IN THIS CHAPTER

- Understanding the basics of animation
- Using the Timeline and Graph Editor for object animation
- Discovering how to animate and insert keyframes in Blender

Chapter 1
Animating Objects

I have to make a small admission: Animation is not easy. It's time consuming, frustrating, tedious work where you often spend days, sometimes even weeks, working on what ends up to be a few seconds of finished animation. An enormous amount of work goes into animation. However, there's something incredible about making an otherwise inanimate object move, tell a story, and communicate to an audience. Getting those moments when you have beautifully believable motion — life, in some ways — is a positively indescribable sensation that is, well, indescribably positive. Animation and the process of creating animation truly has my heart more than any other art form. It's simply my favorite thing to do. It's like playing with a sock puppet, except better because you don't have to worry about whether or not it's been washed.

This chapter, as well as the rest of the chapters in Book 4, go pretty heavily into the technical details of creating animations using Blender. Blender is a great tool for the job. Beyond what this book can provide, though, animation is about seeing, understanding, and re-creating motion. I highly recommend that you make it a point to get out and watch things. And not just animations! Go to a park and study how people move. Try to move like other people so that you can understand how their weight shifts and how gravity and inertia compete with and accentuate those movements. Watch lots of movies and television and pay close attention to how people's facial expressions can convey what they're trying to say. If you get a chance, go to a theater and watch a play. Understanding how and why stage actors exaggerate their body language is an incredibly useful tool for an animator.

REMEMBER

While you're going out there seeing the world like an animator, think about how you can use the technical information in these chapters to re-create those feelings and that motion with your objects in Blender.

Getting Started with Animation in Blender

Before you jump headlong into the world of animating with Blender, it's helpful to understand the general workflow for animation. Then you can see how that workflow applies to Blender. If you recall from Book 1, Chapter 2, Blender has five different editors dedicated to the process of animation. With all those editors available, it might be difficult to guess which one you should be using. Fortunately, this is where knowledge of animation workflows comes in handy. Generally speaking, when you animate, you work in multiple passes, focusing on more detailed movement (and nuances within that movement) with every iterative pass. The basic passes are as follows:

1. **Planning:** This step is frequently done outside of Blender, often with traditional pen and paper. You create a series of drawings that show the key shots in your animated sequence, called a *storyboard*. With a storyboard complete, you have a much clearer understanding of what's going to happen in your animation. You can take the images of your storyboard and assemble them sequentially in a video editor (or Blender's Video Sequencer — see Book 5, Chapter 1 for more on the Video Sequencer) to create a moving storyboard, called an *animatic*. It's also possible to create your storyboard and animatic all within Blender using Grease Pencil objects (see Book 4, Chapter 6) using the built-in Story Pencil add-on, but that topic is a little bit out of scope for this book.

2. **Blocking:** With your plan in place, the next step is *blocking*, where you rough out the biggest movements in your animated sequence. The majority of your time for this step is spent in the 3D Viewport getting things positioned correctly for each major keyframe. The rest of your time is spent in the Dope Sheet, getting the rough timing correct by sliding those keyframes forward and backward in time.

3. **Refining:** With your blocking pass complete, your next step is *refining* your animation. This is the step where you add the little bits of secondary movement that really push life into the objects you're animating. You're still spending a lot of time in the 3D Viewport to work on those keyframes; you're just keying more things and adding more in-between keys to get that refined movement. And again, you're spending the rest of your time in the Dope Sheet where you can adjust the timing on your actions. Often you will have multiple refining passes before you're really happy with the results.

4. **Final Polish:** It's in *final polish* where you really start to use the Graph Editor. This step is where you focus on the nuanced changes between frames. Your focus is tweaking the rate of change, how quickly or slowly the movement happens from one keyframe to the next. Even though this is the final step in the animation process, you often end up spending a *lot* of your time here.

If you're coming to Blender from another 3D animation package, this approach might seem a bit alien. Animators in other software often focus on working exclusively in the Graph Editor. Part of that is because as much as 80 percent of your time as animator is spent in final polish (especially if someone else is doing the planning step). However, another reason, in my opinion, is that the developers of those other packages focused the majority of their attention on the Graph Editors, so their equivalent of the Dope Sheet simply isn't as pleasant to work in as Blender's. And for that reason, many animators choose to spend their time in the Graph Editor in other applications where the more developed tools reside. There's nothing wrong with that workflow, of course, but with all the curves visible in the Graph Editor (regardless of how nice the Graph Editor is), it's often difficult to get a nice high-level understanding of your animation's timing. So it's really beneficial to become friends with the Dope Sheet.

That said, this chapter focuses mostly on the Timeline and Graph Editor because, if you're just starting as an animator, it's helpful to have a visual indicator that something is changing between keyframes, even if you don't know the best way to change it. The curves between your keyframes in the Graph Editor serve this purpose really well, and should help you get a better understanding of how things work in computer animation more quickly. Furthermore, since this chapter focuses on *object animation*, or animating the objects in your scene (rather than controls set up in a rig — see Chapters 2 and 3 in this book for more on rigging), the blocking and refining steps can be done from the Timeline. Then you can jump right to final polish in the Graph Editor. For a more complete look at the animation process, including the use of the Dope Sheet, have a look at Chapter 4 within this book.

Creating the illusion of motion with static images

To effectively do animation, you really need a firm grasp on how moving pictures work. This is the fundamental basis of animation, film, television, and even video games. Without getting into the complex details of psychology and neuroscience, the basics are this: Humans perceive movement as the apparent difference in what we see at two relatively close moments in time.

If you have two images of the same object in different positions, you can create the illusion that the object moves between positions by swapping between those images very quickly. Now, chain a series of those images together and show them

in quick succession. Each image is visible only for a fraction of a second before the next one appears. This rapid swapping of images tricks our minds into seeing movement on the screen.

Who said science isn't fun?

Customizing your screen layout for animation

The Blender workspace specifically set up for animation, named — conveniently — Animation, can be chosen from the workspaces tabs at the top of the Blender window or by pressing Ctrl+Page Up or Ctrl+Page Down until you land on the Animation tab. Figure 1-1 shows the default Animation workspace in Blender.

