to add a material, increasing metallic value and reducing roughness.

4. To check what this material will look like in the final render, change the Render Engine from **Eevee** to **Cycles** in the Render Properties tab (white back-of-camera icon).

5. Change the light from a point source to a “Sun.” Select the light object either in the 3D Viewport or the Outliner Editor area. The default light is named “Light” and its type is Point, so click the Object Data Properties tab (the green lightbulb in Figure 2-14). Under Light, select **Sun** and change the power to **4.0 W/m^2**. Rotate the direction of the sun to be more horizontal using **R** in the 3D Viewport.

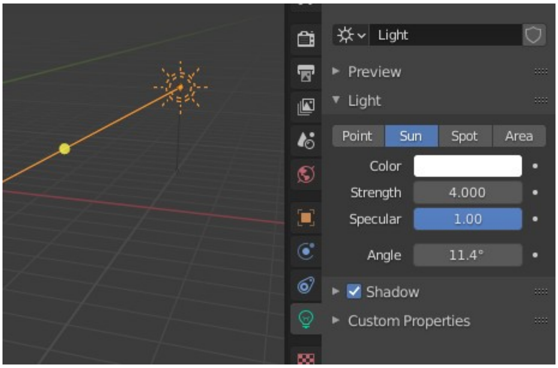


Figure 2-14 Sun settings

6. With the viewport shading set to render, select the upper domes object, tweak the material settings, and see the result.

7. Select the terrain, add a material, and change its **Base Color** to a suitably Martian orange.

Copy and Paste or Duplicate

Pressing CTRL-C then CTRL-V is the general way to copy and paste in most applications, so using that method means one less thing to remember, but if you try right-clicking an object to access the Copy context menu, you'll notice that Blender has an option to “Duplicate selected objects and move them” with shortcut SHIFT-D. To make several copies of an object, that method is slightly quicker and is the standard way you will see objects copied on “how-to” videos online. However, there are subtle differences between the two methods: copy/paste will make duplicates of materials you have in any slots on the copied object, but duplicate won’t.

1. Select then move the manikin you created earlier down to stand on the terrain in the lower domes

2. Copy/paste and rotate it, so you have two characters in conversation.

Viewing the Scene in Camera Walk Mode

As you did in Chapter 1, it’s really useful to “fly” around using the camera walk mode (SHIFT-`) with viewport shading still set to render to see what the scene will look like. To be honest, it’s not very convincing! One way to make it more realistic would be to add some atmosphere to the lower domes to differentiate them from the low-density air outside.

USE MEANINGFUL NAMES

Now is a good time to correct something I ought to have told you to do right from the beginning: change the names of objects from Plane, Plane.001, Cube.003, and Icosphere to more meaningful names like Terrain, Person01, Lower_domes, and Upper_domes. To change them, double-click a name in the Outliner Editor and enter the new name. A menu option and shortcut are available for renaming objects, but the double-click system applies to renaming materials, textures, vertex groups, and so on. Giving objects sensible names as soon as they are created is an important habit to get into.

Another way to improve the rendered scene is with texture and scattered rocks on the terrain. You can do that using geometry nodes, which was used by Quick Fur in Chapter 1, and will be described in Chapter 14, or particle systems, which we'll cover in Chapters 4 and 5.

Volume and Surface Materials

One fortuitous feature of the Wireframe modifier is that you can use it to create a frame but at the same time keep the original mesh, assigning a different material slot to each. In this case we'll use a surface material for the frame so it appears to be metallic, but we'll use a volume material with no surface material for the original object. This will result in light passing through the domes but scattering from the volume within, just as might happen with a glass house containing a damp atmosphere.

1. Select the lower domes object, then the Modifiers tab, and look down at the options on the Wireframe modifier you added previously (Figure 2-15). One checkbox defaults to Replace Original object when the wireframe is created, so un-check that. Another option, Material Offset, allows one material to be used for the wireframe and a different one for the original object. This enables you to set a volume shader to give the effect of light scattering from an atmosphere but still keep the frame as a solid surface, so set this to **1**.

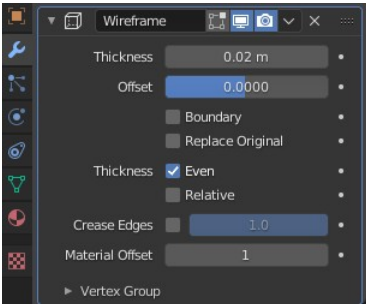


Figure 2-15 Wireframe modifier

2. Switch to the Material properties tab (pink and black sphere, shown in the bottom center of Figure 2-16), add a new material, and rename it to **atmosphere**. Add a New Material Slot with the **+** button (top right of Figure 2-17), then with the drop-down beneath the list of slots, **Browse Material to be linked**, select the material you used for the top dome and corridor, which should still be called Material.001. Rename it to **alloy_frame**. With the **atmosphere** Material Slot selected, change the default Surface shader from Principled BSDF to **Remove** and the Volume shader to **Volume Scatter** as shown in the composite Figure 2-16.

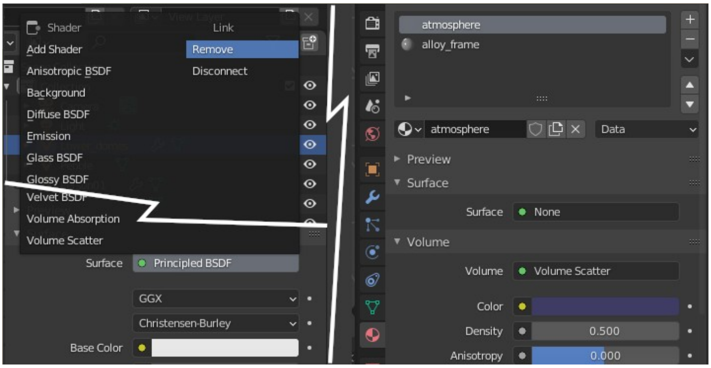


Figure 2-16 Change Material Surface shader to None and the Volume shader to Scatter.

Changes made in the Material Properties tab are reflected in the Shader Editor area and visa versa. This is the reason there is no option “None” available when you select a shader. If you view the nodes in the Shader Editor, it is clear that there are two options: the node can be deleted from the graph or else the link connecting the node to the Material Output node can be disconnected.

Adjust the color and density of the atmosphere material in the Material properties tab, and if you need to further refine the frame's Base Color, Metallic, Roughness, or other inputs, select the material in the second slot to adjust those settings. To speed up the process of fine-tuning the shader settings, you can define a small render region.

1. Select the Output Properties tab (white printer icon, second from top in Figure 2-16) and tick the **Render Region** checkbox.

2. Toggle to **camera view** in the 3D Viewport, then set a small rectangle to render: select **View** ▸ **View Regions** ▸ **Render Region**.

3. If you're not trying to get an absolutely photographic result, switching on Denoising and reducing the number under Sampling in the Render Properties tab is often worthwhile.

NOTE In "The Wireframe Modifier" on page 39, I suggested you switch to use Cycles rather than Eevee rendering. Eevee is the default, generally faster method, and it is possible to use it for volume shading. However, until Blender 4.4, Eevee could be applied only to a bounding box, not the actual 3D shape of the object, so it would be incapable of doing what we want here. The later versions of Eevee with ray-tracing are very good in most circumstances.

---

4. Adjust the **Surface Color** for the background in the World Properties tab (pink globe icon) to a dark blue.

I know the photos sent back by the various Mars rovers show a surprisingly Earth-like sky, but I decided the contrast of the dark sky with the volume shading of the habitat looked better. An alternative you could experiment with would be to set the Surface Color to Sky Texture type Nishita, and fix the sun elevation and strength as well as air, dust, and ozone density to create a generally dark sky with a slight glow near the horizon.

When working with materials, keep in mind that objects have a list of "slots," each capable of holding a material, but the materials themselves are held centrally so multiple objects can use them. The pool of available materials is accessed from the little drop-down below the list of slots (Figure 2-16). In the next chapter, you will see how different materials can be applied to different groups of vertices or faces.

Conclusion

You now should have an inkling of how useful modifiers can be. You should know where to find them and understand what at least a few of them can do. In the remaining chapters of this book, you'll use modifiers to do a host of jobs, and it’s no exaggeration to say that their mastery is crucial to becoming a competent Blender user.

While getting to grips with modifiers in this chapter, you also encountered some unrelated, but important, snippets of information about materials. Not only is is possible to reuse the same material for several objects, but each object also can utilize multiple materials. The inner workings of the Wireframe modifier hid the details, but in the next chapter, you'll explicitly assign different materials to different parts of the mesh.

You also applied a fundamentally different type of shader to the volume of the material output and disconnected the surface shader completely. I mentioned in passing that the node view reflected the relationships between the different components of the material in a more logical way. In later chapters, you will work with more complicated materials, and you will come to appreciate the benefits of the node view of the Shader Editor.

In the next chapter, you will use some Blender objects that are not meshes. Instead of being constructed from vertices, edges, and faces, you will use curves and surfaces that have been defined by curves.

3

NON-MESH OBJECTS

Exploring surfaces and curves defined by functions

In the first two chapters, you learned how to switch from Object view to Edit view, and when editing you could see that each object was constructed from vertices, edges and faces. Objects of this type are meshes, but when you added them you will have noticed that Blender has many other types of object beyond meshes, cameras and lights.

In this chapter you will add curves and text to your repertoire and find alternative, better, ways to create some of the objects you worked on in the first two chapters

Curves

Firstly a word of reassurance: In the chapter subtitle I announced that curves were defined by functions, but those functions are internal to Blender and you don't need to do anything mathematical in order to use them! Think of the difference between meshes and curves as similar to the difference between dot matrix fonts and TrueType fonts. When you zoom in on a mesh the change in angle of edges at each vertex becomes more and more noticeable but with a curve, as the magnification is increased the resolution can also increase so that it always looks smooth.

Convert Text to Curve

To get a bit more of a feeling for the relationship between meshes and curves, and to witness the simplicity of converting one type of object to another, you will generate a curve using text. Later in this chapter you will learn how to manipulate Text objects but for the moment you are just using this as a quick way to create complicated curves.

1. Start Blender and delete the default cube. then select **Add ▸ Text** which will create a Text object with the word "Text", notice that the icon in the Outliner Editor is a letter "a". Convert the text with **Object ▸ Convert ▸ Curve** which will appear to do nothing but the icon in the Outliner Editor will have changed to a curve. Switch to Edit mode and select the Object Data Properties tab which looks like a green curve connecting two little squares, visible at the bottom of Figure 3-1.

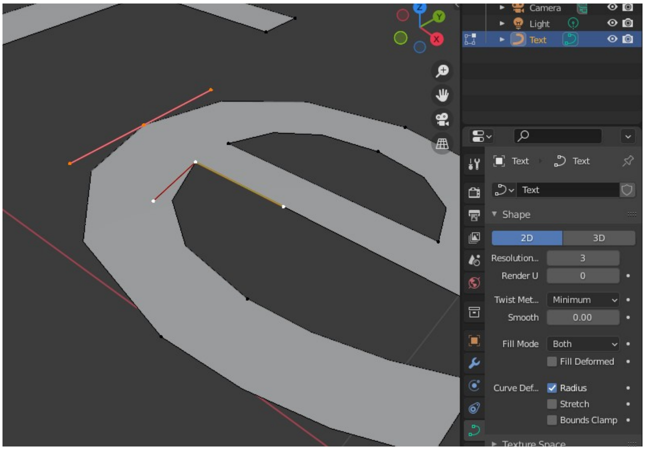


Figure 3-1 Editing Curves

2. If you zoom in, you will begin to see that the curve is constructed from straight edges which would seem to contradict my earlier explanation. However in Blender, rather than the resolution changing automatically with the view, it can be adjusted to suit the needs of the user. See how it changes as you increase and decrease the Resolution in the Object Data Properties tab. When you subsequently convert a curve into a mesh this value will control how many vertices are generated between the control points.

Control Points

3. To see how the curve is controlled by the different types of control point, select a control point in the middle of a curve then select a different one at a sharp corner, such as the ones selected in Figure 3-1. You will see that handles (the lines controlling the shape of the curve) are different colors: pink is used for Aligned which is the default type, red is Free and yellow is Vector, you can change the handles using **Control Points ▸ Set Handle Type**.

4. Experiment with moving, scaling and rotating the control points and the ends of each handle to get a feel for how you can use them to obtain the required curve. Because the curve is set to 2D all the transformations are confined automatically to the XY plane. If you toggle 3D the fill will disappear but, unsurprisingly, you will be able to form the curve into a 3D shape.

Beveling Curves

Your first real exercise of this chapter is to construct a ring and a mask woven from gold wire each one made using curves. You will learn how to create realistic reflections and lighting using an Environment Texture as the background - much simpler and more realistic than the synthetic window you constructed in Chapter 1. The final render should look like Figure 3-2. Links to the blend files and videos for the exercises in this chapter are at *get-into-blender.com 3.NON-MESH OBJECTS*

Figure 3-2 Golden rings and a wire mask

You will need two curves to define the ring. One for the axis of the shape, which will be a simple circular curve, and one for the profile, or section, to be extruded around the circle.

1. Start a new file and add two curves to define the ring. Select **File ‣ New ‣ General** (there is no need to save your work), delete the default cube then select **Add ‣ Curve ‣ Circle** for the ring and **Add ‣ Curve ‣ Bezier** for the profile.

2. Modify the profile curve to suit your ring design. With the Bézier curve selected switch to Edit mode and set the view to orthographic from above, the Z direction. Now create an interesting profile for the ring. I decided that it would be a token of love given to a steam engine enthusiast so I made it in the shape of train track but you can choose anything you like, presumably the inside should be smooth against the wearer's finger. Use E to extrude the end control point then G, R and S to form the shape. My rail section is visible in the center of Figure 3-3.

3. In order to make the ring solid you need to join the ends up using **Curve ▸ Toggle Cyclic**. When you are happy with the profile, still in Edit mode, position and rotate the curve so the outside half of the profile is to the left of the Y axis and the inside half to the right, as in Figure 3-3.

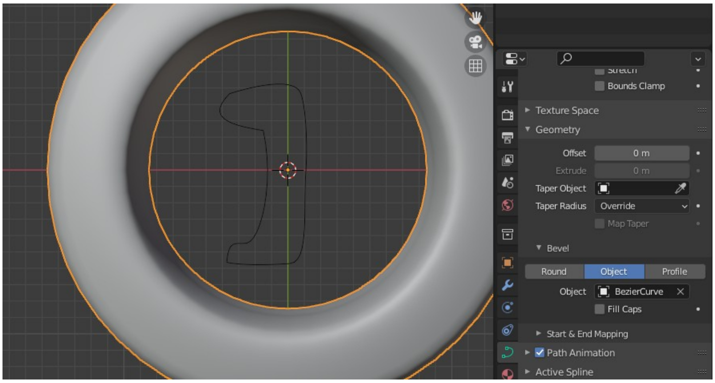


Figure 3-3 Profile along the length of a Curve using an Object for the Bevel

4. To make one curve use the a different curve as a profile, you need to specify it as a Bevel. In Object mode select the Bézier circle and in the Object Properties tab open Geometry then Bevel and select **Object** then choose the Bézier curve, shown on the right of Figure 3-3.

Making 3D Text

You now have a 3D surface in the shape of a ring but, to make it really special, add a message on the outside.

5. Create the text as a new object. Select **Add ▸ Text** then switch to Edit mode, delete “Text” and replace it with a word of your choice.

6. You can make the text bend round a curve using a modifier. Back in Object mode open the Modifier Properties tab and **Add Modifier ▸ Deform ▸ Curve** choosing the Bézier circle as the Object.

7. Now give the text some depth. In the Object Data Properties tab (green “a” icon) under Geometry, increase the extrude amount.

8. In order to orientate the text correctly open the Object Properties tab (yellow square, just above the wrench) and set X rotation to **90** and Z rotation to **180** and tweak the Z and Y Location and the Scale so that the text protrudes on the outside but not on the inside.

9. When it all looks good select **Object ▸ Convert ▸ Mesh**

Correcting Normals on Curved Surfaces

If you look at some of the letters with curved edges you will see that the combination with the curvature around the ring has produced ugly edges running across the letter, even if you change to **Object ▸ Shading Smooth** the effect is still visible and, possibly, made worse. You can improve this a lot by:

10. Switch to Edit mode, select all vertices then **Face ▸ Beautify Faces**. The rearrangement can be seen clearly in Figure 3-4

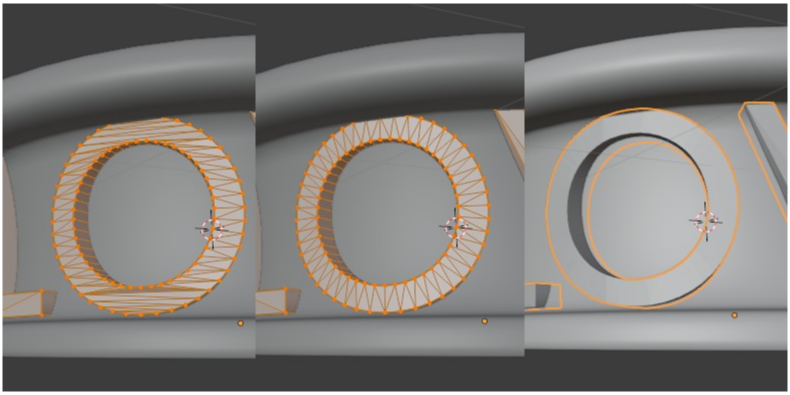


Figure 3-4 Beautifying an ugly arrangements of faces

The problem of ugly lines running across faces that should be flat, or very gently curved, is a perennial issue with 3D modeling, especially with hard surfaces such as needed for modeling machinery, furniture or jewelry. The general solution is to add extra vertices next to sharp edges to confine the effect of the curving surface normals to the part of the mesh near the edge, often by adding a bevel.

NORMALS

The normal represents the direction that the surface is facing at any any given point which effects shadows and reflections. When Shading Flat is selected then the normal direction is the same for the whole of each face and there is a sharp change at the edge. Shading Smooth, on the other hand, makes the normal change gradually across each face so that there is no sudden change at the edge.

One thing to note here is that when you extrude a 2D curve then convert it into a mesh, as you just did with the text, Blender creates separate meshes for the front, back and sides. The benefit of this is that it solves the problem of the normals softening the edges when using smooth shading but, as I mentioned in the last chapter, it produces unrealistic looking sharp edges.

11. In order to see the effect of the following changes, set shading to smooth. In Object mode select **Object ▸ Shading Smooth**.

12. In order to add a controllable bevel to your lettering you first need to merge the meshes together. In Edit mode select all the vertices then **Mesh ▸ Merge ▸ By Distance** which should report, at the bottom of the screen, that a number of vertices have been deleted.

13. Back in Object mode add a Bevel Modifier and increase the Segments to **3**. Figure 3-5 shows the three stages moving from split edges to continuous mesh to a beveled edge.

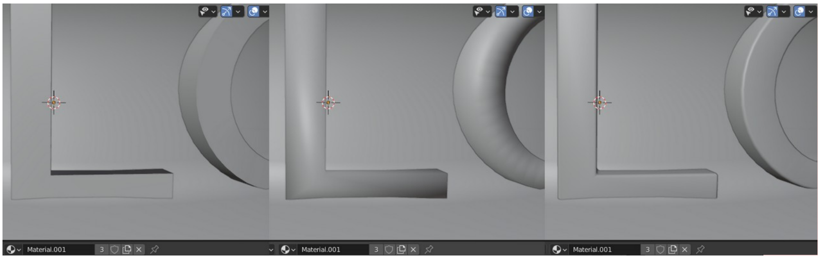


Figure 3-5 Joining split edges and adding a bevel

As it stands the text and ring are not joined together which makes it inconvenient to move and rotate them in the scene. But there is a problem; in order to join them as you did with the ring and the mount in Chapter 1, or to use a boolean modifier as you did in Chapter 2, you would have to convert the Bézier circle into a mesh.

Parenting Objects

The method you will use here to effectively combine the two objects is to make the Bézier circle into the parent of the Text object.

14. Select both objects, *but in the right order*. In Object mode select the text *then* select the Bézier circle as well using SHIFT-LMB. The order of selection is important here: the Text should be highlighted in a slightly more orange color than the Bézier circle object, which is “active”.

15. Now set the parent child relationship. Click **Object** ▸ **Parent** ▸ **Object** and you will see in the Outliner Editor that the text object has moved inside the hierarchy of the Bézier circle object. When you move or rotate the ring, the Text stays attached.

This is the alternative method I referred to in Chapter 2 when you applied the Boolean modifiers to make the coffee cup. You could instead have parented the smaller cone, the torus and the cylinder to the larger cone, then hidden them in the Outliner Editor. This technique is more flexible but at the expense of complexity and the potential to inadvertently move a carefully positioned object.

16. Add a gold material following the same steps you used in Chapter 1. Because the ring and the text are two different objects you will have to specify the material for both of them but as you can select the same material for multiple objects you only need to set up the material properties once.

Drawing a Curve onto a Surface

To construct the mask you will make a mold in the rough shape of a face then draw Bézier curves onto the surface.

1. Create the mold object. Move the ring to one side for the moment and select **Add** ▸ **Mesh** ▸ **Ico Sphere**, scale it into a flat oval shape then, in Edit mode, move vertices to form a ridge for the nose, brow and lips and indents for the eyes. It only needs to be very approximate.

2. Add a new curve object and set it up ready for drawing wires. Create a new Bézier curve object with **Add** ▸ **Curve** ▸ **Bezier Curve** and in the Object Data Properties tab under Geometry

and Bevel (the right hand part of Figure 3-6) select **Round** for the Bevel geometry with Depth set to **0.4** and Fill Caps on.

3. Each time you draw will create a new length of wire so you don't need the original three control points and can delete them. Switch to Edit mode, delete all the existing vertices then select the Draw tool from the Toolbar. To make the lines draw onto the surface of the icosphere you need to open the Sidebar and in the Tool tab set the Depth to **Surface**. Draw by LMB and dragging the mouse as in Figure 3-6.

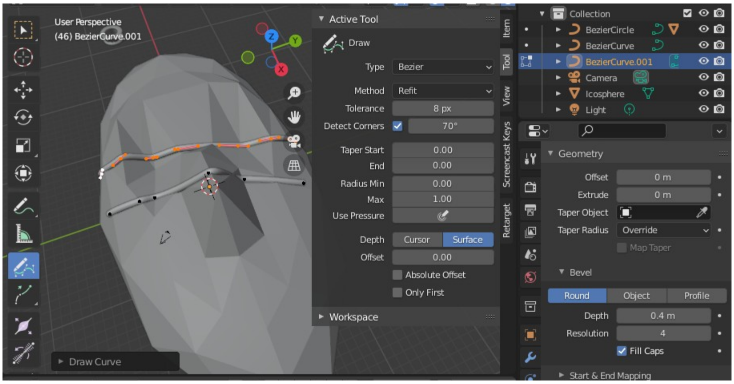


Figure 3-6 Setting Draw Curve onto a Surface

4. Draw curves in alternating directions to give the impression of weaving and, for added interest, alter the Radius of some of the wires. If you just change the Depth of the Bevel you will see that all the wires change, which is not what you want. However, in the Toolbar you can select the Radius Tool and change the size of specific control points as shown Figure 3-7.

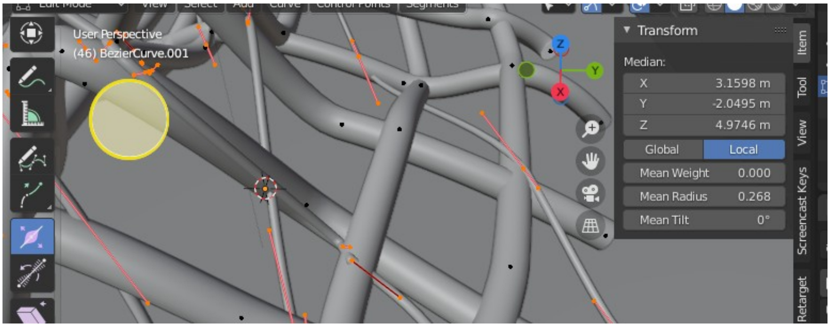


Figure 3-7 Changing the Radius of control points

5. To change the thickness of several wires you need to select one control point on each wire you want to change then expand the selection to all the control points on each wire by pressing CTRL-PLUS several times, or better, **Select ▸ Select Linked** . This process is sufficiently repetitive that you will quickly start using the shortcuts CTRL-L for selecting and ALT-S for scaling the radius.

6. Hide the icosphere you used as a former, which you can do by un-checking the eye and camera icons in the Outliner Editor.

The Outliner Editor has several optional columns that control which objects are visible on the 3D Viewport or appear in the final render, as well as more complicated features required for compositing such as holdout, which you will learn about in Chapter 13. The drop-down menu for setting these is called Filter and has an icon that looks like a funnel in the Outliner Editor header.

NOTE On occasion there can be too many buttons and options to fit into the header, for instance if you move the divider of the 3D Viewport to the right, or when using Sculpting mode or Texture Paint mode. When this happens hold the mouse pointer over the header and scroll using the mouse wheel.

---

7. Set the Material to the same gold you used for the ring.

8. Add a Cube object as a display plinth with a suitable Material. If you look at Figure 3-2 you will see that I adjusted the Subsurface value and Color as well as the Roughness to give a plastic effect.

Environment Lighting

The final touch, which you can also see in Figure 3-2 is to use an Environment Texture for the background and set this to provide the illumination and reflections for the scene. As a rule, using the background texture as illumination gives a much more realistic effect than even complicated lighting arrangements, and it’s easy to set up.

9. Open the World Properties tab of the Properties Editor (pink globe icon) and click on the little yellow dot next to Color under Surface then select **Environment Texture**

10. Now you need to use the Open button to find a suitable image file. If you are on Windows and used the installer then browse to *C:\Program Files\Blender Foundation\Blender 3.03*, otherwise to the location where you unzipped the Blender download. Inside that folder browse further to *3.03/datafiles/studiolights/world/interior.exr* where the first directory will the same as the version of Blender that you downloaded, in this example 3.03. There are a few textures included with Blender but you will be able to find many online by searching for "HDRI environment". Software is also available that will allow you to create a bespoke background by stitching together your own photos.

The simplest way to get the full effect of the the shadows and reflections is to use Cycles rendering but you will probably find that, even with denoising switched on, you need to increase the Sampling in the Render Properties tab to get rid of the "fireflies". Eevee will be nearly as good but in the Render Properties you must turn on **Raytracing**. When you are happy with the overall effect, render the scene and save the blend file.

Surfaces

There are several ways to use Curves to generate graceful surfaces very efficiently, and for the rest of this chapter you are going to try three different approaches to model the shape of a sailing boat hull and sail. But, before you launch into the exercise, search online for "lofting sailing boat hull" and get an idea of the shape you are aiming for. It doesn't have to be precise but essentially the profile should have a very narrow V shape at the prow, opening out to a fairly flat bottomed U shape in the middle and stern.

Curve Tilt and Radius

For the first approach to forming a hull you will try using a variation of the method you used for the ring, but adjusting the "Tilt" as well as the radius of each control point to create a twisted shape.

1. First of all add a profile curve to use as the bevel object. Save your last work and start a new blend, delete the default cube and add a Bézier curve, renaming it "rib".

2. Shape the rib so it looks like half of a boat section, one end needs to be exactly on the origin. Switch to Edit mode and view

in orthographic mode looking from the Z direction. Move one control point to the origin with X and Y both zero using grid snap absolute. Remember you can use CTRL while moving to do this very easily when you need to position a single vertex to a grid point.

3. Move the other point so it is above and left of the origin then rotate and scale the points so the line forms a smooth curve. You can see the rib curve at the top left of Figure 3-8.

4. Create a curve for the keel and orientate it so that a mirror modifier in the X direction will create the full hull shape. In Object mode add another Bézier curve, rename it "keel" then in Edit mode, with both Control Points selected rotate it so it lies along the Y axis using R, Z, 90 then LMB followed by R, Y, -90 then LMB If you rotated the rib or keel in Object mode then you will find everything behaves rather strangely but you can remedy it by **Object ▸ Apply ▸ All Transforms**.

5. With the keel selected, in the Object Data Properties tab, set the Bevel to use the rib object.

6. Now you need to modify the size and rotation of the bevel to make it look like a boat. In Edit mode select all the control points and **Segments ▸ Subdivide** so you have four or five in total. Try to tweak each point to form a boat shape by adjusting the Radius, Tilt, Z position, X rotation and Scale.

Figure 3-8 shows my attempt. You will probably find that this method, though good for some shapes, doesn't provide the flexibility required for a boat's hull.

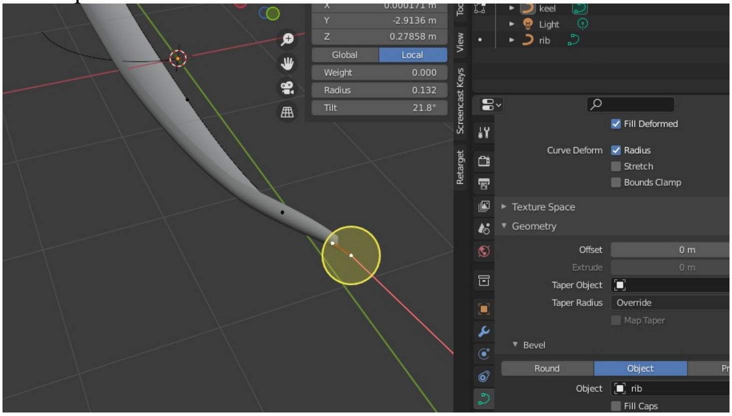


Figure 3-8 Using Tilt and Radius to form a boat shape

Don’t struggle on for long, the next method will give you more flexibility to design surfaces of this type.

Nurbs Surface

For the second attempt at forming a boat hull you will use a nurbs surface. This can be very powerful, but getting the shape you want takes a bit of practice.

1. Delete the rib and keel objects then add a nurbs surface with **Add ▸ Surface ▸ Nurbs Surface**. Switch to Edit mode and you will see a relatively small surface suspended inside a frame of sixteen points.

2. It will be easier to construct a shape for half the hull (you will add a mirror modifier) if you make the surface reach right to the outer control points. Open the Object Data Properties tab (green surface icon near the bottom of Figure 3-9) and under Active Spline tick the Endpoint U and V options.

3. Now move all the points to one side of the origin by selecting all the points and moving them to position one edge of the surface along the X = 0 plane. Move them approximately first, then select just the points on the mirror plane and set their X value using grid snap absolute then move along the Y axis while pressing CTRL.

4. It’s a good idea to add the mirror modifier now before you start moving the control points so can see the finished shape as you work.

5. Move the control points to form the hull. The relationship between point locations and surface shape takes a bit of getting used to, so don’t alter the Y positions of the points much, apart from at the top and bottom of the bow. To appreciate the 3D shape as you create it I suggest staying in perspective view and moving points with G then SHIFT-Y.

You will find it easier to make a streamlined shape using only the sixteen control points of the starting nurbs surface but in figure 3-9 you can see that, after forming the basic shape, I added some more points by extruding the edge twice in the V direction (the pink lines running across the boat) in order to flare the gunwale slightly and wrap the surface over to form side-decks. Leave the stern open, you will fill it after the surface has been converted to a mesh.

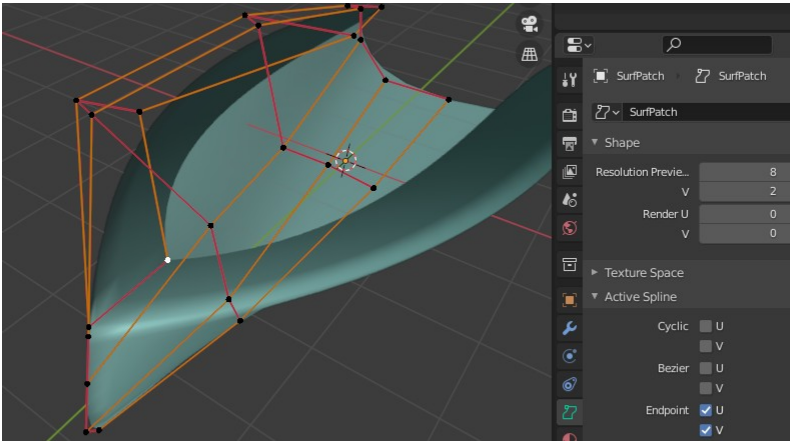


Figure 3-9 Boat shape using Nurbs Surface

6. When you're happy with the shape, use **Object ‣ Convert ‣ Mesh** which will also apply the Mirror Modifier.

7. To form the cabin structure, in Edit mode select the edge loop at the inside edge of the side-decks using ALT-LMB. In orthographic view with X-Ray on deselect the rear half of the vertices by pressing CTRL while box selecting them. Extrude upwards, scale inwards and rotate slightly about the X axis. You can see in Figure 3-12 the proportions I used.

8. Create a new face for the cabin roof by selecting all the vertices that define the edges (which should already be selected after the previous step) then select **Vertex ‣ New Edge/Face from Vertices**. This is such a common action that you will almost always use the shortcut F. This will create a single face which is often fine, you will look more closely at the different tools for building mesh topology in Chapter 7 but for the moment keep it simple – the boat will be too far away to see much detail in the final render.

9. Add the transom by selecting the edge loop of vertices around the open back of the hull and filling with a single face as you did for the cabin roof. This might result in some messy edges where the side-decks fold over but, for this exercise, you can leave them as they won't be visible.

Bridging Edge Loops

For the mast and sail you will convert a series of Bézier curves to meshes then join them using Bridge Edge Loops. At this point you can probably see that it would be very efficient and easy to make a curved and tapering mast from a Bézier curve with a round bevel and a realistic sail from a nurbs surface. However the technique of bridging a series of edges can be useful in many different places, in fact it would have allowed a method of construction similar to the "lofting" drawings of sailing boat hulls that you found online. You could have constructed a series of ribs, each with a different shape but the same number of vertices, with equal spacing between each rib, then built a surface joining them all together.

10. In Object mode add a new Bézier curve and rename it "sail".

11. One end of the sail curve will become the section of the mast. Use the orthographic view from the Z direction and switch to Edit mode then extrude the end control point twice so you can form the mast.

12. Change the control point where the sail meets the mast so the curve can bend sharply using **Control Points ▸ Set Handle Type ▸ Free** as shown in the detail of Figure 3-10.

13. Duplicate the curve twice, and move the copies up to the mid-point and the top of the sail. Scale, rotate and flatten the sail profile appropriately: the top of the sail will be narrower, flatter, and twisted more relative to the bottom. You will need to keep swapping the view between the Z and Y direction as you do this.

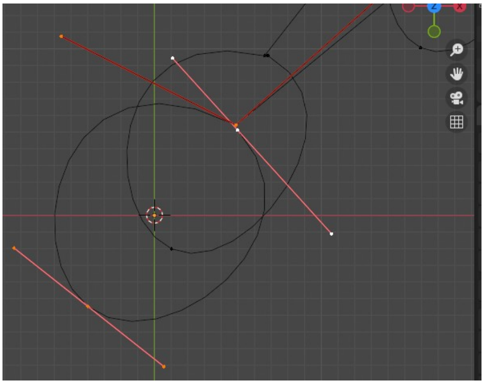


Figure 3-10 Sharp bend between sail and mast using Free Handle type

14. The curves need to be converted into three one dimensional meshes, essentially edge loops, before they can be bridged to form a surface. In Object mode run **Object ‣ Convert ‣ Mesh** then back in Edit mode select all the vertices then use **Edge ‣ Bridge Edge Loops**.

15. In the Adjust Last Operation pane you can increase the Number of Cuts so the generated surface is a bit smoother.

16. Finally scale and rotate the sail then position it on the boat, setting the boat as the parent of the sail as you did for the text on the ring earlier in this chapter.

Multiple Materials per Object

The default white material might not be a bad choice for a fiberglass hull or a sail but the overall impression would be much improved by differentiating the mast and the side-decks.

17. With the boat selected switch to Edit mode and select just the vertices for the side-decks - this is a good instance where selecting Edge Loops would be very quick and efficient.

18. You now need to assign a material to a selection of vertices. Still in Edit mode open the Material Properties tab then, if you have a Material Slot with a Material defined double-click and rename it “deck”. If there is no Material yet, add a new Slot then add a new Material and rename that. If you compare Figure 3-11 below with Figure 1-5 you will see that there is a line of buttons for assigning and selecting vertices by Material which is only visible in Edit mode. With the deck material selected in the list of slots, click **Assign** to associate the Material to the selected vertices.

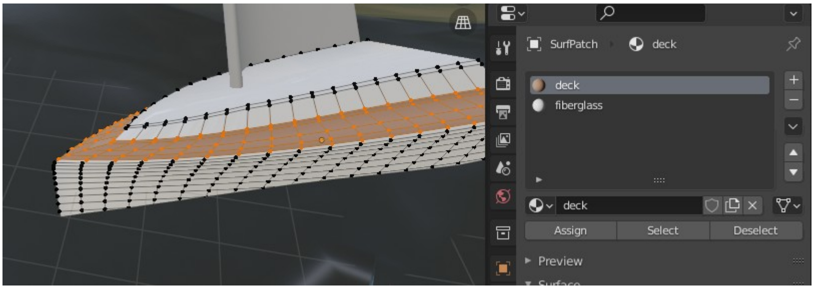


Figure 3-11 Assigning Materials to Vertices

19. Adjust the Base Color, Roughness and other shader parameters to suit the scrubbed teak look of the deck Material.

20. Now create a different material to assign to the parts of the boats that are not deck. Add a new Material Slot and a new Material and rename it “fiberglass”. With your cursor back over the 3D Viewport select the cabin roof and hull by using **Select ‣ Invert**, however if you are in vertex selection mode as in Figure 3-11 this will leave a gap between the vertices selected for the deck and the fiberglass. So change to face selection mode, deselect all faces with ALT-A then in the Material Properties tab select the **deck** material and click **Select**. Now when you invert the selection there will be no gaps and you can select the **fiberglass** material and click **Assign** with the correct faces selected.

21. Reduce the roughness of the fiberglass shader to make it look like plastic.

22. Go through the same process to allocate Materials for the sail and mast.

Ocean Modifier

To make suitable surroundings for your boat you can quickly generate some waves.

1. Create a plane to hold an ocean modifier. In Object mode use **Add ‣ Mesh ‣ Plane**, rename it “ocean”, then add the Ocean Modifier to it. In the settings increase Repeat to **5** in the X and Y direction, change the Size to **2.0**, the Spacial Size to **20** and, under the Waves section increase the Choppiness to **3.0**. Note that you don’t need to scale or subdivide the plane itself so long as the Geometry is set to **Generate**, the modifier does that for you, indeed you could could have added any type of mesh.

2. Add a Material for the ocean, change the Base Color to a dark blue-green, reduce the roughness and increase the Transmission to **1.0**

3. Check the view from the Camera and move the ocean so no edges are visible. Pose the boat so it looks suitably dynamic and rests on the surface. Alternating rotation using R and RR is a useful technique for this.

Rendering with Eevee

Because the scene lacks a lot of detail such as rigging on the boat or surface foam on the sea, this is a good example to try using the Eevee render engine, which is less photo-realistic but much faster. You can also add some atmosphere using environment lighting, haze and camera depth of field.

4. Use an environment texture again for background and lighting. Delete the default point light then add an outdoor environment texture, as you did for the ring and mask exercise. In Figure 3-12 you can see that I used *3.03/datafiles/studiolights/world/sunrise.exr* but turned the Strength down.

5. To add a slight haze as the waves recede into the distance, add a cube with setting the material surface shader to **None** and volume shader to **Volume Scatter**, as you did for the Mars habitat. Scale the cube up and move it so it fills the field of view from the camera. Position it so the boat is just inside, but on the camera side. Adjust the density to an appropriate level.

6. When using Eevee, in order to enable the boat reflection on the water, in the Render Properties tab tick **Raytracing**.

7. Finally, to make the scene slightly more photographic, select the Camera object and in the Object Data Properties tab (green cine camera icon) enable Depth of Field, reduce the Aperture F-Stop to around **0.1** and either manually adjust the Focus Distance or use Focus on Object.

Figure 3-12 Final render: haze at dawn

Using Eevee can often produce good results with only a slight reduction of realism and taking a fraction of the time of a Cycles render.

Try opening the blend file for the ring and mask that you saved earlier and change the render engine to Eevee, then experiment with the optional features in Render Properties to see which ones make a difference and how near you can get to the Cycles result.

Conclusion

Curves provide Blender with a quick method to generate graceful shapes that would be hard to make using mesh tools alone. In this chapter you very quickly dabbled with a few applications of curves and saw how they could be created from Text objects as well as being converted to meshes. As you cover different topics in this book you'll also find other uses for curves where they are used as "guides": in Sculpt and Paint mode they can define the stroke of a brush, the Curve Modifier can deform a mesh to align with it, as you used with the text on the ring in this chapter, and when animating, constraints can be set so that objects or bones slide along or deform to match curves.

When you converted the lettering to a mesh you saw some of the problems that can arise when trying to make large smooth surfaces with realistic edges and you used some of the techniques available in Blender to improve them.

You learned how different Materials could be allocated to specific sets of faces within a mesh and, for the final render of both exercises, you used an Environment Texture to provide the background, lighting and reflections for your scene. Both these techniques can be very effective for adding realism, as are the other two modification you made to the standard setup: Camera Depth of Field and Volume Scatter.

Finally, when you added the Ocean Modifier to the plane you will have noticed that there are still a lot more Modifiers I haven't mentioned yet. In the next chapter you will use two of these, the Array and the Curve Modifiers, to great effect. You will also encounter "empty" objects and, for the first time, use photographic images to texture the surfaces of the models you construct.

4

ARRAYS, MATERIALS AND TEXTURES

Making patterns of an object from a single instance

Blender has a surprising number of ways to generate multiple copies from a single object: Array Modifier, Instancing, the Spin Tool, Particle Systems and Geometry Nodes all allow this to be done. In this chapter you will use the first four.

In Chapter 2 you tried out several modifiers from the second of the four columns in the drop-down list but you may have noticed that I skipped over the first one in the list; Array. The reason for delaying its introduction till now isn't because the array modifier is less useful than the others, but there are some subtleties to its use that deserve more explanation.

The exercise for this chapter will be to construct a rusty steam-punk refrigerator and then to form an enormous sculpture by stacking different circular arrangements of the refrigerators in a parody of a Megalithic structure. The final result should look something along the lines of Figure 4-1

Figure 4-1 Scrap Henge

Downloading a Texture Asset

Before launching into the modeling process on Blender you need to get hold of an image file that you can use to texture the surface of your refrigerator. I used the phrase "free download rusty metal seamless texture" in an online search and found lots. The "free download" part is to help find images that are not subject to tight commercial conditions and restrictions, but read the terms and conditions anyway when you find a good looking website. "seamless" just means that the image can be tiled edge to edge without showing a discontinuity at the joins. Of course you could just take your own photo and make it seamless with almost any image editing software. Save the image file in the folder system where you intend to store the blend file for this chapter's exercise.

The ability to find resources online is definitely one worth cultivating, but if you don't find anything to you liking you can download the image I used from *get-into-blender.com 4.ARRAYS, MATERIALS AND TEXTURES*

A Material with Multiple Shaders

Making a convincing scrap refrigerator Material will be your first task. Using the rust image for the base color gives the impression of mild steel plates, but what we want is the original white or silver finish with rust showing through around the corners and edges, which get more wear. One way of doing this would be to assign different materials to different vertices as you did for the boat in the last chapter, but that gives a hard division between the two Materials instead of an uneven pattern where the rust is gradually eating into the original finish. A better approach is to create a single Material with two Principled BSDF Shaders which are combined together using a Mix Shader where the factor for mixing will take into account the "pointiness" of each vertex as well as having some noise added to make it look more natural. Links to the blend files and videos for this chapter are at *get-into-blender.com 4.ARRAYS, MATERIALS AND TEXTURES*

8. Start Blender, leave the default Cube selected and change the workspace to **Shading** using the tab right at the top of Figure 4-2.

9. You will have the default Material visible as nodes in the Shader Editor area, rename it now to "rusty_fridge" either in the Material Properties tab or at the top of the Shader Editor.

Bumps Using the Normal Input

Your material will have two shader outputs, one for the rust and one for the smooth surface. The rust shader will need the rust image for the basic color but you can also use the image to create a pseudo bumpiness. In the Shader Editor you will add three nodes and connect them to the Base Color and Normal of the default Principled BSDF Shader as I have in Figure 4-2

10. Add a node to supply the rusty texture. Select **Add ▸ Texture ▸ Image Texture** then click on the little folder icon and browse to the location of your rusty metal image.

11. The brightness of the of the texture will be a stand-in for the height of the bumps, this can be extracted using a different node. Use **Add ▸ Converter ▸ Separate HSV** and connect the Color output of the image texture node to the input.

12. The brightness values need to be converted into a normal map using a bump node so use **Add ▸ Vector ▸ Bump**. Connect the Value output of the HSV node to the Height input, and the output to the Normal input of the shader. This synthetic bump image uses the value output but you could experiment with the hue or saturation, alternatively there is a node to convert to RGB and it might be interesting to try each of those outputs, choose whichever gives the most interesting result.

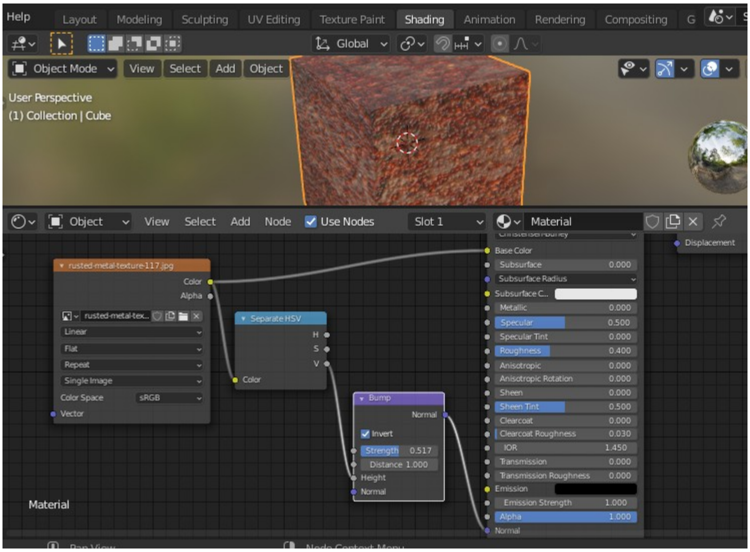


Figure 4-2 The first part of the rusty_fridge Material

The Mix Shader

13. Once you're happy with the basic rust, add another Principled BSDF Shader which will become the non-rusted part of the material. Also add a Mix Shader to combine them.

If you add the mix shader first you can "drop" it onto the pipe connecting your existing shader to the Material Output node, and this will automatically rearrange the nodes for you. Make sure, when you add the second Principled BSDF Shader that you land in a space not overlapping any existing pipes, then connect its output to the first input of the Mix Shader. You will see that the connection from the first Shader automatically swaps over to the second input. The three shaders are the nodes with green tops on the right of Figure 4-3

14. On the mix shader slide the Fac to 0.0 you should see a plain white cube and when you slide to 1.0 it should be fully rusty. If you see the opposite to this, drag the dots on the ends of the input pipes to the Mix Shader to swap them as described above

Figure 4-3 shows the full layout of the nodes so you could use that to position elements as you add them to save constantly moving things round. But I advise you to add the components one by one, following the steps below, so you see the effect of each - and the potential pitfalls.

Pointiness

15. You are going to control the mixing factor using the sharpness of edges, which can be obtained from the geometry node. Select **Add ‣ Input ‣ Geometry** then connect the Pointiness output to the Base Color of the plain white Principled BSDF Shader and set the mix Fac to 0.0.

You're using the pointiness as the base color just to check what's going on - when you get the worn edge effect showing in black and white you can connect the output to the Fac input of the Mix Shader. At this stage there will be no visible edge effect for three reasons.

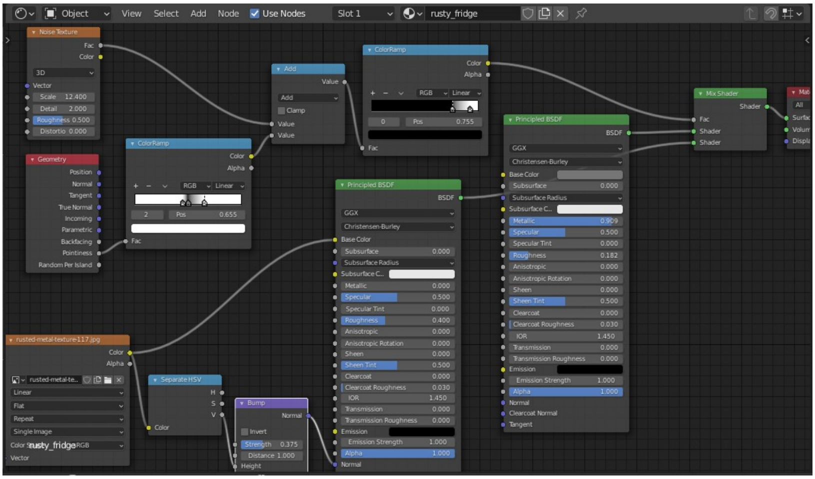


Figure 4-3 Full node tree for the rusty_fridge Material

The first is that the pointiness scale goes from 0.0 for a vertex that is at the bottom of an acute hole to 1.0 for a vertex on the tip of a pin point. For a vertex on a flat area the pointiness is 0.5 but for normal edges and creases the pointiness will be very close to 0.5. So you need to magnify the scale in the region around 0.5 and this is one of the uses of the very versatile color ramp node.

The Ubiquitous Color Ramp Node

16. You need to add a color ramp node then move the stops so that all of the variation in output take place over a narrow band of input near to 0.5. You want the output to be maximum either side and only drop to 0.0 where the vertices are flat. Select **Add ▸ Converter ▸ ColorRamp** and drop the node onto the pipe coming out of the Geometry node, use the **+** button to add a third stop then slide the stops so one is on **0.5** and the others are on either side, then change the colors using the color picker at the bottom of the node so the sequence is white, black, white, as you can just about make out in Figure 4-3.

