Introduction to Animation

The best way to learn about animation is to start animating, so you'll begin this chapter with the classic exercise of bouncing a ball. You'll then take a closer look at the animation tools Maya software provides and how they work for your scene. You'll do that by throwing the axe you created earlier. Finally, you'll tackle animating a more complex system of parts when you bring your locomotive to life.

Topics in this chapter include:

- **Keyframe animation—bouncing a ball**
- **Throwing an axe**
- **Path animation**
- **Replacing an object**
- **Creating and deforming text along a path**
- **Rigging the locomotive, part 1**
- **Animating a catapult**

Keyframe Animation—Bouncing a Ball

No matter where you study animation, you'll always find the classic animation exercise of creating a bouncing ball. Although it's a straightforward exercise and you've probably seen it a hundred times on the Web and in other books, the bouncing ball is a perfect exercise with which to begin animating. You can imbue the ball with so much character that the possibilities are almost endless, so try to run this exercise as many times as you can handle. You'll improve with every attempt.

Animating a bouncing ball is a good exercise in real-world physical motion as well as in cartoon movement. First, you'll create a rubber ball and create a proper animation hierarchy for it. Then, you'll add cartoonish movement to accentuate some principles of the animation techniques discussed in the ultra-fabulous Chapter 1, "Introduction to Computer Graphics and 3D."

Creating a Cartoon Ball

First, you need to create the ball, as well as the project for this exercise. Follow these steps:

1. In a new scene, begin with a poly sphere, and then create a poly plane. Scale the plane up to be the ground plane.

2. Press 5 for Shaded mode.

3. Move the sphere 1.0 unit up in the *Y*-axis so that it's resting on the ground and not halfway through it, as shown in Figure 8.1.

4. Choose Modify → Freeze Transformations to set the ball's resting height to 0, as opposed to 1. This action sets the ball's Translate attribute back to 0, effectively resetting the object. This is called *freezing the transforms.* This is useful when you position, scale, and orient an object and need to set its new location, orientation, and size as the beginning state.

Figure 8.1

Place the ball on the ground.
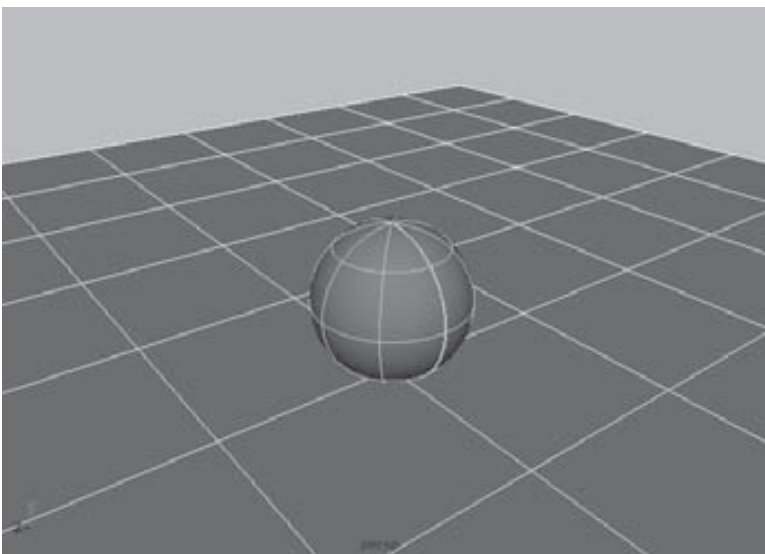
5. Choose File → Project → New to create a new project. Call the project **bouncing_ball**, and place it in the same parent folder as your Solar System project folder. Click the Use Defaults button to create the necessary folders in your project, and then click Accept. Save the scene file into that project.

Setting Up the Hierarchy

To make life easier, you'll set up the ball with three null nodes above it, listed here from the top parent node down: translate, scale, rotate. All the animation will be placed on these three nodes, and not the sphere itself. This will allow you to easily animate the ball bouncing, squashing, and stretching, and moving forward in space.

1. Select the sphere and press Ctrl+G to create the first group. In the Outliner, call this new group **rotate**.

2. With the rotate node selected, press Ctrl+G to create the scale group, and name it accordingly.

3. With the scale group selected, press Ctrl+G one last time to create the translate group and name it accordingly. Figure 8.2 shows the hierarchy.