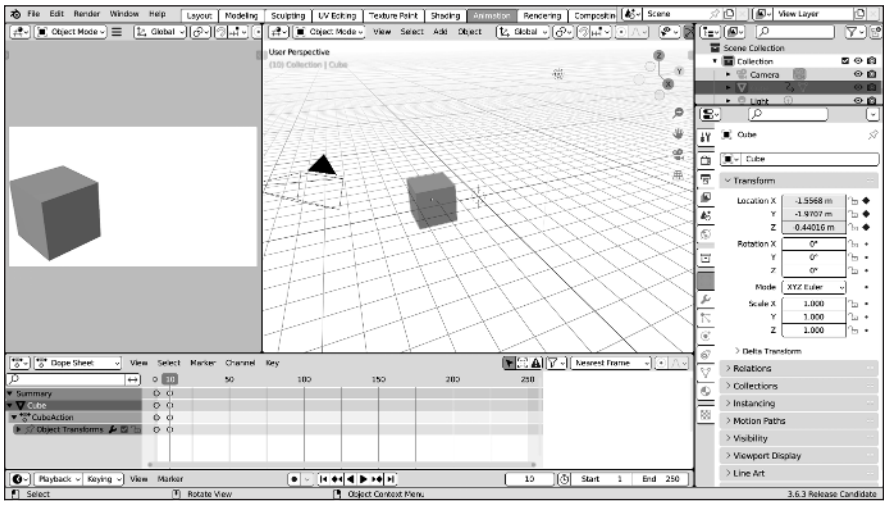

FIGURE 1-1: Blender's Animation workspace is an excellent place to animate.

This workspace isn't too dissimilar from the Layout workspace, but it does have a few differences. In particular, the 3D Viewport is split to show, on the left, a camera view of your scene with all the non-renderable elements turned off, giving you a clean preview of your animation. The Timeline is compressed at the bottom of the window, so all you see are the playback buttons in its header, and above the Timeline is a Dope Sheet editor. (For a full reminder of what each editor does, refer to Book 1, Chapter 2.)

You may be wondering, however, why this workspace has the compressed Timeline and why it's missing the Graph Editor. Recall from the beginning of this section that the animation workflow that works well in Blender is set up so you

shouldn't really need all the different editors visible at the same time. For example, the Timeline and the Dope Sheet have very similar high-level visibility of your animation, and because both of these animation editors have the same scrubbing ability, you only really need the playback buttons from the Timeline. So, in this layout, the Graph Editor isn't really missing. It's just not as necessary when you begin an animation because the workflow typically starts with the Dope Sheet editor.

That said, since this chapter focuses on using the Timeline and the Graph Editor, you may choose to switch the Dope Sheet at the bottom of the Animation workspace to a Graph Editor.

Animating your first scene

It's easiest to understand the animation process by going through a simple example. The following example shows the basic process of animating in Blender and should help explain things more clearly:

1. **Start in Blender with a new General workflow template (File ➪ New ➪ General).**
2. **Select the default cube object and switch to the camera view by clicking the camera icon in the upper right of the 3D Viewport.**
3. **Split the 3D Viewport horizontally and change one of the new areas to the Graph Editor (Shift+F6).**

   I like to make the lower one the Graph Editor because it's right above the Timeline in the Layout workspace.

4. **In the 3D Viewport, make sure that the default cube is selected and choose Object ➪ Animation ➪ Insert Keyframe ➪ Location (you can also use hotkeys for faster access by pressing I ➪ Location).**

   This step sets an initial location keyframe for the cube. I give a more detailed description of keyframes later in the chapter in the "Understanding keyframes" section. The important thing to notice is that the left region of the Graph Editor is now updated and shows a short-tiered list with the items `Cube`, `CubeAction`, and `Object Transforms`. There's also a keyframe diamond at frame 1 in the Timeline.

5. **Move forward in time by pressing → a handful of times.**

   I'd suggest moving forward about ten frames, or so. You can tell what frame you're on in the 3D Viewport by looking in the upper left corner next to the tool buttons. The second line of text there should say something like `(10) Collection | Cube`. The number in parentheses is your current frame.

6. **Grab the cube in the 3D Viewport and move it to another location in the scene.**

   You can either use the Move tool or use the G hotkey to immediately start moving the cube without the need of a transform gizmo.

7. **Insert another keyframe (I ➪ Location) in the 3D Viewport.**

   A set of colored lines appears in the Graph Editor. These colored lines are *f-curves*. F-curve is short for *function curve*, and it describes the change, or *interpolation*, between two key moments in an animated sequence. Each f-curve represents a single attribute, or *channel*, that's been animated. If you expand the Object Transforms block in the Graph Editor by clicking the triangle on the left side of its name, you can see the three location channels: X Location, Y Location, and Z Location. To see the actual curves a little bit better, move your mouse to the large graph section of the Graph Editor and press Home. Your Blender screen should look something like the one in Figure 1-2.

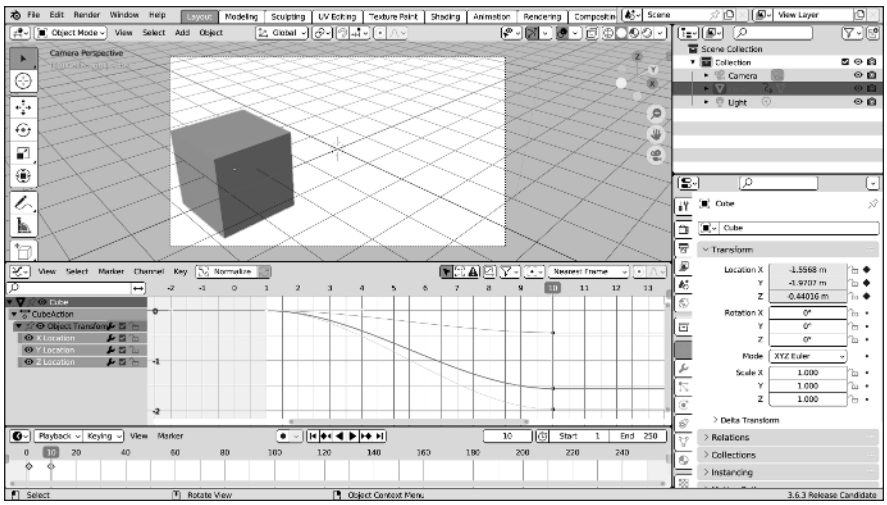

FIGURE 1-2: Animating the location of the default cube object.

Congratulations! You just created your first animation in Blender. Here's what you did: The largest part of the Graph Editor is a graph (go figure!). Moving from left to right on this graph — its X-axis — is moving forward in time. The vertical axis on this graph represents the value of whatever channel is being animated. So the f-curves that you created describe and control the change in the cube's location in the X-, Y-, and Z-axes as you move forward in time. Blender creates the curves by interpolating between the control points, called *keys*, that you created.