This still won't solve the problem because of the second reason; all eight vertices on a cube have the same sharpness.

17. Subdividing will produce some vertices away from edges. Switch to Edit mode and subdivide the cube with Number of Cuts in Adjust Last Operation set to five.

Frustratingly you still won't see the desired effect for the third reason: the Geometry node Pointiness is only available when using the Cycles render engine.

18. In the Render Properties tab set the Render Engine to **Cycles** and, at the top of the 3D Viewport, set the Viewport Shading to **Rendered**. At last you should be able to see black sides and white edges and corners.

19. At last you can connect the output of the color ramp node to the Fac input of the Mix Shader.

20. To complete the Material you need to add some noise and sharpen the cut-off between the original finish and the rusty areas. Use **Add ▸ Converter ▸ Math** and drop the node onto the pipe from the color ramp to the Mix Shader, it will default to Add mode.

21. For the randomness here use **Add ▸ Texture ▸ Noise Texture** and add the node in a space to the left. Connect its Fac output to one of the Value inputs of the Add node.

22. A color ramp can again be used here to control the division between rust and smooth surface. Select **Add ▸ Converter ▸ ColorRamp**, drop the node onto the pipe between the Add node and the Mix Shader then adjust the sliders. The nearer together they are, the sharper the change from white to rust and their position between black and white will determine the amount of rust.

23. Change the parameters in the Noise Texture until you are happy with the result, but you will probably need to make further tweaks after the cube has been converted into a refrigerator.

Constructing the Prototype Object

With the material sorted out you can now move on to modeling the refrigerator. The first modification will be lines of studs and indentations on the refrigerator door, for which you will use an array modifier.

Array with Linear Offset

1. Change the Viewport Shading from Rendered back to **Solid**

2. You will use an array of cylinders to make the indents and studs but, as they will be small, you should reduce the resolution from the default 32. Select **Add ▸ Mesh ▸ Cylinder** then, in the Adjust Last Operation pane, reduce the number of Vertices to **12**, the Radius to **0.01m** and the depth to **0.02m**.

3. At this point the cylinder will be inside the cube so now would be a good time to rename “Cube” to “fridge0” in the Outliner Editor and, for the moment, click on the little eye icon to hide it.

4. To create an array of 8 cylinders, select the cylinder object, add an array modifier, increase the Count to **8**, un-tick the Relative Offset checkbox and tick Constant Offset setting the Distance X to **0.2m**.

5. In the 3D Viewport rotate the object about the Y axis by 90 degrees. Note, this has done what we wanted it to do, you now

have a vertical column of horizontal cylinders, however this may be a little surprising; the array offset specifies an X distance so it would be reasonable to expect a row of cylinder along the X axis, each one rotated by 90 degrees. You will see later that controlling how the array is generated usually requires more care than this.

6. Make the fridge0 more reasonably proportioned. It's a 2m cube at the moment so toggle the visibility back on and scale it down along the different axes until it is the right size and move it up so the bottom face is just on the origin. Making these transformations in Object mode will cause issues later but do it that way so you can see the problems and how to fix them.

7. To create a line of indents, using the orthogonal views and move the cylinder object array so it overlaps the face of the door in a line near the left side, then add and apply a boolean modifier with Operation **Difference** to the fridge0 object using the cylinder for the other object.

8. Repeat the process to form another line of indents towards the right of the door.

9. To create a line of studs, reduce the Array Modifier Count to **4** and apply it near to the left hand line of indents but using a boolean modifier with type **Union**. Delete the cylinder object after applying it for the last time.

Figure 4-4 shows how I arranged the indents and studs but obviously you can use your artistic sensibilities to create whatever pattern you like the look of: lines of rivets round all the panels or, if you had reduced the number of vertices in the cylinder to 6 instead of 12, you could have made an even more steam-punk refrigerator with lines of bolts.

Grooves and Edges Using Edge Loops

10. To make the groove running round the object to represent the door seam use the Loop Cut tool in Edit mode. The way this tool works is that it forms a highlighted edge loop at right angles to whichever edge is nearest to the cursor. When you hold down LMB you can slide the position of the edge loop until you release LMB which fixes it. Position the new edge loop a small distance from the first loop in from the front face as shown in Figure 4-4.

11. You need three edge loops near to each other running around the refrigerator so add one more using the Loop Cut tool just next to the other two.

12. To form a notch running round the door, select the central edge loop using ALT-LMB then scale it down a little bit.

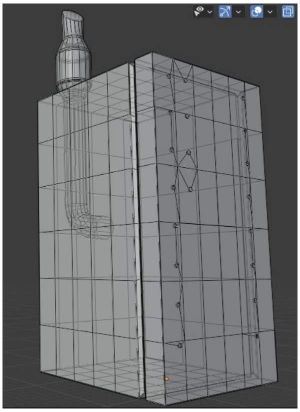


Figure 4-4 The prototype refrigerator

The back of the refrigerator is going to sport a whimsical chrome plated exhaust stack.

13. To form the main straight section, in Object mode select **Add ▸ Mesh ▸ Cylinder** then scale and move it so it is a small distance from the back.

14. Connect the bottom of the pipe with a curved section. In orthographic view from the side, switch to Edit mode and extrude the bottom face of the cylinder and rotate and move it a few times to form a bend, this is the same method you used for the corridor in Chapter 2.

15. To make the silencer and final tube section extrude and scale the top surface of the cylinder a number of times.

16. In Object mode create a new Material for the exhaust stack and rename it “chrome”, setting Metallic to 1.0 and Roughness to 0.0.

You need to use smooth shading for the exhaust stack to make it look cylindrical but when you do that you will see that the edges at either end of the silencer become indistinct and unrealistic. This is the same problem that happened with the Text object in the last chapter which you solved by simply adding a Bevel Modifier. However, because you will produce arrays in different ways using this refrigerator prototype, you cannot Parent the exhaust stack to the refrigerator, you must Join them together into a single object.

Bevel in Edit Mode

This raises the question of how to selectively bevel some edges of an object but not others. For instance, if you add a bevel modifier to the refrigerator object it will reduce the pointiness of the edges and modify the rusty_fridge material. There are several solutions to this problem such as assigning bevel weights to edges or using vertex groups. However for this exercise you will simply modify the mesh.

17. In Edit mode, select the four edge loops at either end of the silencer and bevel them using **Edge ▸ Bevel** followed by LMB then in the Adjust Last Operation pane set Offset to **0.05m** and Segments to **2**.

Joining Objects with Different Materials

18. If you now Join the exhaust stack and refrigerator into a single object, the materials assigned to different vertices will be preserved. In Object mode select first the exhaust stack, then the fridge0 object and use **Object ▸ Join**.

19. Save the prototype refrigerator as a blend file for a later chapter when we will return to the topic of smooth shading, flat faces and beveled edges.

Circular Array

For the first tier of the henge, you will use the third mechanism for Offset positioning with the Array Modifier; Object Offset.

1. In Object mode select the fridge0 object and add an Array Modifier, un-tick the checkbox against Relative Offset and tick Object Offset.

Empty Objects

Although they sound slightly mysterious, *empty* objects are simply a convenient way to hold information about location, rotation and scale, for positioning other elements in the scene say, without the need for an actual object.

2. You will see that this offset mode requires another object for reference, so add an empty object to use to define the relative offset. With the cursor over the 3D Viewport, select **Add ▸ Empty ▸ Arrows**. As you can see there are a variety of appearances for empty objects, they all do the same thing but some shapes will make their use clearer later.

3. Now set the array offset object. Re-select the fridge0 object and in the Object field select the empty object. You should see something odd but to make it a bit clearer increase the Count to 8 and move the empty object a small distance in the Y direction (see Figure 4-5)

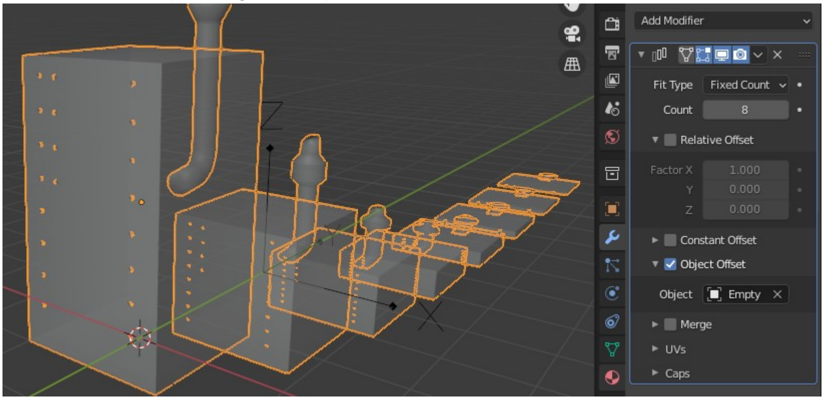


Figure 4-5 Object Offset scaling issue

The reason this happened is because the Object Offset array uses the difference between the Object properties of the fridge0 and the empty. If you want the array to stay the same size you need to return the scale of the fridge0 object to 1.0

4. To effectively convert all the changes made in Object mode as if they had been applied to the vertices in Edit mode, select the fridge0 object then select **Object ▸ Apply ▸ All Transforms**,.

5. In order that the refrigerators are set out in a ring, in orthographic view from the Z direction, move and rotate the empty object.

6. Now try selecting and rotating the refrigerator so the door faces a different direction.

Because the Offset is based on the differences between the rotations and positions of the two objects, the whole circle will change as soon as you rotate the refrigerator!

7. The solution is to rotate the mesh in Edit mode. In the Object Properties tab set the Z rotation back to **0.0** then, in Edit mode, select all the vertices and rotate them about the Z axis.

Now you will see each of the refrigerators rotating nicely around their local center point - in my experience this is the simplest way to work with the Array Modifier; only use modification in Edit mode and avoid any transformations in Object mode.

Arrays Along Curves

Now you will make a linear array of refrigerators then bend them using a Bézier curve, but first, create three copies of the prototype refrigerator to use for each of the higher tiers.

1. Duplicate the fridge0 object and move the copy up so it's clear of the first circular array.
2. Delete the Array Modifier in the copy so you have a single refrigerator, then duplicate it twice.

If you check my render in Figure 4-1 you will see that the second tier of the sculpture is constructed from an array of objects following a spiral using the Curve Modifier, the third tier has five objects as Instances, each at a vertex of a pentagon and the top tier has three refrigerators constructed in Edit mode using the Spin Tool.

3. For the second tier add a Bézier curve object which the array of refrigerators will follow.
4. To make the curve form about one and a half full turns, at a similar diameter to the first tier circle, switch to Edit mode then extrude one more control point. Move, scale and rotate each point using different orthographic views. It won't be possible to make a very regular circle with only a few control points but it doesn't matter too much so long as the length of curve between each point is about the same.
5. Arrange the circle while viewing from the Z direction then adjust the heights and slopes of the control points from the X and Y direction so there is a fairly smooth spiral.

6. First make a vertical array of refrigerators. Back in Object mode, select the second tier refrigerator, add an array modifier and increase the Count to **8**. Use Relative Offset but change the X Factor to **0.0** and and set the Z Factor so that each refrigerator is separated from the next by a small gap.

7. Add a curve modifier, which needs to be below the array modifier in the stack. Select the Bézier curve as the Curve Object and set Z as the Deform Axis.

8. To get the array to start and end on the curve adjust the Z location of the refrigerator object in the Object Properties tab and the Z spacing and Count in the array modifier.

9. You want to make the refrigerators look as if they are resting on the first tier and on each other so adjust the positions of the curve control points.

On a more general level, the technique of making arrays of objects "line up" along curves is occasionally very useful; you can construct cables, hoses and architectural features very quickly from a stack of unit objects and a curve.

Instancing Objects

You will accomplish the third tier of the sculpture using instancing. The method used here creates instances at each vertex of a parent object.

1. First you need to add an object as the parent. Use **Add ‣ Mesh ‣ Circle** then in the Adjust Last Operation pane reduce the number of Vertices to **5**.

2. Scale the circle up and move it above the second tier. Move the third refrigerator approximately to its center.

3. Parent the refrigerator to the circle by first selecting the third refrigerator then SHIFT selecting the circle. The order is important here, the refrigerator should be highlighted in red-orange and the circle in yellow-orange, then run **Object ‣ Parent ‣ Object**. In the Outliner Editor you should now have the refrigerator as a child of the circle.

4. Select the circle object then in the Object Properties tab (orange square icon) under Instancing select **Vertices** and tick Align to Vertices.

5. Move and rotate the circle object to rest the five refrigerators on top of the second tier.

6. Rotate all the outside instances by selecting the central one and rotating that about the Z axis. Note that, as with the array modifiers, you will have issues if you rotate the object rather than rotating the vertices while in Edit mode. The original, central refrigerator will be visible in the 3D Viewport but will not show in the render.

THE BMW CAR DEMO

Instancing is a less frequently used feature of Blender. Though easy to do for simple arrangements such as this, it does have limitations. However, a close relative; making a collection into an instance to scene, can be very powerful as it allows complex assemblies of components to be quickly replicated in multiple locations. There isn't space in this book to use this technique in any of the exercises, however I recommend you download the popular BMW Car Demo from *https://download.blender.org/demo/test/BMW27.blend.zip*, and check out the use of instancing for the wheels (and the whole second car!) This demo also shows clever use of subdivision and mirror modifiers, for instance the body shell including details such as wing mirrors and door handles, only uses 1,765 vertices. Don't worry at this stage if you can't figure out how some of the rather sophisticated modeling works.

Spinning Vertices

For the top tier of the sculpture you will utilize the Spin Tool in Edit mode. On occasion this is a very convenient shortcut but it is less flexible than the non-destructive array or instance methods you just used. The result will be a single mesh so any tweaks you might need later will have to be repeated on each of the refrigerators.

There are two different ways that the Spin Tool operates. If the selected vertices form a mesh with no unconnected edges (you will see this referred to as a manifold), it will create duplicate meshes at intervals around a given angle. If there *are* unconnected edges it will extrude the edges in a number of steps around a circumference.

For this exercise we want the first behavior but as an example of the second: if you had converted the Bézier curve object from the first exercise in Chapter 3 into a Mesh, you could have use the Spin Tool to make it into a ring as shown in Figure 4-6.

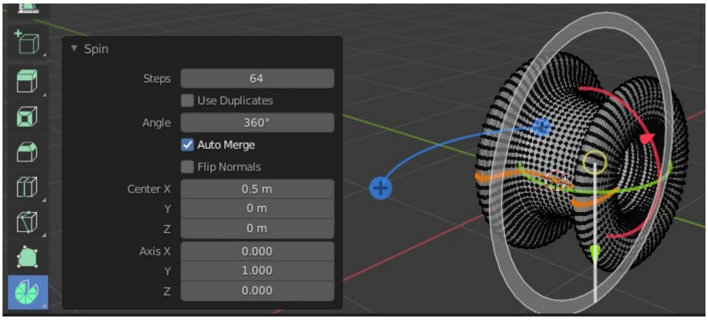


Figure 4-6 Spinning a ring from a mesh

1. Select the top refrigerator object, switch to Edit mode then select all the vertices and choose the Spin Tool from the Toolbar.

2. First create some duplicates by dragging on one of the blue plus buttons at either end of the blue arc (visible in the center of Figure 4-6). It doesn't matter how far you drag it because you will set specific values afterwards.

3. Set the number of duplicates and the angle between them by opening the Adjust Last Operation pane and change the Steps to **2** and the Angle to **240°**. If there happen to be any unconnected edges on your refrigerator (for instance if you used an open-ended cylinder for your exhaust stack) then you will need to tick the Use Duplicates checkbox.

4. To get the refrigerators orientated the way you want change the Center X and Y values.

5. Position the top tier so it's resting on the one beneath by changing back to Object mode and moving and rotating it.

Scattering Objects over the Surface of Another Object

The setting for your sculpture will be a grassy mound with early morning mist against a suitably rural background using an Environment Texture. The grass will be scattered over the mound using a particle system.

NOTE From Blender version 3, the standard way to scatter instances of one object over another surface is Geometry Nodes which will be covered in Chapter 14. However old versions of Blender can get the same effect using particles.

Collections

Before you start adding the components, this would be a good opportunity to introduce collections as an excellent way to tidy up your project and help your workflow.

You used a collection with a boolean modifier to create the Lower_domes object in Chapter 2 and they can also be used for instancing as I mentioned above, however their normal use is as a kind of folder system to help organize complicated scenes. A very real benefit of using collections is the ability to toggle them off in the Outliner Editor which allows you to keep alternative outfits for a character, for instance, in the same blend file but switch between them easily.

You will encounter another important function of collections in Chapter 13 when you use the compositor. Blender can render different parts of a scene as different layers for later reassembly with enormous flexibility, and collections are the mechanism for controlling this.

For the exercise of this chapter you will simply use collections to tidy up your workflow.

1. Make a new collection for the fridge arrays. In the Outliner Editor RMB over Scene Collection then **New Collection** and rename it "fridges".

2. Create another new collection and rename it "ground".

3. Drag and drop each of the refrigerator related objects into the fridges collection. The five sided Circle, the empty and the BezierCurve all belong inside, leaving only the Light and the Camera where they are.

An Object to Hold the Particle System

The mound will be a simpler version of the crater you constructed in Chapter 2, but before you add it, move your Camera so you frame the sculpture in a pleasing way. You can see in Figure 4-1 that I changed Resolution X in the Output Properties tab (printer icon) so that

I rendered a square image. That resulted in only a small amount of hill being visible in the final render – there's no point adding polygons to the scene if they will never be seen.

4. Select the ground collection in the Outliner Editor then add a plane to the scene. It will default to being inside that collection.

5. Rename the plane to "hill" then scale and subdivide it and form a mound in the middle by moving a vertex upward with Proportional Editing on.

The mound will be covered with grass so it doesn't need to be very high resolution or carefully crafted.

An Object to Distribute as Particles

The last method you will use for generating multiple copies of an object is the Particle System. This is a very versatile feature but that versatility comes at the cost of quite serious complexity, and in many cases the settings that need to be adjusted are quite hard to find.

Before you add the Particle system to the hill object you need to make a tuft to be scattered over the surface.

6. Add a plane inside the ground collection and rename it "tuft".

7. Modify the plane to form a single blade of grass. In Edit mode and in orthographic view from the Z direction move the vertices, I subdivided one of the edges twice so I had a total of six vertices to allow better shaping, see Figure 4-7.

8. When it looks OK select all the vertices and duplicate the blade. When the tuft is duplicated by the particle instance the Y direction will be vertically upwards so you need to rotate it around the Y axis so the tuft is later visible from all directions. Scale and move the duplicate to form a second blade.

9. Repeat six or seven times until you have a little tuft, as shown in Figure 4-7. You will need to move some blades up and down in the Z direction too to spread them out in two dimensions.

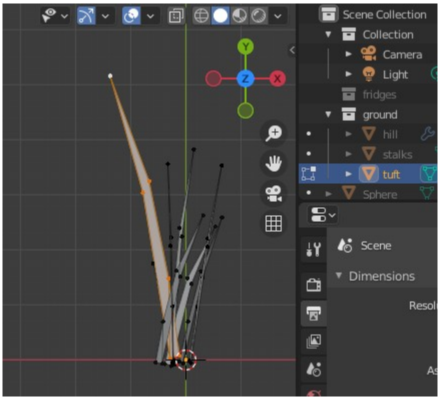


Figure 4-7 The Tuft object

10. To add a particle system to the hill switch back to Object mode, select the hill object and open the Particle Properties tab (the icon is a blue dot connected to three other blue dots). Choose Add Particle System by clicking on the **+** button.

11. Particle Systems are really an animation effect so they are designed to produce particles over a period of time, but you want all of your grass to exist from the start without needing to run the animation so under Emission set Frame End to **1**.

12. The default particle is a glowing dot, so, to change that to tufts of grass, you need to open the Render section then change Render As from Halo to **Object**, make the Scale **1.0** and the Scale Randomness **0.5**. In the Object section below select the tuft as the Instance Object.

13. You can generate more tufts by adding children. Open the Children section and select **Simple** and increase the Radius value to around **0.8** or until the Children objects spread naturally into the spaces on the hill.

14. Create some random rotation as well by ticking the Rotation checkbox then open that section and increase the Randomize Phase value to **0.5** or more.

To increase the variety of the grass it would be good to add a second, slightly different, object as well.

15. Duplicate the tufts object and rename it "stalks".

16. In Edit mode extend one of the blades so it is significantly taller than the others.

17. Back in Object mode select the hill object again and add an additional Particle System, renaming the two System slots and associated Settings with an appropriate “tufts” or “stalks” name.

Note that the Particle System works a little like the Materials setup, there are slots each of which can hold a Setting item, and different objects can share the same settings.

18. With the stalks System selected go through the settings and change the Render, Children and Rotation as you did for the tufts but using the stalks as the Instance Object.

19. Reduce the Number under Emission to **100**.

20. To increase the effect of variability, change the Children Render Amount to **10**, Size to **1.25** and Random Size to **1.0**.

21. Add three Materials, one to the hill and one to each of the grass objects, with different shades of green and brown. If the hill becomes invisible once you add the particle systems to it then you need to tick the checkbox Show Emitter in the Viewport Display section of the Particle Properties tab.

22. Use the Render Region option in the Output Properties tab to check the overall effect when rendered using Cycles. Adjust the colors and Particle System settings until you get good looking grass.

Background

In the World Properties tab choose a suitable outdoor Environment Texture to use as the Color in the Surface section. I used the sunrise image again and turned the Strength down to 0.2 but you can try different options until you find something that shows off your sculpture and grass to good effect.

Adding a Volume Shader

Back at the start of this chapter I mentioned that the Geometry node Pointiness only worked with the Cycles Render Engine, so that should already be set in the Render Properties tab. One consequence of using Cycles or Eevee with ray-tracing is that if you use the Volume Scatter shader in the World Properties, which used to an option with the old version of Eevee, it is too opaque, even on the lowest possible setting. The shader attenuates the light coming from the environment texture into the scene rather than the light being scattered from objects back to the camera, but the result is the same; a black cat in a coal cellar!

A technique often used with Cycles is to add a mist pass in the View Layer Properties tab. However that requires setting up the compositor to do post-rendering image processing, which is a big topic you will cover in Chapter 13.

An alternative though, is to add an object with a Volume Scatter shader so it surrounds everything in the Camera view. This is the technique you used to fill the Lower_domes with atmosphere in Chapter 2 as well as make mist for the boat in Chapter 3 so check back and follow the instructions for those, but obviously don't add a wireframe modifier or any shader on the material's Surface. Figure 4-8 shows the position and size of the sphere with the Material settings I used.

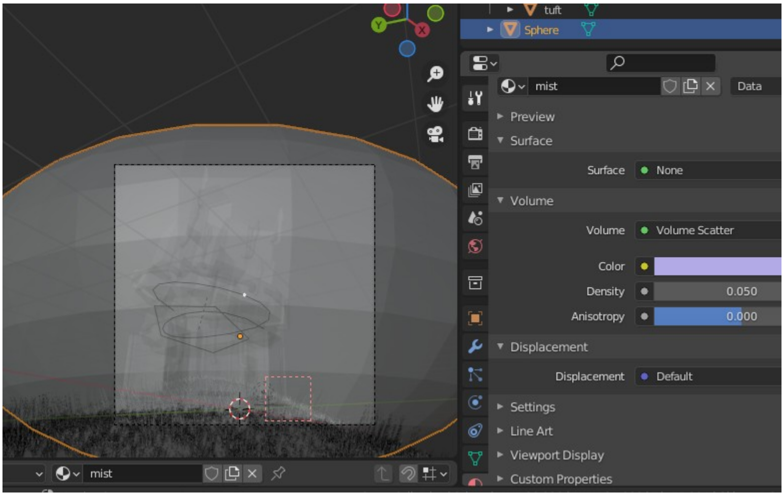


Figure 4-8 The mist sphere

Conclusion

In this chapter you delved into four quite different ways to generate multiple clones from a prototype object. Each technique has strengths and weaknesses and it's hard to be prescriptive over which one to use in a given set of circumstances. My suggestion would be to experiment with all of them and gain a feeling for each one's idiosyncrasies.

Using the Particle System to generate instances of another object is a way to scatter things over a surface. This is how the Quick Fur works, which you applied to the velvet block in Chapter 1, and it would be ideal for spreading rocks randomly over the Martian landscape you created in Chapter 2. In the chapter introduction I mentioned a fifth method of creating multiple instances of objects, *geometry nodes*, which you will use in Chapter 14. The Blender Foundation has stated that geometry nodes will become the standard method for processes such as scattering grass on hills, as well as many other techniques, and in my opinion they are significantly easier to use than particles. As at version 4, the functionality has reached reasonable stability but it's still worthwhile learning about the established particle system.

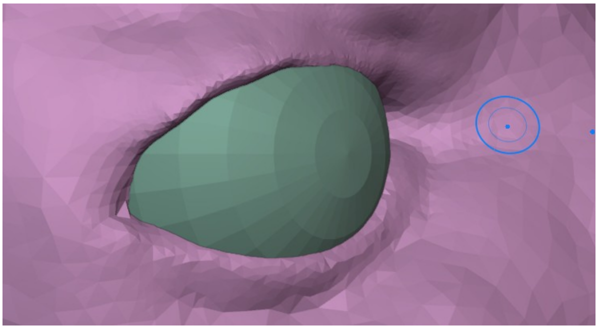
In the next chapter you will learn to use brushes in Blender, for sculpting 3D models but also for Texture Painting. This latter, along with Vertex Painting, allows extra flexibility to be added to the emission of objects by particle systems. You can paint areas to control the Density and Size (as well as many other properties) of generated particles. You could also have used the same method to paint the corners and areas of the refrigerator where you wanted more wear and rust to show through, which would have avoided some of the restrictions caused by using the Geometry node Pointiness.

There is a lot to cover in the next chapter, but at the end I will quickly revisit the Martian terrain as well as the rusty refrigerator material to show you how they could be improved.

21. When the eyes look to be in the correct position select the Person01 object again.

22. Switch back to Sculpt mode and start to build up the eyelids using Clay Strips.

As you try to create a square edge to the lid (where the lashes would grow, see Figure 5-3) you will find that it is hard to make the corner clean, even using **Stroke ▸ Stroke Method ▸ Curve**, as the faces are arranged in a jumble rather than neatly along the contours of



the shape you are sculpting.

Figure 5-3 Jagged eyelid edge from using Dyntopo

This same problem will crop up with the lips and other areas and the solution is the subject of Chapter 7, for the time being don't worry about these few rough edges.

Masking when Forming Larger Details

Ears have a sharp crease and the "clay" doubles back on itself around the top and back. Using a mask will help here.

23. In order to allow a smooth curve to form for the crease around the ear, use a brush in smooth mode first to subdivide the large faces.

24. Build up the area of the ear slightly using Clay Strips while zoomed in to get a reasonable number of small faces.

25. To prevent the head around the ear being distorted, draw using the Mask brush then, with the Grab brush, stretch the edge of the ear out into, well, an ear shape, shown on the left of Figure 5-4.

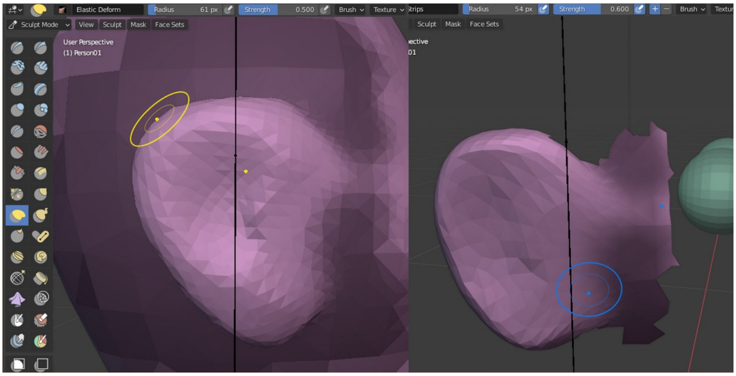


Figure 5-4 Mask around the ears or hide vertices

The Mask can be fine tuned using options from the menu. As well as clearing it when no longer needed you can invert, grow, shrink, smooth, sharpen and alter contrast.

MASKING

Masking is an important part of sculpting and painting on 3D objects, with several methods available. One approach, also shown in Figure 5-4 is to switch to Edit mode, select the vertices you want to work on, then **Mesh ▸ Show/Hide ▸ Hide Unselected**. The hidden vertices will also be hidden when you switch back to Sculpt mode.

An even better method is to use the Draw Face Sets brush to define areas you want to keep separate. Each time you draw with the brush it produces a different color and a new face set, if you want to spread an existing set press CTRL as you start to draw. Once face sets are defined you can hide everything apart from the color under the cursor by pressing H which allows very fast and efficient workflow. However you cannot use face sets at the same time as dyntopo so, at the stage of the sculpting process where you need to subdivide faces, it is better to stick to Mask, or hide vertices in Edit mode.

26. Sculpt the ear further with Clay Strips, adding and scraping away and finally making an indentation for the ear hole with the Draw or Blob brush. Have a look at a photograph of an actual ear before you go too far with this.

27. To define the mouth, build up a flat dome using Clay Strips mid way between the nose and chin, smooth it then add the upper and lower lips, fattening then smoothing to achieve the correct shape. For the sharp crease between the upper and lower lips

you will again improve your work flow using masking as you did for the ears. Also don't forget that switching to Edit mode often allows you to make selections or move vertices in a more precise way than in Sculpt mode.

28. Finally add the shape of the jaw, cheek and chin along with the typical creases running from the edges of eyes, nose and mouth.

29. Make adjustments to the overall shape of the face using the Grab brush with a large diameter.

Fixing the Bone Weights

When you're happy with your sculpt save the blend file. As I mentioned at the start, don't expect to create a work of art, or even anything very realistic. The sculpting process will have scrambled the way the armature controls the mesh so this will need to corrected.

1. To check the damage, switch to Object mode, select the Armature then enter Pose mode.

2. Select the head bone and rotate it and you will see a horrible mess. Most of the vertices don't move, including the eyes, which we know are still a completely different object. That will be the first thing to fix by joining the two objects.

3. Select all the bones and clear all the transforms for the pose.

4. Back to Object mode select the eyes and apply the mirror modifier. Then with the eyes still selected SHIFT-LMB on the Person01 object so both are selected, then **Object ▸ Join**.

The Sphere object should have disappeared from the Outliner Editor and if you go to Edit mode you will see that you can select vertices of the eyes and skin at the same time, it's all one object now.

5. You can use Weight Paint mode as a visual check of bone influence. With Person01 selected, choose Weight Paint from the object interaction mode drop-down.

6. Open the Object Properties tab on the Properties pane (three green dots connected in a triangle).

7. In the Vertex Groups list select a few bones and watch how the coloring changes in the 3D Viewport.

Blue means no effect and red means full effect. You will see that limb bones are affecting both sides of the manikin, and most of the face remains blue when the head bone is selected. There are menu options available in Weight Paint mode to help fix this but it is actually much easier to just re-parent the armature and skin objects. For all practical purposes armature objects should be the parent of the mesh objects they deform, that way you can move the armature around the scene and the model will follow it. The skin modifier doesn't automatically set the generated armature as the parent of the skin mesh, so this is an operation that has to be done anyway.

8. Switch to Object mode and select the Person01 object, then SHIFT select the armature then **Object ‣ Parent ‣ With Automatic Weights**.

Your model may be fine after sculpting but it is possible to have problems because of the complexity of the mesh, essentially vertices that were at the same location as other vertices. This can frequently be cured by merging vertices by distance, however you need to make sure you only select vertices in the head mesh excluding the eyes, so they don't get accidentally fused together.

9. In Edit mode select one vertex on the body then expanded the selection to the whole mesh but excluding the eyes **Select ‣ Select Linked ‣ Linked**

10. Use **Mesh ‣ Merge ‣ By Distance** and, in Adjust Last Operation, increased the Merge Distance until the message at the bottom reports some vertices being removed, but without any dramatic simplification of the mesh.

Merging by distance will normally be sufficient but if the automatic weight assigning still doesn't work there are other options in the mesh clean up menu, such as decimate, that should eventually fix it.

11. Check that the armature now poses the Person01 object as expected. Clear all the transforms from the pose when you've finished.

One thing you might find when you use automatic weight assigning, especially for complicated or unconnected bits of mesh such as the eyes, is that large movements of the bones produce unwanted distortions: vertices get "left behind". Sometimes you can fix this in Weight Paint but often it is hard to find which bone is having an unwanted influence. A useful way to solve this problem is to use Edit mode as shown in Figure 5-5

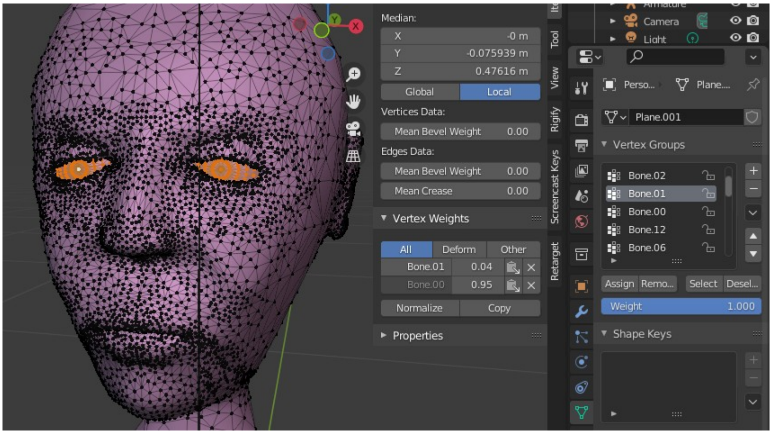


Figure 5-5 Fixing bone influence in Edit Mode

12. In Edit mode open the Sidebar on the right of the 3D Viewport and select the Item tab

13. Select a vertex on the mesh that is showing a problem, for instance the eye, then select a vertex that is behaving correctly, the eyelid. Compare the Vertex Weights for each section in the Sidebar

In Figure 5-5 you can see that the eyes were being influenced by the neck bone Bone.01 with a very small weight factor.

14. Select all the problem vertices then, in the Object Data Properties tab under Vertex Groups (on the right of Figure 5-5) select each bone and either Remove it using the button below or change the Weight value and Assign it so that the Vertex Weights match the correct vertex.

As you can see, using an armature to pose a sculpted mesh has many problems, and a much better approach is to create a new, simpler, mesh over the top of the sculpted mesh and parent the armature to that. You will use this technique in Chapter 7 after reworking this Person01 mesh and in Chapter 12 when you learn more sophisticated rigging techniques.

Texture Painting

Texture Painting allows you to draw on an object using the mouse or graphics tablet and to save the result as a texture for use in materials. Painting is not just limited to solid colors, images can be cloned onto surfaces, and textures can be used to generate patterns on the surface. As well as the base color of the material, the other inputs to shaders such as roughness, displacement and metallic can be painted.

There are two prerequisites for Texture Painting: The first job is to generate the map that controls how the colors on the texture are positioned on the 3D surface. This is referred to as a UV map and you will look in more detail at this in Chapter 8. For now you will keep it as simple as possible so these are the basic steps needed to set everything up ready for painting..

1. To create the UV map select the Person01 object, switch to Edit mode, select all the vertices then **UV ▸ Smart UV Project**, increase the Island Margin to **0.002** then click **OK**

2. The second prerequisite for Texture Painting is to have an image texture available for painting onto. Select the Material Properties tab in the Properties Editor, create a new material and rename it to "person_base". Change the Timeline Editor panel at the bottom to Shader Editor, make it larger, then add a texture node to the material with **Add ▸ Texture ▸ Image Texture** and connect the Color output to the Base Color input of the shader.

3. In the Image Texture node, create a new image by clicking **+NEW**. For the moment leave the resolution at 1024x1024 (though you will see that produces a very pixilated material). Give the image an appropriate name such as "person_base", un-tick the Alpha check-box, set the Color to an approximate skin tone then click **OK**. Switch the Viewport Shading to Material Preview to see the result.

4. Select the Texture Paint preset from the Active workspace option in the menu at the top and the Active Tool and Workspace Settings tab in the Properties Editor on the right. On the left, in the Image Editor pane (the icon is a little paint can) you will be able to see the 2D texture being filled in, as you paint on the 3D object. You can also paint directly onto the texture in this window which is occasionally useful for tidying up areas.

The Image Editor pane also shows the UV unwrapping: notice that the projection is not smart enough to know which areas needed to be large because they might contain a lot of detail, and which areas could be shrunk down relatively small.

5. Back in the 3D Viewport try painting on the figure, use F and SHIFT-F to change the brush size and strength as you did when sculpting.

6. Change the color of the paint either using the tool controls in the header at the top of the editor, or in the Active Tool and Workspace settings tab. Most of the options are duplicated in the two places and many are the same as for sculpt brushes. However, before Blender 4 it was possible to specify Material or Single Image in the properties tab, but now this must be done in the header, see the top of Figure 5-7.

7. Change the Falloff to the square edged profile and draw two contrasting colors next to each other.

Sometimes it is still hard to paint different colors next to each with a sharp edge between, though a higher resolution image would obviously help. In those cases it can be useful to mask the mesh as you did in sculpting.

8. To mask an area you need to switch to Edit mode, set face selection then select a group of faces. Switch back to Texture Paint mode and toggle Paint Mask using the button just to the right of the object interaction mode drop-down (the drop-down you used to switch between Object, Edit and Texture Paint mode). The Paint Mask icon is a solid square in front of an outline square.

Vertex Groups

In order to swap from one set of faces to another set, then back again, for instance to paint eyes, hair, skin, shirt, pants and shoes it would good to be able to save the selections for re-use. This is exactly what *vertex groups* do, although their use extends far beyond; almost all modifiers can use vertex groups and some, for instance the armature modifier, depend on vertex groups entirely.

Just as when assigning vertices to materials, vertex groups used as masks really define the faces between the vertices, so the vertices at the edge of the hair group, say, should also be in the group defining the adjacent skin.

9. To create a vertex group for masking as you paint a shirt, in Edit mode use face select mode and select the the faces to define a shirt for the manikin. Using X-Ray and orthographic view will help this.

10. To add a new group, in the Object Data Properties tab, under Vertex Groups click the **+** button on the right. Double click to rename it from Group to shirt. With the shirt group selected, **Assign** the selected vertices using the button below.

11. Repeat the above process to create groups for hair, eyes, pants, shoes. Because the eyes are unconnected meshes you can select a single face on each eye then click **Select ▸ Select Linked ▸ Linked**

12. Finally use the **Select** button beneath the list of vertex groups to cumulatively select the faces for all the groups, **Select ▸ Invert** then create a group for the skin. This is why you needed to change to face select mode, if you had defined the vertex groups in vertex select mode there would be gap when you inverted the selection.

13. Switch between Edit and Texture Paint mode using the different groups as masks, painting with a suitable color.

The result is probably rather uninspiring - everything looks like it's made from plastic! Adding a Texture to the brush might improve that:

14. In Texture Paint mode open the Active Tool and Workspace setting tab and scroll down to Texture and add New. The texture will be called "Texture" and will show as solid black which will prevent the brush from adding any color.

15. A brush texture is subtly different, and actually much more powerful, than a simple texture such as you have used previously as part of materials. It requires a second stage to specify how the texture is to be generated; select the Texture Properties tab (pink and black checkerboard right at the bottom) and in Type select **Voronoi** and change the Coloring to **Position**.

When you draw with this brush it will combine the pattern and colors of the texture with the color selected for the brush. The size of the pattern depends on the size of the brush so as you zoom in and out the pattern size will change relative to the object you are painting unless you change the size of the brush at the same time.

16. Paint the shirt, pants and shoes using different textures and colors.

17. Change the Texture to Type Wood with Pattern set to Band Noise and paint the hair.

One thing you might find, when over-painting with a different texture brush, is that the black color "blots out" the design that you have already done, even if you turn the brush strength down with SHIFT-F. Often you want the black part of the texture to be transparent, in which case you need to:

18. In the Texture Properties tab, under Color, tick the check-box against Color Ramp. The positions of the sliders can also be used to make the pattern sharper or softer, as well as the color and alpha value at each stop.

Displacement texture

Once you have some interesting base color textures drawn onto your model it would be fun to add a displacement texture as well.

19. Switch the Active workspace preset back to Layout using the menu at the top. In the Shader Editor at the bottom add a second Image Texture node and create a new texture renaming it to "person_displacement" with a black default color (back is 0.0 displacement, white is 1.0 displacement and, if the scale of the object is small, even low values can look extreme). Connect the Color output to the Displacement input of Material Output node. You might get a better result, with finer control, if you add a math multiplication node between the displacement texture and the material output.

20. Change back to the Texture Paint preset using the menu at the top then click on the texture selector in the header, visible at the top of Figure 5-7. This should display the name of the new image texture you just created but sometimes Blender doesn't refresh this straight away. Change the Mode from Material to **Single Image** and make sure person_displacement is selected. Draw with a fairly dark gray in the hair area.

21. Draw displacement onto the jacket area as well using the same wood texture but with reduced turbulence. In Active Tool, Texture settings you can adjust the angle and scale of the pattern to create the effect of horizontal bands on a quilt.

If there are multiple textures involved with a .blend file, then you need to tell Blender which one to paint over by either selecting it in the Image Editor drop-down, clicking on the node in the material Shader Editor or, much more reliably, specifying it in the Texture Paint header

When you draw the iris and pupil on the eyes you will find that there is a shortage of pixels to give anything like a sharp edge, even if you draw directly onto the 2D image texture in the area on the left. In Chapter 8 you will see how to solve this problem by using different UV unwrapping and a different material for highly detailed areas such as eyes.

Pack your textures or save them

After you have finished creating the base color and displacement textures for your character you would be forgiven for assuming that they would be included in the blend file when you save it. However, this is not the default behavior of Blender. You must either save textures as image files or pack them into the blend, though, if you try to exit the blend file, you will be warned and given an opportunity to save any unsaved images.

22. The **Save All Images** button is now hidden inside the material or texture selector in the header, visible at the top of Figure 5-7. Click on that button now.

This is the most straightforward way to avoid losing painted textures and the data is packed into the blend file. Frequently you need to save a texture as a separate file, for instance if you want to use it in different blend files later or edit it with an external application. To save an image as a separate file:

23. In the Image Editor area select each image with the drop-down then from the menu **Image ▸ Save**

To pack textures, that were previously saved separately, into a blend file:

24. In the main Blender menu **File ▸ External Data ▸ Pack All into .blend** or better, set this behavior as default so you don't forget later to tick the check-box **File ▸ External Data ▸ Automatically Pack into .blend**

NOTE At some stage you will find that an object in one of your scenes has turned to a strange pink or purple color. This is a sign that there is a missing image file referenced from a material in the blend file.

There are options under **File ▸ External Data** to help find and link missing image files, as well as options for packing and unpacking images in different ways. In general, problems with external images only become apparent with larger projects, blends imported from other application and online resources or where images need to be processed externally.

Improving the end result

There are many deficiencies with attempting to paint skin this way, not least is the fact that photographic images are readily available and can be incorporated into skin materials quite easily. However there are two things you can do to improve things a little. Firstly, subsurface scattering better represents the fact that skin is actually a rather yellowish gray color but the blood underneath makes the overall effect pink. However the default BSDF shader produces a rather “plastic” look that needs to be toned down in many areas. Secondly, eyes are highly reflective and need to have a much lower roughness than the rest of the skin:

1. Change back to the Layout workspace using the tab at the top of the screen. Set the Viewport Shading to Material Preview.
2. Select the Person01 object and open the Material Properties tab.
3. You can copy and paste one material to another using the tiny drop-down menu button to the right of the material slots list, just visible in Figure 5-6. However there is a quicker way, using the button, just to the right of the current material name, with an icon of two overlapping squares. First rename the material from Material.001 (or whatever default name was created) to skin.

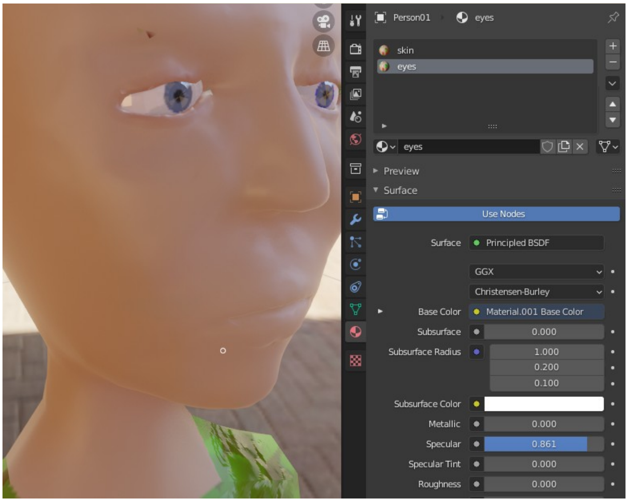


Figure 5-6 Copy the skin material settings to the eyes material

4. Add a new material slot using the + button to the right then select the skin material from the drop-down below.

5. Click on the the little New Material button to the right of the drop-down then rename the new material from skin.001 to eyes. Both the skin and the eyes material now use the same texture file you just painted, so you can add a subsurface color to the skin one and increase the reflection for the eyes.

6. First you need to assign the vertices appropriately. Switch to Edit mode, de-select all vertices, select a vertex in each eye then **Select ▸ Select Linked ▸ Linked**. In the Material Properties tab **Assign** the selected vertices to the eyes material.

7. With the eyes material still selected decrease the roughness to zero and increase the Specular value to above 0.75

8. All the vertices not in the eyes already use the skin material so you don't need to assign those.

In reality the amount of subsurface scattering varies from lips to cheeks to chin so this would be a good instance where it would be useful to be able to paint a texture which can then used in a more general way. This is the technique I referred to in the previous chapter for controlling the worn areas of the refrigerator rather than relying on the Geometry node pointiness. You can see the screen layout for the following instructions in Figure 5-7.

9. Change back to the Texture Paint workspace using the tab at the top of the window then swap the panel on the left from Image Editor to Shader Editor.

10. Create a new image texture node and rename it to “person_subsurface” with initial color white. The default method of subsurface scattering, Christiensen-Burley, gives a good result for skin, and you will mainly need to tone it down by texture paining darker colors.

11. Connect the Color output to the Subsurface Scale input of the shader as shown in Figure 5-7.

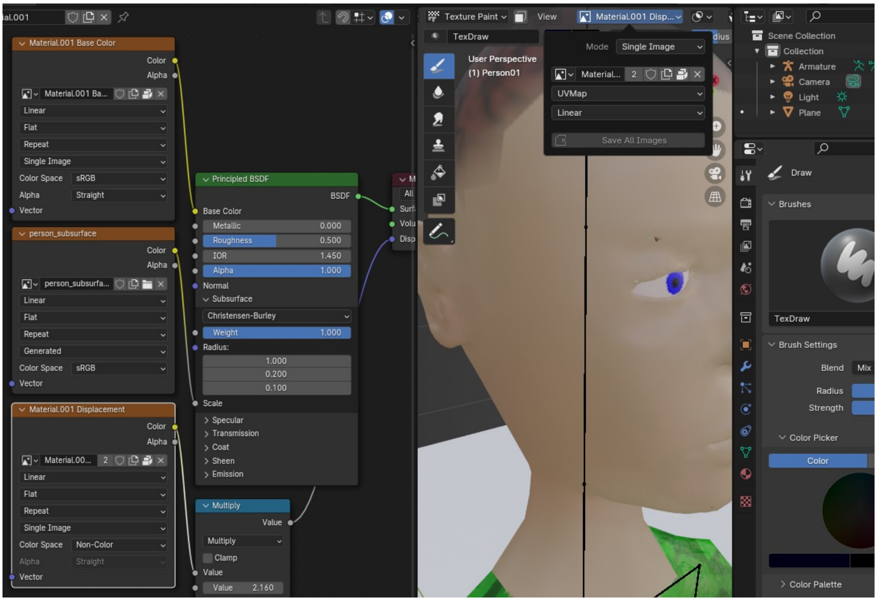


Figure 5-7 Painting a subsurface map texture

12. With Viewport Shading set to Material or Rendered, draw areas of decreased subsurface scattering onto the Person01 object. The whiter the paint color the stronger the effect.

Using an Image Texture as Blend Factor

Try using this technique on the rusty refrigerator material from the last chapter. You can replace the Geometry and first ColorRamp nodes with a hand painted texture representing areas of wear as in Figure 5-8. As this method doesn’t rely on the cycles render engine the texture painting can be done with Viewport Shading set to Material Preview which is normally much more responsive.

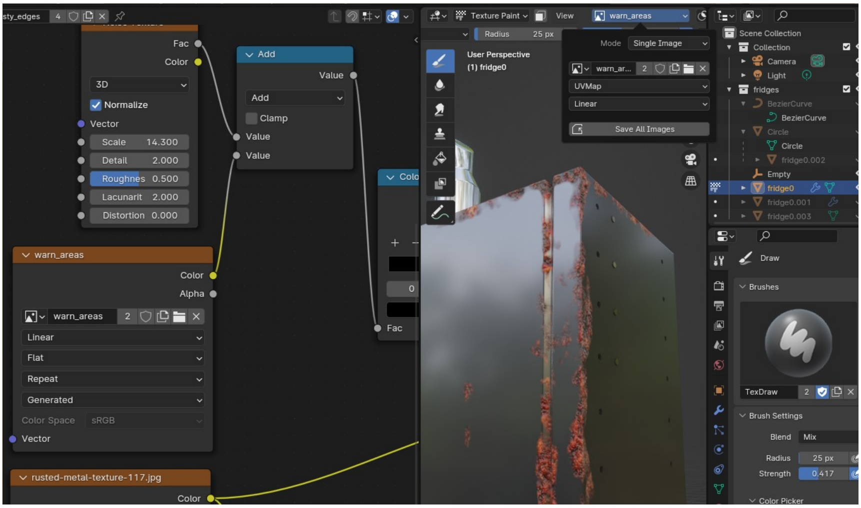


Figure 5-8 Painting areas of wear

Although texture painting will probably work after a fashion with the default UV map from the original cube, it will be much better if you generate a new one following the first step under the Texture Painting heading in this chapter..