Figure 8.2

The ball's hierarchy

As you animate, you'll quickly see why you've set up a hierarchy for the ball, instead of just putting keys on the sphere itself.
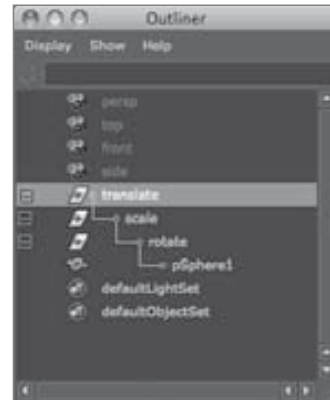
Animating the Ball

Your next step is to keyframe the positions of the ball using the nodes above the sphere. As introduced in Chapter 1, *keyframing* is the process of setting positions and values at particular frames of the animation.

You'll start with the *gross animation*, which is the overall movement scheme, a.k.a. *blocking*. First, you'll move the ball up and down to begin its choreography in these steps:

1. Press W to open the Translate tool, select the translate node, and move it up to the top of the frame, say about 10 units up in the *Y*-axis and 8 units back in *X*-axis at (-8,10,0). Place the camera so that you'll have some room to work in the frame.

2. Instead of selecting the Translate attributes in the Channel Box and pressing S as you did in Chapter 2, "Jumping in Headfirst, with Both Feet," to set keyframes on the planets, you'll set keyframes for translation in an easier way.

   Press Shift+W to set keyframes on Translate X, Translate Y, and Translate Z at frame 1 for the top node of the ball (named translate). To make sure your scene is set up properly, set your animation speed to 30fps by choosing Window → Settings/Preferences → Preferences to open the Preferences window or by clicking the Animation Preferences button (⊞) next to the Auto-Key button. In the Settings category of the Preferences window, set Time to NTSC (30fps). A frame range of 1 to 120 is good for now. Figure 8.3 shows the ball's start position.
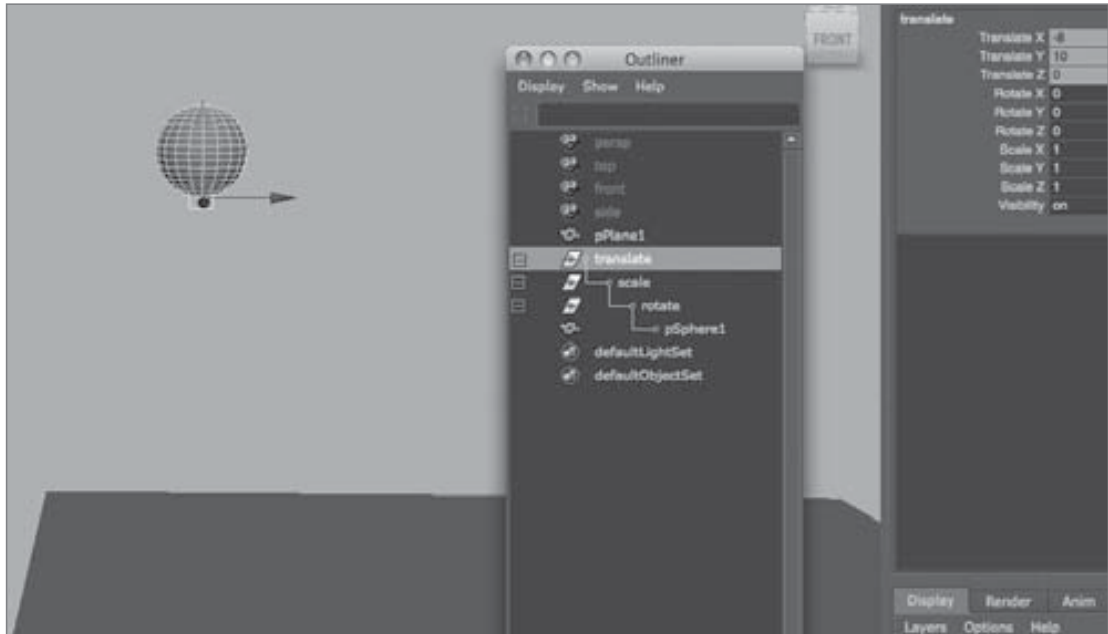
3. Click the Auto Keyframe button (⊶) to turn it on; it turns red. Auto Keyframe automatically sets a keyframe at the current time for any attribute that has changed since its last keyframe for the selected object or node.

For the Auto Keyframe feature to work, you have to set an initial keyframe manually for each of the attributes you want to animate.

4. Disregarding any specific timing, go to frame 10, and move the ball down in the *Y*-axis until it's about one-quarter through the ground plane. Because you'll be creating squash and stretch for this cartoon ball (see Chapter 1 for a brief explanation), you need to send the ball through the ground a little bit. Then, move the ball about 3 units to the right, to about (-5,-0.4, 0). The Auto Keyframe feature sets a keyframe in the *X* and *Y* axes at frame 10. Remember, this is all on the translate node.