To understand interpolation better, flash back to your grade school math class for a second. Remember when you had to do graphing, or take the equation for some sort of line or curve and draw it out on paper? By drawing that line, you were interpolating between points. Fortunately, you don't do any of that graphing work when animating. That's what we have Blender for!

You can see the result of your work on this example for yourself by playing back the animation. Click the Play button in the Timeline (alternatively, you can press Spacebar from just about anywhere in Blender's interface). A green vertical line, called the *timeline cursor*, moves from left to right in the graph. As the timeline cursor moves between your keyframes, you should see your cube move from the starting point to the ending point you defined in the 3D Viewport.

To stop the playback, you can either press Spacebar again or you can press Esc. If you press Spacebar, the timeline cursor stops in place. If you press Esc to stop, the timeline cursor jumps back to the frame you were on when you first started playing. You can watch the animation in a more controlled way by *scrubbing* in the Graph Editor or Timeline. If you look at the top of each of these editors, you should see a slightly darker region with numbers running horizontally; this is the scrubbing region. To scrub, left-click in the scrubbing region of either the Graph Editor or Timeline and drag your mouse cursor left and right. The timeline cursor follows the location of your mouse pointer, and you can watch the change happening in the 3D Viewport.

Understanding keyframes

As I cover earlier in this chapter, in traditional animation, a whole animated sequence is planned out ahead of time using a script and *storyboards* (drawings of the major scenes of the animation, kind of like a comic book, with arrows to indicate camera direction or character movement). Then an animator goes through and draws the primary poses of the character. These drawings are referred to as *keyframes*, or *keys*. They're the poses that the character must make to most effectively convey the intended motion to the viewer. With the keys drawn, they're handed off to a second set of animators called the *inbetweeners*. These animators are responsible for drawing all the frames between each of the keys to get the appearance of smooth motion.

Translating the workflow of traditional animation to how you do work in Blender, you should consider yourself the keyframe artist and Blender the inbetweener (at least to start). On a fully polished animation, it isn't uncommon to have some kind of key on every frame of the scene. In the quick animating example in the preceding section, you create a keyframe after you move the cube. By interpolating the

curve between those keys, Blender creates the in-between frames. Some animation programs refer to the process of having software create these in-between frames as *tweening*.

Inserting keys

To have a workflow that's even more similar to traditional animation, it's preferable to define your keyframes in the 3D Viewport. Then you can use the Timeline, Dope Sheet, or Graph Editor to tweak the change from one keyframe to the next. And this workflow is exactly the way Blender is designed to animate. Start in the 3D Viewport by pressing I to open the Insert Keyframe menu (you can also choose the slower route of choosing Object ➪ Animation ➪ Insert Keyframe). Through this menu, you can create keyframes for the main animatable channels for an object. I describe the channels here in more detail:

- **Available:** If you already inserted keys for some of your object's channels, choosing this option adds a key for each of those preexisting curves in the Graph Editor. If no curves are already created, no keyframes are inserted. Once you start animating, this is often a really good shortcut to have.
- **Location:** Insert a key for the object's X, Y, and Z location.
- **Rotation:** Insert a key for the object's rotation in the X-, Y-, and Z-axes.
- **Scale:** Insert a key for the object's scale in the X-, Y-, and Z-axes.
- **Transform combinations:** The next set of options insert keyframes for various combinations of the previous three values. Your choices are as follows:
  - Location & Rotation
  - Location, Rotation, & Scale
  - Location, Rotation, Scale, & Custom Properties: This option includes setting keyframes for custom properties that you set on your selected object. Custom properties are covered in more detail in the next chapter, as well as in Book 6, Chapter 2.
  - Location & Scale
  - Rotation & Scale
- **Delta Location/Rotation/Scale:** The preceding keying options use absolute coordinates for keying animations. *Delta* keys, however, use coordinates relative to your object's current location, rotation, and scale. Delta keys can be useful if you want to have many objects with the same animation, but starting from different origin points. Of course, if you choose Available after choosing Delta keys, Blender is smart enough to know that you want to use relative rather than absolute coordinates.

» **Visual Transforms:** Insert keyframes for location, rotation, scale, or any combination of these transforms, but based on where the object is visually located in the scene. These options are explicitly made for use with constraints, which I cover in the next chapter.

The basic workflow for setting keyframes in Blender follows these steps, regardless of whether you're animating objects, properties, or controls for an animation rig:

1. **Insert an initial keyframe (I or Object ➪ Animation ➪ Insert Keyframe).**

   A keyframe appears at frame 1 in your animation (assuming you're at frame 1 when you insert your keyframe).

2. **Move forward 10 frames (press → ten times).**

   This puts you at frame 11. Of course, when doing a real animation, your keys aren't always going to be ten frames apart, but this is a good way to rough in your keys. You can go back later and adjust your timing from the Timeline, Dope Sheet, or Graph Editor. To move forward or back one frame at a time, use the ← and → keys. Of course, you can also use the Timeline, Dope Sheet, or Graph Editor to change what frame you are in.

3. **Grab your object and move it to a different location in 3D space using the Move tool.**

   Again, this example is assuming that you're animating your object's location. The same steps apply if you're animating rotation, scale, or some combination of transforms.

4. **Insert a new keyframe (I).**

   Now you should have f-curves in the Graph Editor that describe the motion of the cube.


TIP

You can insert keys in an easier way using a feature called Auto Keying. Like its name indicates, Auto Keying automatically creates keys when you make changes in your scene. To enable the Auto Keying feature, look in the Timeline. To the left of the playback controls in the Timeline's header is a button with a circle icon on it, like the Record button on a DVR. Left-click it to activate Auto Keying. Now, you can simply use the transform tools in the 3D Viewport as you move forward in time and keyframes are automatically inserted for you. Pretty sweet, huh?


REMEMBER

By default, Blender uses the Location, Rotation, & Scale keying set (there's more on keying sets later in this chapter) for inserting keyframes when Auto Keying is enabled. This is worth noting because, if you're not careful, Auto Keying can insert keyframes that are unnecessary or (worse) not wanted. It's for this reason that when I use Auto Keying, I like to insert my initial keyframes manually. Then I use

the Available keying set for auto keying. The next section covers keying sets and how to choose which one you're animating with.

Animating [almost] anything

A really cool feature in Blender is the concept of "[almost] everything animatable." You can animate nearly every setting or attribute within Blender. Want to animate the skin material of your characters so that they turn red with anger? You can! Want to animate the Count attribute in an Array modifier to dynamically add links to a chain? You can! Want to animate whether your object appears in wireframe or solid shading in the 3D Viewport? Ridiculous as it sounds, that, too, is possible!