Painting Particle Density

The other promise I made earlier was in relation to the scattering of rocks on the Martian landscape from Chapter 2. To do this you need to follow the steps you used in Chapter 4 to create grass on the hill but, obviously, using simple rock shapes. In Figure 5-9 you can see that I have used two different sizes for rocks (they are the same shape, basically a cube subdivided once then slightly randomized with **Mesh ▸ Transform**). I have also appended the grass tuft object from Chapter 4 and scattered that as a third particle system inside the dome.

Figure 5-9 Improved Martial landscape

Painting particle density requires a slightly non-obvious step involving a type of texture that is different again from images, brushes or procedural textures. It also involves quite a few steps that need to be set up in different places, so follow the instructions carefully. Links to the blend files and videos for this modified landscape are at *get-into-blender.com*. 5. Sculpting and Painting

1. Open the blend file from Chapter 2. Create and rename the boulder object then duplicate and scale it down as a pebble object. Append the tuft object from the blend file for Chapter 4.

2. With the Terrain object selected switch to Edit mode, select all the vertices then **UV ‣ Smart UV Project** then switch back to Object mode

3. Set up the three particle systems following the instructions from the last chapter. Name them “pebbles”, “boulders” and “grass”. For the moment leave them at the default number of particles which will be spread over the whole area.

4. In the Shader Editor panel create three solid black images. Name the images “pebbles_texture”, “boulders_texture” and “grass_texture”. You don’t need to connect these textures to anything in a material but you need to associate them with the terrain object and select each one here to paint onto it.

5. Back in the Particle Properties tab, for each of the three system, scroll down to the Texture section and create a texture for each

renaming them “pebbles_tex”, “boulders_tex” and “grass_tex”. This is the non-obvious step as these textures are rather different from textures you have encountered so far.

6. With each particle system selected in turn click on the Texture Properties tab (pink and black checkered icon at the bottom). This should have defaulted to ParticleSystem with the correct particle texture selected. Ensure that the texture Type is **Image or Movie**.

7. Under the Influence section un-tick General Time and tick Density and also Size.

8. Under the Mapping section select **Coordinates UV** and Map **UVMap**

9. Under Image, Settings select the relevant image you created in the Texture Paint workspace. The Alpha check-box should match the alpha setting you used to create the textures for painting onto.

10. Draw onto the terrain where you want each of you particle systems to appear. In the Texture Paint header you need to change which texture is selected to ensure that you are painting; pebbles, boulders and grass. Occasionally I find that I have to switch to and from Object mode to refresh the relevant particle system.

11. In the particle settings adjust the number of particles emitted, their rotation, randomness and children to get the desired effect.

Conclusion

You have covered a lot of ground in this chapter. Using brushes on 3D models can go far beyond sculpting and creating surface textures. However you also saw that there were potential deficiencies with these techniques.

There is a large element of skill required for sculpting and the more you practice the better you will get, however, large, detailed objects will inevitably need a powerful computer and benefit from the use of graphics tablets. You also found that the jumble of faces produced by sculpting makes it difficult to get clean lines for both the edges of different materials and in order to allow the mesh to be deformed when it is rigged.

When texture painting, the default UV unwrapping is often inadequate for the detail required in areas such as the face. Also the layout of the different skin components has discontinuities in visible locations. As with plastic surgery, the seams need to be hidden at the back of the head or above the hair line, certainly not in the middle of the face.

In Chapter 7 and Chapter 8 you will learn how to overcome these two problems of topology but the next chapter will concentrate on some of the methods in Blender for importing and exporting models.

6

IMPORT, EXPORT AND ADD-ONS

Increase productivity by borrowing and sharing work

One of the appeals of Blender is the massive community of enthusiasts as well as professional artists. This means that any problem you run into is likely to have cropped up before and you will either find an answer on a forum or wiki, or somebody will have made an add-on to get round the issue: There are thousands of add-ons and the majority are free.

Blender is used by many game creators to make 3D assets as well as by artists as a render engine for models made with other 3D modeling software. As a result there are many built-in import and export functions, and when they're missing there's a good chance someone has written an add-on that can help.

In this chapter you will enable some add-ons that are already built into Blender waiting to be used. In addition you will download and install some add-ons as well as some 3D models in various formats.

Lots of Different File Standards

A quick search online will reveal that there is a bewildering number of 3D model file specifications. This situation is related to the evolution of computer hardware and software allowing more complicated information to be stored, but it has been compounded by companies making propitiatory specifications for commercial reasons. You will most frequently encounter Wavefront .obj, Autodesk .fbx and Collada .dae but there are increasingly lots of .blend files available and glTF has become the new standard for transferring complex scenes.

NOTE Blender allows scripts to be included inside files and executed on load. Although automatic execution is disabled by default, there is a real security risk if you do allow them to run. For this reason you must ensure that any file you download comes from a reputable, trusted source. If you have the slightest misgivings about this then certainly do not allow a script to run.

Download and Import an Asset

OBJ Files

The first format to try is Wavefront .obj which is an older specification and holds only information about the mesh and associated materials, so no armatures for rigging characters and no animations. For this exercise you can search for “download free 3d obj angel” and you should get many results. I will use the model in Chapter 12 on rigging and animation so it would be ideal if you download the same one as shown in Figure 6-2. It seems to be available on *free3d.com* and *open3dmodel.com*, probably more, but any character model with arms and fingers, described as low or medium poly and in .obj format would do. The majority of websites offering free downloads require you to register but many do not. Links to the files and videos for the exercises in this chapter are at *get-into-blender.com 6.IMPORT, EXPORT AND ADD-ONS*.

1. Download the .ZIP file from wherever you find it and unzip into the folder you created for the exercises of Chapter 6.

You will see that it contains an .obj file, a .mtl and several .jpg files. The .obj and .mtl files are plain text and very simply formatted so you can open them with a text editor and see the values of vertices, normals and texture coordinates.

2. Open Blender and delete the default cube then select **File ‣ Import ‣ Wavefront (.obj)** and navigate to the location of the .obj file and select it.

The model that appears will display some of the issues of exporting from one application and importing into another. Not all software uses Z for the vertical axis, sometimes Y is up and Z is towards the camera. Also the scale will often vary from one unit equals one meter used by Blender so the result you see will be a giant lying face down as in Figure 6-1.

Figure 6-1 Imported .obj objects can be too large to see the distant parts

3. Rotate it -90 degrees about the X axis and scale it down so it is just under 2m tall. Use the orthographic view to see the grid lines, if you're not sure if the lines are for 1, 10 or 100m you can see the overall dimension in the Sidebar.

4. As should now be second nature: **Object ▸ Apply ▸ All Transforms**

As everything in the scene is relative to other objects it might seem unnecessary to scale it down, however, viewing the object with the camera and illuminating it with lights will be much simpler if you scale it down. The default camera and point light are five or six meters above the origin and can be used as reference when scaling by eye.

The other problem you will frequently encounter when importing files created on different applications is that the materials will not be converted correctly on import. Blender will try to allocate the information from the original materials to the inputs of a Principled BSDF Shader but sometimes the process is too difficult to do automatically. If the surface has patches of partial transparency, as visible on Figure 6-2, in the Material Properties tab under Settings, try switching the BlendMode from Alpha Blend to Opaque or Alpha Clip.

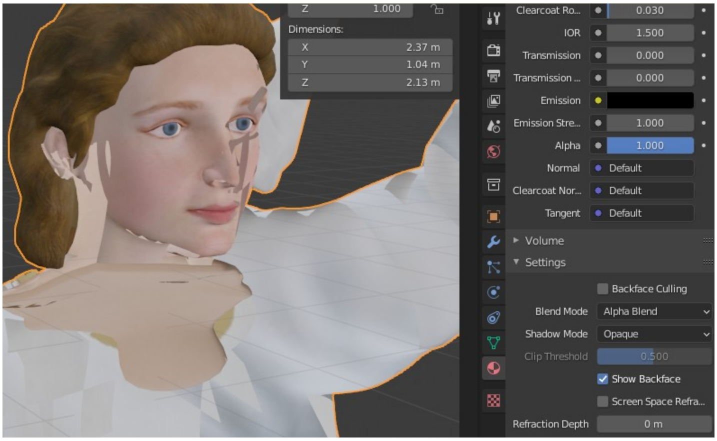


Figure 6-2 Partial transparency due to Alpha Blend

.fbx is another ubiquitous format. It has the advantage that armature and animation information can be incorporated in the file.

FBX Files

To follow the steps in this chapter and Chapter 12 you will need to sign up with *mixamo.com* - it’s owned by Adobe so it’s reasonably reputable and I certainly haven’t been bombarded with spam since joining. Mixamo has a selection of different characters and animations that you can select in combination. As an example of a low resolution mesh character select the Peasant Man and Salsa Dancing as the animation, download it with the default settings (Binary FBX, with skin). Then, as an example of a slightly higher polygon count mesh download character Lola with animation Climbing Up Wall.

1. In Blender start a new scene with **File ▸ New ▸ General**, delete the cube then use **File ▸ Import ▸ FBX** and select the salsa dancing peasant file from where you saved it

NOTE If you use an .fbx file from Mixamo downloaded prior to 2022 you need to do a couple of steps before you click on the **Import FBX** button. In the import options on the right, under Transform un-tick the Use Pre/Post Rotation check-box. Then, Under Armature tick the Automatic Bone Orientation check-box. Without these settings the bones would all be at ninety degrees and, although the animation would work, it would be hard to pose the armature later

The character should be about the right size and the right way up! The armature is obviously present as the bones protrude in various places..

2. Switch Viewport Shading to **Rendered** and zoom in to look at the detail of the face.

3. Open the Overlays drop-down at the top of the 3D Viewport and under Geometry switch on wireframe as shown in Figure 6-3. (The icon is a solid circle overlapping an outline circle.) This draws wireframes over all objects in the scene, to apply it only to a specific object use the option in the Object Properties tab then the Viewport Display section.

Notice how few polygons there are (only 2,400 for the whole model) but there appears to be quite a lot of detail in the rendered material, such as the garments and beard. This has been accomplished by storing surface detail on a normal map image. If you look at the Material in a Shader Editor area you will see there is a specular texture to control shine as well a normal map.

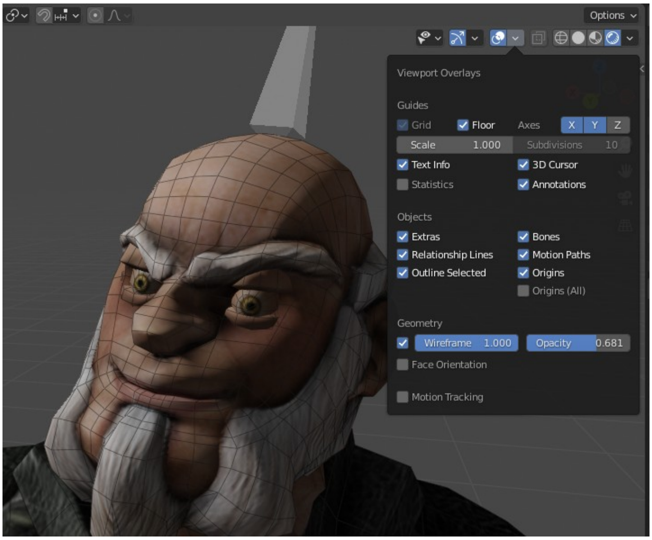


Figure 6-3 Surface detail using a normal map

In Chapter 9 you will learn different ways to "bake" detail from a high resolution mesh such as one created by detailed sculpting, onto a normal map texture. For the time being try running the animation. If you changed the Timeline Editor to a Shader Editor in order to see the details of the Material, change it back and reduce the End frame from the default 250 so it matches the last frame of the dance and it loops smoothly.

4. In order to see the polygon count, open the Overlays again tick the Statistics check-box.

5. Add a Subdivision modifier and increase the levels in the viewport to 1 then 2 then 3, which will generate a mesh of 9,000, 38,000 or 150,000 faces.

6. Run the animation again for each level of subdivision and you should see a noticeable slow-down and, hopefully, appreciate the need for lower resolution meshes for animations, especially in games.

7. Start a new scene and repeat the process for Lola Climbing Up Wall.

This model has more than twice as many polygons as the Peasant Man and if you add a Subdivision surface you will slow the animation even more dramatically. Although the subdivision makes little improvement from a distance, when viewed close up it is quite noticeable, especially where edges are viewed in profile. This is the reason there are two settings for the level of subdivision: rendering a scene can take quite a while so the marginal cost of calculating two levels of subdivision is negligible, but when modeling in the viewport it can be very frustrating.

Compare the arrangement of the mesh on Lola's face with the mesh on the Peasant Man. As well as just dividing the surface into more polygons, they have been arranged to follow contours forming concentric patterns around the lips, eyes, nostrils and ears.

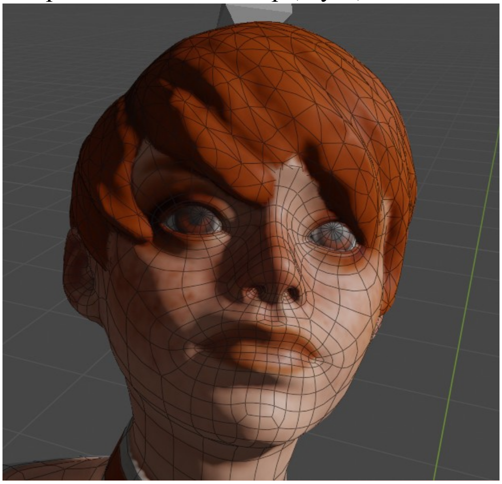


Figure 6-4 Relatively clean topology for Lola's face

By doing this, and by keeping all the polygons as quads, the mesh can be subdivided predictably and can be deformed without causing glitches. In the next chapter you will learn methods for creating clean topology like this.

Export an Asset

Being able to create assets with Blender, then use them in other applications is very useful, but there are obviously limits to the functionality that can be directly transferred. Much of the mesh, armature and animation information is common to many different systems but physics, particles and much of the material nodes is not. A good common denominator format is the .fbx file so try exporting the sculpted manikin you made in the last chapter, then re-import it to see what fell through the cracks.

8. Start a new scene and delete the default cube then append the Person01 object from the blend file you saved at the end of the sculpting and texture painting exercise in Chapter 5.

9. Select both the mesh and the armature then click **File ▸ Export ▸ FBX**. Browse to a location to save the file but, as with importing, do not yet press ENTER or click the Export FBX button yet. There are two non-default options that need to be selected first.

10. In order for the textures that are packed into the blend file to be embedded in the .fbx file you need to set Path Mode to **Copy** and click the inconspicuous button beside it Embed Textures

11. Under the Include section, Limit to, tick the Selected Objects check-box.

12. Now verify the information preserved in the export by re-importing it. Click **Export FBX** then start a new scene, delete the default cube and import the .fbx file you just saved. You don't need to make the changes needed to sort out the bone orientations that you had to do for the Mixamo imports.

13. Check over the imported object. You should find that the armature, mesh, material base color, roughness, specular and so on are OK but the displacement and specular textures have not survived the transfer.

Often the information in the .fbx file is sufficient for your requirements, where you are constructing 3D assets for a video game, for instance. If you need to have more than this, you will have to export all the texture files manually by unpacking them, then reconstruct the material or physics in the destination application by hand.

Add-ons

It is probably fair to say that there is an add-on for almost anything you could possible need to do in Blender. Many add-ons are already included and simply need to be enabled, other add-ons are developed by enthusiasts and are available to download for free, and yet more add-ons are made by professionals and distributed commercially.

The choice of which add-ons you should use depends entirely on the problems you encounter so I will try to avoid coloring your judgment by giving you advice in this book. My approach would always be to get as far as I could without resorting to an add-on in order to understand the issues as thoroughly as possible. Then I would research forums and videos online and use that information to make my choice.

The add-ons you will enable or install as part of this exercise are intended to be “a bit of fun” rather than useful but they should give you a good idea of the general process.

Enable a Pre-installed Add-on

A key part of Blender’s design philosophy has been to keep the core application as general purpose as possible but allow extra functionality to be supplied relatively easily using add-ons. The import and export menu options you used earlier in this chapter are add-ons that are enabled by default and, rather surprisingly, even the cycles render engine is an add-on.

However this flexibility means that add-ons are allowed to introduce menu options, control panels and buttons, almost anywhere in the Blender UI without restriction. The majority of community contributed add-ons, once installed and enabled, can be found in the Sidebar, but not all of them. This means that, in some cases, the only way to discover how to use the add-on is by carefully reading the manual and watching a few videos online. Fortunately, the process of enabling an add-on hosted on *extensions.blender.org* is very straightforward and happens all in one place.

1. Using the menu at the top of the window hit **Edit ▸ Preferences** then the **Get Extensions** tab on the left.

2. You can use search to find add-ons. Leave the default filter using the Official Respositories options at the top then in the search box type “extra” which should find two add-ons, Extra Curve Objects and Extra Mesh Objects. When you open the Add-ons tab on the left you will see that the check-boxes have been automatically ticked in order to enable them, then close the Preferences window.

3. In the 3D Viewport if you select **Add ▸ Mesh** or **Add ▸ Curve** you should now see a host of new options that weren’t there before.

4. As an example try using the Celtic Links curve. Start a new scene, delete the default cube and add an icosphere, switch to Edit mode then squash it to about 20% of its original height.

5. Switch back to Object mode then **Add ▸ Curve ▸ Knots ▸ Celtic Links**, if the last menu option is grayed out it is probably because you didn't have an object selected beforehand.

6. In the Adjust Last Operation panel change Weave Up to **0.05**, Weave Down to **-0.05** and Bevel Depth to **0.07**. The result is shown in Figure 6-5. Select the icosphere and delete it.

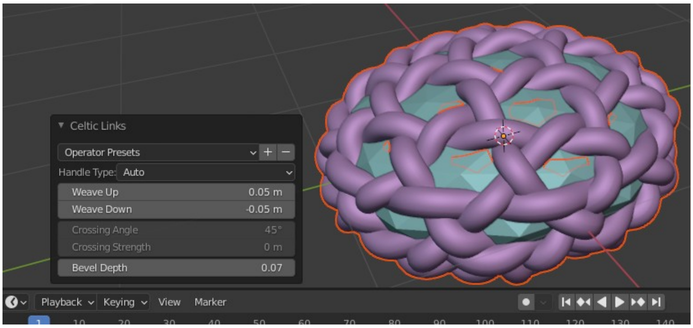


Figure 6-5 Celtic Knot wrapped around an icosphere

As you can see this would have been a very efficient way to generate the gold wire mask you constructed in Chapter 3. I am going to make a quick detour here to show you a slick way to add some dirt in all those cracks formed where the wires cross each other. You will use an input node to the material called Ambient Occlusion which is a way to add realism to computer generated scenes without having to crunch through the the full ray tracing work that is undertaken by the cycles render engine. Ambient Occlusion basically finds, from the perspective of a given point on a surface, how much of the sky is being obscured.

7. To give the knot some context add a plane, scale it up, move it just below the Celtic knot and create a shiny dark gray material for it.

8. There's no point re-inventing the wheel so append the gold material you created for the first exercise of Chapter 3 and select this material for the Celtic knot object.

9. As you did in Chapter 3, change the background lighting to use an environment texture and check what this looks like by setting Viewport Shading to **Rendered**.

10. In the Render Properties tab on the right leave the render engine set to Eevee but tick the check-box for Ambient Occlusion and Screen Space Reflections.

11. Now you can start to modify the material to use ambient occlusion. Switch the Active workspace using the Shading tab at the top of the window, then add an Ambient Occlusion node using **Add ▸ Input ▸ Ambient Occlusion**.

Now see if you can figure out a way to use the AO output of this node to change the color and roughness of the wires where they cross over each other. Try to think through the logic before looking at Figure 6-6. As a hint the AO output will go from 0 where the point is at the bottom of a deep crack to 1 when there is nothing around the surface, you can use two color ramp nodes to map this value to a range for the Base Color from dirty to clean gold and for Roughness from 1 to 0. Remember white is 1 and black is 0.

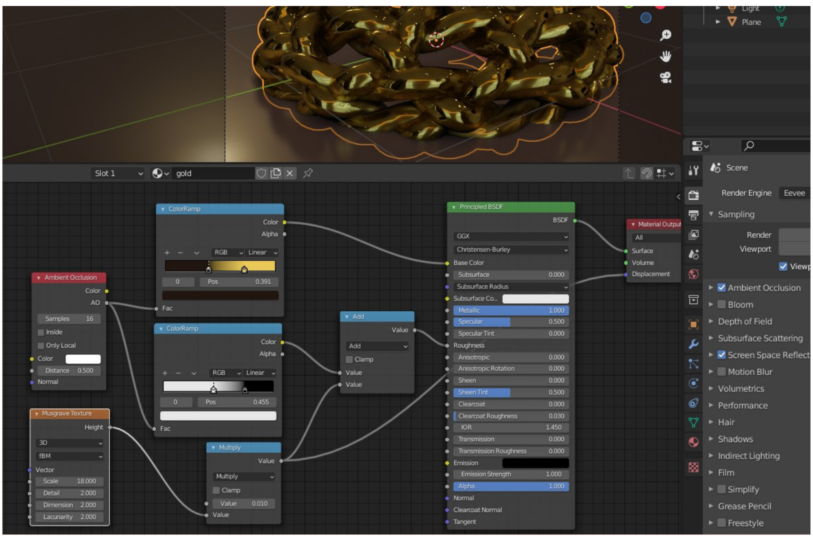


Figure 6-6 Using Ambient Occlusion for dirty corners

By this stage I'm sure you appreciate that you could have used the AO output to combine two different shaders just like you did for the rusty fridge material to get a much better grungy look in the nooks and corners. Also, see what it looks like using the cycles render engine, it is a little bit more realistic but eevee is pretty good. In Figure 6-7 the left half was rendered using cycles, taking five minutes, and the right half was rendered using eevee in five seconds. It's quite hard to see the difference.

Figure 6-7 Cycles on the left, Eevee on the right

Spend a little time experimenting with all the extra meshes and curves. There is so much and I couldn't hope to do the add-ons justice here, however you will use the wall factory in Chapter 12 to provide Lola with something to climb up.

Download and Install an Add-on

Downloading and installing add-ons not held in the Blender repository carries the same warnings as when downloading a .blend file but even more so, in order to use an add-on you *have* to execute someone else's code. My advice would be to stick to add-ons that are popular in terms of reviews on websites and discussion in Blender forums.

Blender add-ons are written using the python scripting language and, although some simple add-ons might be contained in a single file with a .py extension, they are usually distributed as a .zip file. Blender can install add-ons in either form but the normal route uses the zip file without expanding it. Originally, the example I suggested was Oliver Weissbarth's book generator, but as Blender has progressed this add-on has occasionally stopped working. As an alternative I have made a cut-down version saved as *BookGenSimple.zip* in with the blend files for Chapter 6 on *get-into-blender.com*

1. Download from the link and save the .zip file on your computer without unpacking it.

2. In Blender select the top menu **Edit** ‣ **Preferences** then **Add-ons** as before.

3. In the tiny drop-down menu at the top right **Install from Disk** and navigate to the .zip file you just saved then click **Install From Disk**

4. The check-box next to the new entry Book Gen Simple will have been automatically enabled. Close the options window.

5. The original BookGen add-on followed the normal convention of putting all its controls in a tab in the Sidebar. However the simple version uses **Object ▸ Book Gen Simple**. It creates books at the 3D cursor and the number is controlled in the Adjust last Action window. Figure 6-8 was very quick to set up using the wall factory and this book generator. Each book is a separate object so it is quite simple to create random colors for each cover, using the object info node random output and a color ramp, as you can see in the material nodes at the bottom.

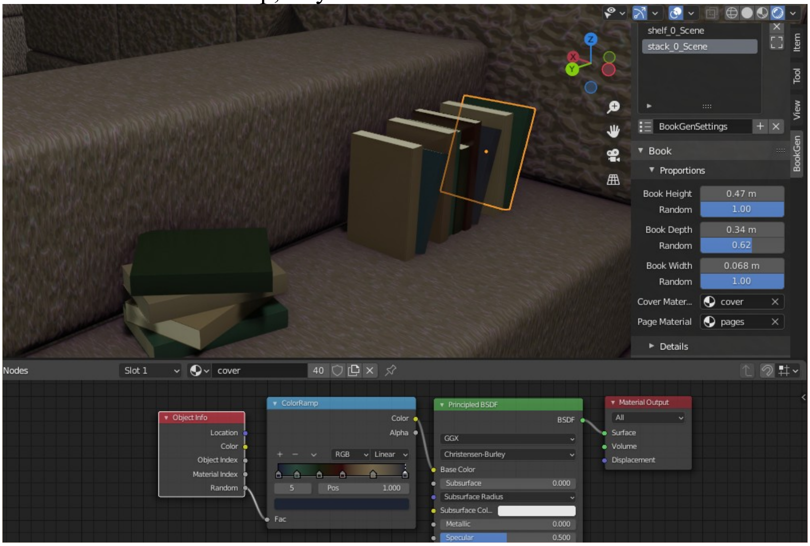


Figure 6-8 BookGen with random cover colors

Conclusion

Although there is enormous scope and variability of resources, the process of importing and exporting to and from Blender is quite straightforward. Using other peoples' models, components and add-ons can, potentially, transform your productivity and push your creativity to a whole new level.

Seeing how other artists have gone about making things is also a good way to learn. You saw the difference between the low poly model Peasant Man and the medium poly model Lola and how they both added surface detail using a normal map. In Chapter 9 you will learn how to transfer the detail from a jumbled, high poly mesh, such as the one you created by sculpting a face in the last chapter, onto a normal map matching a clean, low poly mesh. However, before that, you will need to make that low poly mesh, and that is the subject of the next chapter.

7

MORE THAN ONE WAY TO SKIN A CAT

Building one mesh over the top of anther one

Making a mesh conform to the shape of another one is a surprisingly common job in certain areas of 3D modeling. The process of creating a regular low poly mesh after sculpting is called re-topology but the mesh editing techniques used for it are important for all areas of 3D modeling in Blender.

Over the course of this chapter I will demonstrate the value of making objects comprised from quadrilateral faces as opposed to triangular or higher sided polygons. I will then show different methods for making and fitting clothes to your manikin from Chapter 2 before starting to apply the re-topology process to the sculpted face. As I mentioned when you did the sculpting, there are many excellent, ready-made face meshes available free on the internet, so your objective here is not to reproduce the work of expert modelers and anatomists. Doing re-topology on a complex mesh can be rather slow, and there are several add-ons that could make your life much easier, so you will simply be getting an appreciation of what's involved and learning skills you can transfer to other modeling you do in the future.

After the re-topology exercise you will learn two useful features of Blender that allow you to change the shape of an object by moving a slider, Shape Keys and Drivers. In some ways both of these complement the use of armatures to deform meshes and they would have fitted logically with the work on animation in Chapter 12. Links to files and videos used in this chapter are at *get-into-blender.com 7.MORE THAN ONE WAY TO SKIN A CAT*

The Value of Quads

After a while watching Blender videos online, or reading helpful posts in forums, you might become aware of a general sentiment that "clean" topology is good, and clean topology always consists entirely of quads. This isn't *absolutely* true, like all good rules there are exceptions, but it *is* a good rule!

As a quick exercise make an object from quads then see what it looks like when you convert it to triangles.

1. Start a new scene and, with the default Cube, switch to Edit mode and extrude one of the faces five or six times so each extrusion is about the same size as the original Cube.

2. With the extruded face still selected turn Proportional editing on with a radius of influence a little less than the length of the object. Rotate the end face and move it so the line of cubes forms a curve.

3. Switch back to Object mode and add a subdivision modifier. The result should look like the "banana" in Figure 7-1

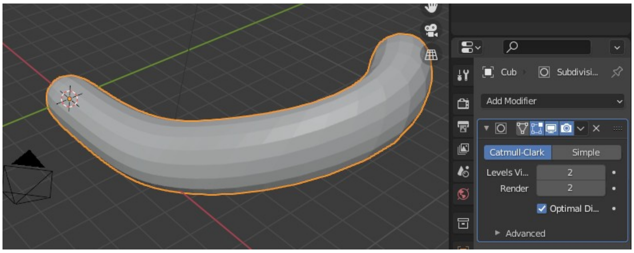


Figure 7-1 Subdivision of quads

4. In Edit mode select all the faces then **Face ▸ Triangulate Faces.**

Notice that the whole banana surface has become subtly eccentric and distorted as shown in Figure 7-2

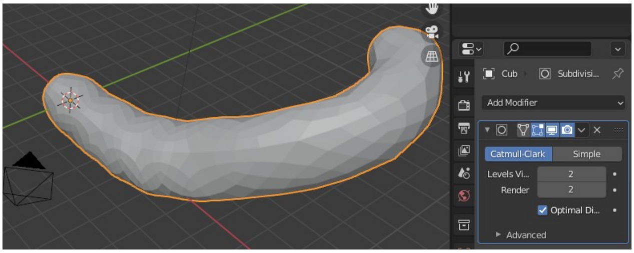


Figure 7-2 Subdivision of triangles

There are further complications associated with meshes that are not constructed from quads and, as you complete the exercises in this chapter, I will point them out.

Create a New Object using Part of Another Mesh

For the first approach to making a mesh following the shape of another mesh, you are going to make a shirt for the Person01object from Chapter 5. You will copy a part of the body mesh, modify it as required, and then *shrinkwrap* it back onto the original mesh.

Converting Triangles to Quads

Before making a copy of part of the mesh you can improve it by running the reverse process of the triangulate faces function you used above.

1. Start a new scene and delete the default cube, you don't need to save the banana! Append the Person01 object from the blend file you saved at the end of Chapter 5.

2. To ensure that the armature is unposed select the armature object, switch to Pose mode, select all the bones and hit **Pose ▸ Clear Transform ▸ All**.

3. Switch back to Object mode, select the Person01 object then change to Edit mode. You will see that the body and arms are built from triangles rather than quads. This is a by-product of using the dyntopo option while sculpting. Deselect all the vertices with ALT-A then try to select an edge loop running round an arm with ALT-LMB on a circumferential edge. See Figure 7-3 You will find that you only select a single edge.

4. Blender can try to convert triangles to quads for you. Select all the mesh with A then click **Face ▸ Tris to Quads**, deselect all the vertices then retry selecting an edge loop round an arm. This time you will find it works, but note that tris to quads didn't convert all of the quads as you can see on the right of Figure 7-3.

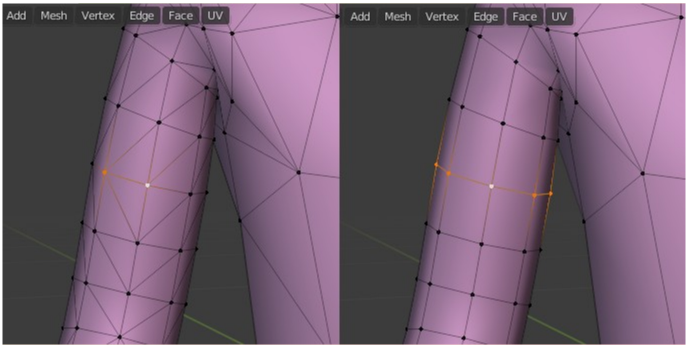


Figure 7-3 Loop Select and Loop Cut only work with quads

The reason that the loop selection didn't work with triangles is because the algorithm works from vertex to vertex by finding the opposite edge from the one joining the previous vertex. This only works reliably for vertices with four, or at the very least, an even number of edges.

Possibly more significant than loop selection is the ability to make loop cuts, a technique often used when building meshes "by hand", both for re-topology and for hard surfaces.

Separating a Selection

You will now duplicate part of the body mesh and use it to create a new shirt object that you can further refine.

5. Switch to orthographic view in the Y direction, toggle X-Ray on then using circle selection select the vertices for a short sleeved shirt. You will add a mirror modifier so you only need half a shirt. Your selection should include vertices on the center line but none from the opposite half. Figure 7-4 shows where I cut off the sleeves and neck, but use your own sartorial judgment.

6. Duplicate the selected mesh using SHIFT-D then LMB without moving the mouse at all. If you move the mouse before clicking, it will move the duplicated mesh, which you don't really want.

7. You can create a new object from the mesh. The duplicated mesh is still part of the Person01 object but the vertices will still be selected, so use **Mesh ▸ Separate ▸ Selection**

8. Switch to Object mode, select the new object and rename it Shirt. You will find that the object is a child of the Armature and, if you check the Modifier Properties tab, it still has an armature modifier though if you pose the character will only pose half the shirt.

9. Add a mirror modifier to create a full shirt. If you try posing the armature now you will find that the mirrored vertices move too, which is not what's required. You will need to regenerate the armature modifier later so, in the Modifier Properties tab, delete the existing one. At this point your shirt should just have a mirror modifier, if the object has inherited any others you added to Person01 experimentally then delete those as well.

10. In the Material Properties tab delete the two existing materials skin and eyes then create a new material, rename it shirt and set the Base Color to a strong color different from the existing skin material. Change the Viewport Shading to Material Preview and, hopefully, you *won't* see your new shirt in all its glory, though you might see some of it with strange banding. The faces of both objects are in the same location so you need to make sure that the shirt is outside the body.

The Shrinkwrap Modifier

11. To ensure the shirt is visible above the surface of the Person01 mesh as in Figure 7-4, add a shrinkwrap modifier to the Shirt object. Set the Snap Mode to **Outside**, the Target to the Person01 object and the Offset to 0.001m.

The shrinkwrap modifier can be a very useful aid when making clothes like this, but it can also be used in many other situations where you need one mesh to line up with the surface of another one.

The two settings, *Wrap Method* and *Snap Mode* are a little hard to understand at first.

Wrap Method controls the way that wrap locations on the underlying mesh are found. For simple meshes like this, the default, Nearest Surface Point is mainly fine, and it's quick to calculate so the modifier can sometimes be left in place on a garment that is being animated and subdivided, to prevent skin poking through. In some locations the calculation may not work perfectly, for instance under the arms of my manikin where the surface folds back on itself. In these cases the Target Normal Project will often give an improved result and for even more difficult situations the Project option can sometimes be needed as it has more fine-tuning available.

Snap Mode controls the way the vertices stick to the surface they are wrapping. The name of each option gives a good description of how each behaves but Inside and Outside need a little clarification. If Outside is chosen then, where a vertex is already outside the target object by more than the Offset amount, it won't be moved, only vertices that are inside the target object (plus Offset) will be moved outwards. This is useful where garments have pleats, pockets or collars that you don't want to get "sucked" in against the target body.

Basic Alterations to the Mesh

You will now add some features to the shirt creating a collar and a breast pocket.

Extrude and Tweak

12. Switch to Edit mode and using front and side orthographic views move the vertex at the neck away from the center line to form an open collar as in Figure 7-4 on the left.

13. Select the vertices around the collar, extrude them in the Z direction then, by rotating the view often and moving each vertex a little at a time, create a folded over collar. For this kind of adjustment it's often handy to switch the select mode to Tweak by long pressing the top entry in the Toolbar. In tweak mode you can select and drag a vertex with one mouse click. The result should look something like Figure 7-4

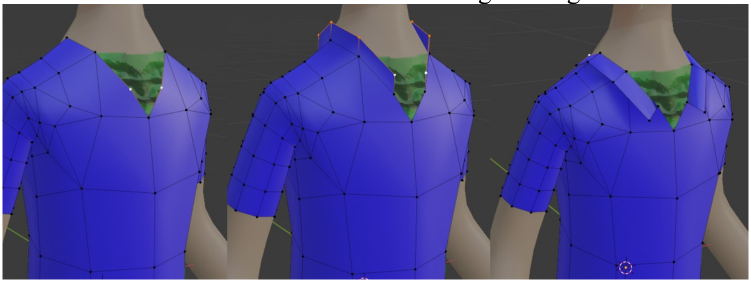


Figure 7-4 Extrude then form the collar

Sometimes it's tricky to see what's happening in Edit mode if there are modifiers, such as shrinkwrap or subdivision acting on the object. You can control which modes show or don't show the modifier by toggling the four buttons next to the name in the modifier controls, in Figure 7-1 and 7-2 you can see that three button are on but the left-most, On Cage, is off. Use the X-Ray view in order to be able to see hidden vertices and, occasionally, for modifiers such as shrinkwrap, it helps to to switch On Cage on. For the subdivision modifier it is usually better to turn Realtime off, so you don't see the effect of the modifier in the 3D Viewport. Juggle these different settings as you work to keep track of what you are changing and what the final result will be.

Usually, when creating a mesh like this, your objective will be to make something as simple as possible, but which can use a subdivision modifier to create the desired detail. The target mesh here, the Person01 object, is rather angular and coarse so the results will be peculiar, but it's still worth checking what the problems will be. The next thing you are going to do is add a breast pocket on one side of the shirt which will break the symmetry.

14. First of all apply the mirror modifier, then add a subdivision modifier and move it to the top of the stack. If the shrinkwrap is applied after the subdivision it will cover the bumps better.

15. Several issues will now be apparent, the most obvious of which is the skin poking through the shirt. Although you will solve that problem with a mask, which you will use later in this

chapter, you should see what the effect is of changing the shrinkwrap modifier Wrap Method to **Target Normal Project** and increasing the Offset to **0.003m**.

Bevel

The second issue is that the collar has smoothed out the sharp bend and reduced its width. This will also be a problem when you add the shirt pocket, as you will see later. The solution is either to add some edges near to the sharp corner to help define the subdivision process or alternatively, add a bevel modifier before the subdivision. There is actually a third options that applies to the subdivision modifier and that is to increase the edges' crease value. For this exercise all three approaches will produce similar results and have similar drawbacks, but you will use the one that creates new edges in the mesh. Being able to see the edges will help you understand and solve the problems caused by non-quad faces later in this exercise.

16. Bevel the mesh by switching to Edit mode and selecting the vertices along the inside edge of the collar. Then select **Edge ‣ Bevel Edges**, move the mouse to create the bevel then LMB to fix the it. Don't worry about getting the right amount for the bevel as you will now open the Adjust Last Operation pane and alter the Width to **0.01** and Segments to **3**. You will have to adjust these values to suit your model, in my case I found that the shrinkwrap modifier was pushing the the underlying mesh through the layer above. Even then I needed to tweak the vertices in a similar way to moving the control points to define the nurbs surface in Chapter 3. Figure 7-5 shows the before and after screenshots.

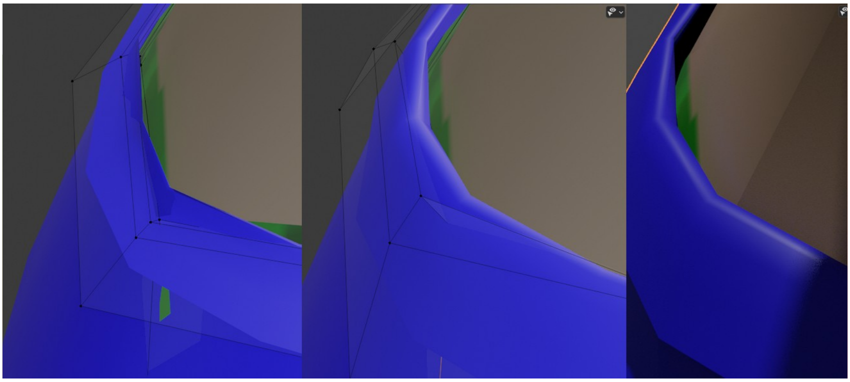


Figure 7-5 Tweaking vertices to fix the bevel for subdividing

The Knife Tool

Your next job will be to create a breast pocket for your shirt using the knife tool to create the extra edges and vertices in the desired location. In the description below I refer to using the tool from the Toolbar menu but when you are making lots of changes to a mesh using different tools you will find it more convenient to use the shortcut. In this case it's easy to remember; K.

17. Still in Edit mode select the Knife tool from the Toolbar and draw a rectangle where the breast pocket will go. See Figure 7-6 stage 1. You will find this easier if you toggle the shrinkwrap and subdivision modifier Realtime option off so they don't distort or hide the mesh in the 3D Viewport. Vertices will be created where you LMB or where the red line crosses another edge. There is automatic snapping to edges or vertices so this tool is very easy to use. When the shape has been defined, press ENTER to fix it.

18. To make the patch shape into a pocket, extrude the rectangle you have created by a small amount as in Figure 7-6 stage 2, but note that when you enable the subdivision modifier it distorts the pocket drastically, as seen in stage 3.

19. The simplest way to stop the edges being smoothed like this is to add a bevel. Select the pocket edges, but excluding the ones crossing the center of the rectangle and bevel them, in Adjust Last Operation set Segments to **1** and an Offset of **0.001**. Now the subdivision and shrinkwrap should look much better, see Figure 7-6 stages 4 and 5. However the faces around the pocket look distorted because they are no-longer quads. That's what you should tackle next, it's the hard part of re-topology, but it's also the most satisfying.

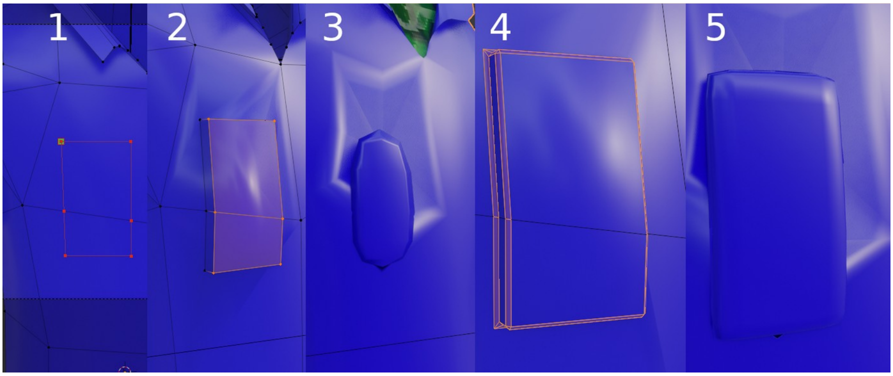


Figure 7-6 Cut, extrude and bevel a pocket

There are two ways to revise the arrangement of edges in an existing mesh. The first is to select edges that are in the wrong place then press DELETE and use the Dissolve Edges option, which leaves a larger face, then add back in edges in the correct location using knife, loop cut, bevel or subdivide. The second method is to delete faces and edges leaving a hole, then fill them back in manually using F.

Both the above methods are fine and useful in different circumstances, but they cannot be easily be mixed without the danger of creating faces on top of each other. Figure 7-7 shows the sequence using the first technique. You will essentially use the other technique for the whole second half of this chapter.

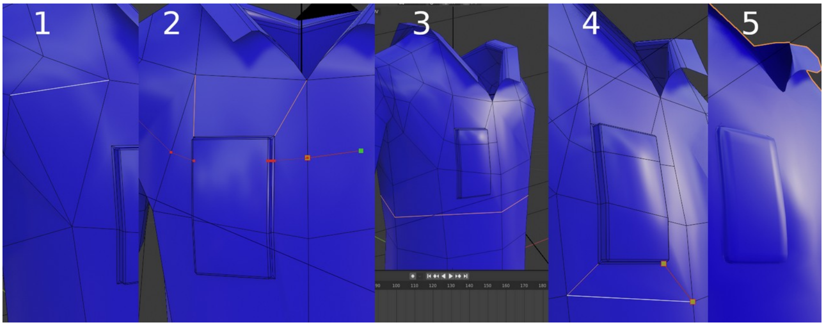


Figure 7-7 Using the knife tool to create quads

20. Select each edge that needs to be removed then DELETE ▸ **Dissolve Edges** you can probably find some edges that will immediately turn two triangles into a single quad.

21. With the knife tool divide up polygons with more than four sides. Your objective is to make quads with corner angles neither too acute or too obtuse. If Figure 7-7 stage 2 you can see that the quad above and to the left of the pocket has three vertices nearly in a line which will limit the ways it can bend, and should be avoided.

22. In Figure 7-7 stage 3 I have used **Edge** ▸ **Loop Cut and Slide** to reduce the gap between the bottom of the pocket and the horizontal edge below it. By Figure 7-7 stage 5 the strange distortions around the pocket have become much improved.

Before you start constructing the mesh for the face it’s worth spending a little time reviewing methods to avoid triangles and also how to cope with bits of skin poking through clothing, which can happen even with the optimum shrinkwrap settings.

Avoiding Triangles

If you look at my mesh in Figure 7-8 stage 1 you can see that, although there are lines of edges looping around the shoulder and under the arm, there are areas where it seems impossible to get rid of all the triangles. The problem is that if a polygon has an odd number of sides it can’t be divided up into quads, even if extra vertices are added in the middle. And, if an extra vertex is added on one side to make the number of sides even, then the polygon sharing that side gains an extra side. Figure 7-8 stage 1 shows my shirt mesh to start with, then stage 2 shows it with all the obvious triangles removed, leaving a problem polygon with seven edges.

There are different ways to solve this problem but they all boil down to converting polygons to even sided ones. Sometimes there are two polygons near to each other that allow edges to be moved so one gains an edge and the other loses one. In this case there is no obvious way to do that, so the simplest solution is to add a loop cut as in Figure 7-8 stage 3 or, better, stage 4.

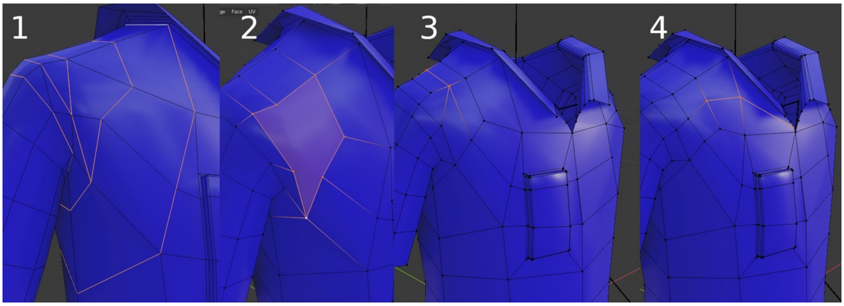


Figure 7-8 Converting polygons to quads

When making the better shaped quads in Figure 7-8 stage 3 and 4 I moved the vertices along edges using **Vertex ▸ SlideVertices** to keep the mesh shape as close to the body shape as possible. Though the shortcut in the menu for sliding vertices or edges is SHIFT-V you can also press G twice, which is more like the normal move shortcut so might be easier to remember.

The Mask Modifier

You may have noticed, while working on the shirt, that bits of the Person01 mesh poke through, however much you adjust the shrinkwrap settings. The classic solution to this problem is to add a mask.

1. Sometimes, on coarse meshes, the mask modifier doesn't work well with the shrinkwrap modifer still on the clothes, so, once you are happy with the shirt topology, apply the subdivision and the shrinkwrap modifiers.

2. Remember you also need to regenerate the armature modifier now the mirror modifier is no longer there, so select the shirt then SHIFT select the armature and click **Object ▸ Parent ▸ With Automatic Weights**

3. Create a vertex group on the parts of the underlying mesh you want to hide. Select the Person01 object then in Edit mode select all the faces that will be hidden by the shirt and add them to a new vertex group "under_shirt".

4. Add a Mask modifier to the Person01 object and specify the under_shirt vertex group. Click on the Invert button as the default behavior is to show rather than hide the vertex group.

For such a coarse mesh as the body of Person01 you will find it difficult to get the balance of vertices to include in the under_shirt vertex group. You will probably have to add and remove them in a process of trial and error.

Re-Topology

For the second half of this chapter you will create a new, clean mesh for the face that you sculpted in Chapter 5. Take a quick look back at Lola's face in Figure 6-4. You can see that the quads form loops around the eyes, mouth, nostrils and ears. Also the edges of quads run along natural ridges or grooves on the face. Something to bear in mind when creating a mesh like this is that it is very easy to subdivide later by adding loop cuts or bevels, so keep everything as coarse as possible to start with. There is a bit of setting up required:

1. You will start your new head topology from the simplest possible mesh, so add a new plane to the scene and immediately switch to Edit mode. All the vertices will already be selected so scale the whole thing down to about the size of an eye and move

the plane up and to one side so it is half in and half out of a cheek.

2. Rename the plane in the Outliner Editor to new_head. Add a mirror modifier to the new_head object, which you can do while still in Edit mode.

3. Change the snap mode to Face and switch it on.

These first three steps are *essential,* the next four just enable you to see what you’re actually doing and they make the process much easier.

4. In the Object Properties tab, Viewport Display section, toggle In Front to on.

5. In the Active Tool and Workspace settings tab, under Options, toggle Auto Merge on. As you work you will probably have to adjust the Threshold value, as well as the merge threshold in the mirror modifier. Coarse settings make it easy to place vertices so they snap together but sometimes you will find that a vertex jumps from where you placed it because it was too near to another vertex.

6. Set the select tool to Tweak mode and the selection mode to vertex *and* edge. It is possible to use more than one selection mode at once by pressing SHIFT when you click on the buttons.

7. Use Viewport Shading set to Solid and in the options drop-down set Backface culling on.

Extruding and Filling

Once you have set up snapping to a target surface, and you can see the mesh as you work on it, you are good to go. For the beginning of this exercise you will use the traditional extrusion and tweaking system. There is much to recommend this method but we will cover some additional tools that are more efficient in some circumstances.

8. Observe how the snapping onto a surface works by moving each of the vertices of the new plane a small amount. Select one of the side edges and extrude it then rotate it and repeat a few times to form part of a ring of quads going around the eye. Figure 7-9 stage 1 shows the ring complete but needing the last vertex to be dragged so it snaps onto the corner of the starting plane.

Bridging Edge Loops

After a bit of practice extruding, rotating, scaling and positioning the edge of the new plane, this method becomes rather efficient. You can probably see how you could tile the whole head using this technique. However for rings of quads such as these around the eye it is possible to use the edge loop bridging method you used to make the sail in Chapter 3. Although you could use just two loops, one at the inside of the existing faces and one on the inside of the eyelid, and set the number of cuts as required, for good topology you should aim to position the edges more precisely.