5. Move to frame 20, and raise the ball back up to about half of its original height and to the right about 2.5 units (-2.5,4,0). Auto Keyframe sets X and Y Translation keyframes at frame 20 and will continue to set keyframes for the ball as you animate.

6. At frame 30, place the ball back down a little less than one-quarter of the way through the ground and about 2 units to the right, at about (−0.5, −0.3,0).

7. At frame 40, place the ball back up in the air in the *Y*-axis at a fraction of its original height and to the right about 1.5 units, at about (1.1,1.85,0).

8. Repeat this procedure every 10 frames to frame 110 or so, so that you bounce the ball a few more times up and down and to the right (positive in the *X*-axis). Make sure you're decreasing the ball's height and traveling in *X* with each successive bounce and

decreasing how much the ball passes through the ground with every landing until it rests on top of the ground plane. Open the Graph Editor for a peek into the ball's animation curves (see Figure 8.4). (Choose Window → Animation Editors → Graph Editor.)
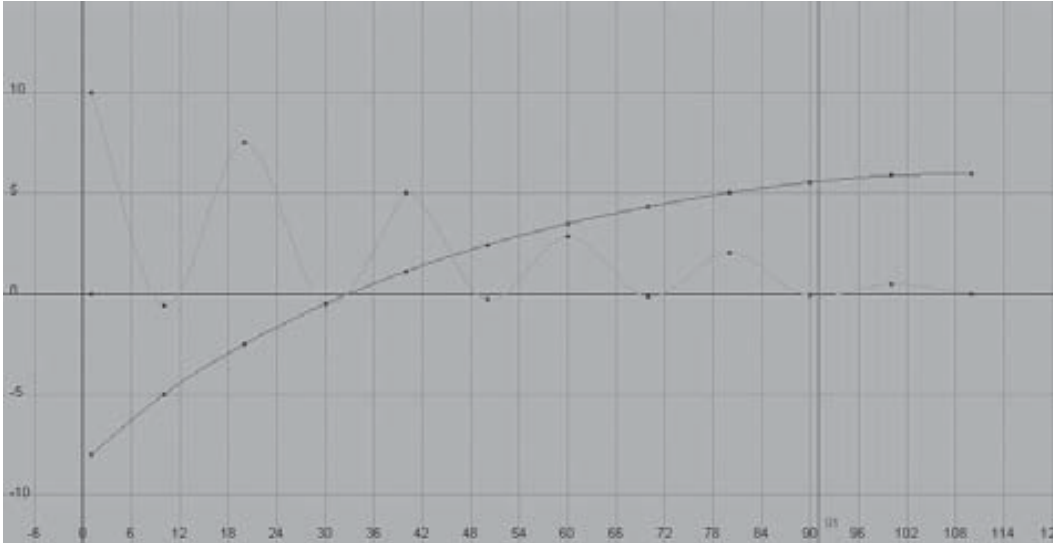
Figure 8.4

The Graph Editor curves for the ball's translate node

By holding down the Shift key as you pressed W in step 2, you set a keyframe for Translate. Likewise, you can keyframe Rotation and Scale. Here's a summary of the keystrokes for setting keyframes:

Shift+W	Sets a keyframe for the selection's position in all three axes at the current time
Shift+E	Sets a keyframe for the selection's rotation in all three axes at the current time
Shift+R	Sets a keyframe for the selection's scale in all three axes at the current time

You'll resume this exercise after a look at the Graph Editor.

The Graph Editor

The Graph Editor is a critical tool for an animator. You provide most of an animation's finesse through work in this window. As a matter of fact, you can animate a number of objects solely through this window. (See Chapter 3, "The Maya 2011 Interface," for an introduction to the Graph Editor.) Using its graph view of where in space and time each keyframe lies, you can conveniently control your animation. Move a keyframe in time to the right, for example, to slow the action. Move the same keyframe to the left in time to speed up the action.

Reading the Curves in the Graph Editor

Understanding what animation curves do in the Graph Editor is crucial to getting your animation right. Using the Graph Editor to *read animation curves*, you can judge an object's direction, speed, acceleration, and timing.

As you'll see later in this chapter with the axe-throwing tutorial, you'll invariably come across problems and issues with your animation that require a careful review of their curves. The ability to see a curve and translate it into what your object is doing comes with time and practice. Be patient, and don't be afraid to open the Graph Editor and animate through it as much as you can. Here are a couple of key concepts to keep in mind.