So, how do you do this miraculous act of animating any attribute? It's amazingly simple. Nearly every text field, value field, color swatch, drop-down menu, check box, and radio button is considered a property and represents a channel that you can animate. If you look to the right of any of these fields, you should see a small circular dot. Click that dot, and it changes to a larger diamond shape as shown to the left margin. When you do that, the field next to your once-circle-now-diamond icon should change color, indicating that you've just set a keyframe for that value. You can also insert a keyframe for a property by right-clicking it and choosing Insert Keyframes from the menu that appears. Figure 1-3 shows what this menu looks like. If the property is already keyed on the current frame, the right-click menu says Replace Keyframes.

REMEMBER

Don't make the mistake of thinking that a property isn't animatable if it doesn't have a keyframe icon next to it. The keyframe icon is included as a convenience feature for frequently keyed properties. If you want to insert a keyframe for a property and you don't see the keyframe icon, try right-clicking to see if inserting a key is an option.

Whether you use the keyframe icon or right-click the property, after you insert a new keyframe for that property, its color changes to yellow (in the default theme), and a channel for that property is added in the Graph Editor and Dope Sheet. When you scrub forward or backward in time, the color of the control changes from yellow to green and the diamond icon changes from solid-filled to just an outline. The green color and different icon indicate that the property is animated, but you're not currently on one of its keys. Insert another keyframe for that property, and BAM! The field's color turns to yellow, the diamond fills in again, and you now have two keyframes for your animated property.

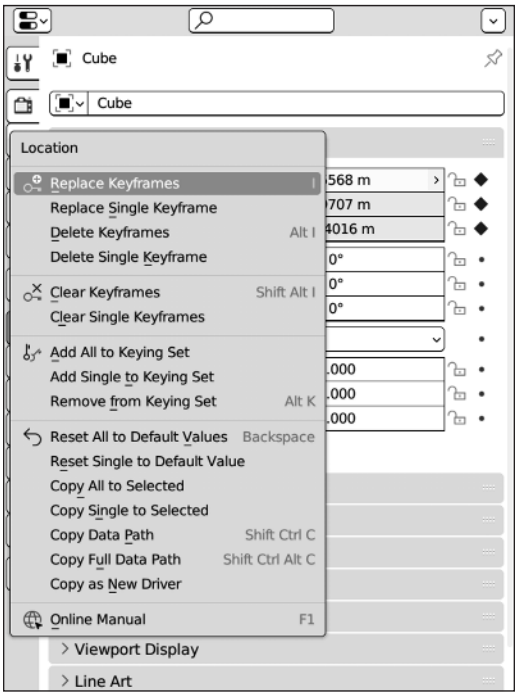

FIGURE 1-3: Right-click any property in the Properties editor to insert a keyframe for it.

TIP

For an even faster way to insert keyframes on properties, hover your mouse over the property and just press I. This trick even works on the renderability icons (known as the *restrict columns*) in the Outliner if you make them visible. How's that for awesome?

TIP

If you want to insert keyframes on multiple properties at the same time (such as the aforementioned restricted columns in the Outliner), keep holding the I hotkey as you run your mouse cursor over those properties. There you go. One key press, many keyframes. An extremely talented Blender animator I know likes to call this "painting Blender yellow."

Working with keying sets

When working on an animation, you can easily find yourself inserting keyframes for a bunch of different channels and properties at the same time. The troublesome thing is that, on a complex scene, these properties can be located all over Blender's interface. Sure, you can insert keys for an object's location, rotation, and scale by pressing I in the 3D Viewport, but you may also need to key that object's visibility from the Outliner, the influence of a constraint in Constraint

Properties, and the object's material color from the Shader Editor. They're all over the place! It wouldn't be difficult at all to overlook one while you work, resulting in an animation that doesn't quite behave the way you want it to.

This load can be lightened by using more advanced rigging features (see Chapters 2 and 3 in this book), but even that doesn't really solve it. A complex character rig can have hundreds of bones and controls to animate. Manually keeping track of all of them for multiple characters in a scene can be a nightmare. (I know. I've tried. We all have, at least once.)

You know what would be really nice? It would be great if you could make a custom group of a bunch of properties so you can insert a keyframe for all of them at the same time, kind of like the Location, Rotation & Scale option when you press I in the 3D Viewport. Well, guess what? That exact feature exists in Blender. It's called a *keying set*.


TECHNICAL STUFF

Actually, the Location, Rotation, & Scale option *is* a keying set. It's one of a handful of preconfigured keying sets that ship with Blender. In fact, all the options in the Insert Keyframe menu (I) are keying sets.

Using keying sets

To use keying sets, start with a look at the left side of the Timeline's header. The first button after the Editor Type menu is the Playback rollout. For the time being, you don't need to worry about the content of this rollout. The one you *are* interested in, however, is the next one, the Keying rollout. Click this button, and you should see a rollout like the one shown in Figure 1-4.

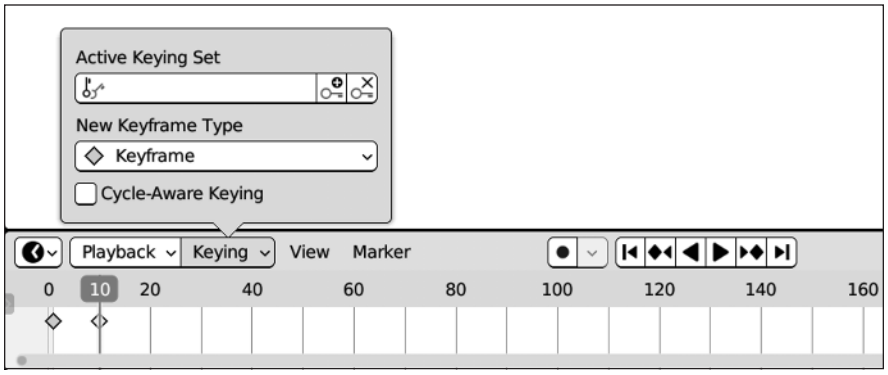

FIGURE 1-4: The Timeline's Keying rollout has controls for choosing your active keying set and inserting keyframes within that keying set.

For keying sets, the first row in this rollout is the most relevant. The Keying rollout contains three widgets for working with keying sets:

- **Active keying set:** The field for this drop-down menu is empty by default. Left-click it to choose one of the available keying sets to be your active one.

  If you have an active keying set, you can make this field empty again by left-clicking the X to the right of your keying set's name.

- **Insert keyframes:** Left-click this button to insert a keyframe at the current frame on the Timeline for every property in the current active keying set. If you don't have an active keying set, a little error pops up to let you know.