There are natural ridges and creases around the eye and you should aim to make the edges of the new topology coincide with these.

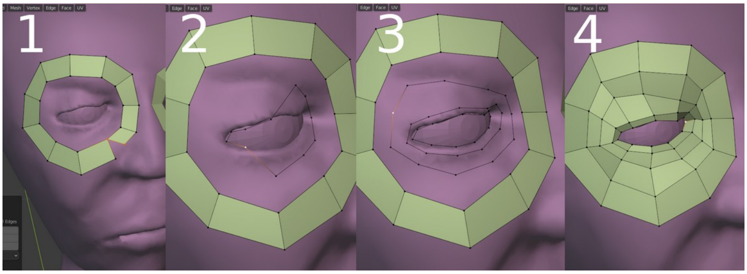


Figure 7-9 Extruding edges and bridging edge loops

9. First make the edge on the inside of the eyelid by selecting the edge loop around the inside of the ring of quads you just created, then duplicating it and scaling it down a little. For this stage of your work it is worth toggling the Auto Merge off in the Active Tool and workspace tab, as you want vertices to stay where you put them.

10. Move each vertex of the new ring to lie at the junction of eyeball and the inside of the eyelids as in Figure 7-9 stage 2.

11. Repeat the process for the ridge on the edge of the eyelid and for the crease above and below the eyelid as in Figure 7-9 stage 3

12. Select all four loops and **Edge ‣ Bridge Edge Loops**. Sometimes the arrangement of vertices on sharply curved surfaces can cause confusion for Blender which might generate crossed edges and overlapping faces. If this happens you have to select a smaller number of loops and fill the area in a number of stages. For instance you should be able to bridge the outside

three loops unambiguously then bridge the inner two edge loops, where there is a sharp fold, with a single ring of faces.

NOTE There are no simple keyboard shortcut for **Edge ▸ Bridge Edge Loops** and **Face ▸ Grid Fill** so it might be convenient to add them to Quick Favorites by RMB on the menu entry. This can then be accessed later by pressing Q

13. To work on other areas of rings, extrude a vertex out towards the ear, nostril, mouth and neck and build up rings of quads in a similar way to the eye. You can see in Figure 7-10, where the faces have been built for the nostrils, that Blender sometimes points the normals inwards instead of outwards. This happens if you bridge loops without any references faces around the edge. You can fix this by selecting all the vertices facing the wrong way, then select **Mesh ▸ Normals ▸ Flip**. However to avoided this problem in the first place you should always extrude out from an edge so that at least one part of the edge loop can be correctly orientated when you start filling in quads.

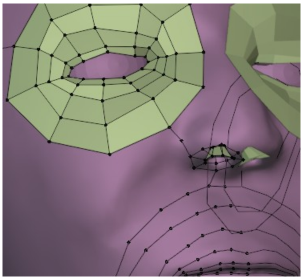


Figure 7-10 Flipped normals around nostrils

Filling with One Face at a Time

In Figure 7-ll stage 1 you can see that filling in the ear with edges along the ridges and grooves runs into a problem where there is a relatively featureless area in the center but there needs to be more detail, and a hole, at one side. If concentric loops continue to be added they will become far to congested below and to the right of the ear hole. Rather than using bridging edge loops, you can add quads and edges one at a time using **Vertex ▸ New Edge/Face from Vertices**, or in practice, the shortcut F.

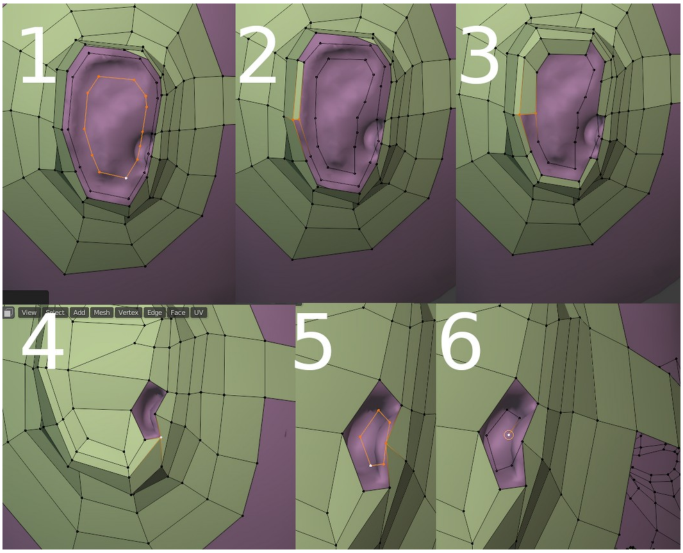


Figure 7-11 Filling in the ear

14. Figure 7-11 stage 1 shows the stage where two loops have been duplicated and scaled down but not yet bridged because they have started to overlap the ear hole.

15. To create a single quad select four vertices or, more easily, two unconnected edges opposite each other, and press F. You can see the new quad formed in Figure 7-11 stage 2.

16. Select one of the sides of the newly created quad which is facing down a “track” of unconnected edges. In Figure 7-11 stage 2 you can see the edge I have selected. Now press F and, not only will a new quad appear, but the next edge down the track is automatically selected so you can press F repeatedly to fill in a strip of quads. This is actually a very fast and controllable way to build the mesh.

17. In Figure 7-11 stage 3 the circumferential quads are being filled as far as the awkward area in the bottom right. In stage 4 the flat area on the left was chosen to have an even number of edges around it, which are now filled in. The hole on the right, also has an even number of edges.

18. Figure 7-11 stage 5 shows what sometimes happens with automatic merging in confined spaces. One of the vertices from the duplicated edge loops forming the bottom of the ear hole has snapped to merge with a vertex on the loop outside it. Don’t

worry if this happens. You can detach the miscreant by selecting the vertex and hitting **Vertex ▸ Rip Vertices**.

Joining Mismatched Quads

A perennial problem in re-topology is joining a neat arrangement of quads in one area with the quads emanating from another area. Figure 7-12 stage 1 shows an example where the smaller quads needed around the nostrils and tip of the nose meet up with the quads from the eye. You can see that an elongated quad between the nose and the cheek has effectively turned the corner from around the eye to across the nose, but the next rows of quads going up the nose will all have to be merged into a single quad. Ideally the next few rows of quads would be fairly square, like the ones in Figure 7-12 stage 2.

The first thing to notice is that there have to be an odd number of rows of quads merging into one row as that creates an even number of sides to the polygon. You can see, in the middle of Figure 7-12 stage 2 that there will be a purple rectangle with six sides once the top vertices are joined. Stage 3 shows one reasonable way to fill it, there are hosts of other possible arrangements.

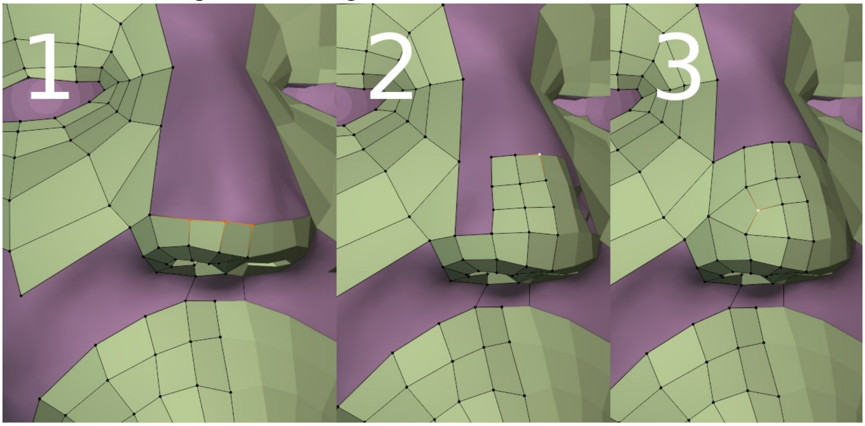


Figure 7-12 Merging one set of concentric faces into another

You can see in Figure 7-12 that the mesh at the top of the mouth could be joined to the nose and eye relatively easily continuing with this extrusion technique but, instead, I will introduce you to a useful alternative.

Poly Build

When you were using the method of bridging edge loops you probably noticed that, after you duplicated a loop, the new vertices didn't snap to the surface until you had tweaked each of them in turn. Even if they were in approximately the correct position when viewed face-on, you had to move them around a bit to force the snap to work. So, in some ways it might have been just as efficient to create each quad individually if that could be done as succinctly as extrusion. Well, in many ways, it can, using the Poly Build tool.

1. Select the Poly Build tool from the Toolbar, then move the mouse around noting when vertices or edges turn blue. Press and hold CTRL and move the mouse around noting when triangles or quads appear. Now hold SHIFT and see when vertices or faces become highlighted red for deletion.

Hopefully you found some difficulty in a couple of locations. This is the reason, in my opinion, that the Poly Build tool isn't the perfect way to do all re-topology. It's often quite difficult to make it select the right edge, vertex or face for extrusion or deletion. However sometimes it's very efficient so it's worth knowing how to use it.

2. Move the mouse until a corner vertex is highlighted blue as in Figure 7-13 stage 1. Now press CTRL to see the new quad, if you press CTRL when an edge is highlighted you will get a new triangle. I find that it is better to press and hold LMB very quickly after pressing CTRL, before poly build changes its mind. The alternative approach is to press CTRL all the time and move the mouse around to get the quad forming where you want it. The latter method is probably quicker in simple areas but the former method seems less frustrating where the topology is already congested. Drag the new vertex to create the correct shape for the quad, as in Figure 7-13 stage 2.

3. To extrude a new quad from one edge of an existing quad, press and hold LMB while an edge is highlighted as in Figure 7-13 stage 3. Move the remaining vertex after extrusion by dragging a single blue dot shown in Figure 7-13 stage 4, exactly the same as when using tweak select.

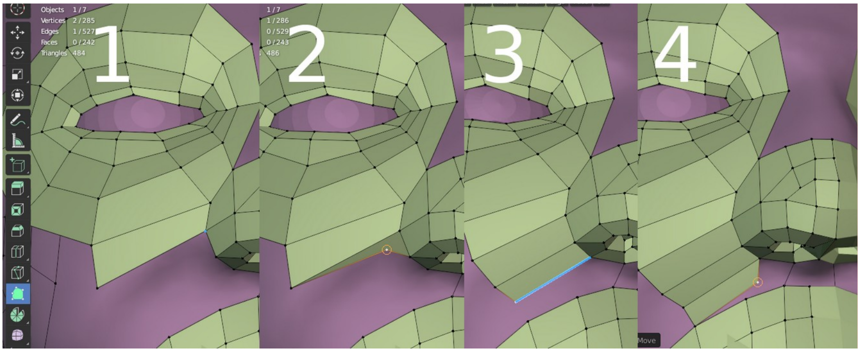


Figure 7-13 Using the Poly Build T ool

Use the poly build system to fill the areas joining together the mouth, nose, eye, ear and neck. Try to figure out how to divide up the shrinking spaces as you expand the areas towards each other.

As an example, Figure 7-14 stage 1 shows the underneath of the chin once I had expanded the rings around the neck up and joined them to the columns that had developed running vertically down between the ear ring and the eye ring.

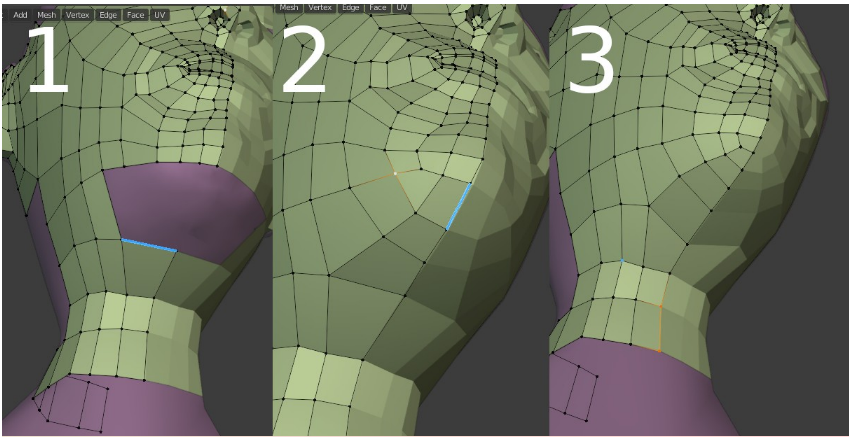


Figure 7-14 Filling under the chin

To start with, the purple area was surrounded by eight edges which could have been divided up into quads, but that would have left a very long edge along the center line, which actually needs a slight curve. To keep an even number of edges around the area to be filled, I had to divide the long edge into either three, as in Figure 7-14 stage 2, or five as in Figure 7-14 stage 3.

This section on using poly build would not be complete without a mention of two other add-ons that are available in Blender extensions, but not enabled by default. The first is F2 which changes the behavior of the shortcut F such that if a corner vertex is selected, in a similar position to the the blue highlighted one in Figure 7-13 stage 1, you can create a new quad in a very similar way to poly building. The other is Polyquilt which gives a range of tools for quickly building up meshes using quads. Polyquilt can give a massive increase in productivity, especially if you have facilities to draw with a graphics tablet rather than the mouse, however you should learn the basic, manual methods first, in order to derive the most benefit from any add-ons that you decide to use.

Grid Fill

Grid Fill is a method for instantly filling areas with mesh, very similar to bridging edge loops. Try using it to complete the mesh for the cranium. In Figure 7-15 you can see the approach I took to create a rectangular area to fill.

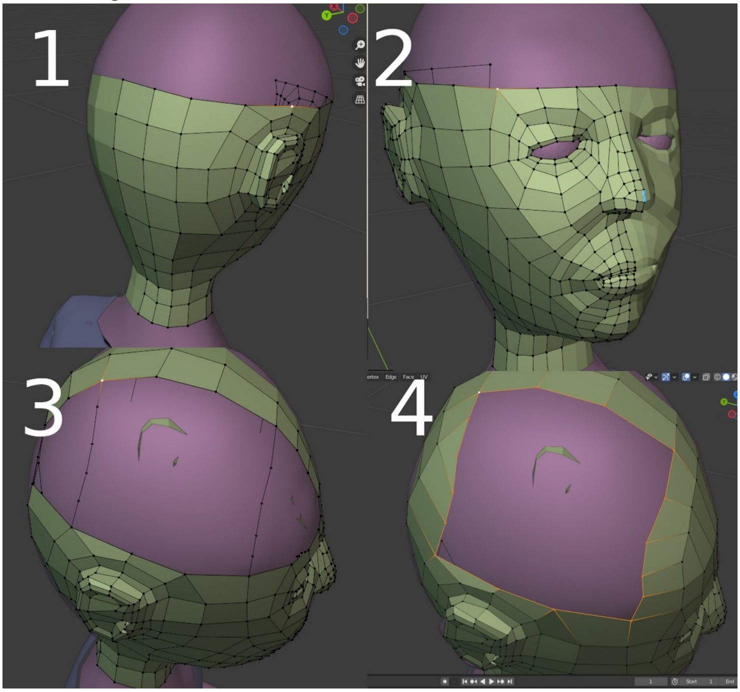


Figure 7-15 Grid Fill the cranium

1. Start by building a line of quads over the top of the head then mark out a rectangle so each side has the same number of vertices as the opposite side as shown in Figure 7-15 stage 3.

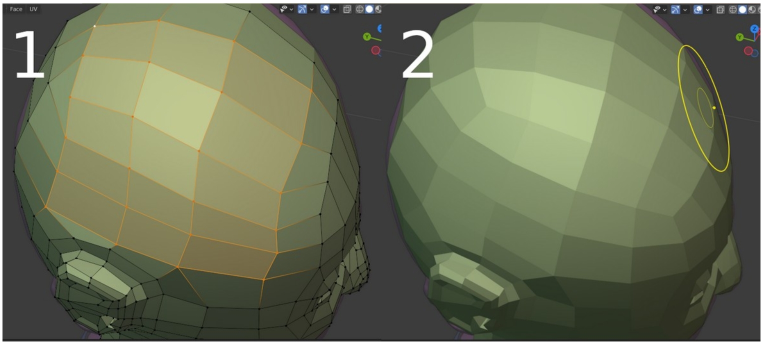
2. Fill the triangular areas at the front and back of the head with quads. You might need to add or remove edge cuts in order to make the number of edges around the perimeter of each triangle an even number. Don't worry if the sizes of quads are uneven, that will make the next section, where you use the relax tool, more impressive.

3. Select the edge loop around the outside of the rectangle ensuring that one of the corner vertices is "selected" in the Blender sense, showing yellow rather than orange. In Figure 7-15 stage 4 you can see that the top left corner of the rectangle is the selected one. You might have to deselect a corner vertex then reselect it.

4. Select **Face ▸ Grid Fill** and the area should be tiled with quads correctly. If the result is arranged diagonally or has triangles in places check that the edge numbers are correct and that a corner vertex is showing yellow. It is also possible to tweak the arrangement in the Adjust Last Operation pane.

Sculpt Slide Relax

Sometimes you will end up with an area of quads that you can see are imperfectly laid out, for instance in Figure 7-16 stage 1 there are elongated rectangles and quads that are almost triangular. Rather than move each vertex individually, you can use a relax brush to quickly



tidy up the geometry.

Figure 7-16 Using Sculpt Slide Relax

5. Switch to Sculpt mode, select the Slide Relax brush and adjust the size and strength so it doesn't accidentally move edges that you have carefully placed on ridges or groves.

6. To relax the edge orientation, press SHIFT and hold LMB as you paint with the brush as shown in Figure 7-16 stage 2. You need to paint carefully and not go too near the ear or eye.

7. You can also move edges towards areas where you want a higher concentration using the same brush without SHIFT.

Shrinkwrapping

Both grid filling and edge bridging, as I mentioned earlier, do not snap the vertices they create to the underlying mesh. If you subsequently relax the mesh in Sculpt mode, rather than moving each vertex individually, the vertices still won't be snapped to the mesh. However shrinkwrapping gives you a quick way to fix this.

8. Switch to Object mode, add a shrinkwrap modifier and, if you have any other modifiers such as mirror or subdivision already there, drag shrinkwrap to the top of the stack.

9. Set the Target of the shrinkwrap modifier to the Person01 object and check that the vertices are being fitted nicely to the target mesh. If there are problems try changing Wrap Method but leave Snap Mode as On Surface. If everything looks OK then apply the modifier.

Fine Adjustments with the Subdivision Modifier

Towards the end of your mesh building exercise, and certainly before you apply the mirror modifier, it's a good idea to add a subdivision modifier to see where you need to add extra topology to help define ridges and creases. The eyebrows, eyelids, ears and lips will probably all need extra loop cuts adding. When doing this final tuning it will be much less frustrating if you turn off auto merging and, for some fine adjustments, snapping.

A final tool that will help a lot as you make adjustments is **Edge ▸ Slide**. Pressing G twice works as a convenient shortcut for this, and as I mentioned earlier in the chapter, this is also the way to move vertices without changing the overall shape of the mesh when you have toggle snapping off or don't have a model to snap to.

Join Two Meshes

To wrap up this exercise you should stich the new head onto the old body and re-apply the armature modifier.

10. Make sure you have saved your blend file.

11. Switch back to object mode, select the Person01 object and select all the vertices apart from the eye meshes by selecting a single vertex then hitting **Select ▸ Select Linked ▸ Linked** then deselect everything below the neck with CTRL and box select.

12. Delete the old head!

13. Back in Object mode select first the new_head object then the Person01 object and **Object ▸ Join**

14. The new_head and old body are all one mesh now but there is a gap between them that needs to be filled. Switch back again to Edit mode and join up the vertices from the top and bottom of the neck. You will have an even number on both sides of the gap so it will be possible to do it entirely with quads. Figure 7-17 shows my attempt.

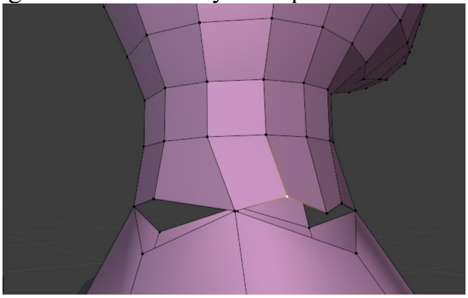


Figure 7-17 Stitching the head onto the body

15. You can merge vertices by selecting two then selecting **Mesh ▸ Merge** and choosing **First** or **Last**, so be consistent keeping either the head or body location and moving the other to match. Change the shape of the new quads to make them more regular by moving vertices using G G.

16. Finally, to recreate the armature modifier and bone weights for the new vertices, in Object mode select the Person01 object first then the armature and select **Object ▸ Parent ▸ With Automatic Weights**.

Shape Keys, an Alternative to Bones

Shape Keys are a way for Blender to store multiple different positions for the vertices of an object. They then allow the user to slide a controller to smoothly transition the vertices from one position to another. In some ways they do overlap with the functionality of an armature but they can be used where vertices need to move in different directions which would be hard to do with an armature, requiring complicated linkages of bones. Shape keys are frequently used to add expressions to the faces of characters such as the Person01 object and you will now make a very simple “smile” shape key.

1. To add a shape key, select the Person01 object then in the Object Data Properties tab under Shape Keys click +. However there always has to be Basis key and this will be the name of the newly created shape key. Click + again then change the name from “Key 1” to “smile”.

2. In the list select the smile shape key then switch to Edit mode. Now, when you move the vertices they will represent the shape of the mesh when the smile shape key is set to 1.0. This means that at a later stage, in order to see, or edit, the basis mesh you must re-select Basis from the list of shape keys. For the moment leave it on smile.

3. You might want to make a crooked smile but it will probably be easiest to set the X symmetry on. Also switch on proportional editing and change the diameter of the influence to something reasonable, say, about the diameter of an eye. In the proportional editing drop-down options you should tick the Connected Only check-box, otherwise you will distort the eyes as you move the eyelids to form the crowsfeet creases when smiling.

4. Now tweak the edges of the mouth, indent the bottoms of the cheeks and crinkle the eyes. When you’re happy switch back to Object mode.

5. Try the slider on the smile shape key as shown in Figure 7-18. You can add other shape keys, and when you edit the mesh with each shape key selected you will create the positions of vertices equivalent to that specific shape key being set to 1.0. However in Object mode you can position several shape keys to different values and the results will all be combined.

6. As an experiment try setting the value of Range Min to **-2.0** and Max to **3.0**. You will find that you can “overclock” the sliders a little, but too far will mangle the mesh!

7. Don’t forget to switch proportional editing and X symmetry off when you have finished editing. Along with forgetting to switch off snapping these are often the cause of Blender “stopping working” and they are inconspicuous settings that will take a long time to spot!

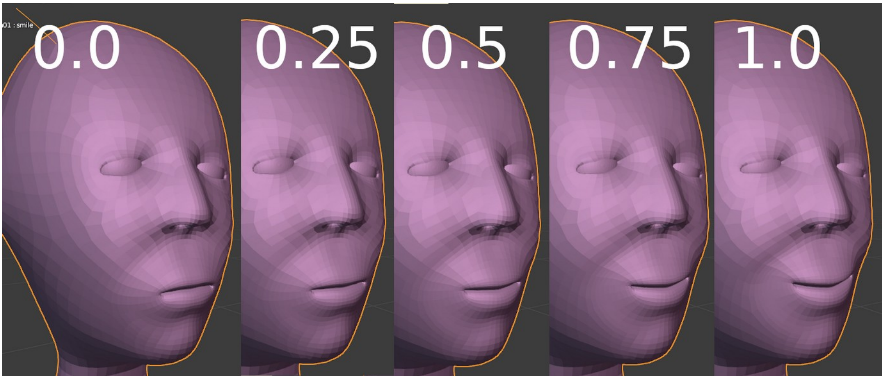


Figure 7-18 The Smile Shape Key

Conclusion

This chapter has packed in a lot of material. I hope you found the process of re-topology to be an enjoyable challenge rather than a frustrating chore, but either way you need to understand the problems, and how to solve them.

If you ever find yourself needing to do much re-topology work then you should invest in one of the various excellent add-ons available, especially as some of them are free! However, knowing how to achieve the results manually, will enable you to judge which tool to use when and allow you to appraise how good they are at doing what they claim.

You used many mesh editing tools, some new ones and some familiar ones being used in a new way or with some additional subtlety. Many of the techniques you used here to make a “soft” mesh that could be deformed cleanly, are actually identical to those needed to make good “hard surface” meshes. Dividing areas into quads, and placing edges in strategic positions is a requirement running through the whole of 3D modeling. Your appreciation of the problems involved, and their solutions has moved incrementally forward and it will help you in many ways over the rest of this book and thereafter.

At the end of the chapter you glimpsed the potential of Shape Keys. They can be used in a multitude of ways beyond making facial expressions. In many ways they belong in Blender's arsenal of mechanisms for controlling animations, so in Chapter 12 you will use Shape Keys again and see how they complement constraints and drivers.

In the next chapter you will unwrap UV maps in a much more controllable way than you used in Chapter 5. And in the following chapter you will learn how to transfer the detail that you sculpted on your original head to a normal map texture for including in the material definition.

8

FLAT PACK FURNITURE

Gaining more control when unwrapping a 3D object to form a flat surface

Blender does a good job of automating the use of UV maps and you can get quite a long way without having to worry about them at all. At some point, though, you will need to apply more control than is available with the default shapes and the smart project option.

In Chapter 5, when you started texture painting, you used smart UV project to create the UV map for the subsequent painting. As the chapter progressed I pointed out problems caused by this imperfect method of unwrapping. In this chapter we will revisit the same model but use the mesh that you created in Chapter 7 and apply some more sophistication. You will see that it is not only possible to make the scale of the UV mapping correspond with the level of detail you want to paint, but it is possible to use different UV maps for different parts of the model and use them to apply different resolution textures to different materials.

Different Ways of Unwrapping

For making a realistic model of human character it would be quite normal to divide the mesh into separate materials for the head, eyes, torso, arms and legs. However your Person01 object is far from realistic so it will be sufficient to just use three materials, head, eyes then everything else. Links to the files and videos for this exercise are at *get-into-blender.com 8. FLAT PACK FURNITURE*.

1. Open the blend file you saved at the end of Chapter 7 with your completed, re-skinned, Person01 object. Save it now with a different name in the folder for this chapter so you don't accidentally overwrite it.

2. Select the Material Properties tab. You might still have two materials left from Chapter 5, if so rename them and add a third one. You want a head, eyes and body material.

3. For this exercise you will need materials set up as in Figure 8-1 for the head and body, and one without the subsurface texture, bump texture and normal input for the eyes. You will probably have a similar setup from before called skin and eyes so modify the skin one and save it as head then use the New Material button, with the overlapping squares icon to the right of the Name field, to duplicate that and rename the duplicate as body and use eyes as it is.

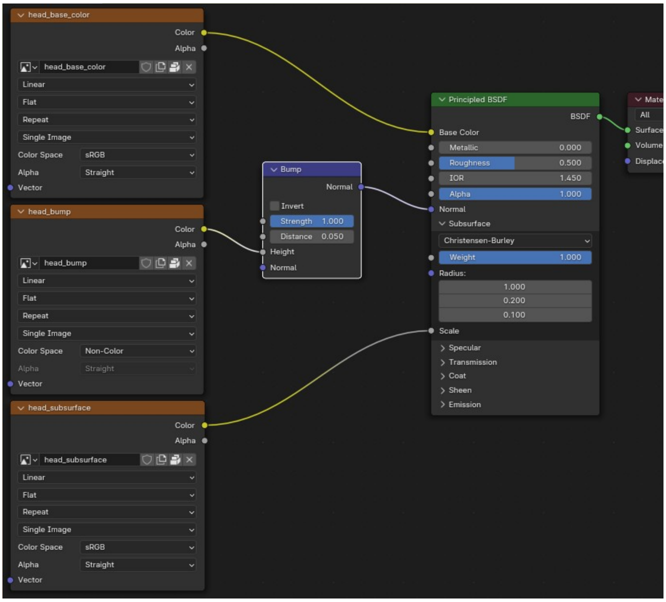


Figure 8-1 Material Nodes for head

4. There are probably existing images from the old file which you can rename. Add some new ones so, in total, you have; head_base_color, head_subsurface, head_bump, body_base_color, body_subsurface, body_bump and eyes_base_color. To get the detail for a realistic face you should make the textures at least 2048 by 2048 pixels, they won't need the Alpha option. You can create images in the node in the Shader Editor panel, using the menu options to the right of the image name. You can increase the resolution of existing images if you reuse some by opening an Image Editor in a work area and using **Image** ▸ **Resize**.

Once the materials are set up ready you need to create the UV maps for the textures to use. The first job is to draw seams where you want the skin to be "peeled" back. The standard locations are round the neck, up the back of the head, down the back of the body, down the back of each arm and down the inside of each leg. The eyes can be sliced around their circumference. See Figure 8-2 for the seams and unwrapping on the head.

5. Switch to Edit mode and select the edges that will be seams. You can do them one seam at a time rather than trying to select them all in one go.

6. Select **Edge** ▸ **Mark Seam** and the selected edges will be marked red.

7. Select the head vertices down to the neck seam, but excluding the eyes. Use **Select** ▸ **Select Linked** ▸ **Linked** then deselect with CTRL box select from the neck seam down, with X-Ray on.

8. Change the bottom screen area to UV Editor then in the 3D Viewport select **UV** ▸ **Unwrap**. You should get something like Figure 8-2 stage 1

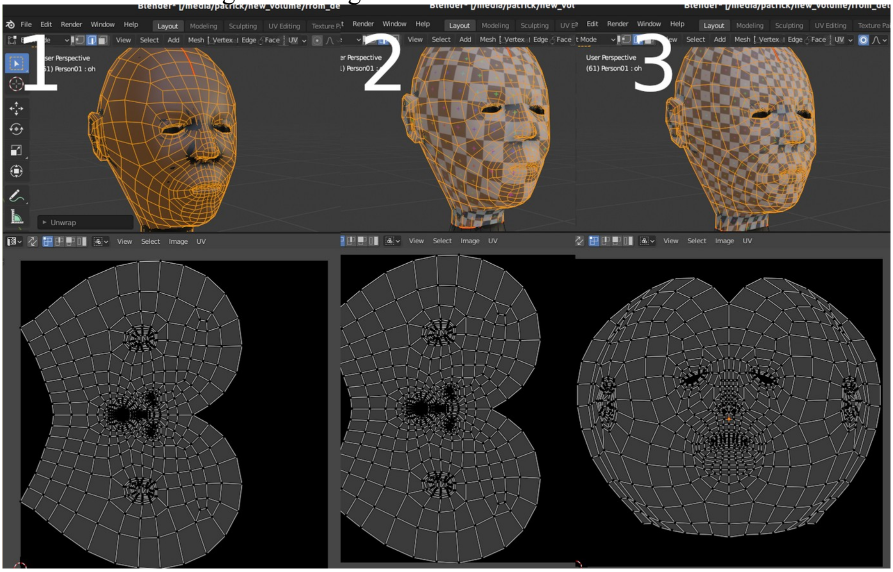


Figure 8-2 Unwrapped Head

It is now possible to edit the UV map in a similar way to editing meshes in the 3D Viewport. However before you do that it’s a good idea to temporarily swap the head_base_color image for a UV grid. Then, if you set the viewport shading to Material Preview, the UV grid will cover the head and you will be able to tell where the UV map is becoming too badly distorted. Your objective is to scale up the central part of the UV mesh so it can contain more detail, at the expense of scaling down the bits of the mesh that are under the hair or behind the head.

9. Change the bottom area back to Shader Editor and make sure you have the head material selected. In the node providing the texture input head_base_color click the button to create a new image then select from the Generated Type drop-down **UV**

Grid and click **OK**. Your head should now look like Figure 8-2 stage 2.

10. Change back to the UV Editor. Select all the vertices by pressing A while the mouse is over the UV Editor, then rotate them by -90 as in the lower half of Figure 8-2 stage 3.

11. You need to keep the squares as regular as possible but make them smaller in the middle of the face. Toggle proportional editing on with linear falloff, then select a vertex on the tip of the nose in the UV Editor. Scale the vertices up, adjusting the radius of the proportional effect so no overlaps or serious distortion occurs. You can see my attempt in Figure 8-2 stage 3.

12. Repeat the unwrapping process for the body and the eyes as shown in Figure 8-3. There is no need to swap the texture to use the UV grid image for these two.

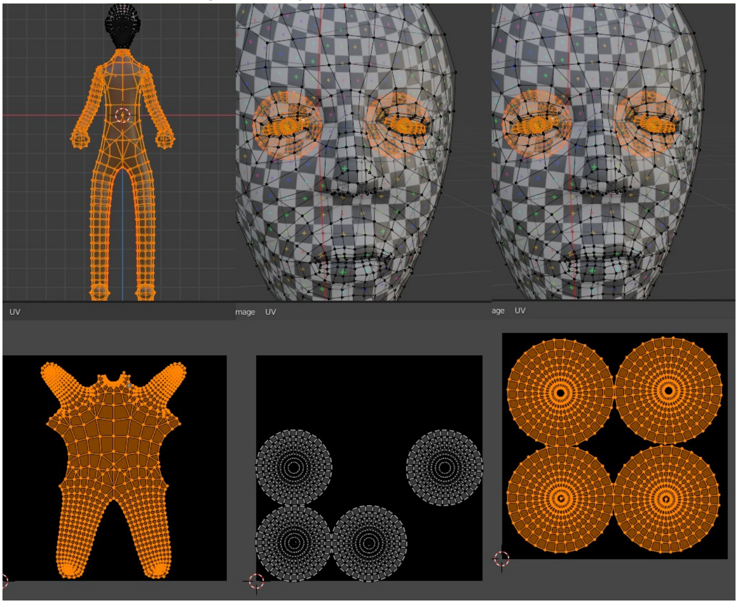


Figure 8-3 Unwrapped body and eyes

13. Change back to the Shader Editor in the bottom area and in the drop-down for the base color texture for the head, select the head_base_color again. Ensure that this node is selected and is outlined with a white line.

You should be set up correctly now for texture painting. In the past I have saved the blend file at this point and, when I opened it later, all the image files I had created vanished. So I always use the Save All Images button in the texture paint settings before I save, just to be on the safe side.

14. In the 3D Viewport switch to Texture Paint mode and in the Active Tool and Workspace setting tab set the brush, color and other settings as you did in Chapter 5. You should see the image texture being over-painted in the Image Editor panel on the left.

You should be able to get much better detail now, and the seams are hidden away, although you will need care to match the different textures across boundaries.

Don't spend too much time painting, as I mentioned in Chapter 5, unless you are an exceptional artist it will be hard to make realistic 3D objects by texture painting. So, for the rest of this chapter you will learn how to use photographs as textures.

Taking Photos for Textures

There are several problems when taking photos to use as textures in Blender materials. The image should ideally contain accurate information about the base color of the object that you are trying to represent. However photos always capture the light being reflected from the surface of the subject, which depends very strongly on the lighting conditions at the time. If you take a photo of something outdoors in strong sunlight, at twilight or under strong artificial lighting, the results will all be very different. In addition, there are almost always shadows caused by the direction of the incident light which will duplicate the shadows calculated by Blender when it does its rendering. Most cameras will also modify the light being recorded to match the sensor, often moving the profile of colors up or down the spectrum to maintain a "white balance". And finally, photos are necessarily a perspective view of just one side of whatever object is being recorded, so even though Blender can unwrap a 3D object to 2D, it's not possible to take that "all around" photo with any normal kind of camera.

Solving all these problems well takes quite a lot of expensive equipment, but it's possible to do a surprisingly good job using just a phone, some reasonable lights and the clever functionality built into Blender. Often flash can help to help fill in shadows, and if you have access to some basic photo-editing software such as GIMP or Photoshop you will be able to improve your results even more.

Applying Images to an Existing Mesh

You will need some photos of a willing subject to use for the head_base_color texture. Take a profile and a frontal image as a minimum but ideally use other pictures as well. You can see the set of mugshots I used in Figure 8-4. Give them distinct, meaningful names such as “full_face”, “face_slight_left”, “right_side” and so on.

Figure 8-4 Mugshots. That’s me looking very serious.

Cloning from Perspective Images

When you used the texture painting in Chapter 5, you may have noticed other tools apart from Draw. Blur, Smear, Clone and Fill all work in a similar way to other painting or image editing software. However there is a useful variation for cloning which allows you to paint onto a texture using, not only a different texture, but a different UV map. This means that it is possible to paint from several different perspective views of a real head onto the head_base_color texture using the clone tool.

1. You now need to set up each of the mugshots with a perspective UV that you can use to clone from while painting. Change the workspace layout to UV Editing using the tab at the top of the window, then from the menu in the UV Editor area choose **Image ‣ Open** and navigate to the first of your images, *full_face.jpg*, for instance.

2. Create a new UV map for this image by opening the UV Maps section of the Object Data Properties tab and selecting **+ New**. Rename the UV map in line with the image, UVfull_face, say. If you look at the top of the UV Editor area you can see that the currently selected UV map is displayed in the header and you can change the current selection here as well as in the Object Data Properties tab.

3. You can now make a UV map to match the view of the head in the image you just loaded. To do this you must adjust the camera position in the 3D Viewport until the model looks as

close as possible to the image showing in the UV Editor. It is much better to do this using the camera view, positioned using walk navigation, because it allows you to change the Focal Length value in the camera Object Data Properties tab. Mobile phone cameras have very short focal lengths so you probably need to reduce the default 50mm setting. Using the camera view also allows you to return to the exact same position you used to generate the unwrapping if you change the view part way through for some reason. Figure 8-9 stage 1 shows my attempt. When it's as good as you can get it, in Edit mode with select mode set to faces and all the head faces selected, run **UV ▸ Project from View**.

4. The UV map is unlikely to line up with the image in the UV Editor area, so you will need to ease it into position. To start with it will be rather a jumble as all the faces from the back are overlapping the ones from the front, as shown in Figure 8-5 stage 1. Start by scaling in the X and Y direction then use proportional editing with a linear fall off, large radius and Connected Only to nudge the UV map to line up reasonably well with the image.

5. You should aim to use only the part of each image and associated UV map, to paint the areas that are clear and undistorted on that image, so concentrate on getting those bits more accurate. Now you can gradually reduce the radius of the proportional editing and line up edges precisely, as in Figure 8-5 stage 2.

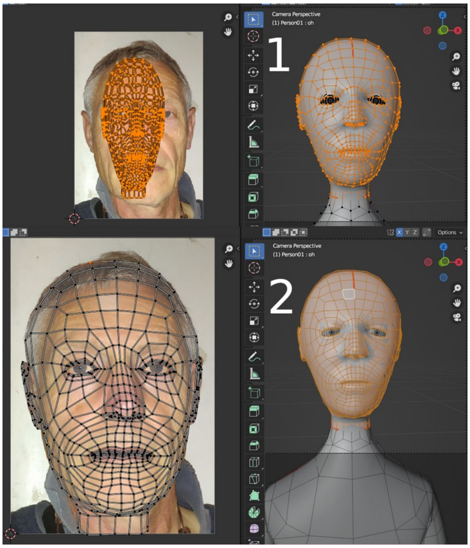


Figure 8-5 UV Projected from View

6. Work your way through each image you intend to use. A critical thing you need to make sure of before generating each new UV projection is that you have selected a new map. If you accidentally leave the UV map set to a previous one, when you hit project from view, it will overwrite all your careful work.

7. You can reuse images taken from the left, say, for painting the right by selecting **Image ‣ Flip ‣ Horizontally** in the UV Editor and using the UV projected from the opposite view.

8. To clone the perspective images onto the head_base_color texture, first specify the destination texture and UV map which you should do in the header of the Texture Paint panel, visible at the top of Figure 5-7. For Mode choose **Single Image**, specifying head_base_color and the original **UVMap**. At this point, in the Shader Editor, make sure you've changed the texture for the head from the UV grid back to head_base_color.

9. To set the source to clone from, select the Texture Painting preset from the workspace options tabs at the top. On the left of the 3D Viewport select the clone tool from the Toolbar then, in the Active Tool Properties tab, Brush Settings tick the checkbox Clone from Paint Slot. Expand the details for this and specify the Source Clone Image texture and Source Clone UV Map. These options will not be available unless you have chosen

Single Image Mode in the panel header. Each time you start cloning double check these settings are correct as well as the ones listed in the previous paragraph.

10. The first image you clone can be done using full strength for the brush and you don't need to be too careful where you paint but subsequent cloning must be done carefully. Use a low strength and build the texture with multiple brush strokes so you get smooth transitions between the different areas.

11. You will probably find some small areas that are inadequately referenced on any one of the mugshots, and leave streaks or sharp changes in color on the head_base_color texture. You can patch these areas by toggling off Clone from Paint Slot so the cloning copies from one location on the object to another, very much like cloning in a normal image editor program. To set the location to clone from use CTRL-LMB. The rather disturbing hybrid of my selfies cloned onto the Person01 head are shown below in Figure 8-6.

Figure 8-6 Full Texture Cloned from Perspective Images

Using the clone tool to create textures for UV unwrapping in Blender is powerful functionality, but it does require a bit of care. Ensuring that the same edges in the UV map line up with the same physical features on the photos is quite hard, and some trial and error is often required to get a good result. You probably found that, even with good lighting, there were shadows and differences in tone when cloning from different photos.

Images as References

While making the UV maps, it may have occurred to you that the result would have been much better if the Person01 mesh had been modified to match the shape of the head in the photos. That is definitely the case, however leave the mesh unaltered for the moment as you will use it in Chapter 9 to bake the normal map from the sculpted head. Instead, you will make a an item of simple furniture from your home using photos as references.

Don’t choose anything too complicated, your objective is to learn a few more modeling skills and some techniques for applying textures as “tiles”. I chose the sofa in Figure 8-7 because it could be constructed from relatively simple shapes, it was all made from the same fabric, it didn’t have a shiny surface which would have been hard to photograph without reflections and because it was light enough for me to be able to turn it round to face the windows.

Figure 8-7 Images of a Sofa

Reference images typically show the subject in profile and face on, and can be used in Blender to help construct a 3D model. Ideally the photographs should have as little perspective as possible, which normally requires them to be taken from a reasonable distance. However, the further the subject of the photo is from the camera, the lower the resolution, so it’s a compromise. For my sofa the two references are Figure 8-7 image 1 and 2.

For this exercise you are going to apply a photo of a small area of the sofa as a material for the whole thing and, in order to make the texture seamless when it’s tiled, the illumination should be as uniform as possible. Figure 8-7 image 3 shows my image. If you have image editing software available you can use that to make your image seamless, alternatively there are websites that offer a conversion facility for free. Failing that, it is quite possible to make textures reasonably seamless using Blender either with the compositor or texture paint cloning and baking.

Modifying the Start Cube

Save the images on your computer and give them meaningful names before you start building the furniture. I will describe the steps as if you were making a sofa but, obviously, you will need to interpret them to match the geometry you're working with.

Building the Geometry

The approach you will take is to put the front and side images into the scene as references then subdivide and extrude the start cube the minimum that will provide a very rough sofa with the right proportions and angles. You will use a mirror modifier and a subdivision modifier then add edges and slide them to get the right amount of curvature and sharpness of edges and creases. You're not going to try to model fine detail like piping or casters.

1. Start a new scene, you need to save the work on the Person01 UV unwrapping and cloning if you didn't do earlier.

2. Bring in your two reference images. First of all switch X-Ray on then in orthographic view from the Y direction use **Add ‣ Image ‣ Reference** and select the frontal image. Scale the image so the sofa is about the right size, the cube is 2m on each side so that's just right for my sofa. Move the image up so the bottom of the sofa is resting on the origin. Switch view to orthographic from the X direction and add the side image and scale it to match the sofa height in the frontal image.

3. You can control how the image is displayed, I find it easier to work if the image is behind the 3D work and only shows in orthographic view. With each image selected, in the Object Data Properties tab (red image icon) set the Depth to **Back** and un-tick the Perspective checkbox.

4. Rename the cube to sofa and add a mirror and a subdivision modifier.

5. Do the rough shaping of the cube. You need to divide it up enough to extrude the back up and the seat down. It will be easier at this stage to turn off the visibility of the subdivision modifier in Edit mode by toggling off the second button. With the cube selected switch to Edit mode and move the vertices so four are on the mirror plane and the overall size matches half the sofa up to the height of the top of the arms. Delete the face on the mirror plane. Figure 8-8 shows the size I used.

Loop Cut

6. Rather than use the subdivide function you've used previously you will add loop cuts where you need them. The whole modeling exercise will require lots of edge loops so I suggest you use the shortcuts CTRL-R to add, ALT-LMB to select and G twice to slide them. In the front orthographic view, add an edge loop and slide it to the position of the inside of the arm. Because of the perspective on my image, this is a little hard to estimate but you can see the overall width of the arm. The vertical side of the back is about in the middle of the arm, so add a second edge loop mid way between each side of the arm. You can see the position on Figure 8-8 stage 1.

7. When you extrude the seat down, you will create some unwanted edges and faces so it will help if you add a horizontal edge loop to match the depth of the seat.

8. You need a final edge inserting to allow the back to be extruded, so switch to orthographic view from the side and add a loop there, see Figure 8-8 stage 2 and 3.

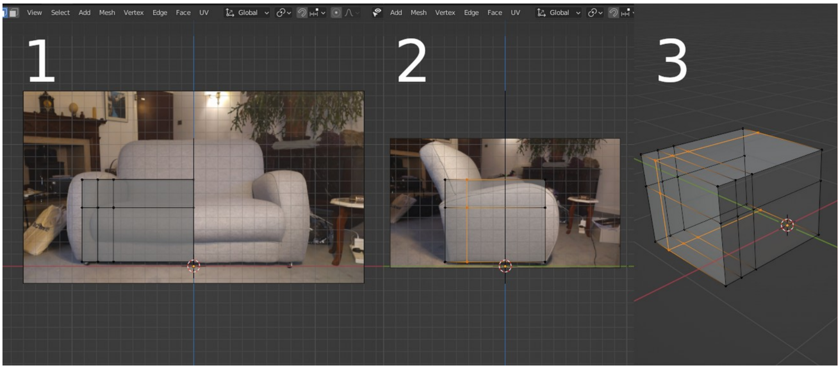


Figure 8-8 Start cube with Edge loops inserted

9. Select the two faces on the top of the backrest and extrude them upwards. Extrude the seat of the chair downwards. Now you will see the extra vertex, edges and face so delete these and fill in the gaps using the shortcut F. Use **Mesh ▸ Merge** to get rid of the gap or overlap at the front of the seat. Figure 8-9 shows the sequence and final result. Extruding the back will also insert an unseen, unwanted face and edge, temporarily disable the mirror modifier to see these and delete them. In orthographic view from the side line up the backrest to match the shape in the photo.

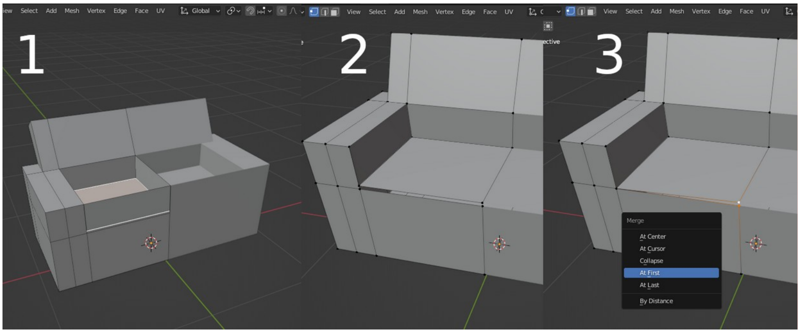


Figure 8-9 Basic extrusion and tidying up topology.

10. Your challenge now is to add the minimum number of edges to produce a likeness of the sofa. When you toggle the subdivision visibility in Edit mode back on, everything will be rather rounded, but judicious placement of edges will create the required balance of curvature. Figure 8-10 stages 1, 2 and 3 show the modeling after edge loops had been inserted. You can see that you need to put three edges close together to form creases around the seat (that place where all the coins and ball-point pens disappear). When the overall shape is correct, delete the extra edges on flat areas where they are not needed, using dissolve edges, and finally sort out the few faces that are no longer quads, using the knife tool. In Figure 8-10 stage 4 you can see on the front of the arm where I had to divide up a six sided polygon.

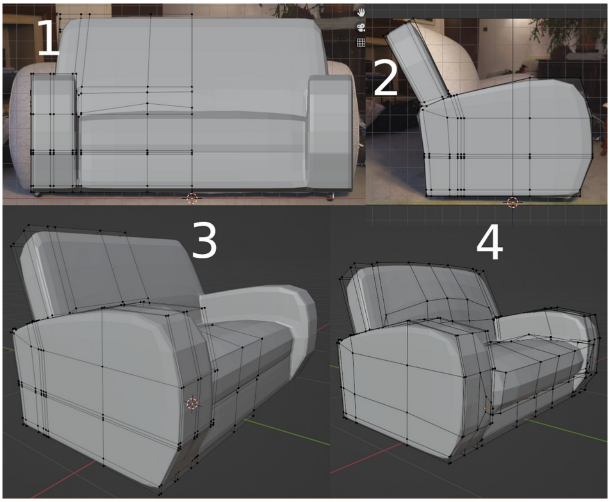


Figure 8-10 The Sofa after adding edge loops.

Define Seams and Unwrap UV Map

11. To generate a UV without too much distortion you need to apply the subdivision modifier with one level.

12. Define the seams for the UV unwrapping. Most of these can be tucked into creases or on edges at the back or bottom, however you will need one or two in visible places on the arms to prevent distortion of the UV mesh when it is unwrapped. You will be able to get a better layout if your seams cut the surface into islands so you can move and scale each one separately. The seams in Figure 8-11 produce islands for each arm, the seat, the back and the bottom. As the bottom will not normally be visible, it can be shrunk, after unwrapping, to leave more room on the image for the rest of the sofa.

13. Unwrap the UV mesh and adjust it to make the most use of the area. Use the tab right at the top of the window to change to the UV Editing preset for the active workspace, then in the 3D Viewport area hit **UV ‣ Unwrap**. You should get a result similar to Figure 8-11 stage 1.