First, the curves in the Graph Editor are like the NURBS curves you've modeled with so far. Instead of CVs on a NURBS curve controlling the curvature, points directly on an animation curve represent keyframes and control the curvature with their *tangency handles*. By grabbing one end of a key's handle and dragging it up or down, you adjust the curve.

Second, the graph is a representation of an object attribute's position (vertical) over time (horizontal). Every place on the curve represents where the object is in that axis; there needn't be a keyframe on the curve. Not only does the placement of the keys on the curve make a big difference, so does the shape of the curve itself. Here is a quick primer on how to read a curve in the Graph Editor and, hence, how to edit it.

In Figure 8.5, the object's Translate Z attribute is being animated. At the beginning, the curve quickly begins to move positively (that is, to the right) in the *Z*-axis. The object shoots off to the right and comes to an *ease-out*, where it decelerates to a stop. The stop is signified by the flat part of the curve at the first keyframe at frame 41. The object then quickly accelerates in the negative *Z* direction (left) and maintains a fairly even speed until it hits frame 62, where it suddenly changes direction and goes back right for about 45 frames. It then slowly decelerates to a full stop in an ease-out.

Consider a single object in motion. The shape of the curve in the Graph Editor defines how the object moves. The object shown in Figure 8.6 is moving in a steady manner in one direction.
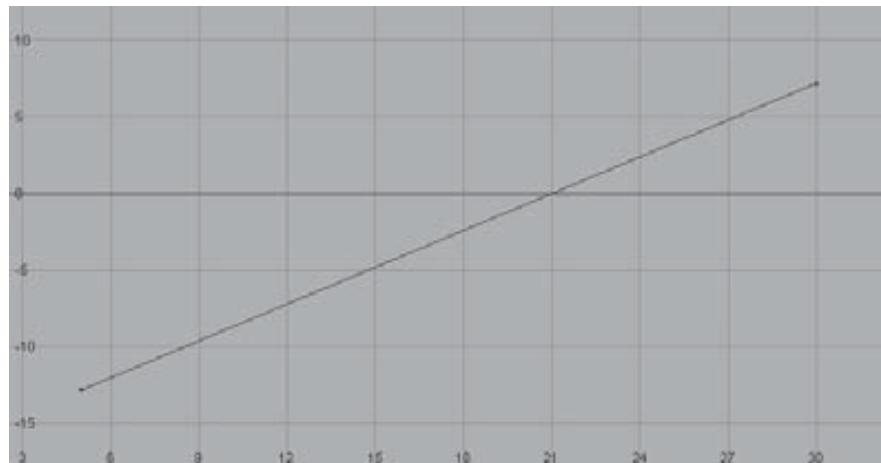
Figure 8.5

An animation curve
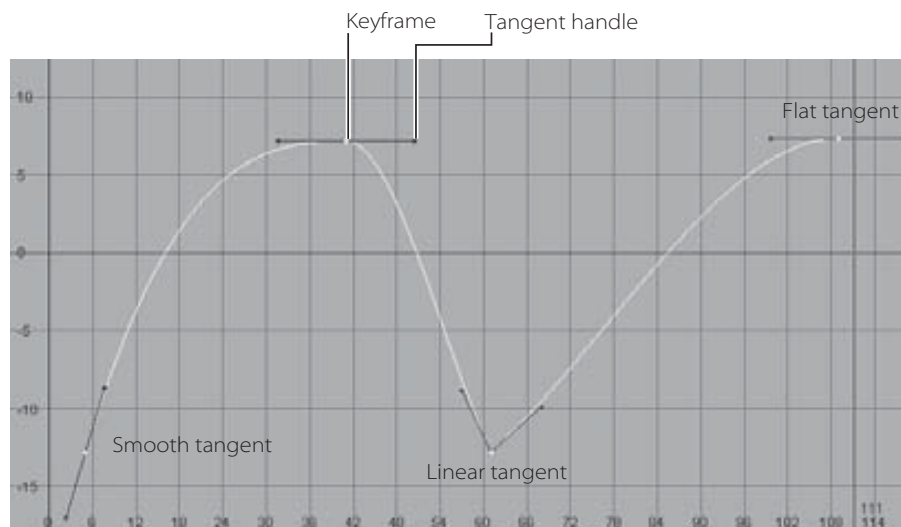
Figure 8.7 shows the object slowly accelerating toward frame 30, where it suddenly comes to a stop. If there is nothing beyond the end of the curve, there is no motion. The one exception deals with the *infinity* of curves, which is discussed shortly.