- **Delete keying set keyframe:** Left-click this button to remove any keyframes from the current frame that belong to any properties in the active keying set.

  If you don't have any keys on the current frame, this button doesn't do anything.

TIP

To quickly choose an active keying set from the 3D Viewport, use the Ctrl+Shift+Alt+I hotkey combination.

REMEMBER

When you choose a keying set, Blender gives preference to the properties in that keying set when you insert keyframes. This means that with an active keying set chosen, you don't get the Insert Keyframe menu if you press I in the 3D Viewport. Blender just quietly inserts keyframes for all the properties in your chosen keying set. This lack of immediate feedback may be disorienting at first, but it makes for a very fast, distraction-free workflow for animating. Also, as mentioned earlier in this chapter, if you have Auto Keying enabled, it uses the active keying set. If you don't have an active keying set chosen, Auto Keying defaults to using the Location, Rotation, & Scale keying set.

Creating custom keying sets

The preconfigured keying sets that come with Blender are handy, but it's also useful to define your own keying sets (especially as your animations become more complex). Keying sets aren't character- or object-specific, so they apply to your whole scene. This means you can use a keying set to insert keyframes for a whole group of characters all at the same time if you want.

Because keying sets are relevant to your whole scene, you can create your custom keying sets from the Scene tab in the Properties editor in the Keying Sets panel. Follow these steps to create a new keying set and populate it with properties to keyframe:

1. **In the Keying Sets panel, left-click the Plus (+) button to the right of the keying set list box.**

   The panel should expand with a whole host of new parameters and options you can adjust. The original empty listbox in the panel should be populated with a single entry, named `Keying Set`.

2. **Adjust your new keying set's settings.**

   Double-click its name in the listbox to rename it to something other than the generic `Keying Set` that you get by default. You can use the Description text field to write a few words that explain the purpose of your custom keying set.

   The Keyframing Settings sub-panel contains three toggleable settings under the label of General Override:

   - **Needed:** Enable this option to avoid adding extraneous keyframes to your animation. If you have a *hold*, where you've set two keyframes and there's no change between them, then as long as that property doesn't change, it doesn't make sense to add more keyframes within that hold. So by enabling the Needed override, Blender adds keyframes only to the properties that change within a hold. If the property didn't change, no keyframe is inserted, regardless of whether you've pressed I in the 3D Viewport.
   - **Visual:** As covered earlier in this chapter, *visual transforms* are the location, rotation, and scale of your object as they appear onscreen, including changes made by constraints. Enable this option and all inserted keyframes in this keying set will use visual transforms for keying.
   - **XYZ to RGB:** This option is enabled by default. It dictates that any new f-curves use red, green, and blue for X-, Y-, and Z-axis channels.


REMEMBER

You may notice that there are actually two toggle buttons next to each of these General Override options. One is the familiar check box, and the other has an icon that looks like a pen over a target. Enable this second toggle button to ensure that the associated property overrides the default keying settings in Blender (set in Edit ➪ Preferences within the Animation section) for cases where the check box is in the on or off state. It's a pretty confusing toggle to think about. However, if you want to change the check box for the property this toggle is associated with from its default state, you need to enable the default override toggle as well. There's more on overrides later in this chapter.

Figure 1-5 shows the Keying Sets panel with a few custom keying sets added in the list box.


REMEMBER

Left-clicking a keying set in the list box of the Keying Sets panel automatically makes that keying set the active one.

At this point, you've created your custom keying set, but it's an empty container. You haven't assigned any properties to it. To add, edit, and remove properties from your active keying set, you need to use the Active Keying Set sub-panel in Scene Properties. It appears within the Keying Sets panel when you have a custom keying set selected in the list box.

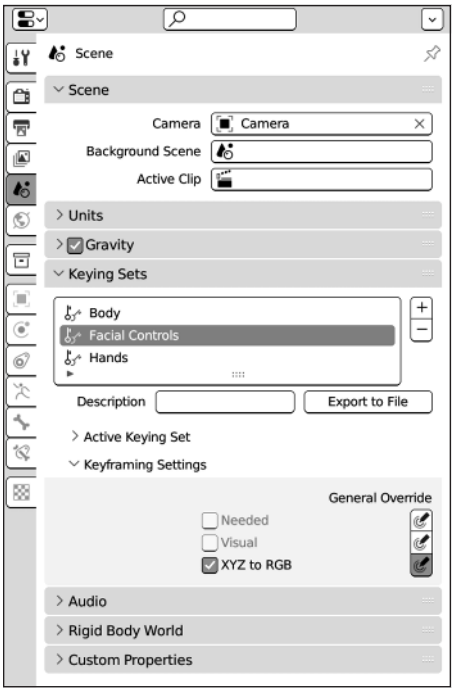

FIGURE 1-5: The Keying Sets panel is where you add new custom keying sets.

To add a new property to your custom keying set, you need to tell Blender where to find that property. You must give Blender that property's *path*, or the way to navigate through your `.blend` file's internal structure to your property. It's certainly possible (and sometimes necessary for more obscure properties) to manually add new paths from the Active Keying Set panel by left-clicking the Plus (+) button next to the Paths list box and get more specific from there. However, that can be an excruciatingly tedious process, even if you have a strong working knowledge of `.blend` file innards.

There's a better, easier way to populate your keying set. Rather than go about the painful, manual way, follow these steps:

1. **In Blender's interface, find the property you want to add to your keying set.**

   This could be the material color of your object, just its Y-axis rotation, or its renderability in the Outliner. It can be any property that's capable of being keyframed.

2. **Right-click the field, button, or check box for that property.**

   Among all the various options available in the menu that appears, there should be either two or three menu items specific to keying sets:

   - **Add All to Keying Set:** If your chosen property is one of a block of properties (such as X, Y, and Z scale), this option can add all the properties in that block to your custom keying set. If the property isn't part of such a block, this option doesn't appear in the menu.
   - **Add Single to Keying Set:** Choose this option to add just the property that you've right-clicked to your keying set.

     If the property isn't part of a block of properties, then this menu item reads as Add to Keying Set.
   - **Remove from Keying Set:** If the property in question already is in your active keying set, you can choose this option to quickly remove it.

3. **Add the property to your active keying set by left-clicking one of the Add to Keying Set menu items described in the previous step.**

   Your chosen property is now a member of the active keying set. You can verify that it's there by looking at the Paths list box in the Active Keying Set sub-panel of the Keying Set panel in Scene Properties. Figure 1-6 shows the Active Keying Set panel with a few paths added.