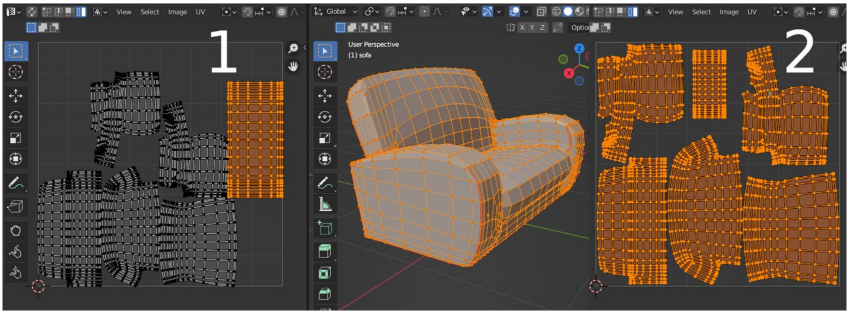


Figure 8-11 Unwrapping the main UV map

14. To scale an island such as the bottom of the sofa and to be able to move all the islands around to optimize the space, there is a useful fourth selection mode, *island*, which has an icon with a solid and outline rectangle, visible at the top of Figure 8-8. To select the island for the bottom of the sofa, in the 3D Viewport deselect everything then select one face on the bottom. In the UV Editor switch to island selection mode and select the single face that you can see there. Now, back in the 3D Viewport, select all the faces so you can see them in the UV Editor. Click again on the single face from the bottom, which is still selected, and you will select the whole island.

In Figure 8-8 stage 2 you can see my re-arranged UV layout and there are a few things worth noting. Not only have I have shrunk the bottom, I have also shrunk the back slightly, as that will also be less carefully scrutinized than the rest of the sofa. More importantly, the layout for the sofa arms look to point in opposite directions, which might be a problem if the fabric was a print, say, where the design needed to be the right way up. However if you look carefully you will see that the UV unwrapping has oriented all the islands so that the bottom of each area of fabric is on the right of the map. If you use an image for the material texture covered in arrows pointing from right to left, on the 3D object, they would go upwards in most locations, see Figure 8-12.

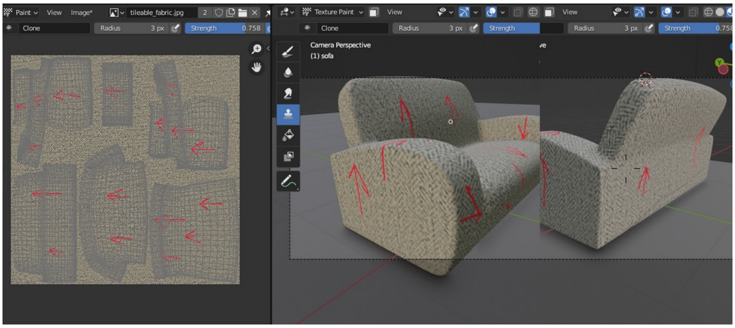


Figure 8-12 Orientation of Panels

The fabric on my sofa has no obvious directionality so orienting the panels is not an issue, however, as you can see in Figure 8-12, the scale is wrong. The square of fabric should be about the same area as the outside of one of the arms of the sofa.

Scaling UV Maps

There are two ways to achieve this. The most obvious, is to scale the UV map up until the image is the right size to match. This what has been done in Figure 8-13.

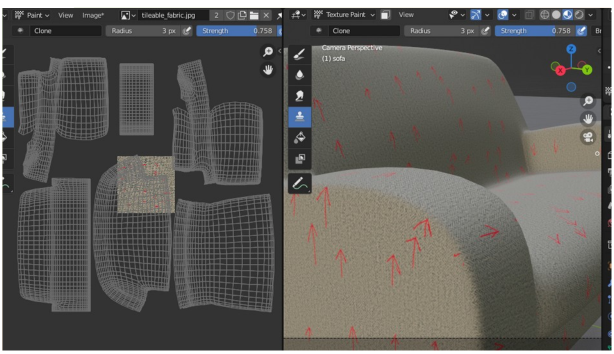


Figure 8-13 Scale the UV up to Reduce the Scale of the Image

This works fine so long as the UV map is only being used for a single purpose. But in this exercise it would be nice to also feed a normal map into the shader so that surface details, such as the piping along the edges, or minor creases, can be represented. The UV map that the normal texture uses must match the area of a single image otherwise the creases will be repeated all over the sofa in arbitrary locations. It would be possible to create a second UV map and feed that into the input of the texture node, but actually it's even easier to just use the default map but scale it up in the shader when it is being used by the base color texture. Figure 8-14 shows the node layout on the left and using the brush Stroke set to Curve for drawing piping on the normal image on the right. I mentioned using curve strokes in Chapter 5 when I introduced the basic workflow for sculpting.

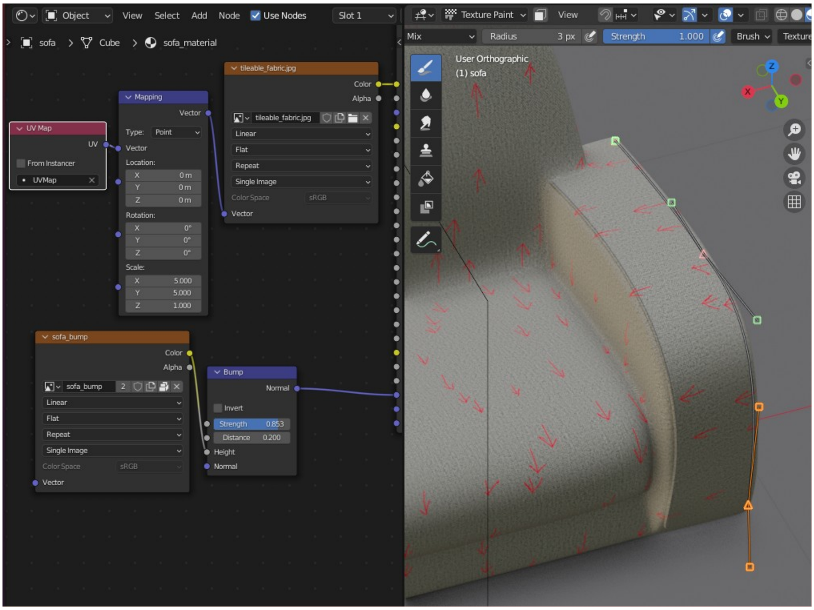


Figure 8-14 Scaling UV Maps for Different Textures in Shader Editor

15. First of all set up the material. Open the Material Properties tab and there should already be the default material that came with the cube, rename it to sofa_material. Change the area on the left from UV Editor to Shader Editor. Add a new image texture node to feed into the Base Color input of the shader then click on the Open button to import the seamless fabric image. Add a subdivision modifier back to the sofa object to smooth out some of the angles.

16. In order to see the result of changes to the material, in the 3D Viewport switch the viewport shading to Material Preview. In the Shader Editor click **Add ‣ Input ‣ UV Map**, it will default to the correct one, but just to make it explicit, click on the little dot with a drop-down selector and choose **UVMap**. Connect the output to the input of the image texture node. This is how you could set different UV maps to be used by different textures in a material.

17. Now you need to multiply the values being fed into the texture node. Select **Add ▸ Vector ▸ Mapping** and drop the node onto the connection between the UV node and the texture node, then change the X and Y scales to **5.0**. This node is also where you could rotate the texture or offset it, if needed.

NOTE For simplicity I have not suggested switching the Type field from the default Point, and normally this is fine. However, especially when you want to rotate the UV map about a specific position, you will normally find it easier to use the Texture option. If you do that you will need to change your frame of reference for the translations. i.e. scale of 5.0 with Point mapping becomes 0.2 with Texture mapping.

Now you can add some surface displacement to the sofa by painting a texture to represent the piping along the seams, edges or cushions, or other details to add realism.

Drawing onto a Bump Texture

18. To draw details onto a bump map you need to a new image textures, and click **+ NEW** to create a new image. Call the image “sofa_bump” and increase the resolution to 2048x2048, again you don’t need the Alpha channel. Add a new bump node to go between the sofa_bump texture node and the normal input of the shader, with the texture setting the Height input. You will probably need to fine tune the Distance and Strength of this node once you’ve done some texture paining.

19. Ensure that the sofa_bump image texture node is selected then change back to texture painting. If the sofa_bump texture was not selected as the target you will draw onto the wrong image. As I mentioned above, drawing using the brush stroke set to Curve will help with smooth curves like the piping on the seams. Also, switch to orthographic view using the button under the camera view button, visible on the right of Figure 8-14. This will prevent long lines drawn on the 3D object being distorted by perspective and resulting in tapered lines on the 2D texture.

20. Don’t forget to click Save All Images as you work.

Conclusion

You will now see that there are endless possibilities. You could add areas of wear or dirt by altering the roughness or you could mix in a different texture, adding a print design with a different repeat spacing from the basic fabric with yet another sample using a higher repeat as a bump map for the weave, and so on and so on.

Being able to paint onto a 3D object by cloning from a perspective image is also amazingly powerful, especially when you can map the locations from the photo so easily to the object surface. Making quick, medium poly, models, for use in the distance of a scene, or as game assets becomes very fast indeed.

Behind both these techniques is a requirement to make a good quality UV map and that takes a bit of practice. It also helps if you have an instinct for where to make the minimum number of seams to give the least distortion. I imagine tailors would be good at it!

You also did some more traditional modeling of a mesh by adding edge cuts to control the shape of a subdivision surface. Despite the speed and efficiency of using boolean and bevel modifiers, often you need to get stuck in and change the mesh. As with many things in Blender, if you find you are doing a lot of modeling this way there are add-ons that can help you. However, having the skills to do the work manually is a prerequisite to finding the best tool for your needs.

In the next chapter you will learn two different approaches to baking normal textures from detailed meshes.

9

NORMALS, BUMPS AND DISPLACEMENT

Capturing surface detail as a Normal Map

Normal maps are used throughout the world of 3D modeling to render objects with the appearance of much higher resolution than their meshes actually hold.

At several places in this book I mentioned normals, such as when describing problems that could arise if faces had more than four sides, or when explaining the difference between smooth shading and flat shading. In this chapter I will flesh out some of the principles behind reflections and shading on surfaces in computer generated images. I will then take you through two ways to generate a normal map from a more detailed mesh and discuss some of the ways to combine these normal maps with other detail such as painted or procedural textures.

Reflecting the Light

When I first introduced the idea of meshes as 3D objects I said that they generally consisted of lists of vertex locations, along with data defining the faces and edges in terms of those vertices. In the last chapter you investigated some additional data stored in the model; the texture coordinates of each vertex, but the other significant element is the normal direction.

The normal direction is the way the surface faces at any point and is used to calculate the way light bounces from the surface. Superficially each vertex would have a normal so that every point between them could be calculated by interpolation. However it quickly becomes apparent that there must be multiple normal directions at each vertex, one for each face corner, because otherwise it isn't possible to represent a change in direction from one side of an edge to the other, which is needed for flat shading.

Normal maps extend this logic by providing a way to vary the normal direction between vertices. For low and medium poly models they add enormously to the realism, as you saw with the rust on the refrigerator material in Chapter 3. Figure 9-1 shows a low poly floor and wall without and with normal map applied.

Figure 9-1 Low poly shapes without and with normal maps.

Baking a Normal Map

The classic approach to making normal maps is to construct a very high resolution model, often using sculpting to create fine detail, then "bake" the detail to an image texture that can subsequently be applied to a low poly version of the model. This scenario sounds just like situation you have arrived at with the face you sculpted in detail in

Chapter 5, re-meshed in Chapter 7 and textured in Chapter 8, you have a ready-made example to work with. Links to the files and videos for this exercises in this chapter are at *get-into-blender.com 9.NORMALS, BUMPS AND DISPLACEMENT*

Part of the difficulty when baking in Blender is in the terminology. You may feel that you are baking *from* the detailed mesh *to* an image texture on the low poly mesh. However you are required to make the detailed mesh an *active* object by clicking on it first then make the low poly object the *selected* object by clicking on it second, you must then tick the check-box that specifies "Selected to Active". That might sound to be the wrong way, but it's what you have to do.

1. Open the blend file from the end of Chapter 8; the head painted using selfies. Make sure that the armature has all the pose transforms reset and that, in the Object Properties tab, neither armature or mesh have moved, rotated or scaled. To avoid accidentally overwriting your previous work, save the file now with a different name in the folder you will use for this chapter.

2. The baking process will look in each material of of the selected object for a selected image texture node that can be overwritten with a normal map. To set this up, change a window area to the Shader Editor and for each material in the Person01 object make sure that there is an image texture node disconnected from the shader ready to have the normal map baked to it. The head and body materials will have an image texture node already created, such as "head_bump", but you need to create one for the eyes as well otherwise the baking process will look for another texture, find the one you painted for the base color, and overwrite that with a generated normal map. Which you don't want. Each material should have just one unconnected image texture node, so check around to make sure you didn't leave one lying around when you were experimenting!

3. In each material, make sure that each of the unconnected image texture nodes for the normal maps has the Color Space drop-down set to **Non Color** and click on each node so it is selected. The selected node is indicated by a white outline.

4. Now bring in the detailed mesh you will bake from by appending the sculpted mesh from the blend file saved at the end of Chapter 5. It will arrive with all its dependencies such as materials and an armature but you don't need these. Delete the armature, which has probably been called Armature.001. There should be two materials skin and eyes.001, which will probably

also have been renamed to avoid clashes with existing materials. Delete the eyes.001 material then go through the skin material setting the base color to plain white and deleting any texture nodes. This is mainly so you can differentiate the two models as they will both be in the same location.

5. Baking can only be done using the cycles ray tracing engine so, in the Render Properties tab set the Render Engine to **Cycles** then, lower down, expand the Bake section and set the Bake Type to **Normal** and tick the Selected to Active check-box. Expand the Selected to Active section and set the Extrusion distance to **0.1**. The extrusion distance is the amount to expand the selected surface that you are projecting onto, in this case the Person01 object. The extrusion distance needs to be more than the size of any bumps on the high resolution surface.

6. Select *first* the sculpted object *then* the Person01 object, the order is crucial here, then click **Bake** in the Render Properties. So long as no error message pops up this should work, but it will take a while as images are baked for each of the materials.

7. When the baking has ended switch the Shader Editor area to Image Editor and look at each of the three image textures set to capture the normal map. They should all have produce a generally light blue result with parts tending to purple-red and parts tending to yellow-green. The results for the body and the eyes can be ignored as we haven't really set the system up for those materials. If your baking has resulted in parts of the head_bump image having significant patches of orange then that probably means that the Extrusion distance wasn't enough. This is a tedious process on a low powered computer but you just have to incrementally increase the distance until the amount of orange drops. I explain why this happens in the next section.

8. To see the results of your normal map on the material of the 3D object, switch the Viewport Shading to Material Preview. Hide the sculpted object in the Outliner Editor and change the Image Editor back to the Shader Editor then, with the head material selected, use **Add ▸ Vector ▸ Normal Map** to create a new node, and connect the the output of the head_bump image texture to the input of the normal map node, and the output of the normal map node to the normal input of the shader.

9. It might be hard to see much difference at all but you can increase the Strength of the normal map node to values greater

than 1.0 to emphasize the effect. Just to verify that it's doing what you expect, do some additional sculpting of the detailed mesh. Switch the visibility of the sculpted object back on in the Outliner Editor, select it and switch to Sculpt mode. Add some strong grooves and ridges on the sculpted face then change back to Object mode, select the Person01 object as well, disconnect the normal map node from the shader and run the bake again. When you reconnect the normal map node to the shader you should be able to see the changes to the surface. It will probably show better with Viewport Shading as Render, especially if you set up unflattering lighting striking the object tangentially.

10. Remember to save your image files, either by packing them, saving them individually or using the button in Texture Paint mode!

The sequence of operations required to bake a normal map from one object to another is, in my opinion, very non-intuitive and rather precarious. If any of the steps are missed or done in the wrong order then the baking will fail. Sometimes, even if everything is correct, the baking will fail, in which case the only solution seems to be to save the blend file, close Blender then start it up again.

Why are Normal Maps Blue?

At this point I should point out that the texture image names used in the materials, such as head_bump are misleading for this exercise. Bump maps vary from black to white to represent the amount that the surface detail varies from the flat plane of the mesh. Where you drew on details such as the piping on the sofa in the last chapter that was appropriate. However normal maps are different shades of blue because they represent something else.

The direction that the surface faces can be represented by an arrow, or vector with three components X, Y and Z where X is the amount the arrow points from side to side, Y is up and down and Z is outwards. These three components can be stored in an image file by allocating X to the red value of the pixel, Y to the green value and Z to the blue value. The range of values available for each color on a standard image file is 0 to 255, so a red value of 0 is interpreted as facing completely to the left and 255 as facing completely to the right, and similarly with the green value ranging from straight down to straight up.

The blue value is always adjusted to keep the length of the arrow the same, with 0 interpreted as straight inwards and 255 as straight outwards. However, because it only makes sense for the surface to be pointing outwards, rather than inwards, the values of blue component should always vary from 128 to 255. If there is an error in the calculation of the normal directions, such as where a bump on the high resolution mesh is sticking through the low resolution mesh onto which it is being projected, then the blue component switches to a value lower than 128 and the normal map texture has patches of orange. This is a frequent problem when baking normal map textures and the solution is usually to increase the separation between the low and high resolution meshes by increasing the Extrusion distance.

Multiresolution, Another way of Sculpting

There is an alternative method of baking a normal map which relies on a modifier that you haven't encountered yet, the *Multiresolution* modifier. This modifier has some similarities to subdivision however it allows the high resolution mesh to be sculpted and the details then baked to a normal map for use with a lower resolution mesh.

It may seem illogical at first, but a common work flow is to start with a basic shape, sculpt it to a good approximation of the final model, form a new mesh using re-topology over the top of the sculpted mesh, then finally, use a multiresolution mesh to sculpt all the fine details for baking to a normal map. Although there appears to be duplication, in that the sculpting is done twice, in fact the first sculpting concentrates on larger scale features and can be fairly "broad brush". The second sculpting carries on where the first finished, adding more subtle details, and so there isn't necessarily any duplication of work.

Lets try using multiresolution to bake a normal map for the Person01 mesh you created in Chapter 8.

1. Save the blend file you worked on earlier, in the folder for this chapter. *Don't* overwrite the version you opened at the beginning of the exercise because you now need to open that again.

2. Select the Person01 object and add a Multiresolution modifier and click **Subdivide** three times. Each time you press this button it increases the number of faces exponentially so you need to be very careful. My model starts at 3k then goes to 11k, 40k, 158k, 627k, 2,507k with each subdivision, so unless you have a reasonably powerful computer it is easy to crash Blender.

3. Notice that there are subdivision levels shown for Viewport, Sculpt and Render. This is how the modifier can be used to generate normal maps; when you bake using bake from multires the process will make a normal map to represent the difference between the surface at the level of subdivision in the Sculpt setting, compared with the level of subdivision in the viewport setting.

4. Change the 3D Viewport to Sculpt mode and add a few ridges and creases.

5. Switch the area on the left to a Shader Editor and ensure each material has a non-color image texture node, such as the head_bump texture you created earlier, and that it is the selected node. Baking from multires doesn't have a problem if the image textures are connected to the shader but I often disconnect them before baking, just in case!

6. In the multiresolution modifier turn down the subdivision levels to 0 for Viewport and Render then in the Render Properties tab set the Render Engine to **Cycles,** expand the Bake section, tick the Bake from Multires check-box and make sure the Bake Type is set to **Normals**. Click **Bake**.

7. Change to Image Editor and check that the head_bump image texture looks OK then, back in the Shader Editor connect the output of the head_bump image texture node to a normal map node and then to the normal input of the shader.

8. Usually, with a fairly coarse mesh such as Person01, a subdivision surface modifier would be added to the object. In which case it might make sense to leave the multiresolution modifier in place after baking the normal map, with the level set to 1, or 1 in the viewport and 2 in the render. The level should only be reduce to 1 when baking and, to prevent the blend file being very large, you should click Delete Higher with the viewport level set to the highest you want to keep. This will permanently lose the finer detail from the sculpting so make sure you have baked the normal map before doing this.

Combining Bump Maps with Normal Maps

So what if you want to combine a bump texture with normal texture? This can be done but the result is a slight compromise because of the range that normals operate over, usually it works OK though. As an example you could add some pores to the skin using a noise texture. A more flexible way to do this would involve painting the bump map with the desired texture as a brush, but using a procedural texture will quickly demonstrate the node setup required.

1. The first thing needed is a series of nodes that will give the right kind of surface when fed into the normal input of the shader. In the Shader Editor click **Add** ▸ **Input** ▸ **UV Map** and position the node to the left. To scale the output of the UV map select **Add** ▸ **Vector** ▸ **Mapping** then set X and Y Scale to 300.0. The mapping output feeds in turn to the texture node created using **Add** ▸ **Texture** ▸ **Noise Texture** with Normalize un-checked, then add a bump node by selecting **Add** ▸ **Vector** ▸ **Bump** with Strength set to **0.6** and feed the output into the shader normal input, replacing the input from the head_bump image texture. Adjust the settings to give the right level of detail.

2. To combine the two normals you use vector addition. Reduce the Strength of the normal map node to **0.6** as well then use **Add** ▸ **Convert** ▸ **Vector Math** to add a node to combine the two normals. However this result then needs to be normalized by another Vector Math node with the operation set to **Normalize**. The node arrangement is shown in Figure 9-2.

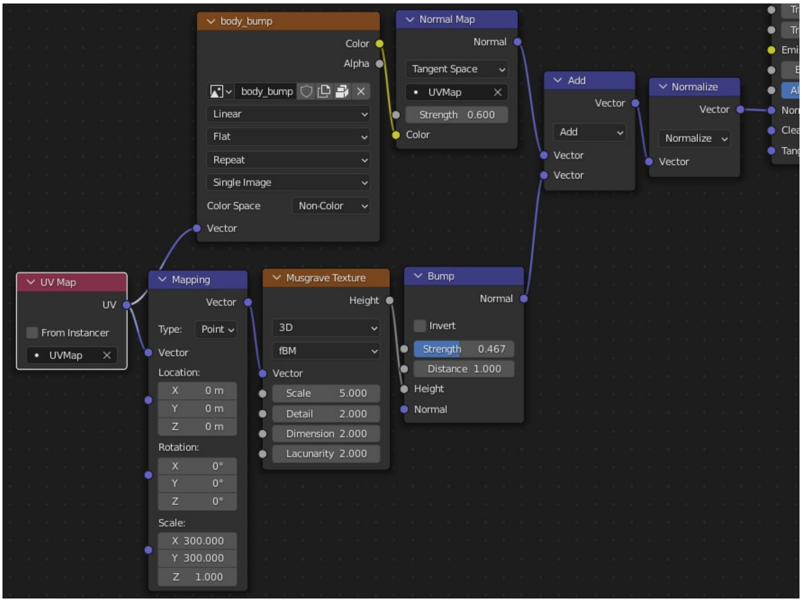


Figure 9-2 Nodes Combining Normal Map and Bump

The render on the left of Figure 9-3 shows a material with the normal disconnected, but with a multiresolution modifier still in place, set to level 1. The middle render has the normal image texture baked from sculpted multiresolution level 3. The right hand render has both the image and the noise texture pores using the shader shown in Figure 9-2.

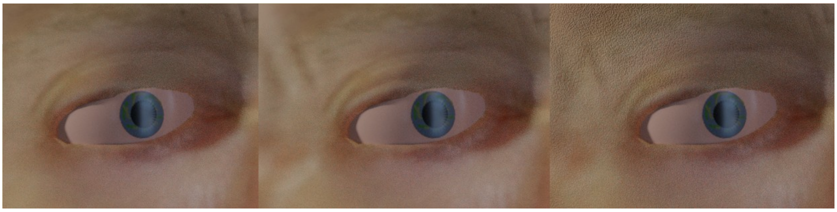


Figure 9-3 Render showing the addition of baked normal and bump map

Applying a normal map to a material shader works really well if the surface of the object is reasonably square to the camera, however, when viewed in profile it is clear that the bumps don't actually protrude. Generally speaking this doesn't cause a serious problem because the normal maps are confined to small surface details, but for gross features that need to be seen from the side, an alternative solution is required.

Displacement

An obvious solution would be to model the alternative surface as a finer mesh using a multiresolution modifier sculpted to provide the required bumps. This would work fine for many situations but where finer control is required, for instance if the strength of the effect needs to be animated, or the bumps are required to move over the surface, then a good solution is to texture paint a bump map and apply that to a displacement modifier.

In the following exercise you will draw some scales as a bump map onto the face of the Person01 object and use a displacement modifier to show them. In order for there to be enough vertices for the displacement to act on you will need to add a subdivision modifier, so this will also re-introduce the importance of the order of modifiers in the stack. In Chapter 11 you will learn about physics and particles and it will become apparent why these functions also have an entry on the modifier stack. Subdivision obviously needs to happen before displacing bumps but you could apply a surface deform modifier to follow a soft body simulation, say, after the displacement to allow the bumps to droop, or before the displacement to get a "crustier" effect.

1. Make sure you have saved your last work, again using a different name from the one you opened, then load the blend file from Chapter 8 once more.

2. First add a subdivision modifier by selecting the Person01 object then in the Modifier Properties tab adding a subdivision modifier with **2** subdivisions. You might need more but you can come back to increase this when you see the results. Too many subdivisions will slow the computer down while you work.

3. Now add a displace modifier, click on the **+ New** to create a new texture and rename it texture_bump. This texture is an internal reference that then needs to be linked to an actual type such as noise or image. Open the Texture Properties tab which will show the settings for the texture you just created and default Type will be **Image or Movie** which is what you need. Expand the Image section and set the linked image as the head_bump image texture you created in Chapter 8.

4. Back in the Modifier Properties tab adjust the settings to the desired levels. By default the modifier expects a texture where mid gray has no displacement, black gives extreme indentation and white extreme expansion. Turn the Strength down to **0.05** and the Midlevel down to **0.0** you can come back and fine tune these later.

5. To allow texture painting scales on the 3D model you will need to use the UV unwrapping you set up in Chapter 8. In the modifier properties change the Texture Coordinates to **UV** and select the main UVMap.

6. Now draw some scales onto the face. Select the Texture Paint tab from the workspace default configurations, at the top of the screen then, in the Active Tool and Workspace settings tab change Mode to **Single Image** with the head_bump as the linked image and the UVMap as the specified map.

7. When you draw on the face you might find, as with painting the particle density in Chapter 5, that the effect isn't actually visible until you switch from Texture Paint mode to Edit mode or Object mode, then back again. For this reason it's probably best to draw with viewport shading as Solid and the Color option in the drop-down as **Texture**, so you can see where you've just painted. To make the scales tilt, I used a texture for the brush with Type **Blend** and set the brush Falloff to **Constant**. To get a graduated circle with a single click you will need to set the

Brush Settings, Texture, Mapping to **View Plane** and to control the size and direction of the scales set Stroke, Stroke Method to **Anchored**.

8. Because the eyes and body are also influenced by any modifiers added to the Person01 object you will find that they also become scaly in a haphazard way. You can fix this by confining the modifier to a vertex group. With Person01 still selected switch to Edit mode, deselect everything, then in the Material Properties tab click on the head material then use the **Select** button to select just the faces on the head, but excluding the eyes. Open the Object Data Properties tab and, under the Vertex Groups section create a new group with **+ New**, change its name to "scaly_skin" and click the **Assign** button. In the displacement modifier setting set the Vertex Group optional field to scaly_skin.

You should be able to create quite large scales using a displacement modifier in this way. One additional advantage over simply sculpting the mesh, is that the texture can be used in the material to modify the appearance. In Figure 9-4 you can see that the higher the displacement value, the more reptilian the coloring.

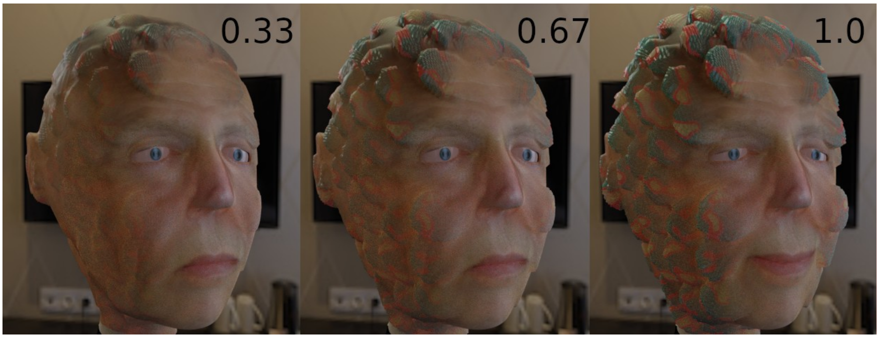


Figure 9-4 Increasing scaliness

You will remember how to use a factor to blend two shaders from Chapter 4, but wouldn't it be really cool to have one "slider" you could adjust, or animate, that would alter both the strength of the displacement modifier and the factor for blending the shaders in the material. Well this can be done using *custom properties* and *drivers*, both very powerful, but less frequently used features of Blender.

9. First add the nodes to create the reptilian effect on the tops of the scales. You might like to try figuring this out for yourself, referring back to the rusty refrigerator material to refresh yourself.

My node layout is shown in Figure 9-5, if you get stuck. There you can see that I use a noise texture with a color ramp for the scale color and and two voronoi textures for the normal. A quick search online will provide suggestions for excellent lizard skins.

The Color output from the head_bump texture is the factor for blending the shaders, but it is modified by passing through a multiply node which has its text input box colored purple. The purple indicates that the value is being changed by a driver and cannot be altered manually.

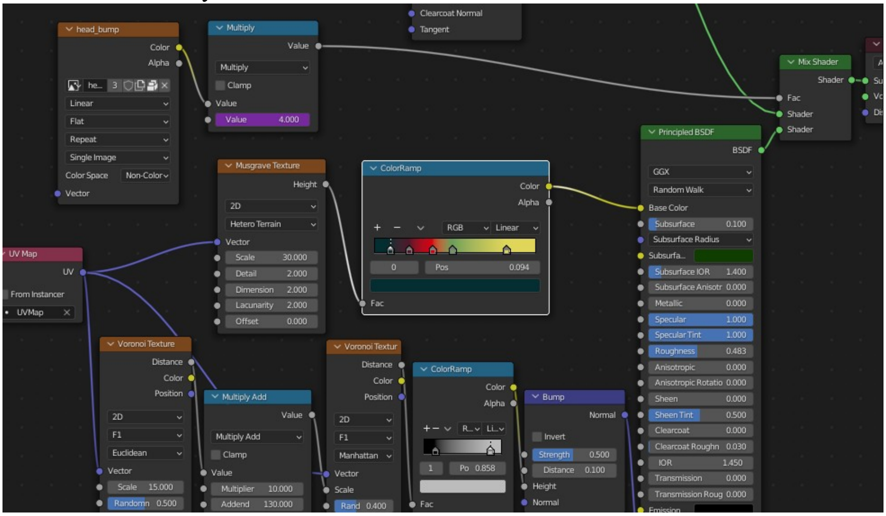


Figure 9-5 Scaliness controlling the blend of two shaders

10. Add a math multiply node and use it to multiply the output of the head_bump image texture before feeding into the Fac input of the mix shader.

11. With the Person01 object still selected, open the Object Properties tab, scroll down to the bottom and expand the Custom Properties section. Click Add and then Edit and change the name of the new custom property to “scaliness”. The other defaults are probably fine but you can see it is possible to set max, min, default and tooltip. Click OK to save the edit.

12. Right click on the value of the scaliness custom property and select Copy as New Driver, then right click on the value field of the math multiply node you just added to the material and select Paste Driver. Open the Modifier Properties tab then right click and paste the driver into the value field of the displacement modifier Strength field as well.

13. The chances are that the displacement strength and material adjustment will need different scales. I'm sure you can already see how you could adjust the value using an additional node in the material but it is also possible to perform quite complicated calculations as part of the driver. Right click on the purple Strength field and, in the menu options, select **Edit Driver** and change the Type to **Scripted Expression**. In the Expression text box below change the entry from scaliness to **scaliness * 0.05**. Change the multiplication value in the material node in a similar way to give the desired effect as you move the slider, I multiplied the scaliness by 4.0 in my example. There is also a graphical method of achieving the same effect using the Show in Drivers Editor button at the bottom of the pop-up window. There isn't room in this book to explore this in real detail.

In some circumstances drivers are extremely useful, they can link almost any value in an object to any other value in the same or any other object. In the exercise you have just completed, you could actually have made the value in the material node drive the modifier strength field directly, or visa versa, and skipped the custom property completely. However it is much better to keep clever mechanisms like this as clear as possible with all the components visible and explicit and nothing hidden away. Even when you are comfortable that you know exactly how everything works, in only a few weeks time you will struggle to remember where the value with the purple background is coming from. When that happens you must right click and edit the driver to rediscover the source.

Conclusion

In this chapter you used two different ways to bake a normal map from a high resolution mesh to a coarser one. Both techniques have their applications and which one you opt to use will depend on both the workflow you are most comfortable with, the resources available for your modeling and the final destination of the finished product. If you are making a low poly asset for use in a game engine you will probably take a different route from the one you would follow for an animation rendered in Blender.

The displacement modifier can be a very useful tool, providing a shortcut alternative to sculpting repetitive detail such as tree bark, rock surfaces or lizard skin. You also used a subdivision modifier to generate the vertices needed for the displacement modifier to act on, and this method of stacking physics and particle effects after subdivision is something you will encounter often in “how to” explanations online.

In the next chapter you will delve further into volume shaders, using them to investigate four dimensional textures, monsters of the deep and glowing writing.

Keyframes

You will learn about animation in Chapter 12 but, as there is so much to cover, the following exercise is a good opportunity to introduce one of the most important aspects of animation; *keyframes*. Keyframes are a feature of Blender that pervades virtually everything and, in an astonishingly simple way, allows you to breathe life into your static scenes. Essentially by holding the mouse cursor over any variable field and pressing I on the keyboard you can specify a value for the field to take at the current *frame*, where frame is a proxy for time. Blender is clever enough to understand that if you press I when certain bones of an armature are selected, and the cursor is over the 3D Viewport in Pose mode, that keyframes should be set for the relevant bones. You can animate object locations, camera properties, modifier fields and material settings. There is scarcely anything in Blender that cannot be animated. For this exercise you will animate the material used by the screen to cast shadows on the sea bed.

1. Select the screen plane and open a window area as a Shader Editor. You should move the view point up so you are looking down onto the surface of the screen. In order for the animation to happen at a reasonable speed you should switch the Viewport Shading to Material Preview and make sure that the screen material Blend Mode in the Settings section is set to **Alpha Clip**.

2. In order to change the location of the texture in its third dimension, to represent time, you need to feed information into its Vector input. First select **Add** ▸ **Input** ▸ **UV Map** then create another node with **Add** ▸ **Vector** ▸ **Mapping** and take the UV map output via the mapping node to the musgrage texture input. The node layout is visible on the bottom left of Figure 10-5.

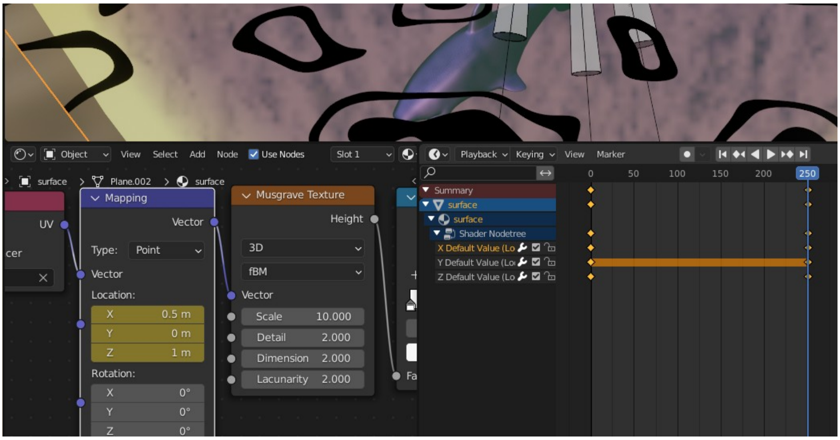


Figure 10-5 Node Layout and Timeline for Animated Shadows

3. Now you are in a position to set a keyframe, but before you do, you should check that the current time is set to frame 1. Swap the editor area to the Timeline and ensure that the vertical blue line is in the right place. At this point the only reason it might have moved is if you accidentally pressed the spacebar which will start the animation playing. You can move the play position to the start point by pressing the **Jump to Endpoint** button on the play controls.

4. Switch back to Shader Editor and, with the mouse over the mapping node Location X, Y or Z field press I. It doesn't matter which of the three location components is under the cursor as all three will be set at this keyframe. All the location values should change to a yellow color.

5. Switch to the Timeline where you should see that an orange diamond has appeared to show the location of a keyframe. If you can't see one it's probably because you haven't selected the component that is being animated, in this case the mapping node, which must be selected and showing a white outline in the Shader Editor. Move the play head to the extreme right end point, which will default to frame 250. Change back to the Shader Editor and where the Location values will still all at 0.0 but they now show green. This coloring indicates a field that is animated but at the specific frame it is being viewed, the value has been interpolated by Blender.

6. Change the Location Z to **1.0** and you will see that the field turns orange to indicate a field that has been changed from an

interpolated value but not yet fixed as a keyframe, it will be forgotten as soon as you change frame. With the mouse over the Location fields press I, and you will see them all change to yellow to indicate that they have been saved as keyframe values.

7. To see the animated values press LEFT a few times. The Location fields will be green and the Z should gradually be changing from 1.0 down to 0.0. Press spacebar to make the animation play and the water surface should move in quite a realistic way. Press spacebar again to stop it playing.

8. The first issue you might notice is that the speed of animation is much slower just after and just before the keyframes. This is because the default interpolation uses easing to avoid jerkiness and sharp discontinuities of movement. To fix this, go back to the Timeline and select both of the keyframes then, from the RMB menu select **Interpolation Mode ▸ Linear**.

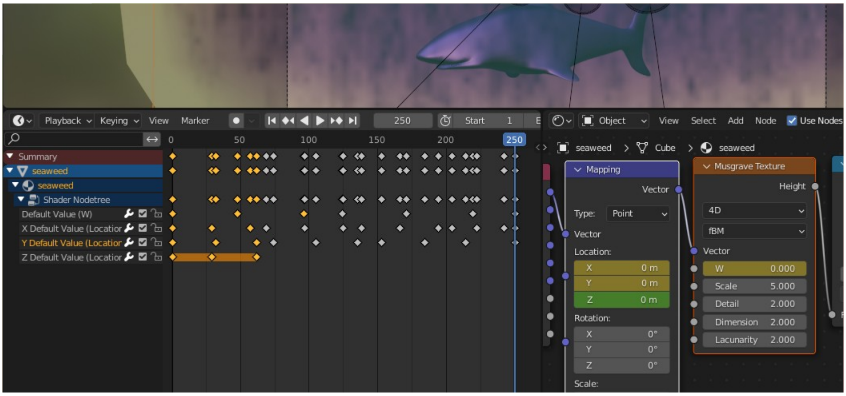
NOTE In the Timeline area the selection and manipulation of keyframes has many similarities with vertices in a mesh. Groups of keyframes can be selected using box selection or circle selection with SHIFT and CTRL having the same effects. It is possible to move or scale keyframes using G or S as well as duplicate and delete using the same shortcut keys as in Edit mode.

9. The second improvement you can make is to give a slight sideways movement to the ripples by increasing the X Location to 0.5 on frame 250. The field will turn orange after you make the change, warning you that the keyframe hasn't been updated with the new value until you press I, whereupon it will turn yellow.

10. Play the animation to verify that it does what you expect.

Using 4D Textures to Animate Volumes

Just as a 3D texture can be used to animate a 2D surface, a 4D texture can animate a volume. You will now try setting up a little bit of sway to the seaweed in front of the shark. Your objective will be to add three keyframes for time in the fourth dimension as well as three keyframes in the X and Y directions. There will be no movement in the Z, or up, directions but, hopefully the overall effect will be of gentle back and forth movement. If you make the time period of each repeat different then when you copy and paste the keyframes they should produce a more random looking effect. Figure 10-6 shows the Timeline



with one repeat of keyframes selected.

Figure 10-6 Node Setup to Animate a Volume Texture

1. First of all, on the Timeline move the play head back to frame 1

2. Select the seaweed object and switch a window area to the Shader Editor. You have already set the noise texture node to take the position vector from the geometry node via a mapping node but that only gives you three dimensions, so where does the fourth one come from in order to change the texture with time? Click on the Dimension field in the noise texture node and select 4D.

3. Now you can see that the fourth dimension is available in the texture node as a field W. With the mouse over the field add a keyframe by pressing I, then move to frame 48 on the Timeline and, back in the Shader Editor change the W value to **0.01** and fix the keyframe. The default animation speed is 24 frames per second so you are aiming to get a slow change to go with the physical swaying movement. On the Timeline move the frame to 96 then add a keyframe with W set to **0.03**.

4. In the Timeline Editor you should be able to see the three keyframes you created. If you can't, remember you need to select the noise texture node in the Shader Editor. On the left of the Timeline is a collapsible hierarchy of the components that are selected and have keyframes allocated. Expand the hierarchy so you can see which line of keyframes corresponds to each animated value, then select the three keyframes you

added to the W value and copy them with CTRL-C. Move the current frame marker to around 120 then paste with CTRL-V. Select just the keyframe on frame 1 then copy and paste it to frame 250 to make the animation loop smoothly.

5. Move the current frame back to 1 then, in the Shader Editor select the vector mapping node that is feeding into the noise texture node. Add a keyframe for the Position fields with them all at 0.0 then add two more keyframes at 30 and 60 with X and Y positions set to arbitrary values between -0.04 and 0.04.

6. On the Timeline you can drag individual keyframes backwards or forwards in time. Use this to change the positions of the X and Y key frames slightly then copy and paste the two variables separately so they are out of phase. If you look carefully at Figure 10-6 you will see how the keyframes are arranged to make the animation appear more random.

When you run the animation for this scene it will probably be rather hard to get an impression of what it would look like running at 24 frames per second. Unless you are using a powerful computer the time to calculate each image, even using the eevee render engine, is quite long. This is a problem with transparent objects such as volumes and also with particle systems or hair which are notoriously slow to render. In Chapter 12 you will learn how to render animations as a series of images and how to then convert those images into a video using the Video Sequencer. For the time being save your project so you can render the animation later.

Manipulating Procedural Textures

In Chapter 13 you will look in detail at ways to make Blender produce non-photorealistic renders, and you will use both material nodes and the compositor to apply hatching to a scene. Figure 10-7 is in the style of a 17th Century etching where a vision has been conjured as a source of light. However, for the following exercise, you will concentrate on one part of that scene; the apparition appearing above the alchemist’s table. Your objective is to create a volume in which the density varies with a random 4D texture that has been stretched radially and, in addition, the density is high in the center, falling off towards the edges. As the light shines through the volume it is scattered brightly from the pattern in the volume give the illusion of rays spreading from a central point.

Figure 10-7The Alchemist at Home in his Study

1. Start a new Blender scene but leave the default cube in place and rename it to rays. Scale it in the Y direction by a factor of 0.1 then, in the Material Properties tab, in Surface select **Remove** and in Volume select **Principled Volume**.

2. Give the volume a textured density as you did in the previous exercises by adding a noise texture node and feeding its output through a color ramp and then a multiplication node into the shader's density input. Even with a high multiplication factor for the density it will be hard to see the pattern so, in the World Properties tab change the background Color to black. Move the camera so it's looking at the rays object square, approximately along the Y axis, then change the light to a spotlight and move and rotate it so it's shining onto the back of the rays object and towards the camera.

3. In order to create a pattern of spokes you need to calculate the angle and the distance of each point in the volume relative the center. For the angle you will use the arctan2 function and for the distance you will use the length function. In the Shader Editor create a node to generate the location of each point in the volume using select **Add ▸ Input ▸ Geometry**, then create a node to split the X, Y and Z values with **Add ▸ Converter ▸ Separate XYZ**, and finally a node to work

out the angle with **Add ▸ Converter ▸ Math**. Change the type of function for the math node to **Arctan2** then connect the Position output of the geometry node to the Vector input of the separator, then the Z output to the first Value input of the arctan2 node and the X output to the second Value input. Connect the output from the arctan2 node to the Vector input of the noise texture node. Your node setup and result should be similar to Figure 10-8

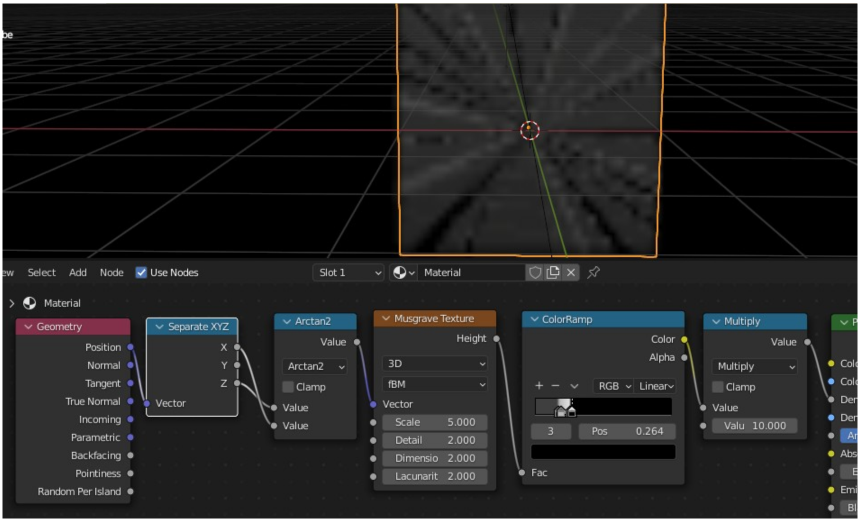


Figure 10-8 First Stage of Creating a Radial Pattern

4. You have only done the first part of the exercise, calculating the radial angle of each point, however there is a problem you should sort out now. Try moving the rays object and you will see that the center of the pattern is fixed to the global origin of the scene. This is because the geometry position is in the global frame of reference so, to get the position within the volume you must subtract the location of the center of the object. To do this use **Add ▸ Converter ▸ Vector Math**, and drop the node onto the pipe between the existing geometry position and the separate XYZ node, then change its mode to **Subtract**. Create a node with a Location output by selecting **Add ▸ Input ▸ Object Info**, and feed that into the other Vector input of the subtract node. The node arrangement should look like the left part of Figure 10-9 and now, if you move the rays object, you should find that the radial pattern moves too.

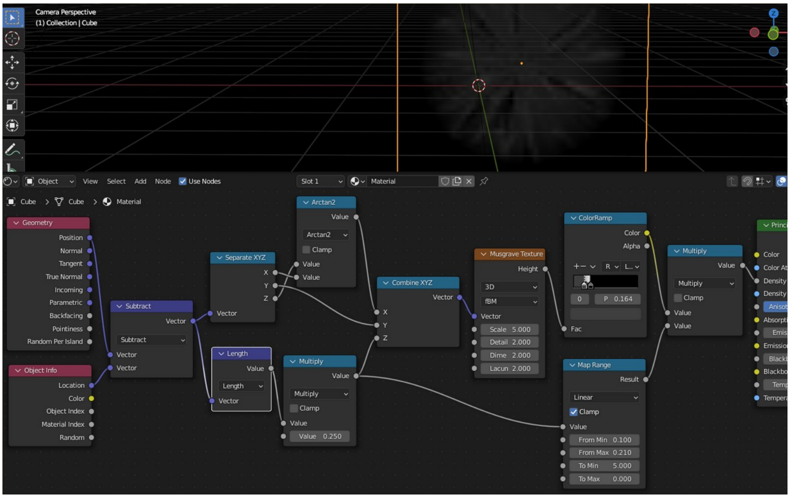


Figure 10-9 Completed Material Nodes for Radial Rays

5. To make the density vary radially, as well as circumferentially select **Add ▸ Converter ▸ Vector Math**, change its mode to Length then connect the output of the subtraction node to its input. You also need to feed additional coordinate values into the noise texture node so use **Add ▸ Converter ▸ Combine XYZ** to create a node and drop it onto the input of the noise texture. Now feed the output of the length calculation into the Z input of the combine node.

6. The pattern you obtain is probably deficient in two respects; the spokes aren't stretched enough and the density doesn't taper to the edges, leaving a sharp square outline. To solve the first issue you can insert a multiplication factor into the Z lookup value you just set up.

NOTE To improve your workflow you should get into the habit of duplicating nodes using SHIFT-D rather than creating them with the menu, especially the common ones like math, vector math and color ramp.

To make the density high in the middle, reducing to zero at the edges you could use some math nodes, or a color ramp node followed by a math node, but in Figure 10-9 you can see that I have introduced a map range node which can sometimes simplify the node tree. You may also notice the other change I made; connecting the Y location into the noise texture lookup. This has the effect of making the density vary with depth, away from the camera, and adds more complexity to the pattern.

Conclusion

This chapter used volume shaders as the focus of two ideas that have uses in other areas of Blender; First you saw how higher dimensional textures can evolve through time and allow you to add life and subtlety to many animated scenes. Next you used the modification of UV maps to generate new patterns from procedural textures, and this method also creates enormous potential for stunning procedural textures. Before you move on to the next chapter try experimenting with materials nodes using different procedural texture to scale or rotate the UV map of another. Checkout some examples on *eldwick.org.uk/get_into_blender 9.NORMALS, BUMPS AND DISPLACEMENT*.

The occasions when you need to use a volume shader with varying density might not be many, but when you do, the techniques covered here can provide unique functionality.

You have seen a little bit of animation so far in the book. In Chapter 6 when you imported the characters from mixamo.com and, a little more proactively, in this chapter when you set keyframes for the material textures. Chapter 12 takes this exploration much further, including how to rig an armature to a model and ways to constrain bones while they move. Before that, in the next chapter, you will learn about the different physics and simulation systems built into Blender.