At the bottom of the Active Keying Set sub-panel is a drop-down menu labeled F-Curve Grouping. This drop-down menu dictates how animation channels are grouped in the Dope Sheet and Graph Editor. You have three choices:

- **Keying Set Name:** This is the default setting. Animation channels in the Dope Sheet and Graph Editor are grouped according to the name of your keying set.

  It generally works well, but if you have a lot of properties in your keying set, this can make that grouping seem a bit like a grab bag of animated properties.
- **None:** You have the option of not grouping the properties in your keying set at all. This can lead to pretty messy animation editors, so it isn't an option you'll choose often, but it's nice to know it's available.
- **Named Group:** This is a very powerful option for organizing a large keying set. If you choose Named Group as your method for f-curve grouping, a text field labeled Group Name appears below the drop-down menu. Type the name of a group in that field. Now, if you choose another property in your keying set and type the exact same group name, both properties are collected together in the Dope Sheet and Graph Editor. This is extremely useful for complex character rigs.

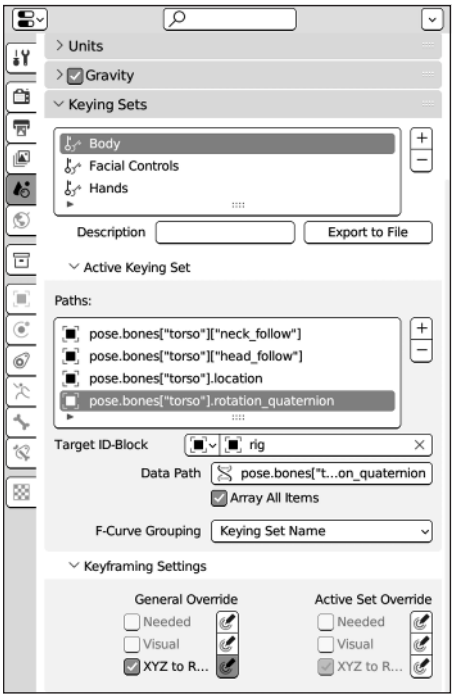

FIGURE 1-6: The properties of your active keying set are listed in the Active Keying Set panel of Scene Properties.

You may notice in Figure 1-6 that, like the Keying Sets panel, the Active Keying Set sub-panel features the same three buttons for Keyframing Settings, only now they're doubled. You have one set under the label of General Override and another under the label of Active Set Override. By adjusting these settings in the Active Keying Set panel, you override the global Keyframing Settings as defined in the Keying Sets panel, but just for that property.

WARNING

The override behavior in the Active Keying Set panel works only in an additive way. That is, if you have Needed disabled as a General Override, enabling it for a specific property as an Active Set Override will override as expected. However, if that same option is enabled as a General Override, currently there isn't a clean way to do an override that disables it for individual properties in the keying set.

Working in the Graph Editor

After you've added keyframes to your scene, the next logical step is to tweak, edit, and modify those keyframes. You can use the Timeline and Dope Sheet to adjust your general timing by selecting the keyframe diamonds you want to move and dragging them to a different frame in the editor (there's more on that in Chapter 4 of this book). But once you've blocked and refined your animation, the next step is the final polish. In that step, your focus is on the interpolation between keyframes. That editing is done in the Graph Editor.

Working in the Graph Editor is very similar to working in the 3D Viewport. Table 1-1 describes the basic controls available in the Graph Editor.

TABLE 1-1 **Basic controls in the Graph Editor**

Navigation	
Middle-click+drag	Moves around your view of the graph.
Ctrl+middle-click+drag	Allows you to interactively scale your view of the curve horizontally and vertically by moving the mouse vertically or horizontally. Diagonal mouse movement adjusts in both directions at the same time.
Scroll your mouse wheel	Zooms in and out.
Shift+scroll	Moves the graph vertically.
Ctrl+scroll	Moves the graph horizontally.
Home	Makes all available keyframes visible in the Graph Editor.
Editing	
Left-click	Selects individual f-curve control points.
Ctrl+right-click	Allows you to arbitrarily add points to a selected f-curve in the graph. Anywhere you Ctrl+right-click in the graph, a new control point is added to the selected channel.
Left-click and drag in the graph area	Box Select (pressing B also works).
A	Selects all control points.
Alt+A	Deselects all control points.
Region Controls	
Left-click and drag in the scrubbing area	Moves the time cursor.
N	Hides and reveals the Graph Editor's Sidebar (similar to the Sidebar in the 3D Viewport).
Left-click a channel in the left region of the Graph Editor	Selects the channel. You may need to expand the blocks in this region to drill down to the actual animation channels (such as X Location, Y Location, and Z Location).

The Graph Editor also has some handy keyboard shortcuts for hiding and revealing channels. They're particularly useful when you want to manage a complex animation with a lot of keyframes and channels. If you select one or more keyframes, you can use any of these hotkeys:

- **H** hides the channels that the selected keyframes belong to.
- **Shift+H** hides all channels *except* for the ones where you've selected a keyframe.
- **Alt+H** reveals ("unhides") all animation channels in the Graph Editor.

TIP

You can also click the eye icon to the left of each channel to selectively hide or unhide its curve from the Graph Editor.

Editing f-curves

The similarities with the 3D Viewport go even further than those described in the previous section. Not only can you select the control points of f-curves in the Graph Editor, but you can edit them like you would edit a 2D Bézier curve object in the 3D Viewport. The only constraint is that f-curves can't cross themselves. Having a curve that describes motion in time do a loopy-loop doesn't make any sense.

For more detailed descriptions of the hotkeys and controls for editing Bézier curves in Blender, see Book 2, Chapter 4. Selecting and moving control point handles, as well as the changing handle types by pressing V all work as expected (if you want to change handle types using the menus in the Graph Editor rather than the hotkey, choose Key ➪ Handle Type). However, because these curves are specially purposed for animation, you have a few additional controls over them. For example, you can control the type of interpolation between control points on a selected curve by pressing T or choosing Key ➪ Interpolation Mode in the Graph Editor's header menu. You get the following choices:

- **Interpolation:** These three options control the general interpolation from the current keyframe to the next, and they're the ones you'll use most frequently as an animator.

  - **Constant:** This option is sometimes called a *step function* because a series of them look like stair steps. Basically, this interpolation type keeps the value of one control point unchanged until it gets to the next one, where it instantly changes. Many animators like to use this interpolation mode when blocking out their animations. This way, they can focus on getting their poses and timing right without the distraction of in-between frames.

  - **Linear:** The interpolation from one control point to the next is an absolutely straight line. This option is similar to changing two control points to have Vector handles.

  - **Bézier:** The default interpolation type, Bézier interpolation smoothly transitions from one control point to the next. In traditional animation, this smooth transition is referred to as *easing in* and *easing out* of a keyframe. Of course, you can always edit the control handles on your Bézier animation curve to customize that behavior to something else.

- **Easing (by strength):** The interpolation options in this column are preset, mathematically defined interpolation methods. You can get the same curve profile by manually editing the handles of a curve with Bézier interpolation, but these easing presets are a faster way of getting the same result.
- **Dynamic Effects:** When things move in meatspace, there isn't always a smooth transition from one pose to the next. When you drop a ball to the ground, it bounces. When you slam on the brakes while driving, the car over-extends, lurching forward before settling back to rest. In many cases, you can (and should) animate this behavior by hand. However, in a pinch, these three Dynamic Effects interpolations can give you a great starting point:
  - **Back:** In this interpolation type, the curve "overshoots the mark" set by the keyframe before settling back where you want it. To see what this effect is like, try slapping the surface of a table, but stopping just before your hand makes contact. If you watch carefully, you should notice that your hand goes a bit farther than you want it to before it finishes moving.
  - **Bounce:** The Bounce interpolation is a fun one. It makes your object appear to bounce before coming to a rest.
  - **Elastic:** Have you ever seen an over-extended rubber band break in half? The loose ends flop all over the place before they stop moving. The Elastic interpolation mode gives your object this kind of effect, like it's stuck to the floppy end of one of those broken rubber bands.

Figure 1-7 shows the Keyframe Interpolation menu.

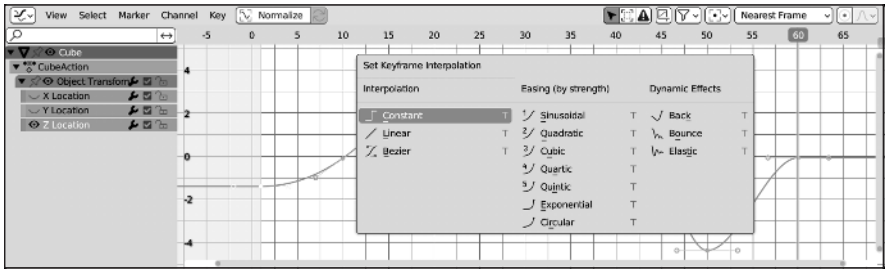

FIGURE 1-7: Changing the interpolation type on selected f-curve control points.

TIP

The interpolation mode options work only on the selected control points in the Graph Editor, so if you want to select all the control points in a single f-curve, select one of those control points and press L or choose Select ➪ Select Linked in the Graph Editor's header menu. Then you can apply your interpolation mode to the entire curve.

You can also change what a selected f-curve does before and after its first and last keyframes by changing the curve's *extrapolation mode.* You can change a curve's

extrapolation mode by selecting an f-curve channel in the left region of the Graph Editor and then pressing Shift+E or choosing Channel ➪ Extrapolation Mode in the Graph Editor's header menu. When you do, notice four possible choices:

- **Constant Extrapolation:** This setting is the default. The first and last control point values are maintained into infinity beyond those points.
- **Linear Extrapolation:** Instead of maintaining the same value in perpetuity before and after the first and last control points, this extrapolation mode takes the directions of the curve as it reaches those control points and continues to extend the curve in those directions.
- **Make Cyclic (F-Modifier):** This option adds an f-curve modifier (covered in the next section of this chapter) that takes all the keyframes in your animation and repeats them before and after your first and last keyframes.

  If you have a looping animation, like a character who never stops waving at you, this option is an easy way to make that happen.
- **Clear Cyclic (F-Modifier):** If your f-curve has a Cycles modifier, it's possible to remove that modifier from the Graph Editor's Sidebar (N), but this menu option is faster.

Figure 1-8 shows the menu for the different types of extrapolation modes, as well as what each one looks like with a simple f-curve.

If you have lots of animated objects in your scene, or just one object with a high number of animated properties, it may be helpful to hide extraneous curves from view so that you can focus on the ones you truly want to edit. To toggle a curve's visibility (or that of an entire keying set), left-click the eye icon next to its name in the channel region along the left side of the Graph Editor. If you want f-curves to be visible, but not editable, select a channel from the channel region and either left-click the lock icon or press Tab. You can also disable the influence of a specific f-curve or keying set by left-clicking the check box.

TIP

As mentioned previously in this chapter, you can quickly mute/hide animation channels in the Graph Editor by using the H, Shift+H, and Alt+H hotkeys with one or more selected keyframes.

If you need explicit control over the placement of a curve or a control point, the Graph Editor has a Sidebar like the 3D Viewport. You open it the same way: Press N or choose View ➪ Sidebar. Within the F-Curve tab of the Sidebar, there's an Active Keyframe panel where you can enter the exact value that you'd like to set your selected control point or that control point's handles, as well as modify that control point's interpolation type. Figure 1-9 shows the Sidebar in the Graph Editor.

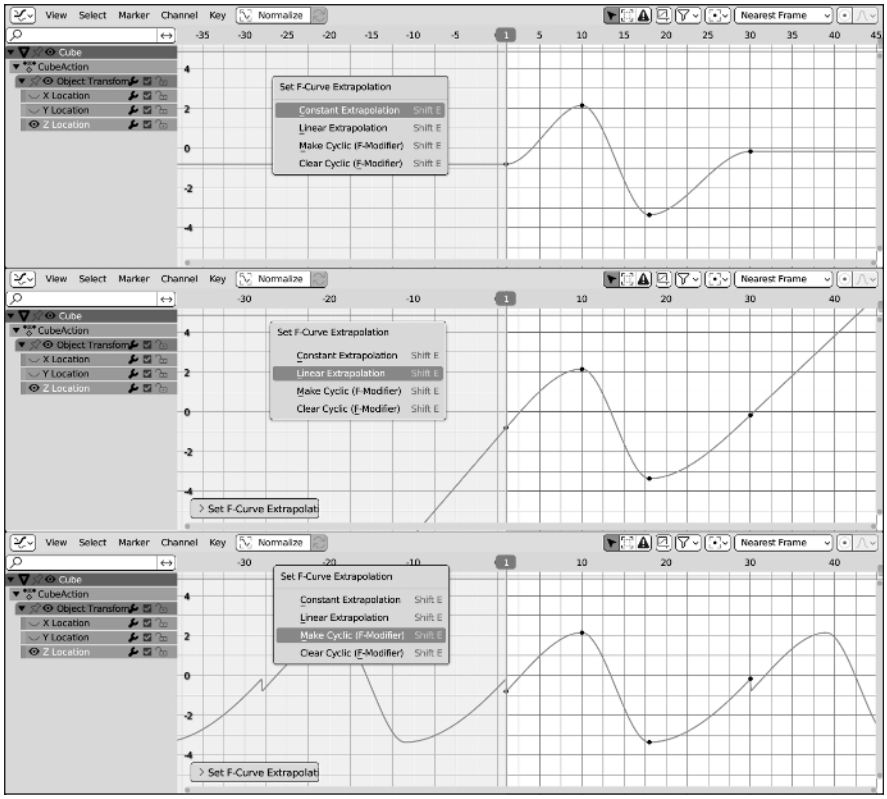

FIGURE 1-8: The four extrapolation modes you can have on f-curves.