11

EARTH, WATER, AIR, FIRE AND HAIR

Using Blender physics and simulation systems.

The software has several sophisticated methods available for producing very realistic special effects and making your animations behave naturally.

You have used particles already in this book to create the distribution of grass on the hill and boulders on Mars. When you tried the quick effects option in Chapter 1, I remarked that the options sounded exciting but might be frustrating to try out before you had learned more of Blender. Well that point has now arrived and I will guide you through the remaining quick effects; explosion, smoke and liquid. Each one has its own quirks and there are different methods required to get them to interact with other objects or to render satisfactorily.

As you work through the interactions of physics, particles and modifiers used by the quick effects, you will try applying some of the force fields to them, before moving on to simulations using cloth, soft bodies, rigid bodies and object constraints. There's a lot to cover so lets get started.

Explosion

Surprisingly, out of the remaining three quick effects, the explosion is the simplest, and produces a visible, if rather unimpressive, result straight out of the box. The menu option is essentially a short cut to add two features to the selected object, the first is a particle system and the second is an explode modifier which chops the object up. You could create your own explosion very quickly by applying the two components yourself, then reducing the number of particles being emitted, setting the start and stop time to the current frame and changing the render type from Halo to None. However the critical step is to ensure that the particle reference in the modifier stack is *above* the explode modifier. Links to the files and videos for the exercises in this chapter are at *get-into-blender.com 11.EARTH, WATER, AIR, FIRE AND HAIR*

1. Start a new Blender scene and add the explode quick effect to the default cube. In Object mode, with the cube selected run **Object** ▸ **Quick Effects** ▸ **Quick Explode**. Press spacebar to run the animation, and again to stop it. Use SHIFT-LEFT to return to the first frame, or alternatively, you can navigate using the Timeline controls in the editor area at the bottom of the screen. Note that, to see the unexploded cube, you need to move to the “zeroth” frame by pressing the left arrow again.

2. There are several issues with the explosion; there's not enough shrapnel, it's not moving fast enough, the pieces aren't spinning at all and there isn't anything bouncing off walls or the floor. The first thing to do is break the cube into smaller pieces and the best way is to add a subdivision modifier to the stack. In order for that to work you must move the subdivision above the explode modifier, set the type of subdivision to be **Simple**, and the number of subdivisions in the viewport to **2**.

3. To increase the particle speed and make them spin, open the Particle Properties tab and under Velocity, increase Normal to **10**, Tangent to **5** and Randomize to **5**. Tick the Rotation check-box then expand that section and set that Randomize to **1** and tick the Dynamic check-box. Now when you run the animation it should look more spectacular but you will find that at the start of the animation the cube is already starting to explode. In the Particle Properties tab change the Emission Frame Start and End to **5**. Also, to stop all the pieces disappearing after 50 frames increase Lifetime to **250**.

4. To create some bouncing, add another cube and scale it up so the camera and light are inside it. With the new cube still selected open the Physics Properties tab and click **Collision**. Make the collisions less elastic by increasing Stickiness, Damping and Friction. The simulation will also be more realistic if you edit the particle system on the exploding cube and in the Physics section under Forces change the Drag and Damp values to **0.02**.

NOTE The physics collision used here is a mechanism to control the interaction of objects in very specific circumstances; it applies to particle systems, cloth and soft bodies. As you will see later, rigid bodies collide with other rigid bodies and fluids need to have a fluid added to the physics properties of any objects with which they interact, with its type set to effector. An issue you might have is that the simulation won't reflect changes to the physics. This is a side effect of the efficient caching that Blender uses, and you sometimes need to trick it by changing a value or adding or removing a keyframe!

5. Once you've got the exploding cube behaving as you want, fine-tune it and experiment with some force fields. The pieces of shrapnel would look better with some thickness which you can achieve by adding a solidify modifier, adjust the Thickness to suit the size of the chunks. To create a force field add a new empty object then, in the Physics Properties tab select **Force Field**. Leave the Type as Force but change the Strength to **-15.0** to create a black hole effect. Try the other types, some will need more setting up to work, but Wind, Vortex and Magnetic should have dramatic effects, though you might have to rotate the empty or change the Strength value. As I mentioned in the last chapter it's possible to add keyframes to almost everything in

Blender so you could make the black hole start after the initial explosion. Try setting the force strength to **0.0** and insert a keyframe at frame 30 then add another keyframe at frame 50 changing the strength to **-15.0**.

Smoke

The next quick effect is smoke, and, although it might be tempting to add this to the existing explosion, everything will be easier to follow if you start a new scene. Smoke uses Blender's fluid simulation system which is very flexible but, unfortunately, requires you to set it up in a different way from the particle system used by the explosion. The main things to watch out for are; setting the physics fluid properties for objects to interact with the fluid, and baking the simulation after you've made changes.

1. Start a new scene, save your explosion so you can make a video using it when you cover how to do that in Chapter 12. With the start cube selected run **Object ▸ Quick Effects ▸ Quick Smoke**. This should produce some reasonably realistic smoke when you play the animation, even with the viewport set to the default solid shading.

A new object has appeared named "Smoke Domain" which defines the volume in which the fluid simulation will be calculated. When you run the animation the smoke might appear to be deflected from the edges of the domain as if by invisible walls and ceiling but that is an effect caused by the simulation having a sharp cut-off at the edge of the domain. The fluid domain is divided up into a number of resolution divisions which defaults to 32, so if you make the domain larger it will make the fluid simulation coarser. However if you increase the number of divisions you will slow down the simulation and use much more memory. So your objective should be to make the domain as small as you can get away with and only increase the number of divisions once you have run the simulation and can see, for instance, that the droplet size is too large.

If you are using an older version of Blender, prior to 3.3, you will run into a problem getting the simulation to re-run and bake in any changes you have made. Run the simulation right through to the end then stop it and return the play head to frame 1. With the Smoke Domain object selected open the Physics Properties tab then scroll down and expand the Cache section. You will see that the Type is set to Replay, which, in theory, will bake the cache as the animation is playing; the first time through might be slow but subsequent runs will be fast. The problem is that the fluid domain doesn't always "know" when it needs to re-bake the simulation. If you have an old version of Blender you will see what this means by selecting and scaling the domain object by a factor of 2. The cube of smoke should have grown in proportion to the domain, and when you run the animation you will simply see a scaled up version of the original animation. To force a re-bake you need to change one of various settings in the domain, so in the Cache section Type select **Modular** then re-select **Replay**. You should see the box of smoke shrink back to match the start cube, and when you play the animation the simulation will be re-calculated.

2. In order to deflect the smoke you need to add a mesh object and define it as an *Effector*. Add a new cube object to the scene, scale it up so it is the same width as the fluid domain then scale it down in the Z direction so it's about half the height of the start cube. Move the new slab object up so it's about a third of the way between the start cube and the top of the domain. In Figure 11-1 you can see the size and location I used.

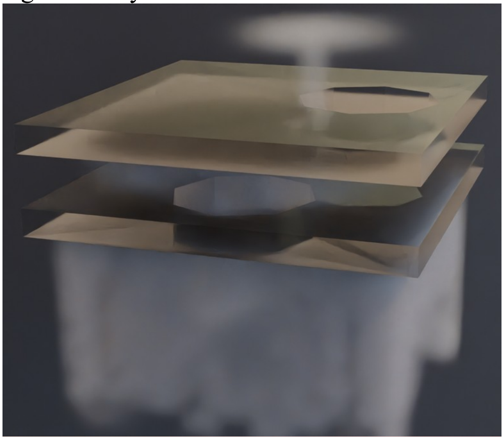


Figure 11-1 Smoke Simulation Rendered with Cycles

3. To make the slab into an obstruction, open the Physics Properties tab and click on **Fluid**, then select **Effector** as the Type. Select the domain object and, if using an old version of Blender, force a re-bake by changing the cache type from, and back to, Replay. Make sure the play head is back at frame 1 then run the animation and you should see the smoke hit the new baffle.

4. Now make a hole in the slab about a quarter of its overall width and over to one side, again, you can see the proportions I used in Figure 11-1 and Figure 11-2. The simplest way to make the

hole is to add a cylinder to the scene, reduce the Vertices to 8 in the Adjust Last Operation panel then, when it is suitably scaled and positioned, apply it as a boolean difference modifier to the slab. In order to see the smoke flowing around the baffle add a new material to it, change the shader to **Glass BSDF** and reduce the IOR to **1.01**. Duplicate the slab, move it up and rotate it 90 degrees about the Z axis. You may once again have to force a re-bake of the fluid simulation to see the effect of these double baffles.

5. We haven't yet looked at the settings on the original cube which is generating all the smoke. Select the cube and open the Physics Properties tab where you can see that the quick smoke function has enabled fluid physics and set the Type to Flow, the Flow Type to Smoke and the Flow Behavior to Inflow. Most of the settings here are self explanatory but the three Flow Behavior options; Inflow, Outflow and Geometry need a little more explanation. Inflow and Outflow are continuous processes that either flow into the simulation, as a source of fluid, or out of the scene as a drain. Geometry is a one-off inflow of fluid in the shape of the mesh so this would be suitable for an object disappearing in a puff of smoke, or to simulate the spillage of a cup of coffee. Try changing the Flow Behavior now to **Geometry** then trigger a re-bake of the simulation, if needed, and run the animation.

6. Finally, see what happens when you tick the Border Collisions check-boxes in the domain object and try adding an empty to the scene with a force field set to Wind, Vortex or Magnetic

Liquid

At this point, as an alternative to using the quick effect to make the default start cube into a liquid, it might be more instructive to try converting your existing smoke simulation.

1. First of all save the blend file so you can use it to make a video later, then delete the empty with the force field as this will be an unnecessary confusion to start with. It would be a good idea to save the blend file now with a different name so you don't accidentally overwrite the smoke version at the end of this exercise.

2. Because a liquid is going to flow downwards you need to move the start cube, with type flow, up to just below the top surface of the domain. Move the two baffles down into the lower half of the domain below the cube.

3. The start cube still needs to have Type set to Flow but the Flow Type should be changed to **Liquid**. Keep the Flow Behavior as Geometry for the moment.

4. The domain also needs to be changed from Domain Type Gas to **Liquid**. This might make it appear solid again in the viewport display, in which case open the Object Properties tab and under Viewport Display change Display As to **Wire**. Move the Timeline to frame 1, force the simulation to re-bake if required, and you should see a box-full of generally blue particles flow out over each baffle, draining through the holes and gradually flooding the bottom of the domain. The left of Figure 11-2 shows the simulation part way through.

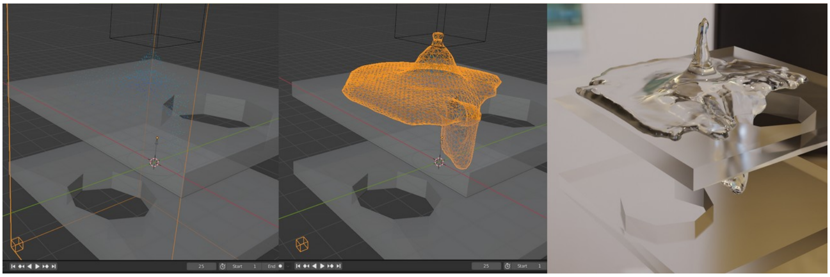


Figure 11-2 Stages to Achieve Liquid Simulation

5. To make the liquid actually look like a liquid you need to specify it as a mesh by ticking the check-box in the domain object Physics Properties tab, and you also need to go back to frame 1 and re-run the animation. The middle of Figure 11-2 shows the same frame but with the liquid as a mesh. At this point you may decide, as I did, that the droplets are too coarse, so increase the Resolution Divisions to 64. Changing this value will trigger a re-bake of the simulation so you don't need to do anything apart from return the frame pointer back to frame 1. Notice how much slower the first run through the animation is.

6. Once the animation has run and the cache has been re-baked, you need to do a few more things to make the liquid look like water. First, in the Object Properties tab for the domain change the Display As setting back to **Solid**, it only needed to be wire to allow you to see the blue particle simulation stage. Next open

the Material Properties tab and change the surface shader to **Glass BSDF**, giving it an IOR of **1.33**. When you view this as a render preview, with the cycles render engine, it should look a little bit more watery but rather lacking in reflections and somewhat turbid. That's because the quick smoke effect added a volume shader to the material, so remove that. You should also set the object shading to smooth and, to provide some more interesting reflections, change the world background color to an environment texture. The right side of Figure 11-2 shows the final render.

These two fluid simulations give you an idea of how to start setting up your own projects, and the kinds of problems you might encounter, however they barely scratch the surface of the potential available. Viscosity, surface tension, foam, spray and bubbles are possible and allow all manner of effects from molten metal to tidal waves.

In the next sections you will move onto a form of simulation mid-way between fluids and rigid body physics; cloth and soft bodies.

Coth

This basically does what it says on the tin. You make a mesh, define it to be made from cloth by adding a modifier and when you run the animation it crumples and deforms like cloth. OK, so there are a few details that you need to set up as well.

1. Save the liquid project and start a new scene, then with the start cube selected, add a cloth modifier to it. When you start the animation playing the cube will fall and disappear off the bottom of the screen.

2. To give the cloth cube something to crumple on add a plane, scale it up then move it down below the cube and add a collision modifier. Now when you animate the scene the cube drops onto the plane and stops, but it doesn't look like cloth.

3. In order for a mesh to deform like cloth it needs to be subdivided to a relatively fine level of detail. So add a subdivision modifier to the cube, select **Simple** and increase the subdivisions to **4**. The animation still won't work yet because the subdivision is happening after the cloth, but when you move the modifier up you should see a crumpling cube.

4. Now you can have some fun by adding internal pressure to the cube. Tick the check-box for the optional Pressure section in the Physics Properties tab then increase the value of Pressure to **5.0**. Now when you run the animation the cube will inflate and bounce realistically on the plane. Duplicate the cube twice and move the duplicates above and slightly to one side. If you try running this you will see that you need to add a collision modifier to two of the cubes as you did for the plane in order to make them interact with each other.

5. Cloth simulation is a very quick way to model objects for static scenes as well as for creating animations. With a little practice cushions, clothes and drapes can all be made very easily this way. Try adding a curtain to your scene in a position where at least one of the bouncing cubes will hit it. Add a plane then scale it up, rotate it 90 degrees about the X or Y axis and move it to the side of the falling cubes where a cube will roll. In order to hang the curtain along its length subdivide it in Edit mode so there are 5 vertices on each edge. Select the vertices on the top edge then, in the Object Data Properties tab, create a new vertex group, rename it pins and assign the selected vertices. Switch back to Object mode and add a subdivision modifier using the Simple method and 3 levels of subdivision. Below this modifier add a cloth modifier then open the Physics Properties tab and, in the cloth settings, expand the Shape section. Select the pins vertex group as the Pin Group and, in order to create some vertical pleats, set the Shrinking Factor to **-0.1**. This will make the cloth expand between the pinned vertices to provide some "fullness".

6. When you run the animation you should see the curtain drape and then deform realistically as it is hit by the cubes. As with the fluid simulations there are many setting to adjust and create anything from a rubber blanket to silken gauze. Mix up some materials as well to create an interesting scene as in Figure 11-3. Note that this render was done using the eevee engine with ambient occlusion, bloom, screen space reflections and motion blur all turned on. Also I added a solidify modifier to the curtain, after the cloth modifier. Sometimes the Blender physics simulation system may struggle with multiple different collision objects and not produce the expected distortions on the cloth. For the example shown in Figure 11-3 I had to make sure that a cube with a collision modifier hit the curtain and that the curtain did *not* have a collision modifier.

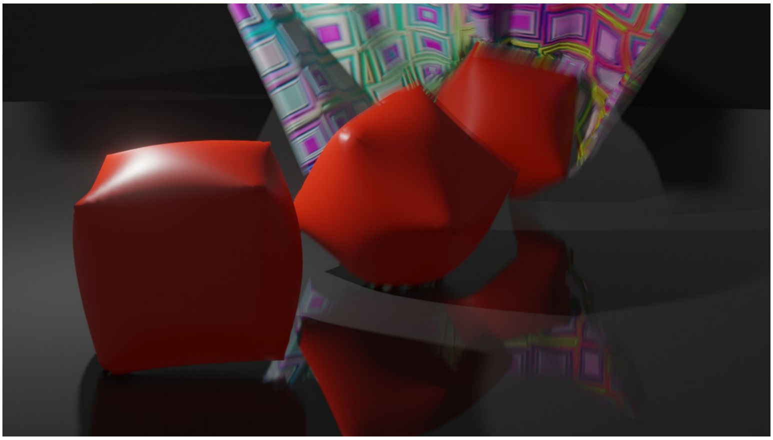


Figure 11-3 Cloth Simulation with Inflated Cubes and a Curtain

Soft Bodies

Unsurprisingly soft bodies lie part way between cloth objects and rigid bodies and one of their most useful applications is to animate parts of a character automatically. Details such as hair, jowls, bellies, loose clothing or antennae should all be animated in time with the movements of the character and this would be very time consuming to do convincingly by hand.

You will apply a method using a soft body to animate the belly of the peasant man salsa dancing FBX model you downloaded in Chapter 6.

1. Start a new scene, saving the cloth simulation for later, don't delete the start cube, this will become the soft body, then import the salsa dancing FBX file following the instructions in Chapter 6. Select the armature then open the Object Data Properties, which has changed to a green stick-man icon, and under Skeleton select **Rest Position**.

2. The peasant man will be submerged inside the cube so select the cube that will become the soft body, and scale it down so it's about the size of the belly area then move it in the Y and Z directions so one side is just inside the man's body as on the left of Figure 11-4, this is most easily done in orthographic view. Just to preempt any confusing behavior later, select **Object ‣ Apply ‣ All Transforms**.

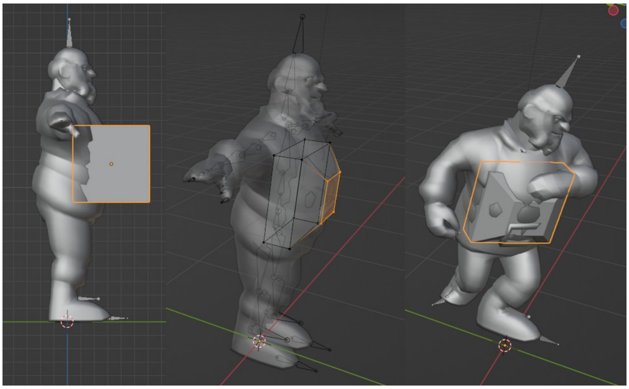


Figure 11-4 Making a Soft Body to Deform the Peasant Man’s Belly

3. In Edit mode add an edge loop then scale and move the vertices to more closely match the shape of the belly as in the middle of Figure 11-4.

4. You need to define a vertex group to use for attaching the soft body to the armature, in a very similar way to creating the pins for the cloth curtain simulation. Still in Edit mode select the four vertices nearest to the man, create a vertex group, rename it pins and assign the selected vertices.

5. To attach the the soft body to the armature you will need to parent it and add an armature modifier. The quick way of doing that, which you used before, was to use the parent with automatic weights option, however that would add a vertex group for every bone in the armature and would try to guess what weight to use for each. In this case you want to just use one bone, the Spine, so the easiest route is, still in Edit mode, select all the vertices then create a new vertex group and rename it Spine. Note that it has to match the name of bone exactly, then assign all the vertices to that group. Switch back to Object mode and SHIFT select the armature as well, the soft body should be highlighted in orange and the armature in yellow, then select **Object ▸ Parent ▸ Object**. The final step is to add an armature modifier to the soft body and specify the armature in the Object field.

6. When you change the armature Skeleton back to **Pose Position** from Rest Position you should see the soft body move with the armature as shown on the right of Figure 11-4. Of course you haven’t actually made it into a soft body yet, so select the soft

body object and add a soft body modifier. All the settings for soft bodies are in the Physics Properties tab so open that and expand the Goal section. In order to attach the pins vertex group to its start position but allow the rest of the body to deform, set the Vertex Group to **pins** and the Strengths Default to **1.0**. To prevent the soft body being too soft, increase the Strengths Min to **0.8** and set Settings Stiffness to **0.9** and Damping to **4.0**. It will also be more realistic if you increase the Object Mass to **2.5**. All of these values, plus others will probably need adjustment depending on the kind of object you are hoping to simulate. Run the simulation to see how it looks.

7. When the soft body is behaving reasonably, it doesn't need to be perfect, you should set up another modifier to transfer some of that movement onto the peasant man object. Select the character skin then enter Edit mode and, in orthographic mode with X-Ray enabled, select all the vertices in the belly area, as in Figure 11-5. Create a new vertex group called belly, but before you assign the vertices soften the edge of the selection by reducing the Weight to **0.25**, click Assign then reduce the size of the selection by pressing **Select ▸ Select More or Less ▸ Less**, or use the shortcut CTRL-MINUS, then change the Weight to **1.0** and assign the vertices.

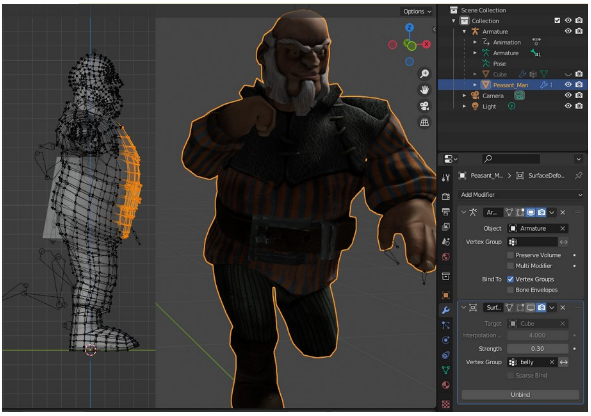


Figure 11-5 Selecting the Vertex Group for Belly Movement

8. To make the vertices move with the gyrations of the soft body you need to switch back to Object mode and add a Surface Deform modifier. Set the soft body, still named Cube, as the Target and, in the Vertex Group field select **belly**. Change the Strength to **0.3** then click **Bind**. In the outliner hide the soft body object in the viewport and the render. The right hand

image in Figure 11-5 shows the two views superimposed; with and without the modifier.

Using soft bodies to animate parts of characters can be very useful but sometimes they behave a little too like jelly and it can be hard to reduce the amount of wobbling. For things like hair, which are relatively light and floppy, it is normally better to follow almost the same procedure as that outlined above, but use a cloth modifier instead of the soft body. One additional problem for dangling features, such as pony tails, purses or sporrans, is that they need to be prevented from swinging inside the character's body. Rather than add a collision modifier to the whole of the character mesh, which will have thousands of polygons and a complicated geometry, it is much more efficient to put the modifier on a simple mesh with only a few vertices and parent that mesh to the armature as a deflector.

Rigid Bodies

In some ways rigid bodies are the simplest concept in the range of Blender physics processes. Rigid bodies collide with each other in a realistic way and can be connected together using different types of links. However the construction of the rigid body constraints is not very intuitive and takes a little practice to get the hang of. More importantly, there can be conflicts in Blender's internal machinery for applying modifiers and other physics systems at the same time as rigid body simulation. For instance, it is much harder, and less reliable, to construct a system of rigid bodies connected by damped springs to simulate the swaying belly of the peasant man than to use a soft body or cloth simulation.

To finish off this chapter you will do one more exercise modeling six spheres rolling into a bowl. This simulation will be used for an animation in Chapter 13 in which the spheres turn into bugs that uncurl and start crawling out of the saucer. There will be five bug spheres which will be joined together into a kind of train using rigid body constraints and there will be a heavier sphere that will be launched around the bowl to jostle them.

A problem you will encounter with rigid bodies, sooner or later, is their tendency to pass through each other. Apart from forgetting to define them as rigid bodies, the most frequent cause of this behavior is due to scaling meshes and not applying the transform to the vertex locations.

1. Start off by making a bowl from a sphere by inverting the top half of it. Start a new scene, delete the start cube then add a new sphere object. Scale the sphere 10 times then scale it down in the Z direction by a factor of 0.5. In Edit mode use orthographic and X-Ray view to select the vertices above half way, scale them down slightly then invert the curve by scaling them in the Z direction by a factor of -1.0. Move the selected vertices down and adjust and scale them, then by selecting edge rings at the top expand and flatten the lip so create a bowl as shown in Figure 11-6.

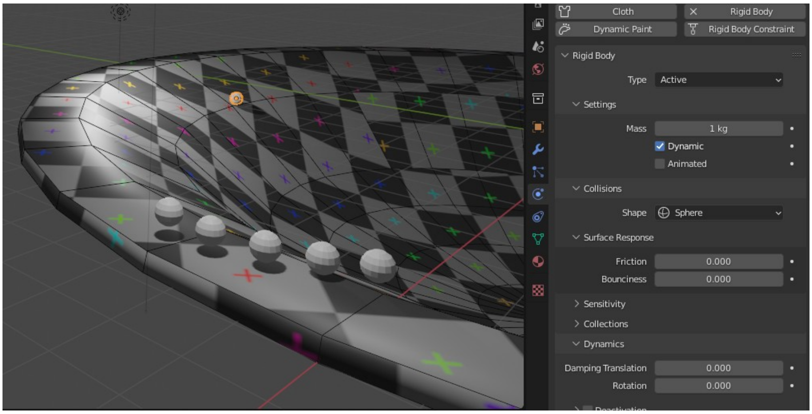


Figure 11-6 Simulating Balls and a Bowl

2. To make the bowl interact with the balls when they're added you must make it into a rigid body. With the bowl selected open the Physics Properties tab and click on **Rigid Body**. There are several things you need to correct with this setup but you will pick them off one by one, that way you can see the symptoms and, if they occur later, you will know how to fix them.

3. Add the first ball as a UV sphere and in the Adjust Last Operation panel change the Segments to **16**, Rings to **8** and Radius to **0.2**. Enable rigid body behavior for this sphere too, then move it up, higher than the rim of the bowl and to one side, so it will drop onto the curved part of the surface.

4. When you play the animation you will see that the bowl and the ball both fall under the effect of gravity. In order to stop the bowl falling you need to tick the Animated check-box in the Settings section of the rigid body. The bowl should stay where it is now but the ball will stop abruptly just above the bowl, level with the top edge. That's because the collider shape defaults to

Convex Hull, but the bowl is, by its very nature, concave. If you position the ball below the rim of the bowl it will be ejected dramatically by the simulation.

5. To make the ball bounce off the actual surface of the mesh change the Collisions Shape to **Mesh**, for the bowl, but for the ball set it to **Sphere**. Now when you run the animation the ball will very likely drop through the mesh, if it doesn't, try moving the start point around, the majority of locations should cause the ball to fall through.

6. You can probably already guess the solution to this problem because it's cropped up so many times before; you need to apply all the scaling you did on the sphere that got turned into the bowl. With the bowl selected hit **Object ▸ Apply ▸ All Transforms** and now you should see the ball interacting with the bowl correctly. If you still have problems it could be that the mesh of the ball is no longer around its center point, in which case use **Object ▸ Set Origin ▸ Origin to Center of Mass (Volume)**

7. There is one other change you should make now, before you start carefully positioning the balls to create an interesting animation. The smooth or flat shading option changes the normal directions that are used by rigid body physics, as well as for light reflection, so you should set the option to smooth shading now.

8. The ball you just made will be the heavy ball to knock into the bugs so keep its weight at 1.0, but to keep it rolling round and round the bowl you should reduce the friction. In the rigid body properties, under Surface Response reduce Friction to **0.1** and Bounciness to **0.5**, under Dynamics reduce the Damping Translation to **0.0** and Rotation to **0.0**.

9. To create the bugs, duplicate the ball and change its mass to **0.1** then duplicate the lighter ball four more times. Position these five balls on the flat edge of the bowl so they are just above the surface and separated by about one diameter. The idea is that the bugs will slowly fall into the bowl and the heavier ball will be fired horizontally so it hits them. You can see the locations in Figure 11-6. Make sure that the balls are not intersecting the surface of the bowl otherwise they will experience a strong repulsive force in the first frame of the animation!

10. To join the bug spheres into a train you need to add four constraints. In Blender each constraint is expected to be added to a new empty object and the constraint specifies the two other objects to be linked as well as all the physical properties of that link. For most rigid body constraints the location of the empty object relative to the two linked objects is critical to its behavior and behaves most reliable if placed mid way between them. Fortunately there is a menu option to automate much of this. Select the first and second bug then use **Object ▸ Rigid Body ▸ Connect**, then select the empty object that has been generated and in the settings for the rigid body constraint change the Type from Fixed to **Point**. Repeat the process until all five balls are joined together.

11. When you run the animation you should see the first ball slowly roll away from the flat edge area and drag the other four with it. You might need to adjust the balls' positions or tweak the mesh of the bowl, but if you do move a ball you should move the empty object that holds the constraint so it's approximately mid way between each of the balls. Notice that the empties do not move with the balls but the constraints are constantly re-calculated as if they had done.

12. To give the heavier ball some starting velocity you will animate its position for the first few frames and then switch back to simulation. Select the heavier ball and, in the rigid body settings tick the Animated check-box under Settings. Make sure that the animation position in the Timeline is set to frame 1 then, with the cursor over the check-box press I to insert a keyframe. With the cursor over the 3D Viewport press I again and select **Location**. Now move the position on the Timeline to frame 5 and move the ball in the 3D Viewport in the direction you want to fire it so it will hit the five bug balls. Insert a keyframe here by pressing I and again selecting **Location**. In the rigid body un-tick the Animated check-box and insert a keyframe for that by pressing I with the cursor over it. If that was all done correctly then, when you play the animation you should see the ball set off circumferentially round the bowl.

13. It is very unlikely that you will hit the bug balls first try. Often, if you do hit them, either the heavy ball or the bug balls fall out of the bowl. I found the easiest way to get the right direction and speed was to view the bowl from straight above and keep moving the first frame location of the heavy ball, remembering to reset the keyframe each time by pressing I. Because you have

a keyframe to turn animation off on frame five you will not be able to move the location of the ball at that point unless you temporarily tick the Animated check-box. To increase the initial speed increase the distance between the first frame position and the fifth frame position. I also had to move all five bug balls a small amount, along with their constraint objects, to get the timing right.

14. In the end I managed to get two collisions within 250 frames. If you want to run your animation longer then you can change the End field in the Timeline, however there is a separate location where the end of the simulation is set, and this defaults to 250 frames. To change this, open the Scene Properties tab, which has a cone and sphere icon, expand the Rigid Body World section and then the Cache section where you can set the Simulation End field to match the end of your animation.

NOTE The Scene Properties tab also contains settings that allow you to fix problems with physics in the rigid body world. Increasing the accuracy of the simulation can prevent fast moving objects "tunneling" through other objects and similar effects, but at the cost of slowing down the calculations. There are similar controls available for fluids, cloth and soft bodies in the Physics Properties tab.

You will use this scene in Chapter 13 as the basis for a short video using cartoon style shading so there's no point working on the materials and other details now, just save the project along with the other blend files for this chapter.

Conclusion

You have glimpsed each part of the wide range of physics engines built into Blender. You should have an appreciation for the potential traps, and how to avoid them but you probably feel there are several different mechanisms all nearly doing the same thing, so it might be worth taking a step back and reviewing how they fit together.

There are strengths and weaknesses to each of the different simulation systems and knowing which one to use in a given circumstance is an important skill to learn. At a superficial level there are two categories of physics; one uses vertices as its components and one uses individual objects. The vertex level simulation includes everything apart from rigid bodies and a characteristic is that there are different forms of baking, but the baked information is not then accessible, it exists to speed up subsequent animation runs. For rigid bodies the animation information consists of object locations and rotations and this can be integrated more easily with animation keyframes. You used this facility to apply an initial impulse to the ball in the last exercise and it is possible to convert the whole simulation back to an animation by baking keyframes, which you will do with the bugs in Chapter 13.

For cloth and soft body physics the distinction between the two systems becomes much less distinct. It is possible to create bouncing cubes using rigid bodies, cloth or soft bodies and your choice will depend on how important it is to see realistic deformation of the mesh, the complexity of the scene and the requirements for other constraints on objects' movement.

The relationship between physical simulation and animation is very close. You have been adding keyframes to animate aspects of Blender quite extensively, both in this chapter and the previous one, and in the next chapter you get to grips with animation in a much more comprehensive way. You will learn how to build an armature from scratch and how to rapidly fit an armature to an existing mesh in order to animate it.

12

ANIMATION AND RIGGING

From building and animating armatures through to rendering and video production.

Creating 3D models that can be posed quickly and easily is a first step to making impressive animated scenes. But the process of controlling all the components including the camera, and then re-building the rendered frames back into a finished video are also critical stages.

By this stage of the book you have seen objects being animated in the 3D Viewport several times. What's more you are now fairly familiar with the process of adding keyframes to a property, such as when you applied an initial velocity to the ball in the last chapter. This chapter will provide some depth to your understanding of keyframes and introduce alternative editors that can make editing scenes simpler. However, before that, you will learn some basics of rigging by making an armature to control your own model and then by adding a standard armature to a previously un-rigged figure.

Rigging Hard Surfaces

The armatures that you have used so far have all been controlling soft surfaces. The manikin you made in Chapter 4 and the peasant man and lola that you imported as animated .fbx models in Chapter 6 all deformed smoothly at the joints as you rotated the different bones.

Sometimes you need to move parts of an object without inducing any bend or stretch, such as where the material is supposed to be steel or plastic. In the following exercise you will make a robot arm consisting of six very simple meshes, see Figure 12-1. You will then add an armature and see how to constrain and then animate its movement. Links to the files and videos for the exercises in this chapter are at *get-into-blender.com 12.ANIMATION AND RIGGING*

1. Start a new scene and keep the cube, renaming it “robot_arm”. The cube will be the platform of the arm, so switch to Edit mode and move the bottom four vertices up so they are level with the origin and scale the top four vertices in a little.

2. You will now add five more meshes while in Edit mode so they will be part of the robot_arm object but the vertices will not be connected. First make the two arm sections by selecting **Add ▸ Mesh ▸ Cube** then scaling the mesh down in the X and Y direction with S then SHIFT-Z then **0.2**. Move the new cube in the Z direction so it’s above the platform then duplicate it and move the duplicate in the Y direction. Now create the cylinders for the joints, after adding the mesh open the Adjust Last Operation panel and change the Vertices to **16**, Radius to **0.2** and Depth to **0.4** then move the cylinder in the Z direction until it is almost completely protruding from the top surface of the platform. Duplicate the cylinder, rotate it about the Y axis by 90 degrees, move it in the Z direction so the axis is about level with the top surface of the original cylinder then duplicate the second cylinder and move it in the Z direction so it’s clear of the other components.

3. Now you should arrange all your components into a robot arm shape using the orthographic view along the X axis as shown on the left of Figure 12-1. So long as you specified the Z or Y directions when you did the movements in the last paragraph, everything should lie in the plane X = 0, but check this by viewing along the Z or Y axis. You can select all the vertices in each of the meshes by first selecting one vertex then running **Select ▸ Select Linked ▸ Linked**. But note that you will need to do that selection many times during this exercise so you might want to adopt the shortcut CTRL-L.

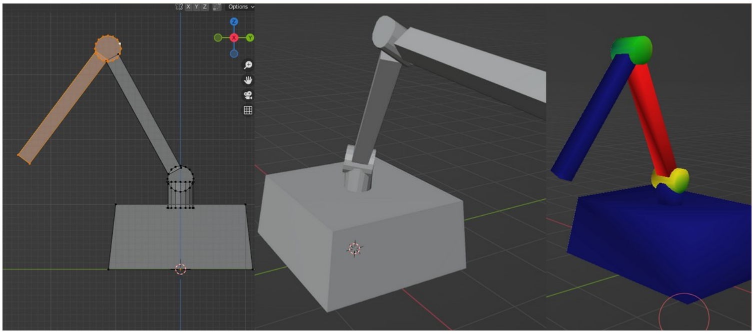


Figure 12-1 The Robot Arm Showing Distortion of the Mesh

4. You will now add an armature to control the robot arm. Switch back to Object mode and select **Add ▸ Armature ▸ Single Bone**. Keep in orthographic view from the side and switch back to Edit mode. The starting single bone of the armature will be the "root" bone for positioning the whole robot arm assembly and you are now going to extrude three more bones for the remaining moving parts. The first bone will be for rotating the arm about the vertical axis, the second will be for rotating the first stage of the arm about a horizontal axis and the third bone will be for rotating the last part of the arm. The cylindrical hinges represent the mode in which the robot arm is intended to bend.

NOTE Armature bones are directional, in that each has a root, about which they will rotate when posed, and a tip. In Edit mode it is possible to select either the tip of a bone, in which case extrusion will cause a new bone to branch from that tip, or the whole bone can be selected, in which case a new bone will also be created from the tip of the previous bone in the chain, if one exists.

The default bone representation is the octahedron, as used here, to make it clear which end is the root and which the tip. There are several other options to represent bones, including the stick method you saw in Chapter 4, and in addition it is possible to create custom shapes to use for each individual bone as you will see below when you use rigify.

5. Select the top end of the initial bone and extrude in the Z direction until the tip of the new bone is at the center of the horizontal cylinder. Extrude the next bone and guide its tip to the center of the next hinge cylinder. Finally extrude the last bone so its end is at the end of the robot arm. On the left of Figure 12-2 you can see the positions of the bones.

6. To set up the parenting, modifier and vertex groups, switch back to Object mode and select first the robot_arm object then the armature. Use **Object ‣ Parent ‣ With Automatic Weights**, then select the armature, switch to Pose mode and try rotating various bones.

7. You will find that there are two problems as shown in the middle of Figure 12-1. The first is that the joints all behave like ball and socket with no restriction of how far, and about which axis, rotation takes place. The second issue is that the cylinders become distorted as the arm is posed, and move relative to the cuboid section of the arm. Switch back to Object mode, select the robot_arm object and switch to Weight Paint mode. In the Object Data Properties select each of the vertex groups and see how the color varies. On the right of Figure 12-1 the weights are being shown for vertex group Bone.002, some vertices are being affected by more than one bone and, for any given bone, some vertices are being affected more than others. This isn't what you want for a supposedly solid component.

Resetting Vertex Groups

8. First remove all the vertices from all the groups. Switch to Edit mode and select all the vertices, then, in the Object Data Properties tab select each bone in turn and click **Remove**. Now, select a vertex on the platform mesh and extend the selection to all linked vertices, then select the Bone vertex group in the Object Data Properties tab and click **Assign**. Repeat this process with the vertical axis cylinder and Bone.001. For Bone.002 select a vertex on the next cylinder as well as the first part of the arm before selecting linked vertices and assigning them. Finally

select the last cylinder along with the last part of the arm to assign to Bone.003.

Bone Constraints

9. Now when you pose the armature there is no distortion but the hinges still have too many degrees of freedom. You need to fix that by setting bone constraints. With the armature selected and in Pose mode select Bone.001, the one that should only rotate about a vertical axis, then open the Bone Constraints Properties tab and select **Limit Rotation**. It might seem logical to set the X and Y limits to 0 for Max and Min then limit Z to -45 to 45. And this would appear to work, but only so long as the platform wasn't tilted at all. That's because the default frame of reference for the rotation limits is World Space and you want your bones to rotate relative to the position of the previous bone, which is Local Space, so you must change that for each bone. However before you start trying to set the limits for each of the bones you should open the Object Data Properties tab, and in the Viewport Display section tick the check-box next to Axes. Now you can see that Bone.001 should only be allowed to rotate about its local Y axis and the next two joints should only rotate about their local X axes. The actual values of the Max and Min will depend on the starting rest pose and the configuration of the joints so you will need to figure those out by a process of trial and error.

10. The robot arm is now behaving like a piece of machinery and can be posed by rotating each bone. However it's quite a long-winded process if you wanted, say, to make the tip of the robot arm follow the outline of a square on the floor. To do tasks like that, very quickly and easily, Blender has an inverse kinematics solver available as a bone constraint, which you will now add to your robot arm. Inverse kinematics, or IK, is widely used in game engines and other posing software to attach a hand or foot to a given location, notoriously to prevent feet sliding over the ground as a character walks. To act as a target for the inverse kinematics, add an empty object to your scene and move it somewhere in front but within reach of the robot arm. Select the armature, change to Pose mode and select the last bone, Bone.003. In the Bone Constraints Properties tab add an inverse kinematics constraint and set the Target to the new empty object. The tip of the bone should reach to the empty but you will probably see the whole robot arm tilt. This is because you

haven't limited the IK solver to just the last three bones. Change the value of Chain Length to **3**.

Inverse Kinematics

11. When you move the empty object around, the robot arm will follow it but the bone constraints for the joints have stopped working. This can be rather confusing at first, however the reason is that there are additional IK controls within each bone's Bone Properties tab. Not only is it possible to add constraints for joint rotation but joints can have variable stiffness to resist rotation in some directions rather than others. For this exercise just go through the Inverse Kinematic settings for the three bones in the chain and set the rotation constraints to match those set up as bone constraints. You will see that there are useful arcs drawn in the 3D Viewport to represent the range of movement.

12. On the right of Figure 12-2 you can see that I added three different materials using the vertex groups of my robot arm and also a bevel modifier, the lighting is using an environment texture. Animate the robot arm by moving the empty object and setting keyframes for its location. You could do all kinds of fun things such as following the outline of lettering or following the seams of a simple mesh that was "appearing" one face at a time using a build modifier.

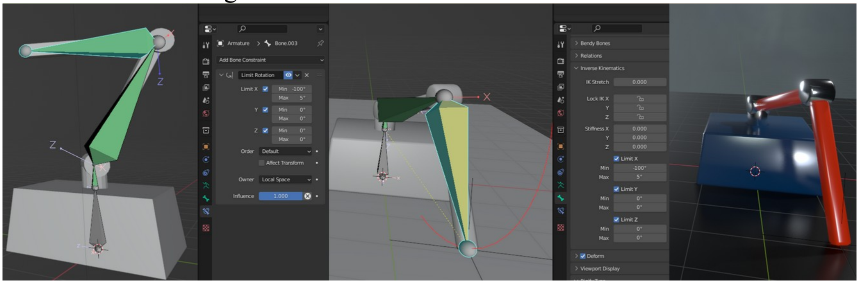


Figure 12-2 Setting the Robot Arm Joint Constraints

Once you have set up some materials and an interesting animation for the robot arm it's time to render the animation.

Rendering an Animation

1. In the Output Properties tab under the Output section open the file browser and create a new directory in the location for this

chapter's files then click **Accept** to set the directory. Unless you have a powerful computer it's probably worth setting Resolution percent to **50** in the Dimensions section as well. Leave the Render Engine as Eevee in the Render Properties tab. Note that the rendering is still set to produce png images rather than a video format and this is the preferred option, partly for quality reasons but also, if something goes wrong or you need to stop the render part way through, you can carry on rendering from the next frame at a later stage.

2. Using the main menu at the top of the window select **Render ▸ Render Animation**. You will see each frame as it is being rendered with the time taken for each, so you can calculate how long the whole job will take. Normally it takes long enough to make a cup of tea, sometimes long enough to go to bed and check again in the morning!

The Video Sequencer

3. Switch one of the editor windows to a Video Sequencer and from the menu choose **Add ▸ Image/Sequence**, then browse to select all the images rendered previously and create a strip.

4. At this point it would be possible to do various adjustments and compositing but for this exercise simply save the sequence as a video. Open the Output Propterties tab and browse in Output the same folder you saved the frames and change the File Format to **FFmpeg video**. Finally, from the top menu select **Render ▸ Render Animation** again.

The FFmpeg video format is the most flexible to use and the default encoding is Matroska, however, because of the mess of different standards and intellectual property claims you might find that your particular version of your particular operating system won't play the video. You can either find a video player that will play the .mkv file you created or you can choose from a wide variety of different encoding formats such as MPEG-4, AVI, Quicktime or WebM. If you get stuck there is lots of help available online.

Now you know the basic way to generate a video you can convert the simulation exercises from Chapter 11 to artistic masterpieces, and start to build up your youtube following.

Rigging a Standard Armature to an Existing Mesh

Being able to find a free 3D model online and quickly add an armature to modify its pose, or to animate it, can be a very useful skill. For instance, if the angel you imported in Chapter 6, looked just right but you wanted it to be sending a text.

The job of rigging is actually very easy to do in Blender with only a couple of potential gochas. Normally it is done on a character mesh that is symmetrical, standing in standard pose and the benefits of that are two-fold. Firstly it's only half the work as positioning bones on one half of the armature can be reflected onto the other side. Secondly the rotations of joints and distortions created are much more predictable when starting from the standard position. However the process you need to go through will be essentially the same as the steps for this exercise, and, as I said above, being able to "tweak" an un-rigged model can be very useful on occasion.

1. Start a new scene, save the robot arm project and delete the start cube. Use the top menu **File ▸ Import ▸ Wavefront (.obj)** and navigate to where you downloaded the files for Chapter 6.

2. Convert the model to the correct size and orientation by scaling to just under 2 grid lines and rotating about the X axis by -90 degrees. I'm sure you know this by now, but if you scaled and rotated in Object mode you need to apply the transformations.

Rigify

3. The rigify addon is installed and enabled in Blender by default after version 2.8 and this includes some ready-make skeletons. Add one now using **Add ▸ Armature ▸ Rigify Meta-Rigs ▸ Human** then scale the armature down so it matches the angel, as near as you can. You will delete the bones for posing the face so the main points of reference are the location of the shoulders and length of the arms. You have probable already done it, but apply the transformations.

4. With the armature selected switch to Edit mode and roughly line up the armature with the mesh. For this first stage I suggest you stick with box select mode and select parts of a limb to move and rotate as one block, positioning in stages. For instance, first select the whole of the right arm and rotate it to about the correct orientation and position it so the elbow is in the right place, then select just the forearm and hand and adjust it so the

wrist is nearly in the right place, and finally select just the hand to rotate the wrist to match the angel's hand. Obviously you need to move the view position around to help with this but you will also benefit from switching between normal rotation, in the plane of the screen, and track-ball rotation using shortcut R twice. On the left of Figure 12-3 you can see the final positioning of the right hand using this technique.

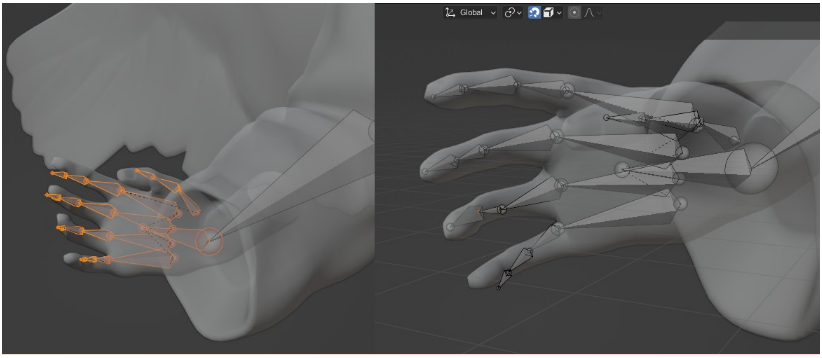


Figure 12-3 Positioning the Bones of an Armature

5. The detailed face features of the human meta-rig are too difficult to match to the angel's face, partly because the pose is not symmetrical, but you can do without them for this exercise. Carefully select all the face bones, without selecting the head or neck bones (the bones you need to keep will be called Spine.006, Spine.005 and so on) and delete them. The Face bone may be hidden by the Head bone.

Snap to Volume

6. You should be able to get the feet lined up by selecting the three bones involved and rotating and moving them as described above. To position the knees you need to select just the ball representing the joint between the femur and the shin bone which can be moved without disturbing the position of the ankle or hip joints. On the right side of Figure 12-3 I have selected the first joint from the tip of the right ring finger. However notice too, at the top of the image, that I have switched snapping on and set the type of snapping to *Volume*, this is crucial for rigging animals. The other setting you may find helpful is to change the selection mode to Tweak, the shortcut for quickly changing from one selection mode to another is W. You should now find the rigging of all the finger bones really quite easy. When

you've finished, don't forget to turn snapping back off otherwise you will be scratching your head when you come to pose the armature later.

7. When all the bones are lined up you might be tempted to parent the angel object to the armature using automatic bone weights, as you have done before, and you would then be able to change the pose. However this would be hard work, especially to get the finger joints to move realistically. So, instead of this, you will convert the armature to a rigify armature and then use that to pose the model. Select the armature in Object mode then open the Object Data Properties tab, expand the Rigify Buttons section and click on **Generate Rig**. If you use the blend file from the *get-into-blender.com* website you might find that you need to **Update Metarig** first.

8. Sometimes there is an error while the rig is being generated, usually caused by a gap in the chain of bones that rigify expects to be continuous. If you look at the human meta-rig armature you will see that there are some bones, such as the collar and thighs, that are separated from their parent by a dotted line. This allows extra articulation in the skeleton, for instance the upper leg will rotate about its root but the root can also be moved slightly without causing any connected bones to rotate. Unfortunately there are several bone ends that rigify expects to be coincident, some in the face and one between Spine.003 and Spine.004. The easiest way to make one end of a bone coincide with the end of an unconnected one is to set snapping to Vertex, so, if you get an error of this type when you tried to build the rig, use snapping to position the bone ends and try again.

THE 3D CURSOR

If you search online for help with problems converting an armature to a rigify rig, you might well find a suggestion as above, but with instructions to first snap the 3D cursor to the position of one bone end and then to snap the second bone end to the position of the 3D cursor. If you look in the menu for the armature in Edit mode you will see, under Armature Snap, a series of options including the ones involving the 3D cursor.

Up to this point in the book I haven't mentioned the 3D cursor, not because it isn't sometimes useful, but the number of occasions where you need to use it are much reduced compared with the earlier versions of Blender. The 3D cursor is the default spawn point for objects added to the scene and it is used by several of the mesh transforms such as Bend, Warp and Spin.

9. Once the rigify armature has been created you can parent the mesh to it for posing. First hide the metarig armature object which will not be used any more, then select first the angel mesh object, then the rig object, and select **Object ▸ Parent ▸ With Automatic Weights**.

10. Try posing a few bones to ensure everything worked correctly. First double-check you turned off snapping and proportional editing, then make sure you can see all the bones by select the rig object, and in the Object Data Properties tab, under the Viewport Display section tick the In Front check-box. Switch to Pose mode, select the left hand IK bone, which is a rounded rectangle and colored red, and move and rotate the hand. You should find that the angel bends their arm and shoulder in a fairly smooth way. Now try moving and rotating the head, neck, abdomen, hips and so on. Some of the bones will create more severe distortion than others, so you have to be careful with those. The fingers have a strange wire with a spatula sticking out of them. This can be rotated and, as with the other bones, the fingers will rotate about the point where they are attached to the palm, however if you scale these bones the fingers will curl and uncurl which is a very useful feature for posing. Remember, in order to rotate in the plane of the screen use a single press of R and to track-ball rotate press R twice.

11. The next step in this exercise is to construct a short animation in which the angel pulls a mobile phone out of their back pocket, glances down at it, touches the screen, puts the phone back in their pocket and returns to the start pose. First of all you need to

make the phone. Add a cube the scene, scale it down to an appropriate size and shape, add a bevel and then a subdivision modifier, add a new material with black base color and reduced roughness and finally set to smooth shading. Position the phone behind the angel's back, there's no need to model the pocket or other details, the hand will just reach around out of sight and reappear holding the phone. You should also make sure that the camera angle doesn't view the screen of the phone or the angel's face in too much detail.

12. Now add some keyframes for the movement of the angel's arms and head. Select the rig object and switch to Pose mode, then select the right hand IK bone and insert a keyframe by pressing I and selecting **Location, Rotation & Scale**. Move the current frame pointer in the Timeline on to 50, which is about 2 seconds then position the right hand about where the right, back pocket would be. Rotate the hand so the fingers are pointing downwards and the palm is twisted towards the angel's back. The hand will be hidden from the camera but the IK process might propagate twist back up the chain of bones which would impact on the rotation of the elbow and shoulder. Insert a keyframe here. Move to frame 100 and position the right hand in front of the angel's chest, somewhere it would be comfortable to read a text message, and insert a keyframe. Don't worry about positioning the fingers at this stage, you will do those later.

13. To return the phone and strike the original pose you will copy these three keyframes. Make the Timeline area a little larger then select the first keyframe, duplicate it using SHIFT-D and move the duplicate to frame 250. Duplicate the keyframe at 50 and put a copy on frames 75, 175 and 200, the same position from frame 50 to 75 and from 175 to 200 represent the pause while the phone is being taken out of, and reinserted into, the back pocket. Duplicate the keyframe at 100 to frame 150 which will give the angel about two seconds to see the message and press the acknowledge button.

NOTE The Timeline shows the keyframes for each of the selected objects in the scene that have had any keyframes previously added. On the left is a collapsible hierarchy of detail, and if you expand that, you will see that there are keyframes for each component, enabling, for instance, the X rotation to take place prior to the Y rotation. When you copy or move keyframes, especially if more than one object or bone is selected, you need to be careful that you only select the ones you intended.

14. Run the animation, and where the movement is awkward, or attempts to pass through the angel, position the hand correctly and insert a keyframe until you have created a plausible sequence of movements.

It turns out to be rather hard to animate humanoid armatures without making them look a bit like the robot arm in the first exercise. For many animation sequences, such as the two Mixamo models you downloaded in Chapter 6, the positions and rotations of bones have been inferred from motion capture of human actors. This results in very realistic movement, but it creates lots of keyframes, and is extremely difficult to reproduce by hand. You already know one way to improve the result by deforming the mesh using physical simulation of cloth or soft bodies, but for this animation you will use the Graph Editor and see the functionality that is available there. Before that, however, you need to make the phone attach itself to the angel's hand while it's out of sight, you will then be able to animate the left hand and the head.

The Child Of Constraint

15. To make the phone move as if attached to a bone in the armature is actually quite easy using object constraints and there are several ways to make an object copy the location, rotation or transforms of a bone in an armature. The problem is you don't want the phone to be positioned *exactly* where the hand bone is, it needs to be offset into the palm of the hand and getting that to work is much easier using the Child Of constraint. Move the frame pointer along to 70, then in Object mode select the the mobile phone object and open the Object Constraints Properties tab. Select **Child Of** and set the Target field to **rig** and Bone to **hand_ik_R**. Add a keyframe for the Influence field set to **1.0** by pressing I with the cursor hovering over it. Move back one frame to 69, change the constraint Influence field to 0.0 and set a keyframe for that.

16. You now need to set the position and rotation of the phone for these two frames. At frame 69 the phone should still be behind the angel's back, so press I with the cursor over the 3D Viewport and select **Location, Rotation & Scale**. Move forward to frame 70 and position the phone correctly relative to the hand. You will sort out the finger positions later so try holding your own phone and see where it lines up relative to your palm, it only needs to be fairly approximate. Insert a keyframe here as well, then, in the Timeline duplicate the keyframe at 70 to 180 and 69 to 181, the reverse order for letting go of the phone.

17. Sort out the fingers on the right hand by moving to frame 69 then selecting the rig armature and entering Pose mode. Select the spatula controls for all four fingers and the thumb then enter a keyframe so they keep their starting positions. Move on to frame 70 and pose each finger by scaling and rotating the spatula controls, once each finger is correct save a keyframe for it. It might be easier to see what it looks like by moving forward to frame 140, say. If you enter the keyframes at this later time then you must move them back to frame 70 by dragging them in the Timeline area, see Figure 12-4. When it looks good, duplicate the keyframes in the Timeline in reverse order to 180 and 181.

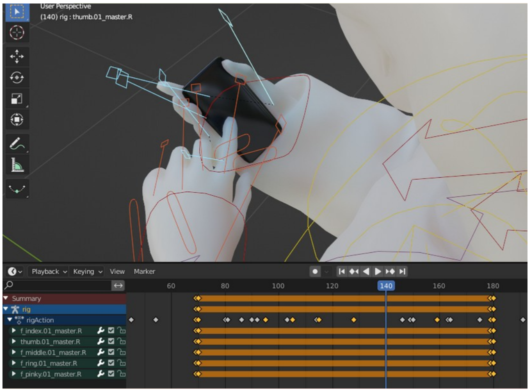


Figure 12-4 Keyframes for Gripping Phone

18. Now that you have a phone with which to interact, you should animate the left hand, the head and the neck. I did a little role play to get the timings of when the angel might start to move their left hand, and glance down. It seemed more natural to do most of the phone extraction and lifting up before moving the left hand to press the button and glancing down at the screen.

You could add some slight shoulder or torso movements as well if you thought they made the animation more realistic. The left hand is already held in a pose suitable for touching a screen but, again, you might like to add a little movement to add some life.

The Graph Editor

19. The Graph Editor area extends the functionality of the Timeline by representing not only the time of each keyframe but also the value at each point and the shape of the curve between the keyframes, see Figure 12-5. Change the Timeline area to a Graph Editor then, with the rig armature in Pose mode, select the right hand IK bone. As you can see there is a lot of info squashed in here so there are two ways you could reduce the clutter; the first is to hide the channels you're not editing by clicking on the eye symbols on the left, the second is to adjust the ends of the sliders to set the range of the area to match the graph.

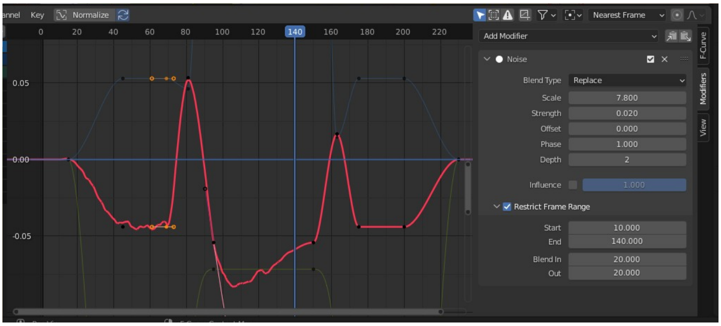


Figure 12-5 Adding Noise Using the Graph Editor

20. This editor is a good place to make subtle, accurate changes to an animation, it would have been ideal for aiming the ball in the last chapter, but for this exercise you will just use it to add a little noise to the arm movement. Each curve can have a series of modifiers applied to it in a stack which is accessed using the right sidebar, open that now either by dragging the inconspicuous arrow or using the shortcut N. Select the channel for the Z Location, and from the Add Modifier drop-down, click **Noise**. Slow the the noise down by increasing Scale and reduce the amplitude by reducing Strength. Increasing Depth to 2 will make the noise more detailed, and you can make the shaking start after frame 1 and finish before frame 250 by setting values in Restrict Frame Range. Adjust the settings and run the

animation until you get the right level of variability to stop it looking too much like a robot arm but short of delirium tremens!

21. Once you have all the animation working well, with the right amount of noise, you can add some artistic style and create another video. These were the features I added, but you shouldn't feel constrained by this list: A marble material for the angel, mainly different scales of noise texture with Normalize un-checked for subsurface color and bump map. A shape key to rotate the eyes and eyelids, animated to flick downwards when the head tips forward. An environment texture of a suitable indoor space, a cloth plane to animate the right wing, and finally, camera animation. To add keyframes to the camera, select it in the outliner then switch to camera view in the 3D Viewport, set the frame in the Timeline area, move the camera using the walk navigation controls, and save keyframes by pressing I and choosing **Location & Rotation**.

Creating a More Complex Video Sequence

For the third exercise of this chapter you are going to take the two characters you downloaded from Mixamo and use their animations in a creative way. The first problem is to stop Lola's hands from moving while they should be gripping a hold, which you can do using an IK constraint. You then need to apply the climbing up the wall animation to the peasant man, and when they both reach to top of the wall you need to apply the salsa dancing animation to Lola so the two characters can celebrate together. You will use the wall factory add-on to build the castle walls and a cloth simulation to create a rope.

1. Save the last exercise and start a new scene. Delete the start cube then import the .fbx file of Lola climbing a wall which you downloaded for Chapter 6. Add two empty objects to the scene and rename them left_hold and right_hold.

Making Hands and Feet Stick

2. You now need to find where would be a good place to fix the holds. Run the animation one frame at a time by pressing RIGHT, and when the left hand looks to be latching onto a hold move the empty up so it lines up with the tip of the hand bone. You will need to change the Viewport Display option for the

armature to In Front in order to see all the bones. Use orthographic view from the side and the front to help position the empties. As you move the animation on, the hand will slide slightly but that's the point of doing this exercise so don't worry.

3. You will add an IK constraint for each hand to make them stick to each of the holds, and you will animate the strength of the constraint from 0.0, a few frames before the hand lands on the hold, to 1.0 at the start of holding on, then 1.0 at the end of holding on, to 0.0 at the end of letting go. Because the animation loops very nicely, the constraint for both hands should be 0.0 for the first frame and the last frame, otherwise there will be a discontinuity in the hand positions. Move back to the first frame, select the armature, switch to Pose mode and select the right hand bone. In the Bone Constraints Properties tab select **Inverse Kinematics** then set the Target to **right_hand** and Chain Length to **4**. The right hand is actually on the hold in the first frame but set the Influence to **0.0** and add a keyframe, move forward three frames, set the Influence to **1.0** and add the next keyframe. The frame to start letting go is when the left hand starts to curl over its hold. Move forward to that frame and insert another keyframe with Influence still at **1.0** then allow the motion to blend smoothly into the animation by setting the Influence to **0.0** ten frames later. Figure 12-6 shows the sequence of keyframes.

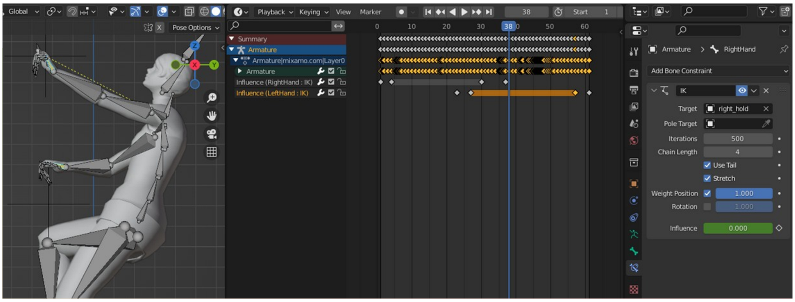


Figure 12-6 Keyframes to Control IK Influence

4. Repeat the procedure for the left hand. The first keyframe should be for Influence 0.0 ten frames before the hand reaches the hold and, at the end of the loop the influence should drop smoothly to 0.0 again by the last frame. When you run the animation the hands should stay in place once they're grasping a hold.

Ideally the animation would last for several loops, to allow you to record a few seconds of continuous climbing, and you can probably see one way to do that. It would be quite easy to copy and paste all the keyframes so they were repeated a number of times, then in the Graph Editor, increase the values of the Z component of the armature root bone so that the second loop continued from the end of the first, and the third continued from the second. However there are deficiencies with that technique, the first one being that the empty objects controlling the IK constraints would have to be moved up at the end of each loop. The second problem is more practical, where you are using a short animation repeated many times, for instance the legs of a bug scurrying along, it becomes very hard to manage a scheme based on copying and pasting.

Baking Constraints to an Animation

The best solution is to bake the animation, including the IK constraints, into a series of posed positions, which is essentially the form of the animation imported in the .fbx file. Blender then gives you the option of creating an *action* from the animation, which you can can use to make a *strip* to repeat multiple times. You can also apply the action to other armatures, so long as they have the same bones.

5. To bake the animation, select the armature and enter Pose mode. Select all the bones then from the menu click **Pose ▸ Animation ▸ Bake Action** and tick the four check-boxes Only Selected Bones, Visual Keying, Clear Constraints and Overwrite Current Action. If you don't tick the last option, you will preserve the animation imported with the .fbx, and Blender will generate a second action with the generic name "Action", "Action.001" and so on. This might be useful if you were going to make different versions of the animation, each requiring tweaking, but in this case it will be fine to just overwrite it. Bake Data should default to Pose but make sure the Start Frame and End Frame are correct, then click OK.

6. After a pause the keyframes should have been overwritten and, when you run the animation, it should look no different. However if you check in the Bone Constraints Properties tab with either hand bone selected you will see that the IK constraints have gone and the two empty objects can now be deleted. If you select both of the hand bones and look in the Timeline area you will see that the two channels of keyframes are still there, but underlined in red to show there is an error. Delete each of the lines.

7. There are two more editor types that you will need to use for more complicated animations. The first one to look at is the Dope Sheet which allows you to see keyframe positions for a variety of different types. Change the area at the bottom from Timeline to Dope Sheet, then change the Editing context drop-down from Dope Sheet to **Action Editor**. As shown in Figure 12-7. At the top of the area some options have appeared; Push Down, Stash and a drop-down to Browse Action to be linked to the object. At the moment there is just one action, but now is a good time to rename it to climbing_baked_ik. When you import the Peasant Man .fbx there will be another action and this is where you will be able to swap them, to make the Peasant Man climb and to make Lola dance.

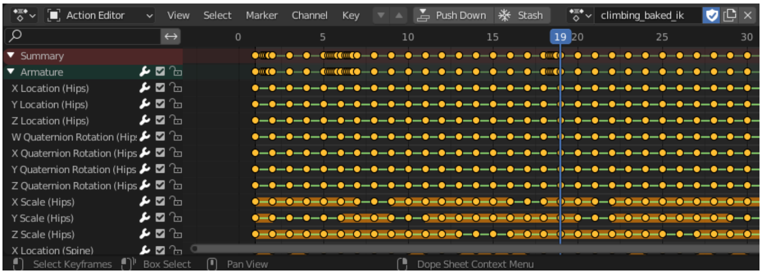


Figure 12-7 The Dope Sheet Editor

8. In order to make an action strip which can be repeated a number of times change the editor type to Nonlinear Animation. This view has a different section for each animated object in the scene, and each section has a list of actions in orange, for this scene there is just one object and one action. On the left of Figure 12-8 you can see that there is a button; Push Down Action that will convert the action to a strip as shown on the right of Figure 12-8, press that now for the climbing_baked_ik action.

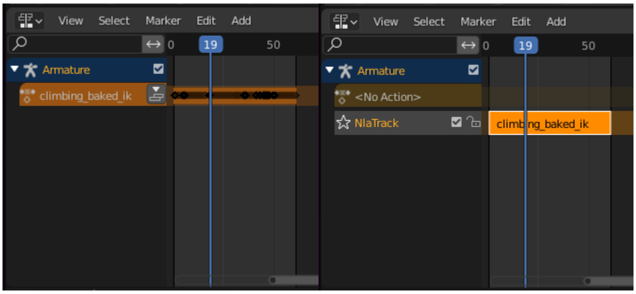


Figure 12-8 The Nonlinear Animation Editor

9. The bright orange block can now be moved, split, duplicated and generally used for building more complicated animations. The idea is that other animations can be overlaid on the basic one. As an example, you could make a short animation of just the neck and head to make Lola look over her shoulder for a fraction of a second and if you placed that strip above the climbing it would override the default climbing animation, just for those bones. If there were face bones, or shape keys available you could then make Lola's face either smile, or grimace in horror, as a third layer of composition. The potential, and complexity, of this animation editing facility is enormous but, for now, you simply need to duplicate the existing strip twice using the RMB context menu. The copies will be placed in a new track above the existing one so drag them down so they are all in a line and delete the newly created, but now empty, tracks by clicking the track name on the left and pressing DELETE.

Animating Parent Objects

10. If you now change to the Timeline view and increase the End frame so you can see all three strips playing you can see that you still need to correct the start position for the second and third strips. The tidiest way to do this is to have one empty object that can be moved around the scene to provide the start location for the animation, and a second empty that will move up by the correct distance at the end of the first and second cycles. Create two empty objects and rename them lola_start and cycle_start. Select first the armature object then the cycle_start object and hit **Object ▸ Parent ▸ Object**, then, in a similar way parent the cycle_start object to the lola_start object.

11. In the Timeline move to the last frame of the first animation cycle and with the cycle_start object selected add a keyframe for its location by pressing I with the cursor over the 3D Viewport. Use the orthographic view from the side and zoom in so Lola's head is about half the height of the screen, then pan the view so the tip of Lola's nose if exactly in line with the top edge of the window. Move forward one frame then move the cycle_start in the Z direction until Lola's nose is in the exact spot it was at the end of the previous cycle. Add a location keyframe for the cycle_start here. Repeat the process at the end of the second cycle, adding a keyframe in the same location, and on the next frame a location that keeps her nose in the correct place. Now you can move the lola_start empty to different locations on the castle walls and she will climb there.

Simulating a Rope

Now that the climbing animation is sorted it's time to construct the rest of the set. Having a rope dangling from Lola's waist will add a dynamic element so do that first.

1. Add a plane to the scene and rename it rope, then scale and move it so it's where the floor would be behind Lola's back, you can judge the size from Figure 12-9. In Edit mode subdivide the plane four times then, with edge selection enabled, delete rows of edges so you are left with a serpentine shape. Move four or five vertices so the end vertex is just behind Lola's waist. Finally create a vertex group containing just the end vertex called "mixamorig:Hips", this has to match the bone name in the armature but you can also use it as the pin group in the cloth.

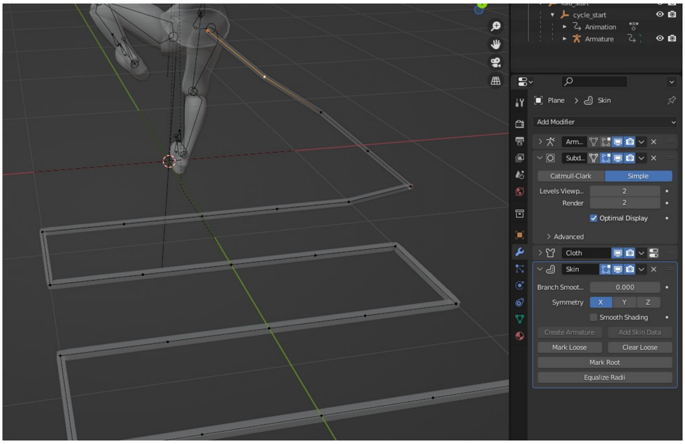


Figure 12-9 The Rope

2. The rope object needs to be parented to the armature object as an object, you don't want the bone weights to be generated. Then you need to add modifiers in the order; 1 Armature, targeting the armature object. 2 Subdivision, set to Simple and Levels to 2. 3 Cloth. 4 Skin with smooth shading. Reduce the skin to a realistic thickness by selecting all the vertices in Edit mode and using **Mesh ▸ Transform ▸ Skin Resize**. In the Physics Properties tab open the Shape section of the cloth simulation and set the Pin Group to **mixamorig:Hips**. Check that the simulation runs correctly, you should also change the shading to smooth and set the material to a more rope-like color. Having a one dimensional rope like this is quick to simulate but difficult to get collisions working well, but for the purposes of this short video it will be fine.

3. To make the castle wall, enable the Extra Objects add-on then hit Add Mesh Extras Wall Factory, then move and scale the section of wall to match Lola's climbing. Add two array modifiers to the wall object, one repeating in the X direction and the other repeating in the Z direction. You might need to tweak the factors from exactly 1.0 or -1.0 so you don't have gaps. To make the blocks expand to touch each other without leaving cracks you can select all vertices in Edit mode then **Mesh ▸ Transform ▸ Shrink/Fatten**. You will probably want to add a bevel modifier and create a suitable stone material. To avoid distortion of textures on the end of the blocks you should smart unwrap the UV map and feed that as an input to the texture nodes.

4. Use an environment texture for the background and adjust the strength to give it night-time lighting. I used the eevee engine for speed but set ambient occlusion and motion blur on. In retrospect the motion blur was a mistake because of the jump frame of the cycle_start empty object. I also reduce the frame size to 50 percent which rendered each frame in about one and a half seconds on this desktop computer with a reasonable, if old, graphics card. To add a little more realism I also added a tiny bit of noise to the camera X and Y rotation, for a hand-held feel. Render two or three videos of Lola climbing quite high on the wall, as image sequences that you can mix later.

Animating a Second Armature

5. The final product will include the Peasant Man character climbing up the wall after Lola so make sure the lola_start object is located reasonably up the wall then import the .fbx as you did for Lola. Open the Dope Sheet in Action Editor mode and rename the convolutedly named animation to salsa_dancing. From the Browse Action to be linked drop-down select **climbing_baked_ik**. To make your life easier, at this point, also rename the armatures in the Outliner Editor to lola_armature and man_armature.

6. Change to the Nonlinear Animation view and follow the same process you used for Lola. Push down the climbing_baked_ik action to create a NlaTrack then duplicate the strip twice, drag them in line and delete the empty tracks to keep it tidy. You also need to create two empties, as you did for Lola and parent the man_armature to the man_cycle_start empty and parent man_cycle_start to the man_start empty. Follow the same procedure to animate the movement of the man_cycle_start empty with four keyframes so the animation runs without discontinuity for three cycles.

7. Record some video of the man climbing. It would be nice to attach the bottom end of the rope to his Hips bone but the simulation with collisions becomes rather slow so it's probably sufficient to increase the Air Viscosity in the rope Physics Properties tab, to stop it swinging around so much.

8. When you have enough footage of climbing, in the Dope Sheet change the animation for both armatures to salsa_dancing. In order to stop the characters jumping upwards part way through the dance you also need to disable the animation for both the

cycle_start empty objects. Do that in the Dope Sheet either by clicking the X next to Browse Action to be linked or deselect the check-box on the channels list. In the Timeline change the end frame to match the length of the dance then move both characters to the top of the wall and record some video of them dancing.

9. When you have all the footage you need, create your final masterpiece using the Video Sequencer as shown in Figure 12-10. There are many controls available here, and it's hard to do more than scratch the surface in this exercise, however two things may be useful. The first one you may want to use is the fade effect which you can control using the right Sidebar in the Compositing section. You can set keyframes for the Opacity in the same way you have used to animate other Blender values. The second one is to slow down a strip, or part of a strip. For this you need to click Add Effect Strip Speed Control from the area menu, again, there are many controls available in the Sidebar.

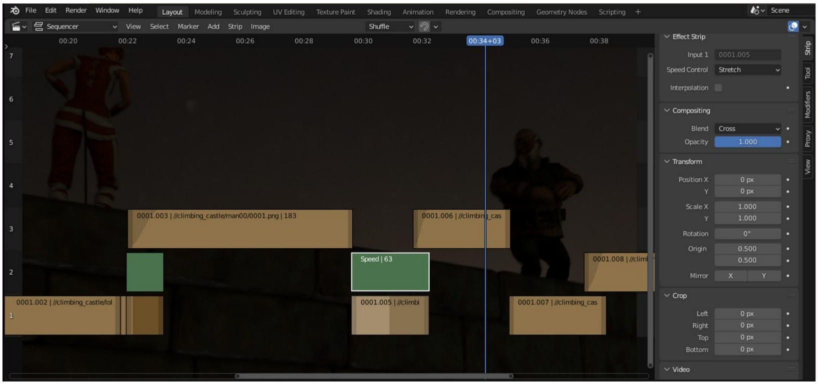


Figure 12-10 The Video Sequencer

Before closing the chapter there is one job that belongs here. When you used rigid body simulation in Chapter 11 to bounce balls off each other as they rolled around a bowl, I mentioned that you would use them later in the book, and that the balls would turn out to be bugs that would uncurl themselves and scuttle back up the sides. Well now would be a good time to create the meshes, armatures and animations for those bugs.

Creating and Animating a Bug

This will be another opportunity to do some bread-and-butter mesh work, building on the lessons learned when you made the furniture in Chapter 8 and the shark in Chapter 10. This time you will make most of the geometry by extruding faces of the start cube. As the bug will be symmetrical you can use some shortcuts to build the armature and you will animate the legs using the Follow Path object constraint.

1. Start a new Blender scene, saving your previous work, but leave the start cube as the basis for the bug mesh, and rename it bug. The balls in the simulation were radius 0.2 so the bug should be about 0.6 long, 0.2 high and 0.4 wide. You can see the proportions in Figure 12-11 stage 2, and when curled up in stage 3. With the cube selected enter Edit mode and scale the vertices down to the size of the middle leg section. To help with the animation later, design the bug so it is walking on a surface running through the origin as shown in Figure 12-12. Extrude and scale sections in the Y direction for other body sections, the head and tail. Once you have enough divisions along the length of the bug use mesh bisect as you did in Chapter 10 then add a mirror and a subdivision modifier and set shading to smooth.

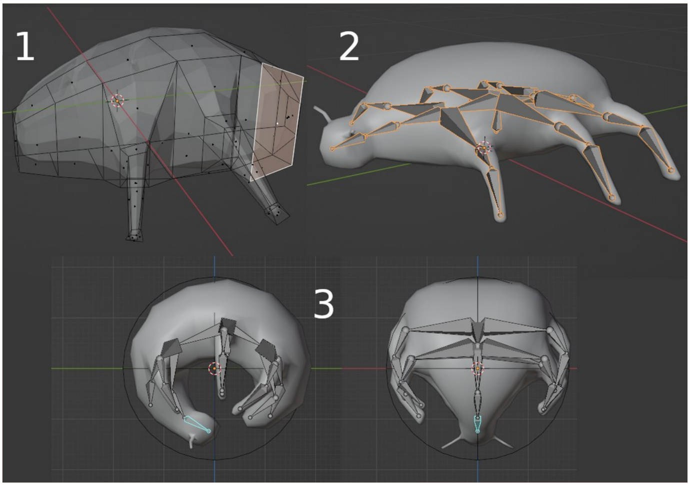


Figure 12-11 Crafting and Rigging the Bug

2. Extrude, scale, move and rotate the faces from the sides of the bug to form the legs and add extra edge loops to help define the shape and sharpen edges as needed. The modeling doesn't need to be too detailed but, as the bug will be curled into a ball,

involving large deformations, you should make sure that the visible parts are composed of quads. When it looks sufficiently bug-like, switch back to Object mode and apply the two modifiers.

3. Because the bug is symmetrical you only need to construct one half of it. Start by adding a new armature object, set it to display in front of the mesh, then switch to Edit mode in orthographic view from the X direction, and scale the initial bone down so it's about a third of the height of the bug, as visible in Figure 12-11 stage 2. Extrude this forwards four times and backwards three times to make the spine – if that's the right term for an invertebrate. In orthographic view from above extrude three bones from matching nodes on the spine to the points where each leg joins the carapace, renaming each bone "leg1.001", "leg2.001" and "leg3.001". Now, when you extrude these bones the suffix will increment, and if you set snapping to volume, you can position the bones to the center of the legs as you extrude them. Remember to turn snapping off afterwards otherwise you will run into problems when you try to move control points or objects later.

Symmetrize an Armature

4. Once you have three legs rigged you can quickly generate the mirror image but first you need to append ".L" on the name of each leg bone, and there is a handy tool for that too. Select all the leg bones then from the menu run **Armature ▸ Names ▸ Auto-Name Left/Right**. Now, with all the bones selected, click **Armature ▸ Symmetrize** to generate a symmetrical rig with the left sided bones named with an L suffix and the right sided bones an R suffix. Finally generate the rig as you did previously with **Parent ▸ With Automatic Weights**

5. Before you set up the scuttling animation, create the pose with the armature curled up, changing to the unposed state as an uncurling animation. With the armature selected and in Pose mode, view the bug from the side in orthographic view. Move the spine bones to make the mesh as spherical as possible then view from different angles and fold the legs in. In Figure 12-11 stage 3 you can see the two orthographic views, and also that I added a rigid body with a spherical collider to help position the bug inside, though you will need to remove the rigid body when

you run the animation as it will make the mesh fall off the armature! Select all the bones then add a keyframe for the rolled up pose at frame 1, move to frame 30 and select **Pose ▸ Clear Transform ▸ All** and add a keyframe for all bones in the flat pose. Check that the animation runs as expected then, with the Dope Sheet set to the Action Editor, rename the animation to uncurl and push it down onto the NLA stack.

The Follow Path Constraint

6. You will now create a second strip to make the legs move, so click on **+ NEW** to create the action and rename it scuttle. When you move the frame pointer you will find that the NlaTrack for uncurl is still being followed because there is no action yet set above it in the stack. Move to frame 1, in Pose mode select all the bones, clear the transforms and add a keyframe for the unposed state. To set up leg movement, switch back to Object mode and add a Bezier circle object then, in Edit mode scale, rotate and move it to the path that the end of the front left leg should follow, as shown in Figure 12-12. Add an empty and reduce its size in the Object Data Properties tab so it matches the scale of the bug, then add an object constraint **Follow Path** setting the Target to the Bezier circle you just added. Click the button **Animate Path** to automatically create the modifier in the curve. You now just need to select the armature and, in Pose mode select the end bone of the front left leg. Add an inverse kinematic bone constraint setting the Chain Length to **2** and the Target to the empty.

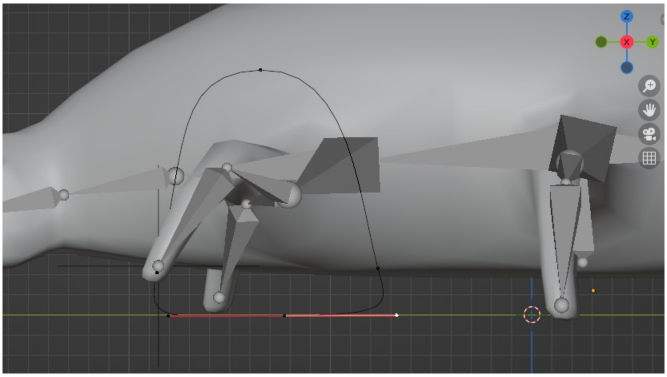


Figure 12-12 The Bezier Circle to Control an IK Constraint

7. Check that the leg follows the curve without any awkward bending, you will probably need to adjust the position and shape of the curve. There is a 50 percent chance that the animation will run in the wrong direction, if that's the case, with the curve in Edit mode, select all the control points and run **Segments ▸ Switch Direction**. The default speed of the animation is also rather slow so select the curve object and open the Graph Editor then in the Sidebar on the right open the Modifier tab. Increase the Coefficient for X^1 to **12.0** which should make the legs spin round a bit more frantically. Once the leg is behaving properly make sure you have renamed both the curve and empty to something related the the front left leg, such as BezierCircle.L.001 and Empty.L.001 then duplicate them and position them to drive the other legs. Remember to change the L.004 to R.001 when you duplicate the first leg and empty for the opposite side. In order to make the legs move out out of phase, so that there are always three points of contact on the ground, for each empty change the Offset value in the follow path constraint. You need to get the timing right so the lifting leg doesn't hit the leg that's about to land, though the legs will be whizzing round, and the bugs will be small enough, that it will be hard to see details on the final video.

8. Once you are happy with the animation, it's time to bake the constraints to actual bone rotations, but first save the blend file just in case. It will be much harder to adjust once it's been baked, so don't tick the overwrite option when baking. When you're ready, select the armature, and in Pose mode, with all the bones selected, run **Pose ▸ Animation ▸ Bake Action** and tick just the three check-boxes; Only Selected Bones, Visual Keying and Clear Constraints. In the Dope Editor rename the action to scuttle_baked_ik and push it down as a new NLA strip.

9. To check it all worked correctly change to Nonlinear Animation view, select the scuttle_baked_ik track by clicking on it and drag it so it starts at the end of the uncurl strip. In the Sidebar you might have to change Blending to **Combine** to allow the lower strips to influence through the strips above them. When you play the animation the bug should uncurl and then scuttle, the empties will still go round and round the curves but they no longer control anything. You could delete them but I suggest you keep them until you have the whole sequence in the can. Save your file!

There is one more step needed for the bugs in the bowl exercise, which is to make the bugs follow paths as they scuttle up the sides. However, as that will be part of the final rendering process, you will implement it in Chapter 13, when you work on cartoon style materials.

Conclusion

You've covered a lot of ground again in this chapter. Building an armature by hand, from scratch, has given you a little more insight into how they work and, hopefully, will enable you to spot problems when they occur. You also saw how you could use armatures to bring your machines to life by making wheels rotate, levers swing and pistons slide, and how bone constraints simplify so much of the difficulties of posing and animating armatures.

Fitting an armature to a mesh is a requirement for both soft bodies and hard machines, so being able to do this rigging quickly and easily has enormous benefits to your productivity. The exercise with the imported .obj file of the angel showed you how this can be done and it also introduced you to posing with the rigify system. As well as rigging and posing, you used object constraints as yet another way to synchronize different parts of your scene and automate parts that would be difficult to construct manually.

The castle wall exercise took some existing animations that were imported to Blender as part of an .fbx file and showed you how you could modify and re-apply them in different ways to different objects. You used four different representations of animation keyframes; Timeline, Graph Editor, Dope Sheet and Nonlinear Animation, all with slightly different applications. To start with you will need to keep referring back to your notes to remember what functionality is where, but in time you will learn your way around them and value their flexibility.

In the next chapter you will concentrate on non-photorealistic rendering and the compositor. You will add the finishing touches to the bug animation and learn about outlines, cell and hatched shading. The compositor tends to be undervalued by people starting to use Blender but it allows a great deal of different effects and optimizations that would be impossible to do otherwise.

13

UNREALISTIC ASPIRATIONS

Creating non-photorealistic materials and the many uses of the compositor.

Using 3D models to create cartoon style images gives a massive productivity boost to artists compared with manual methods, but achieving a reproducible, hand-drawn, effect turns out to be quite hard to do. Fortunately there are several features of Blender that help the process.

In this chapter you will use a few new techniques of combining nodes in the Shader Editor to create “cell” shading, which is the term used to describe blocks of uniform color applied as if by a simple printing process or by a color wash. It is used in comic books and many animations, including manga and anime. If you search online for help on this subject you will find a plethora of clever systems, most of which are more complicated than the ones you will use here. However I suggest you start simple and get an understanding of what’s going on, you can then add extra features as and when you find they are necessary. You will then look at ways to add hatching such as might be used for pencil or ink drawing, and also, the grease pencil line art system that Blender has for adding outlines.

The second half of the chapter will cover some of the uses of the compositor. You will see how different parts of the scene can be rendered onto different view layers, and how the layers can then be recombined. The use of nodes in the compositor has many overlaps with their application in material shaders, but you will briefly sample some of the filter and distortion nodes and see how they can be introduced into the data flow. Links to the files and videos for the exercises in this chapter are at *get-into-blender.com 13.UNREALISTIC ASPIRATIONS*

Cell Shading

The first method of cell shading described here is essentially to take the output of the standard principled BSDF shader material and feed it through a color ramp converter to create a hard division between two areas of solid color. It will be easier to see how this works using a simple example.

Color Ramp

1. The shading effect will be more obvious on a rounded shape so start a new file in Blender and add a subdivision modifier to the start cube with two levels of division.

2. To set up a very simple arrangement of material nodes as shown in Figure 13-1, increase the size of the bottom window area and switch it from Timeline to Shader Editor. Change the name of the material from Material to comic. To see the results of changes to the node arrangements you will need to change the Viewport Shading to **Rendered** with the Render Engine set to **Eevee**.

3. The default material has just two nodes, the shader and the material output, so move them further apart to leave room for three more nodes to be inserted between them. Now select the following menu options placing each node onto the connection between the shader and the output node;
Add ▸ Converter ▸ Shader to RGB,
Add ▸ Converter ▸ Separate Color,
Add ▸ Converter ▸ ColorRamp and
Add ▸ Converter ▸ Combine Color The nodes to separate and recombine the color should be set to **HSV** with Hue and Saturation connected directly from one to the other but the Value output passed through the color ramp node. Set the ramp interpolation drop-down to Constant and adjust the slider to make the cut-off between black and white part way up the side of the rounded cube.

4. You can probably see how this method could easily be extended to having three or four divisions rather than this simple black and white one. Try adding another three stops to the color ramp node and edit the base color of the BSDF shader to different brightnesses of yellow, say.

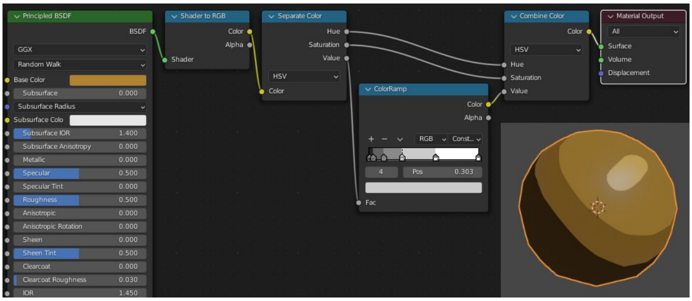


Figure 13-1Simple Cell shading using the color ramp node

This simple color ramp method is perfect for many situations, and might produce quite satisfactory results if used for the bug uncurling animation that you will tackle later. However there are disadvantages that arise from the inability of the color ramp node to take variable inputs for the colors and positions of its stops, which means that each color ramp needs to be adjusted individually for each material and lighting conditions. This arrangement also produces blocks of uniform color that tend to blot out surface detail, apart from in the area where one cell changes to another.

The Map Range and Mix RGB Nodes

Node groups tidy up the layout, so when you make complicated materials using lots of nodes, there is an obvious visual improvement. However the real power of groups is that, if designed properly, they can be reused many times. This will be one of the real benefit of switching from the color ramp node to the map range and the mix RGB nodes, as they can both use variable input values.

1. Edit your existing material by deleting the color ramp and the combine color nodes then add two more to replace them by selecting **Add ▸ Converter ▸ Map Range** and **Add ▸ Color ▸ MixRGB**. The output of the map range node is used for the factor input of the mix node. You need to set the From Min and From Max values very close together to give a sharp cut-off at the edge of the cells. In the mix node change the colors to two shades of yellow, putting the darker one above the lighter one.

2. The first, obvious, deficiency of this system over the color ramp node, is that there isn't an obvious way to insert additional stops with different shades. One solution is to change the interpolation of the map node from Linear to **Stepped Linear** when you do that you will see an extra input value; Steps, which you can change to **3.0** to give intermediate blended colors. You will need to change the From Min and From Max values to make the cell divisions fall more evenly.

3. The advantage of this technique over the simple color ramp is when you want to build up layers of shading, such as for the cross-hatching you will use later. For those situations the more general solution, using grouped nodes will be required, but in order to see the potential of this technique, quickly try moving the color input out of the mix node. Create a new input by selecting **Add ▸ Input ▸ RGB**, place the node to the left of the shader and feed its output into the Base Color input. Change the color of the new input RGB node and use the eye-dropper to set it equal to the lighter value in the mix node. Now drag another two connections from the input RGB node to both color inputs of the mix node. You need to darken the Color1 input by adding a node with **Add ▸ Color ▸ Bright Contrast**, dropping that onto the connecting pipe, and reducing the Bright value. Finally add a magic texture and a bump node to set the normal of the shader. Reduce the Distance value of the bumps to **0.05** to give

a more subtle effect. The full node layout and the resultant material are shown in Figure 13-2.

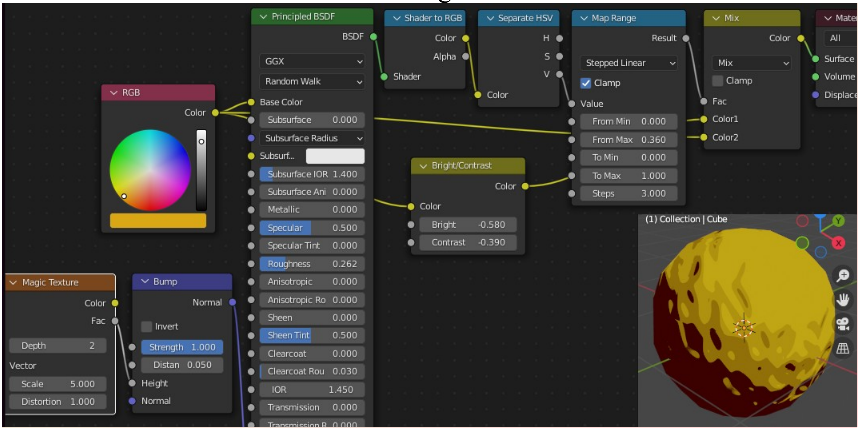


Figure 13-2 More complicated Cell shading using map range and mix RGB nodes

As you tried to fine-tune the settings you probably found that it was difficult to get the correct range of colors, especially where the roughness was low, as highlights or reflections are typically a different color from the bulk material. The neatest solution to this problem is to use *node groups*.

Node Groups

As mentioned above, groups are a way to organize nodes but, by doing so, they help you to think through the logic of the material as you build it. For instance, in the cell shader you are making now, it would make sense to have a node group that took inputs of: First the threshold value for the edge of this boundary between cells. Second the color to use use where the material is darker than the boundary. Third the color to use if lighter than the boundary. These nodes could then be stacked one after the other, each taking the result of the previous node for the darker color and introducing a lighter color for the other. In this way each group is equivalent to a stop of the color ramp node, the arrangement is shown in of Figure 13-3 where the three node groups are arranged on the right side of the image.

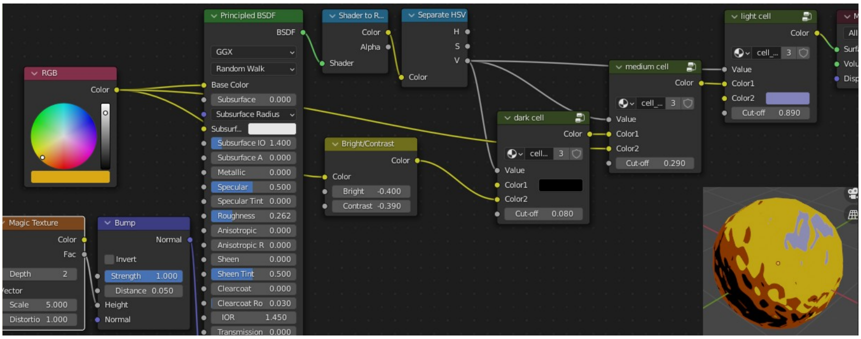


Figure 13-3 Using multiple node groups to produce a cell shader.

4. Select the map range node as well as the mix RGB node then from the menu **Add ‣ Group ‣ Make Group.** The original network of nodes has now been replaced by the contents of the new group with the additions of an input and an output node. On the right, in the Sidebar, you can alter properties of the group such as adding new inputs, renaming them and setting default or limiting values.

5. For the general purpose version of this group you need to add an input to control the cut-off point. In the sidebar select the Node tab then add a new input by clicking on the relevant **+**. Change the name of the new input to Cut-off, the default to **0.5**, Min to **0.0** and Max to **1.0**. Connect the Cut-off input to the From Max input of the map range node, set the From Min to **0.0** and the Steps to **1.0**. The layout of nodes within the group is shown in Figure 13-4.

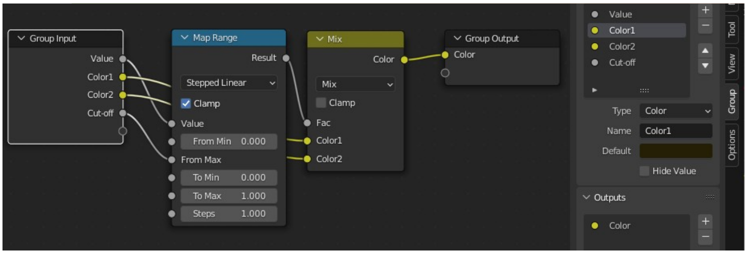


Figure 13-4 The Nodes inside the cell_shader_stop node group.

6. To finish editing the group node click on the button on right of the header with an icon of an upward pointing arrow. To return to editing the contents of a group you click on the connected nodes icon in the top right corner of the node, which you can just make out in Figure 13-3. When you select on a node that is

a group, if you open the Properties section of the Node tab in the Sidebar, you can change the name to something more meaningful, in this case call it cell_shader_stop. In the Node section above you can also change the unique name for each instance of the group, as well as the label to be displayed at the top of the node. Using meaningful names will make your life much easier in the long run. Duplicate the prototype node twice using SHIFT-D and connect them up as in Figure 13-3

There is much sophistication you could add to this basic node group. For instance, if you wanted to give the cell areas slightly blended edges, as if created using a water color wash, you could change the interpolation method to linear and calculate the From Min value as a fraction of the Cut-off using a multiply node. For the moment, however, there is sufficient control and complexity for the current exercise, so we will move on to look at some inking techniques.

Hatched Shading

There are two more mechanisms to investigate before finishing off the bugs-in-a-bowl animation, the first is how to add hatched shading to a material. In the node group you just developed you mixed the two colors in a fairly intuitive, pro-rata way. The effect was the equivalent to changing the transparency of the top color so it blended with the one underneath. However for hatched shading you need the black lines of each layer to accumulate on top of each other, and this can be achieved by changing the mode of the mix node to *darken*.

1. Make a duplicate of the cell_shader_stop group node and drag it down below the other three. To the right of the drop-down and name of the node group there is a number and, to the right of that, a shield icon. The number represents the occasions that the node group is being used anywhere in the current blend file, which should be 4. Click on the number to make a new copy of the node group, and to allow you to edit it without altering the three already being used in the cell shader. Rename the node group from cell_shader_stop.001 to hatched_shader_stop.

USERS IN BLENDER

Many components, including materials, image textures, animation actions and node groups can be used in multiple locations in a blend file. Bender keeps track of which components are required by other objects, and shows the number of users at the side of each component wherever it is listed. When you save a blend file, Blender strips out unused clutter to save disk space and improve running efficiency when you reopen it. More often than not, this is just what you want, and you don't have to think about it, but sometimes you need to keep unused resources available. One example of this would be where you created a blend file as a library of animations, materials or node groups that you could append into other projects when you have a requirement.

The way that Blender allows you to control the users, and consequently whether the resource will be saved, is through the series of buttons beside the resource name. The first button shows the number of users, if there are more than zero, and clicking on this will create a new instance of the original. The second button has a shield icon and this will create a fake user for the resource to prevent it from being discarded when you save the file. In some locations there are additional buttons such as one showing two overlapping squares which will create a new resource, a button with a folder icon for browsing files to load or one with an X icon for deleting the resource.

2. The hatched shader will operate in a similar way to the cell shader, it will take an input of background color and it will overwrite it with a hatched texture wherever the value of the color output by the principled shader is below a cut-off value. Click on the top right corner of the hatched_shader_stop to edit its contents and change the mix node from Mix to **Darken**. In order for the hatched texture to be blended at darker values, you need to invert the output, on the map range node change the To Min value to **1.0** and To Max value to **0.0**.

3. The hatching will be provided by a noise texture node but in order to build up layers of shading on top of each other you need to rotate the direction of stretching. This means that you won't need a second color input but you will need one for the rotation angle. In the Sidebar Group tab remove the Color2 input and add one called Rotation with Type **Float**, Default **0.0,** Min **-pi** and Max **pi**.

NOTE Within Blender, angles are held as radians but in some entry fields, such as the Object Properties tab, the entry and display is provided in degrees. For those situations you can specify which format you are using by following a number with the letter D or R. It is also possible to enter formulae into any value field in Blender so long as they are legal python expressions. You could have typed **radians(-180)** and **radians(180)** and Blender would have calculated the range using the python radians function.

4. In order to construct the hatched texture you need to add some more nodes: UV map, vector mapping, noise texture and a color ramp, all of which you've used before several times. You need to use the rotation as the Z value of an XYZ vector as the vector mapping node requires a vector input, add this by selecting Add Converter Combine XYZ. To simplify the vector rotation operation, set the Type field of the vector mapping node to **Texture**. The node layout for the group is shown in Figure 13-5.

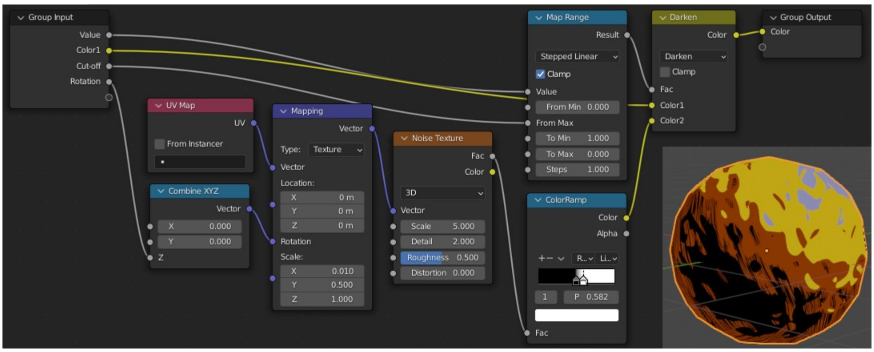


Figure 13-5 The Nodes inside the hatched_shader_stop node group.