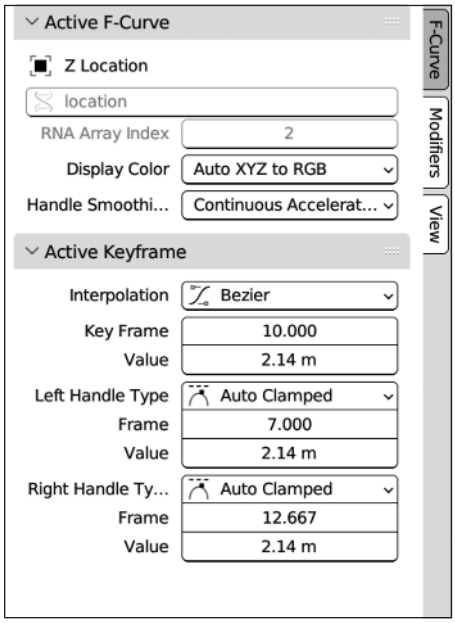

FIGURE 1-9: The Sidebar (N) in the Graph Editor.

Often, you run into the occasion where you need to edit all the control points in a single frame so that you can change the overall timing of your animation. You may be tempted to try and box-select to select the strip of control points you want to move around. However, a cleaner and easier way is to select just one of the control points on the frame you want to adjust and press K in the Graph Editor or choose Select ➪ Columns on Selected Keys. All the other control points on the same frame as your initial selection are selected.

Using f-curve modifiers

One of the biggest appeals of computer graphics in general — and computer animation specifically — is the prospect of letting the computer do all the hard work for you. Or, at least you want the computer to handle a big chunk of the boring parts. So if Blender's Graph Editor gives you the ability to procedurally manipulate f-curves with modifiers much in the same way you can with meshes, that's something worth knowing about.

The preceding section addresses modifiers when covering the cyclic curve extrapolation mode (Channel ➪ Extrapolation Mode). The Cycles modifier described there is one of a handful of f-curve modifiers that you can add. The following is a brief rundown of each of the f-curve modifiers that Blender offers:

- **Generator:** This modifier generates a new curve based on basic mathematical formulas for lines, parabolic lines, and some more advanced curve shapes. Unless you use additive mode, it completely disregards any existing keyframes on your f-curve.
- **Built-In Function:** This modifier generates a curve based on a different set of mathematical formulas than the Generator f-curve modifier. The formulas in this modifier are more naturally cyclic, like sine waves and tangent curves. Like the Generator f-curve modifier, this modifier also completely disregards any existing keyframes on your f-curve unless you use additive mode.
- **Envelope:** Though it can be a bit difficult to control, the Envelope f-curve modifier is extremely powerful. It's like the envelopes in audio processing software. You can add control points to define upper and lower thresholds for your f-curve. Any values in the f-curve beyond those thresholds (outside of the envelope) are attenuated (reduced so they fit within the envelope's bounds). I often couple this modifier with a Noise f-curve modifier to have more control over when the noise influence is added to my f-curves.
- **Cycles:** This f-curve modifier has nothing to do with the Cycles renderer. Instead, as covered previously in this chapter, this f-curve modifier duplicates your existing keyframes, repeating them over time. This is the modifier that gets added when you choose Channel ➪ Extrapolation Mode ➪ Make Cyclic (F-Modifier).

- **Noise:** With the Noise f-curve modifier, you can add random variation to your f-curve.

  Used judiciously, this modifier can bring a lot of life to an animation. For example, you could animate a car rolling along a bumpy road without actually creating those bumps.

- **Limits:** This f-curve modifier is similar to the Envelope f-curve modifier, except you can't set arbitrary control points. You just have minimum and maximum X- and Y-axis control values for your f-curve. Also, instead of attenuating your f-curve when it exceeds those controls, the Limits f-curve modifier simply clips, or cuts off, the overshot segments of the f-curve.

- **Stepped Interpolation:** Using the Stepped Interpolation f-curve modifier, you can give your f-curve a stepped appearance as it goes from one keyframe to the next. It would be like using constant interpolation on your f-curve and inserting a keyframe every few frames between the main keyframes of your animation. This can be useful if you want to give your animation a strobed or stop-motion look.

Figure 1-10 shows the panel for each of the f-curve modifiers.

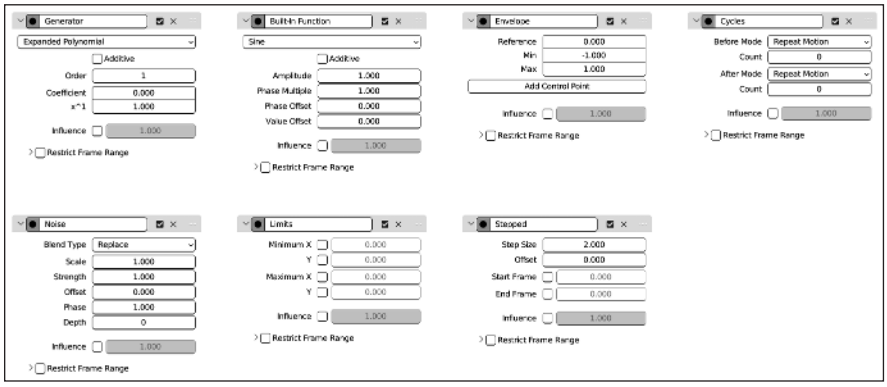

FIGURE 1-10: Blender offers seven different f-curve modifiers that you can apply to f-curves in your animation.

REMEMBER

Like the modifiers that affect your 3D objects (see Book 2, Chapter 2), you can stack f-curve modifiers so each one contributes to the one before it. For example, you could start with the Built-In Function f-curve modifier on your object's Z location to have your object move up and down sinusoidally. Then you could add a Noise f-curve modifier so it stutters a bit as it moves up and down.