5. Close the edit of the node group by returning to the parent node tree and position the new hatched shader to take the output of the last cell shader as well as the value output from the original shader. Adjust the rotation and the cut-off to give a pleasing result.

At this point you could further enhance the hatching by, for instance, allowing the thickness of the lines to be controlled with another map range node instead of the color ramp. Additionally the rotation of the lines could be made relative to the curvature of the surface, by using the normal output from a texture coordinates node. But for the moment leave the material as it is and save the blend file so you will be able to append the node groups into the bug animation project.

Line Art

At the start of the previous section I alluded to two mechanisms, and the second one is Blender's *Grease Pencil Line Art*. This is a convenient way to add lines to specified edges by creating grease pencil objects with various modifiers. In previous versions of Blender this would have been done using freestyle lines, but they have the disadvantage of being harder to control and can't be previewed on the 3D Viewport, only appearing when the scene is rendered. If you are constrained to an old version of Blender then I'm afraid you will have to find resources online to help you sort out their intricacies.

1. First of all you need to add a new Grease Pencil object with a Line Art modifier. In the file you've been working on for the cell shading shading exercise, make sure that the rounded cube is selected then in Object Mode select **Add ▸ Grease Pencil ▸ Object Line Art** and a new object called LineArt will appear in the Outliner. In the camera view the line runs round the outside of the Cube object but from other view positions you can see that it's a 3D object stuck to to the rounded cube's surface.

2. To change the line style and thickness, select the LineArt object and open the Modifier Properties tab where you can see the modifier that was generated automatically. Try adjusting the Line Thickness to different values then set it back to **25**.

3. In the modifier expand the Edge Types section and you will see that the default is to draw lines around contours but none are apparent on the bumpy surface. This is because the surface lumps are only being created by adding a normal texture rather than deforming the mesh surface. To fix that, select the Cube again then open the Shader Editor and delete the magic texture and bump nodes feeding into the principled shader. Open the Modifier Properties tab and increase the subdivision levels to **4** then add a displace modifier and click **+ New** to make a new texture to control the distance. The default new texture will be called Texture so open the Texture Properties tab and change the Type to **Magic** then, back in the Modifier Properties tab change the displacement Strength to **0.2**.

4. You should see that some lines have appeared around the new bumps. However the lines are rather thick and have ugly square ends. To improve this situation we can use a different Grease Pencil object for the lines drawn inside the object, away from its outline, with a smaller thickness and with tapered ends. Select the LineArt object and rename it LineArt_silhouette, then in the Modifier Properties tab change the Edge Types to Silhouette in the first drop-down, and un-tick the other options.

5. Add a new line art object using **Add** ▸ **Grease Pencil** ▸ **Object Line Art** as above, and rename it LineArt_contour. If you didn't first select the Cube object you will need to first change the Object value in the Modifier Properties tab to **Cube**. Add a second modifier with type Thickness, tick the Uniform Thickness check-box and reduce the Thickness to 15. Expand the Influence section, tick the Custom Curve check-box and create a profile that goes from small to big to small again. The settings and resultant lines can be seen in Figure 13-6

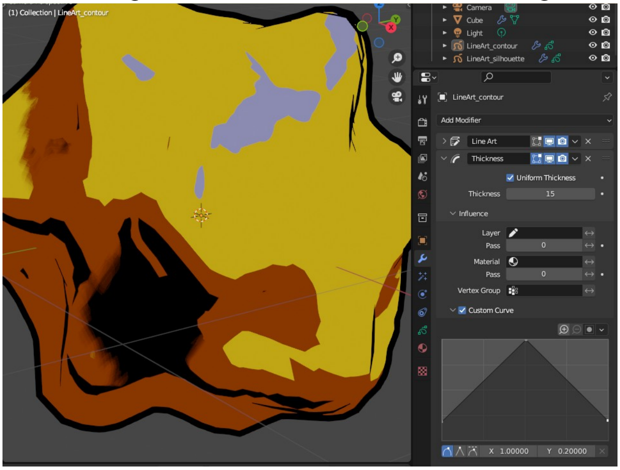


Figure 13-6 Line Art settings.

As you added the thickness modifier you saw that there are multiple features of line drawing that can be controlled by stacking multiple modifiers. The permutations are enormous and, if you are interested in using Blender for graphic art, you will find many amazing works online to inspire you. I will leave you to explore this topic further on your own, for the moment just save the blend file so you can append the material for the bugs in the next section.

Assembling the Project Elements

At last it's time to put together the physics, animation and materials to create a cartoon style video of the bugs rolling down then climbing out of the bowl. The first job that needs to be done is to substitute the bug mesh and armature from Chapter 12 for the simple spheres used in Chapter 11.

1. Open the the last blend file from Chapter 11 which should be the physics simulation of the balls rolling around the bowl. There are three ways you could approach switching the bugs for the balls. The first would be to snap the cursor to the location of each ball, appending the bug, snap its location to the cursor, add a rigid body and constraint to the bug then remove the original balls and run the simulation again. The next logical approach would be to keep the existing simulation but hide the balls and add an object constraint for each bug to copy the location and rotation of the corresponding ball. The third option, and the one I will guide you through here, is to bake the physics of each ball to a keyframe animation and then, after appending the five bugs, add the corresponding animation in the dope sheet action editor. Before you bake the animation you probably need to do a bit more fine-tuning. Ideally you want all the bugs to have stopped rolling by the time the uncurling animation starts so you might need to increase the friction of all the surfaces in the Physics Properties tab. I added keyframes to the friction values so they increased from 0.1 to 1.0 between frame 149 and 150. I also increased the Rigid Body World, Cache to 350 frames in the Scene Properties tab. After the friction changes you will probably have to re-aim the heavy ball, try using the Graph Editor to tweak the location of the ball at the second keyframe.

2. When you have the physics running well and the balls nearly stationary by the end of the animation, select the five bug balls and the heavy ball, then from the menu run **Object ‣ Rigid Body ‣ Bake to Keyframes**. Save the blend file now to the folder used for this chapter. The baking process removes the rigid bodies from the objects involved, however there are still some hanging animated properties for the removed constraints which might cause Blender to crash if you trigger certain processes. Make the Timeline Editor pane larger and expand the left hand outliner. If you see any entries underlined in red then delete them now, and then delete the four empty objects. In the

Outliner Editor, toggle the visibility and rendering off for all the bug.00x balls.

3. Append the bug object from the last exercise of Chapter 12. Even though the object you import is a mesh, the armature, animations and materials used by it will also be brought into this file. To make Blender help with renaming as you duplicate the bug, rename the armature to bug_armature.001 and the mesh object that it parents, to bug_mesh.001. Now select the armature and the mesh and duplicate them four times, you should end up with sequential suffixes.

4. Select bug_armature.001 then open the area at the bottom of the screen as Dope Sheet with the Action Editor context selected. From the drop-down for choosing actions select **bug.001Action** that was generated by baking the keyframes for the first ball. Switch to the Nonlinear Animation and scroll down to find the tracks for bug_armature.001, move the two imported tracks; uncurl and scuttle, to the right of the baked rolling ball animation. Repeat this procedure for the other four bug_armatures. The order of the tracks for the uncurl and and scuttle animations is important as the positions of bones in a higher track will override a lower one. If the bugs roll down the bowl uncurled or don't scuttle after uncurling move both the actions into one strip. The alternative is to toggle a track off while recording the animation before it, then toggle it back on to record the remaining animation.

5. When you run the animation you should see the bugs replicating the collisions and rolling just as the balls did. You may find that the bug diameters are not quite the same as the rigid body colliders and the bugs either float above, or sink into, the surface of the bowl. It would be quite easy to move the positions of the animation keyframes up or down using the Graph Editor, however it is even easier to adjust the Z location of the bowl object. Getting the correct height will be easier if you use Viewport Shading as Rendered and check the position of the cast shadows.

Putting together the full animation in one unbroken shot would be pretty cool but it will be much easier to assemble the video from several components. To achieve that you will first set some keyframes for the camera location and rotation so it follows the action and finally zooms in close, as the bugs come to rest. You then need to set the materials for the bugs, ball and bowl to produce anime style rendering, and subsequently render a sequence of images for the first part of the video.

For the next stage you want the bugs to start in frame 1 in the same locations and rotations that they finished the previous animation. You then want the bugs to play the uncurl action at the same time as they transition from whatever rotation they end up in, to the right one for the scuttling sequence. For the scuttling you will add five Bézier curves drawn onto the surface of the bowl and set an object constraint on each bug to follow a curve.

6. Select the camera and set the current frame to 1 then use walk navigation to position it at the start and insert a keyframe for position and rotation. Move forward through the animation so you get a good balanced view of the action without dramatic camera movements that will give the viewer sea-sickness. Add enough keyframes to prevent the camera swinging the wrong way but not too many that you have sudden transitions. By the final frame you want the bugs fairly filling the frame.

7. Append the material named comic from the blend file you worked on at the start of this chapter. You could also append two LineArt objects but there are several modifications needed so it simpler to create two new Grease Pencil objects as described below. Select each bug mesh and set the material to the appended comic. For the bowl and the heavy ball, also select the comic material but then click on the overlapping squares button just to the right, to create a new material for each object. You can now change the base color, roughness, metallic and other properties of these two objects. I used a magic texture for the base color of the bowl but set the cell input color as blue, as you can see in Figure13-7.

8. In the Outliner select the default collection called Collection then in the 3D Viewport add two line art objects with **Add ‣ Grease Pencil ‣ Collection Line Art**, rename them and adjust the modifier properties as before but, for LineArt_silhouette set the Edge Types to **Individual Silhouette**. Adjust the Line Thickness values and do a few trial renders to make sure everything works as expected, then render the whole animation as an image sequence into a sub-folder for this exercise. Depending on the power of your computer you might want to reduce the resolution to 50% in the Output Properties tab, it could take quite a while to render 350 frames.

9. When you have the required shot in the can, save the blend file with a different name. For projects like this where something might go wrong or get accidentally deleted, it makes a lot of sense to do lots of incremental saves so you can go back to a working restore point. Put the current frame marker on the last frame, then in the Nonlinear Animation area, for each bug, push down the baked physics animation onto the stack. Mute the physics animation track and un-mute the uncurl and scuttle tracks. Now when you run the animation the bugs should stay where they are but uncurl and scuttle their legs.

10. You need to add keyframes for the location and rotation of each bug armature at frame 1 as it is now, and at frame 30 rotated onto its stomach, disentangled from any adjacent bugs and pointing outwards from the group. The two keyframes are shown in Figure 13-7.

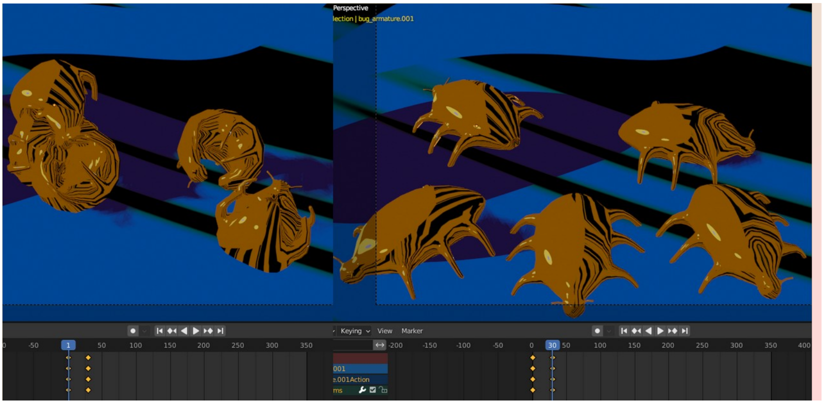


Figure 13-7 Keyframes for curled and uncurled locations and rotations.

11. The bugs each need a path to follow. Stay on frame 30 and check which bug is number 001 then, in Object mode add a Bézier curve, rename it bug_track.001, switch to Edit mode, delete all the vertices then select the Draw tool. In the Sidebar, Tool tab set the Depth option to **Surface**. Draw a curve starting just in front of the bug, in a circuitous route up to the rim of the bowl. Change to the selection tool, select the vertex nearest to the bug and move it as close as possible to the bug's origin; the yellow dot visible when the bug is selected. Adjust the handles of the vertices near the bug so the curve is in line with the direction the bug is pointing.

12. Now you have a line, the bug needs an object constraint to make it follow it, but only after frame 30. Return to Object mode, select the bug_armature.001and, in the Object Constraint Properties tab, add **Follow Path**. Set the Target as **bug_track.001**, the Forward Axis as -**Y**, tick the Follow Curve check-box and click **Animate Path**. Change the Influence value to **0.0** and set a keyframe for this at frame 30. Move on one frame to 31, change the Influence value to **1.0** and set a second keyframe here.

13. If you run the animation you should see the bug suddenly jump, at frame 31 to an offset position part way along the curve. There are two corrections needed; first, delay the start of the path animation so it commences at the end vertex on frame 31, second, the location and rotation of the bug_armature need to all be set to 0.0 between frame 30 and 31. Select the bug_track.001 curve object then change the bottom window area to the Graph Editor. Open the Sidebar and select the Modifiers tab where you can see the details for the generator function created when you clicked on the animate path button. The function is a straight line starting from -1 with a slope of 1.0, so by frame 31 it has moved 30 percent of the way along the path. Fix this by halving the slope, change the x^1 value to **0.5**, and delay the start by changing the Coefficient to **-15**. The bug will now start moving at frame 30 and get to the end of the line at frame 230.

14. To make make the bug actually stay on the track instead of being offset, move to frame 30, select the bug_armature.001 object and open the Object Properties tab. There should already be a keyframe here giving a yellow background for the location and rotation values. Advance on one frame to 31, change all the location and rotation components to 0.0 and set a keyframe here. If you use the shortcut, you will need to press I over one of the location values as well as over one of the rotation values.

15. Watch the bug carefully all along its path. If you see the bug floating above or sinking into the surface, or alternatively, rotating to one side, then select the curve object and switch to Edit mode. Carefully move the relevant vertices up or down and rotate the bug by adjusting the tilt. For exercises like this you will probably find the shortcut CTRL-T more convenient than the Toolbar or menu options.

16. Add paths and constraints for the other four bugs following the description above. When they're all animating to your

satisfaction you should set up the camera animation. In the Timeline delete all the keyframes you set up for the rolling animation, apart from the last one and move that to frame 1. Run the animation on for fifty frames or so then position the camera using walk navigation and set another keyframe. Try to get some close-up views of the bugs as well as making sure to see that there are lots of bugs all running around independently, take care that the camera always moves smoothly.

17. Finally render the frames to a different sub-folder of this chapter's storage location. Save the blend file then open a new project using **File ▸ New ▸ Video Editing** and import the two sequences then render in video format. When you import the images make sure that they are sorted by file name as Blender will keep them in the order they appear in the file browser window.

The Compositor

Many users of Blender never venture far into the intricacies of the compositor, which is a shame because, not only does it allow many enhancements to the finished render, but sometimes it can speed up the process significantly.

To use the compositor beyond the simple application of filters, requires the use of view layers which create different sources of image information to composite together. It is possible to render a scene to multiple view layers with each layer being used for different collections of objects and collecting different information such as lighting, depth and masks. One of the options that can be varied between layers is the number of samples used for rendering with the cycles engine, and this allows the background objects to be rendered at relatively low resolution, but the central focus of the scene to use a much higher sample rate. Different layers can have noise reduction, blur, sharpen, glow, color correction and many other filters applied.

View Layers

The creation and switching of view layers is controlled using an inconspicuous drop-down at the very top right of the Blender window. By default there is just one view layer called ViewLayer and, very often that will suffice, but in this exercise you will use two layers; a background containing some glass objects and a foreground containing a large diamond. The problem with transparent and reflective objects is that they can produce a lot of noise when rendered, which requires a high number of samples to resolve. The denoiser in Blender version 3.0 onward is amazingly good, as you will appreciate if you look at Figure 13-8, but in some circumstances it can blur details, such as the highlights on the top of the red cut-glass block. In this exercise the foreground will be rendered with a high number of samples and the background will use a low number of samples and denoised in the compositor, then a small amount of the noisy render will be mixed back in.

1. Start off by adding a few objects to the scene. I used the add-on; extra objects, which was still enabled from Chapter 12, and has a diamond mesh option, but a normal ico sphere would be fine. Add a cube with beveled edges and some five sided cylinders as table and background objects. Change all the materials to Glass BSDF with low roughness and light tint, for the cube use a procedural texture and bump node to create a slight normal pattern. Finally in the world settings add an environment texture. You can see in Figure 13-8 a detail from the scene I set up, but the actual layout isn't too critical.

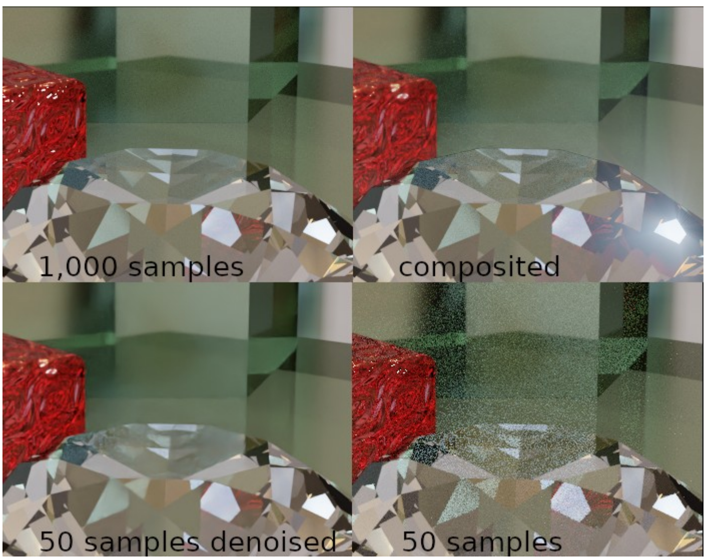


Figure 13-8 Comparison of different sample numbers and effect of denoising.

2. Once you have the scene set up it's time to create the two view layers and configure some collections to use them appropriately. At the top right of the window rename ViewLayer to background_layer, then create a new layer by clicking on the

button to the right with the overlapping squares, and rename the new layer as foreground_layer.

3. You now have two layers but there will be no real indication of how they effect anything. Open the Render Properties tab and set Render Engine to **Cycles**, change Max Samples under the Render section from 4096 to **50** and un-tick the Denoise check-box. When you render an image you will see that the whole scene is rendered twice and at the top of the Blender Render pop-up window you can click on a drop-down to see the results of rendering each layer. At the moment the two layers are the same.

Collections

4. Divide the objects into two collections. At the moment everything is in a collection called Collection, rename that to background_objects. Right click on the Scene Collection and add a new collection, rename it foreground_objects then drag and drop just the diamond object into it.

5. At the top of the window select the background_layer as the current view layer and un-tick the check-box next to the foreground_objects collection. Then select the foreground_layer as the view layer and un-tick the check-box against the foreground objects. Now, when you render you will see a difference between the render results, but it's not really what you want. The lighting is missing from the foreground render and the background render has no shadows cast by the foreground object, besides which, there is no composite result recombining the two layers.

Holdout and Alpha Over

6. A neat solution to this problem is to use *holdouts* to enable the renderer to skip over the pixels where objects in a given collection should be. With the foreground_layer still selected as the view layer tick the check-box next to the background_objects collection to re-enable it, then right click on that collection and select **View Layer ‣ Set Holdout**. In order to get reflections and shadows from the foreground object, change the view layer to the background_layer and re-enable the foreground_objects collection here too.

7. Before you render there is one more setting you need to make; open the Render Properties tab and, near the bottom, expand the Film section and tick the Transparent check-box. When you render you should get the background with all the objects shown but the foreground will just have the diamond with everything else the gray checker pattern to indicate transparency. You are in a position to reassemble the image in the compositor.

Compositing

8. Enlarge the area at the bottom of the window, change the editor type to Compositor then tick the Use Nodes check-box in the header. The default is a single render layers input node connected to a composite output node. Select the render layers node and duplicate it then create a new node with **Add ▸ Color ▸ Alpha Over** and drop it onto the connection leading to the composite output node. Connect the image output of the duplicated render layers node to the other input of the alpha over node, then using the drop-down list at the bottom of each node, select the **background_layer** for the input to the top input of the alpha over node and the **foreground_layer** for the bottom input.

9. In order for the composite result to be any different from the one for the background layer, you need to increase the samples when rendering the diamond. Make sure you have the foreground_layer selected as the current view layer, then in the View Layers Properties tab scroll down to the Override section and change the value of Samples from 0 to **1000**. You also need to transfer some more information for the background layer to use in the compositor later, so select the background_layer as the current view layer then in the View Layer Properties tab, Passes section, under Data, tick the Denoising Data check-box, and under Light tick the Environment check-box. When you render you should see that the background is noisy but quick, and is overlaid by a less noisy diamond. However the environment texture background is missing.

10. Back in the compositor area there is an addition that will allow adjustments to be checked much more quickly. From the menu select **Add ▸ Output ▸ Viewer**, put the node next to the composite output node and connect the image output from the alpha over node to this node as well. In the Sidebar on the right,

open the View tab and tick the Backdrop check-box, you can zoom and position the result using the controls in the tab. When something produces an unexpected result it is often useful to direct intermediate results into the viewer node until you find the step in your network that is causing the problem.

11. In order to denoise the background image select **Add ▸ Filter ▸ Denoise** and drop the node onto the output from the background_layer. The quality will be improved by connecting the normal and albedo inputs as well. To blend the denoised image with the noisy one use **Add ▸ Color ▸ Mix** and use the first input from the denoiser node and the second input from the background_layer directly. Adjust the factor to give the best result, and try different mixing mode such as darken or lighten, to reduce blurring without showing too much noise. The environment texture needs another alpha over node taking its first input from the Env output of the background_layer and the second input from the color mix node. The output of this node goes into the first input of the original alpha over node. You node layout should look like Figure 13-9 below, with the exception of the sun beams and lighten nodes applied to the foreground layer.

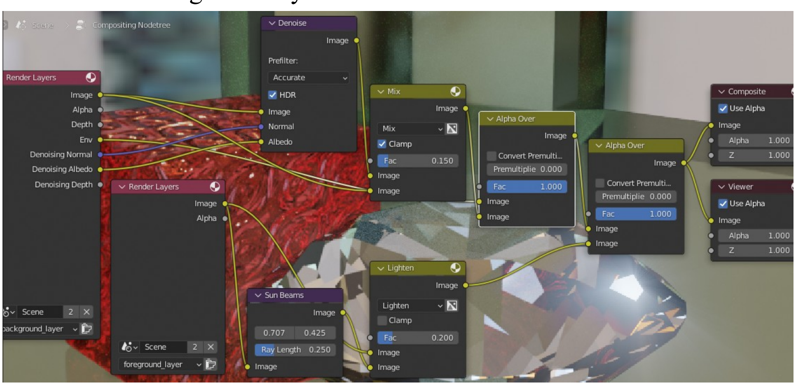


Figure 13-9 The compositor node layout.

The sun beams effect added in this example is just one of a vast number of filters, distortions, masks and other effects available in the compositor. The full scope is similar to the functionality of specialist image processing software such as Photoshop, indeed the compositor has an input node to read an image or movie from file and an output node to write one, so it would even be possible to use Blender to process photographs or film clips. However there isn't enough space in this book to investigate all the details, so I suggest you do some experiments on your own. Where a filter or color processing node has a

Fac input, you can use the output of a matte mask node, often blurring the mask to soften the edges, which makes it very easy to do things such as reducing saturation towards the edges of the image.

Back in Chapter 10 I mentioned that the correct way to avoid the spotlight shining onto the shark or the sand was to use the compositor. You might now have some idea how that could be achieved, it is a little more complicated than the example you have just completed. There need to be two view layers and the spotlights need to be moved into their own collection. In the main view layer the spotlight collection is completely disabled and creates no illumination or shadows. In the spotlight view layer the main collection has holdout set so that the compositor can just use the version without any spotlight effects. The seaweed object needs to be illuminated by the spotlights as well as the sun, and it should cast a shadow over the sand and the shark. To achieve this the seaweed should be put into a third collection which is enabled and rendered in both layers, but the version with the spotlights will be composited in front of the version without.

In situations where the foreground object is large, and the background objects are being rendered at a high sample rate, the time spent rendering the foreground twice might become quite significant. In those cases, to avoid duplicated rendering, you can set holdout for the foreground object in the background view layer, however the alpha over node will then leave a faint, one pixel wide line around the foreground object. To avoid that you must add the colors of the two view layers and then set the alpha value to the sum of their individual alphas. Figure 13-10 shows the node layout for a simplified version of the previous exercise without denoising or sun rays.

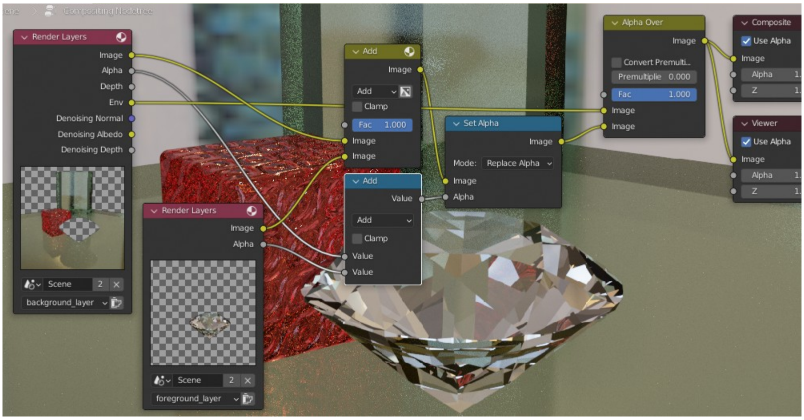


Figure 13-10 The node layout for compositing layers with complementary alpha values.

As you can see on the left of Figure 13-10, the area of the diamond is this scene is small relative to the rest of the background objects, so the benefit of not rendering it is only a few percent, but sometimes the speed increase will be significant, so this is a mechanism to keep in mind.

Conclusion

Making non-photorealistic images is an important creative use of Blender, and a significant proportion of the software development is directed in support of animator workflow. The approach you took for the cell and hatching materials was just a simple starting point, and when you search online you will find of a host of different methods, but now you have an understanding of the way they work, you will be able to pick and choose the best ideas.

The final assembly of the cartoon video, with the curled up bugs climbing back out of the bowl, used several sophisticated techniques and, if you managed to get everything to work perfectly first time, you should feel proud of yourself. Even if you had to re-read sections, or use the support information for this chapter, you will have learned a lot from this exercise, probably more than if you hadn't encountered problems.

The compositor, by its very nature, is the logical final stage of producing an image with Blender. To use it effectively you need to build on many other features, and it is no coincidence that this chapter has come at the end of the book. As I mentioned above, there are many filters and effects that I don't have space to describe, but you should spend some time investigating them on your own. You transferred the environment texture and denoising information to the background view layer, but you will have also seen that it is possible to pass freestyle lines, cryptomatte masks, normals, shadows, UVs, lighting information and much more, all of which enables you to do some amazing things.

The next chapter, the last one in this book, is dedicated to all the things in Blender that I didn't have time to cover in detail. Of course there wouldn't be space in a single chapter to do more than itemize all the subjects that I missed out so, instead, I will concentrate on two important features that are being actively enhanced with each Blender release; geometry nodes and grease pencil.

14

WHAT NEXT

Geometry nodes, grease pencil and all the features that couldn't be squeezed into the book.

There are so many useful and effective ways to use Blender that there just isn't space to explore them all properly. In this chapter I provide a basic introduction to two significant areas not covered elsewhere.

Both geometry nodes and grease pencil are relatively new arrivals for Blender. Although version 2.79 provided the ability to draw with grease pencil, the power and flexibility has increased dramatically with each release since. Geometry nodes were introduced in version 2.9 but they have since been completely overhauled with many new nodes available, and a very different method of linking the nodes together. As at version 3, both geometry nodes and grease pencil have reached a level of stability where minor details may change, and usability will certainly improve, but the core functionality should stay the same. Links to the files and videos for the exercises in this chapter are at *get-into-blender.com 14.WHAT NEXT – Geometry Nodes and Grease Pencil*

Geometry Nodes

A stated aim of the Blender developers is to transfer much of the configurable aspects into graphical, node based systems, as you experienced with material shaders and the compositor. The grand name of this project is *everything nodes* and, although nodes have already been used for many years in other parts of Blender, the first release of functionality under the new heading is geometry nodes.

The number of things you can do with geometry nodes is growing all the time, but it embodies many of the characteristics of modifiers along with the particle system ability to scatter instances of other objects over surfaces. The scope of geometry nodes is already quite amazing and, as more of the existing mesh and curve modifier become available as nodes, it will undoubtedly become the default way to accomplish several of the exercises in this book.

In the first exercise you will use geometry nodes to scatter rocks and boulders over the surface of another object. This is essentially the same result achieved by the particle system in Chapter 4 and Chapter 5 but with greater flexibility, control and, once you get the hang of it, much easier to use.

The Geometry Nodes Modifier

As with physics and particles, the starting point for working with geometry is placing a modifier in the stack, and, as with those other two systems, the order of the modifiers will make a difference to end result.

1. Start a new Blender file, then, to set the screen up with a 3D Viewport, Geometry Node Editor and Spreadsheet, use the Active Workspace menu at the top of the window and select **Geometry Nodes**. Delete the start cube and add a plane mesh

object. In the Adjust Last Operation pane change the Size to **20** m.

2. In the Modifier Properties tab add a **Geometry Nodes** modifier, then click on **+ New** to create a group for this object. There will be no apparent change in the 3D Viewport but a simple node tree will appear in the Geometry Nodes Editor.

Mesh Objects Created Within the Node Tree

3. The first divergence from the workflow using particles is the ability to create meshes within the node tree. For more complicated meshes, such as the tufts of grass, it is better to pull the geometry into the node tree from external objects, but for simple shapes such as rocks, they can be generated procedurally. From the menu in the Geometry Nodes Editor select **Add ▸ Mesh Primatives ▸ Ico Sphere** and drop the new node onto the pipe running form the input to the output node.

There are a couple of things to note here. The original plane has vanished, replaced by the ico sphere, and you can see that the input node is now disconnected from anything. In order to see the plane as well as the rock the two geometries need to be joined back together, which you will do later using the Join Geometry node. The other thing you may have noticed is that, although there are some similarities, the options under Add in the menu are much more extensive and arranged differently from the options for material nodes or compositor nodes. Although I will give explicit menu selections in my instructions for this exercise, you will find that you need to use the search option when you are working on your own. Another useful feature that was added in Blender version 3.1 is the ability to drag a colored connecting dot on the side of a node into a space, when released it will present you with a list filtered by compatibility.

4. You are now going to add some nodes to make the regular ico sphere a little more random as shown in Figure 14-1. First of all add a new node by selecting **Add ▸ Geometry ▸ Set Position** and drop it onto the output connector from the ico sphere node. Notice that there are three other diamond shaped inputs to the new node which indicate that they use a different type of data from the circular inputs and outputs. The diamond connections relate to a finer level of detail within the geometry, in this case each vertex in the mesh can be selected, positioned or offset individually. The introduction of this mechanism was a major change from the first release of geometry nodes and, as a result,

much of the online help is rather confusing. Hopefully, in time, this situation will remedy itself.

5. Add a node to create a random offset using **Add ▸ Utilities ▸ Random Value**, change the Type to **Vector**, the Min to **-0.2** and the Max to **0.2**, then connect it to the Offset input of the set position node. Finally position a node to transform the resultant mesh, after the set position node, using **Add ▸ Geometry ▸ Transform**, changing the Scale value in the Z direction to **0.6** to flatten the rock.

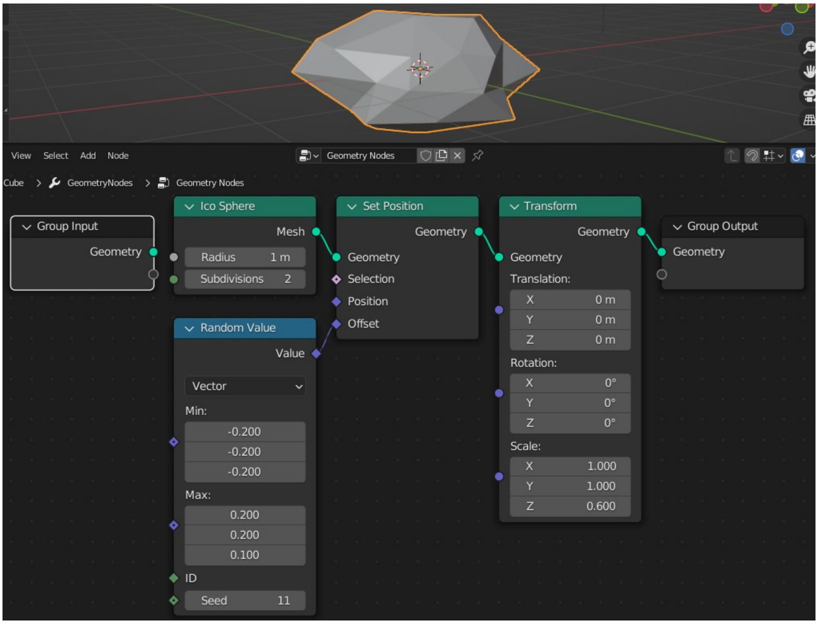


Figure 14-1 Using geometry nodes to create a rock

6. In Figure 14-1 you can see that I increased the ico sphere subdivision to 2. Try doing that then change the Seed value in the random value node to see the variety of results. A better alternative is to return the ico sphere subdivisions to 1 then insert a new subdivision surface node after the set position node using **Add ▸ Mesh ▸ Subdivision Surface**.

7. Before looking into how the rocks can be distributed over a surface you need to learn a little more about the usage of data relating to vertices within a mesh. At the moment the surface of the rock is flat shaded and this cannot be changed to smooth by toggling the setting on the original plane object. However, there

is a node that can be used so add that after the transform node using **Add ‣ Mesh ‣ Set Shade Smooth**.

8. For the sake of this exercise, imagine that, because of the extreme temperature difference between night and day on Mars, only the upward facing surfaces should be smooth, while the downward facing surfaces remain flat. This feat can be achieved relatively easily using nodes. The first node you need is one that can generate data about the direction that each face is pointing. As you have seen before, this property is the normal direction, so select **Add ‣ Input ‣ Normal**. It may seen unclear how this general sounding node can know which mesh it is supposed to providing the data for; and, if you mouse over the output of the node you will see the prompt "The socket value has not been computed yet". This is the clever aspect of the geometry nodes system in its final evolution, the specific sources of data are only set once the node has been connected to other nodes.

9. In order to create a selection criterion based on the normal direction you need to find whether the Z component of the vector is positive which you can do by adding two more nodes by selecting **Add ‣ Vector ‣ Separate XYZ** and **Add ‣ Utilities ‣ Compare** setting the Type to **Float**, Operation to **Greater Than** and the second input Value to **0.0**. Connect the Result output to the Selection input of the set shade smooth node. The layout is shown in Figure 14-2

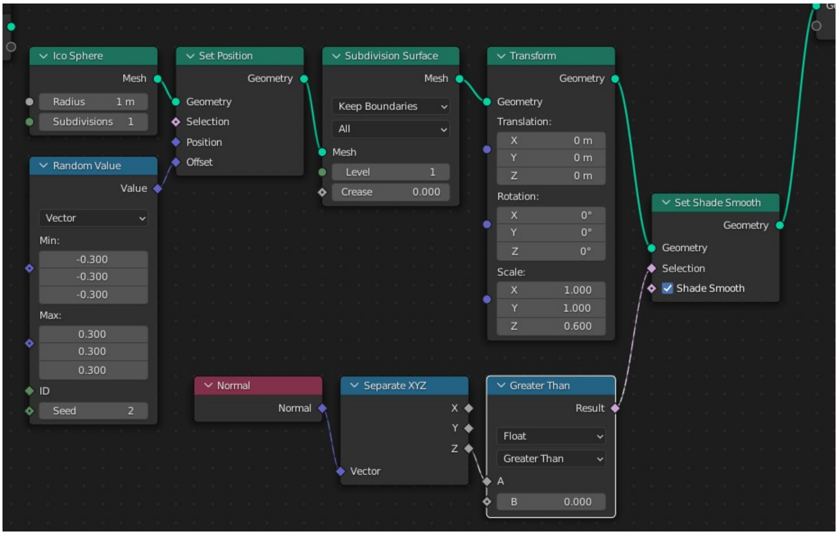


Figure 14-2 Applying selection criteria to a node

10. You've now made a set of nodes that can be converted into a group, and used to generate a variety of rocks for scattering over the terrain. Box select all the nodes apart from the input and output, then hit **Add ▸ Group ▸ Make Group**. The editor is now showing the node tree for the new group, so, to give control over the rock being generated by the group, drag the Scale input from the transform node to the unconnected socket of the input node and drag the Seed input from the random value node to the next unconnected socket. Return to the parent node tree using the up arrow in the header bar and change the name from NodeGroup to something meaningful such as rock_generator.

11. Now it's time to construct the mechanism for scattering the rocks over the plane. First of all disconnect the rock_generator node from the output node and reconnect the input Geometry output, you should now see the original plane again. Insert a node to generate locations for the rocks using **Add ▸ Point ▸ Distribute Points on Faces** and, provided the viewport shading is still set to solid you will see a large number of diamonds appear. Insert a second node to create instances at each point with **Add ▸ Instances ▸ Instances on Points**, at which point all the points will vanish. Finally connect the Geometry output of the rock_generator node to the Instance input of the instances on points node.

12. There are two obvious problems with this initial result; the rocks are too large and they overlap each other, so, before introducing variation in shape, rotation and size, you should correct this first setup. Change the Scale values in the rock_generator node to **0.1**, **0.1** and **0.06** , and change the scattering method on the distribution points on faces node from Random to **Poisson Disk**. Now gradually increase Density Max value of the same node until there are no gaps. Many of the rocks will be overlapping each other again, but Poisson disk distribution has another variable that can stop that happening; gradually increase the Density Min until all of the rocks just stop overlapping.

13. At this stage you need to combine the original plane with all the generated rocks, you want to produce a variety of rocks, the rocks should be rotated randomly about their Z axis and that axis should be rotated so that each rock sits flat on the surface. The latter point is not a problem while the plane is flat but for an undulating surface, it will be important. Insert a node after

the instances on points, using **Add** ▸ **Geometry** ▸ **Join Geometry** then drag the Geometry output from the group input node to the join geometry node. Notice that there is a single input to the join node but it expands to take as many geometries as you want.

14. You should now see rocks spread over the original plane, but they are embedded too far in. Open the rock_generator group and in the random value node change the third Min value to **0.2** and the third Max value to **0.8**. This will bias the Z offset upwards. Return to the parent node tree and duplicate the rock_generator four times using SHIFT-D then insert a node onto the connection from the rock_generator node leading to the Instance input of the instance on points node using **Add** ▸ **Geometry** ▸ **Geometry to Instance**. Tick the Pick Instance check-box on the instance on points node then connect all of the rock_generator outputs into the geometry to instance node. Finally change the scale and seed values for each of the rock_generator nodes and you should see the scattered instances change too.

15. To make the rocks lie flat on the surface connect the Rotation output of the distribute points on faces node to the Rotation input of the instances on points node. Then, to make the rocks rotate randomly around their Z axis insert a rotate node into the connection you just made, select **Add** ▸ **Utilities** ▸ **Rotate Euler** and change its method of rotation to **Axis Angle** with the Axis **0.0**, **0.0**, **1.0**. You now need to feed a random value into the Angle input of the rotate Euler node, use **Add** ▸ **Utilities** ▸ **Random Value**, setting the Min to -**pi** and the Max to **pi**. The node tree and resulting 3D view are shown in Figure 14.3

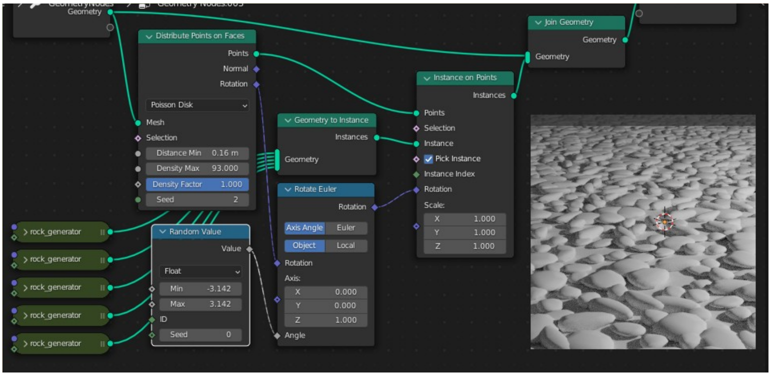


Figure 14-3 Scattered rocks on a plane.

Image Textures in Geometry Nodes

To finish off this exercise introducing geometry nodes you will add textures, allowing you to paint hills and valleys onto the plane object, as well as painting the density of rocks and boulders. In many ways the process is simpler than using the particle system but there are one or two quirks that are hard to figure out from scratch.

1. Although you know how to use normal modifiers to subdivide and displace a mesh, you will now do that process within the geometry node tree. Select **Add ▸ Mesh ▸ Subdivide Mesh**, add the node to the Geometry output pipe from the input node and increase the subdivision Level to **4**. Add a second node after the subdivision using **Add ▸ Geometry ▸ Set Position** this node will use the Z offset from an image texture.

2. It would be possible to feed image data directly into the Offset input of the set position node and the red value of a pixel would be used for the X offset, the green for the Y offset and the blue for the height. However the standard practice is to draw the height texture in gray-scale, so a better option is to use a combine XYZ node to confine the offset to the Z axis only and to use a multiplication node to allow the height scale to be controlled. In a space to the left of the node tree position a node using **Add ▸ Texture ▸ Image Texture** click **New** to create a new image and set the Color to mid gray, to allow valleys as well as ridges, with HSV **0.0**, **0.0**, **0.5**, un-tick the Alpha check-box and rename the image to height_map. Create a second node with **Add ▸ Utilities ▸ Math**, change the mode to **Multiply**, use the Color output of the image texture node as one input and change the other value to **10.0**. Create the third node with **Add ▸ Vector ▸ Combine XYZ**, set the X and Y values to **0.0**, connect the output of the multiply node to the Z input and the output vector to the Offset input of the set position node.

3. At this point the plane will have probably floated up above the rocks which has happened because both the distribute points on faces node as well as the final join geometry node both need to take their input from the same displaced mesh. Reconnect the nodes so they match the arrangement shown in Figure 14-4. Now it’s time to paint some terrain.

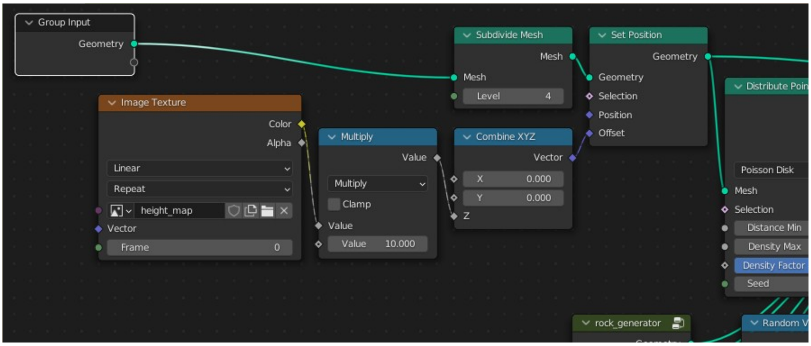


Figure 14-4 Using an Image Texture to displace a plane.

4. In order to see what you're painting set a material for the plane object in the Shader Editor add an image texture node using the height_map image as the base color. You also need to change the viewport shading to Material Preview. Make sure that the image texture node is selected in the shader then, using the workspace tabs at the top of the screen switch to **Texture Paint**. As soon as you start to paint you will see that the UV map used in geometry nodes is incorrect, this is because the UV data has to be fed into the node tree as an additional input.

5. Switch the workspace back to Geometry Nodes and drag a link from the Vector input of the image texture node for height_map, back to the empty socket on the group input node. In the Modifier Properties tab on the right, you will see that a new input variable has appeared here called Vector with a spreadsheet icon and three components all set to 0.0. The icon is a button that toggles between a single value and an array of data, click on the button then click in the empty input field that appears next to it, and select **UVMap**. To give the variable a more meaningful name go back to the Geometry Node Editor, select the group input node then open the Sidebar and in the Group tab, change the Name from Vector to uv_map. If the node tree was intended for reuse then you would connect other variables to the input, such as the scale factor for the height offset. The interpretation of the UV map will give confusing results if the mesh is not subdivided enough, so watch out for this.

6. Now, when you change back to texture painting you should see the terrain rising or falling with the value of the color you apply, the rocks rotating to rest flat against the surface however the

surface slopes. To add an image texture for controlling the density of the rocks, switch to editing the geometry node tree and duplicate the the height_map texture node, then create a new texture for it by clicking on the little button with an overlapping squares icon next to the image name. Set the default color of the new texture as black and name it rock_density. Connect the Color output directly to the Density Factor input of the distribute points node and connect the uv_map from the input node to the Vector input of the rock_density image texture node. Go back to texture painting and choose the rock_density as the referenced image, paint where you want rocks to appear. Crucially, when you have finished painting, remember to click the **Save All Images** button!

There are so many different things you can do with geometry nodes that it is hard to limit the scope of this exercise, but from now on you should be able to find your way a little more easily, you will also be able to follow the excellent examples available online.

Grease Pencil

Blender's grease pencil is a bit of an oddity. In many ways it can be viewed as a rather clunky 2D drawing application, except that it draws in the normal 3D space, with which you are now familiar. Also, once you understand how grease pencil works, your general knowledge of Blender converts the clunkyness into flexibility and productivity as you can use many methods you already know from editing, sculpting and modifiers.

Using grease pencil for 2D, hand-drawn, animations is one of its big selling points and there is much information available online for using the specialist tools such as onion-skinning and interpolation. However, in this chapter, I will concentrate purely on basic steps needed for creating a single image, which is a prerequisite skill for anyone setting out to make an animation.

Strokes

The grease pencil object consists of a number of strokes, each of which is a series of points. The points can be shown either as lines connecting the positions of each point, or as dots, or as image textures. Each point along a stroke has a size and opacity and the points can be used to define the outside of a polygon which can be filled with a color.

As well as being situated in the 3D space of the Blender scene, there are many settings to control the material, the brush, the layer and vertex color of strokes. Learning where the settings are, and how to use them will the main focus of this exercise.

The Drawing Plane

Before starting to draw, it's a good idea to set up Blender to make your life as easy as possible. To make everything explicit I will go through all the steps, starting from a new general purpose Blender file.

1. Open a new file and delete the default cube. Set the view to orthographic along the Y axis by clicking on **-Y** on the gizmo. Set the camera to match this view so you can easily return here later, select **View ▸ Align View ▸ Align Active Camera to View**. In order to see the colors as you draw them you need to set the viewport shading to **Rendered** and, in the World Properties tab, under the Surface section change the Color to white. Finally, select the light object and in the Object Data Properties tab under the Light section change the type to Sun and the Strength to **3.0**. In the 3D Viewport rotate the direction of sun light so it shines onto the XZ plane as viewed from the camera. Later, when you learn about grease pencil layers, you will find that there is an option to use the actual stroke color without being effected by lights, but for the moment, this setup will provide a good starting point.

2. After rotating the light return to the camera view then add a new empty grease pencil object by clicking **Add ▸ Grease Pencil ▸ Blank**. Switch to Draw mode, but before you start drawing anything look up at the header where you will see a selection of new options. For the moment just concentrate on the two that define the location of strokes, the first shows a set of axes and the word Origin, the second show a computer monitor and the word View. Change View to **Front (X-Z)** to ensure that strokes are laid down as if drawing onto paper held in the X-Z plane through the origin.

Brushes

3. You will now draw three different lines in order to see the first place you can control grease pencil strokes. With the mouse LMB held, draw a rough circle, don't leave a gap between the start and the end as that will prevent the flood fill working later.

Notice that the line is rather fine and not completely opaque. To make the line broader press F while the cursor is over the 3D Viewport, then increase the diameter of the brush and draw a second closed loop overlapping the first. From Blender version 3.5 onward, the line that is drawn matches the diameter of the brush, but on previous version you just have to estimate the size of the brush.

4. To change other features of the brush, open the Active Tool and Workspace settings tab. Although all these controls are duplicated in the editor header, the layout there is cramped and some options are rather cryptic so it will easier, to start with, using the tab on the right. You will see that the default brush is the Pencil, expand the Brush Settings section below the brush image and check how it is set up. The first variable is the Radius which you increased before drawing the last stroke. Notice also that the Use tablet pressure button, to the right of the value, has been enabled and a graph is shown below. If you use a tablet to draw, then the harder you press the thicker the line, up to a maximum at the value of the Radius field. Below the Radius value is the Strength, which controls the alpha, or opacity of the stroke. For the pencil brush, once again, the opacity is pressure controlled but note that the maximum value, which will be used when drawing with the mouse, is 0.6. If you look where the lines you drew overlap you will be able to see the first one through the second.

5. To draw the third line click on the pencil icon and choose **Ink Pen Rough** then in the Brush Settings section change the Radius to **200**. The thickness of this brush has non-linear sensitivity to tablet pressure but the Strength is 1.0, you will not be able to see through it. Draw another closed loop overlapping the other two.

Materials

6. Below the Brush Settings is the Color section, where the color picker is set by default to light green. However the lines you drew were black because the default setting is to use Material, change that now by clicking **Color Attribute** then draw a fourth closed loop overlapping the other three.