The object in Figure 8.8 begins moving immediately and comes to a slow stop by frame 27, where the curve first becomes flat.
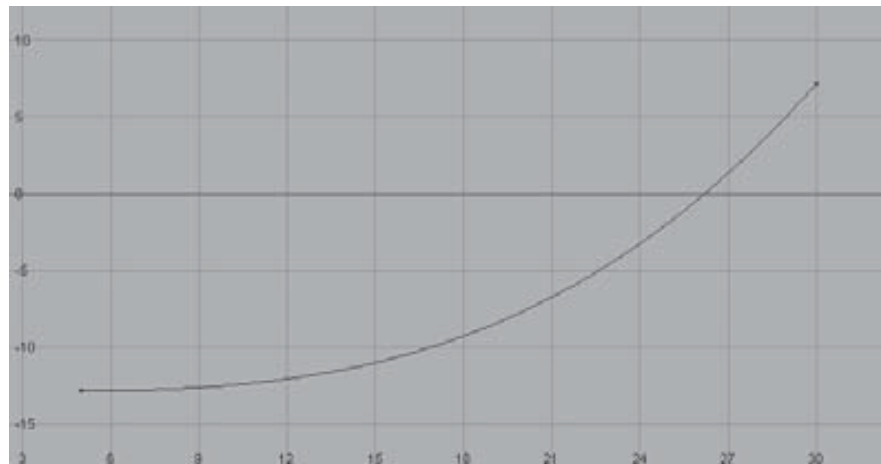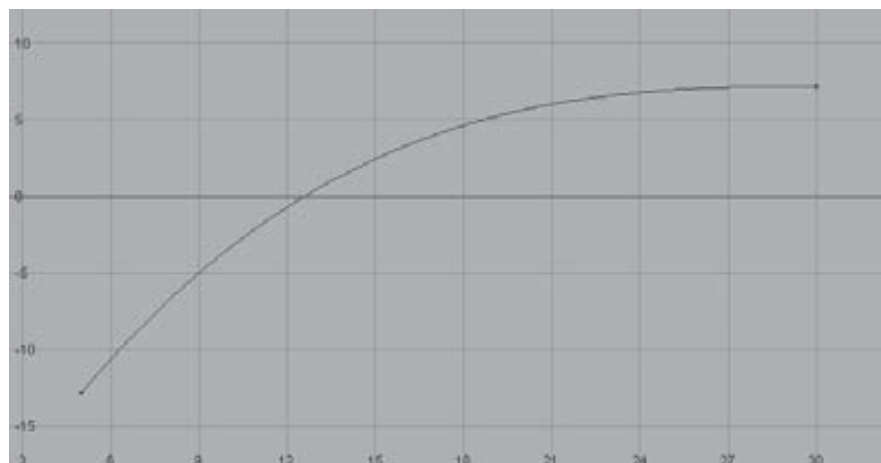
Cartoon Ball

Now, let's apply what you've learned about the Graph Editor to the bouncing ball. Follow these steps:

1. Open the Graph Editor, and look at the ball's animation curves. They should be similar to the curves in Figure 8.4, shown earlier.

2. Notice how only the *X* and *Y* axes' translates have curves, and yet Translate Z has a single keyframe but no curve. It's from the initial position keyframe you set at frame 1. Because you've only moved the sphere in the *X* and *Y* axes, Auto-Key hasn't set any keys in the *Z*-axis. This is better than pressing S to set keys on everything, so if you don't have animation on something, it doesn't get keys. Keep your scene clean.

3. Play back the animation, and see how it feels. Be sure to open the Animation Preferences window. Click the Animation Preferences icon (▣) to set the playback speed to Real-Time (30fps). You'll find this icon in the Playback section in the Timeline category.

4. Timing is the main issue now, so you want to focus on how fast the ball bounces:

   - The ball is falling too fast initially, although the second and third bounces should look fine.
   - To fix the timing, move the keyframes in the Graph Editor. For the *X* and *Y* axes, select the keyframes at frame 10 and all the others beyond on both curves. Move them all back two frames. (See Figure 8.9.)

As the ball's bounce decays over time, it goes up less but still takes the same amount of time (10 frames) to go up the lesser distance. For better timing, adjust the last few bounces to occur faster. Select the keys on the last three bounces and move them, one by one, a frame or two to the left to decrease the time on the last short bounces. (See Figure 8.10.)

Figure 8.9

Move all the key-frames for both curves to the right to slow the initial fall by two frames, but leave the timing the same for the rest.
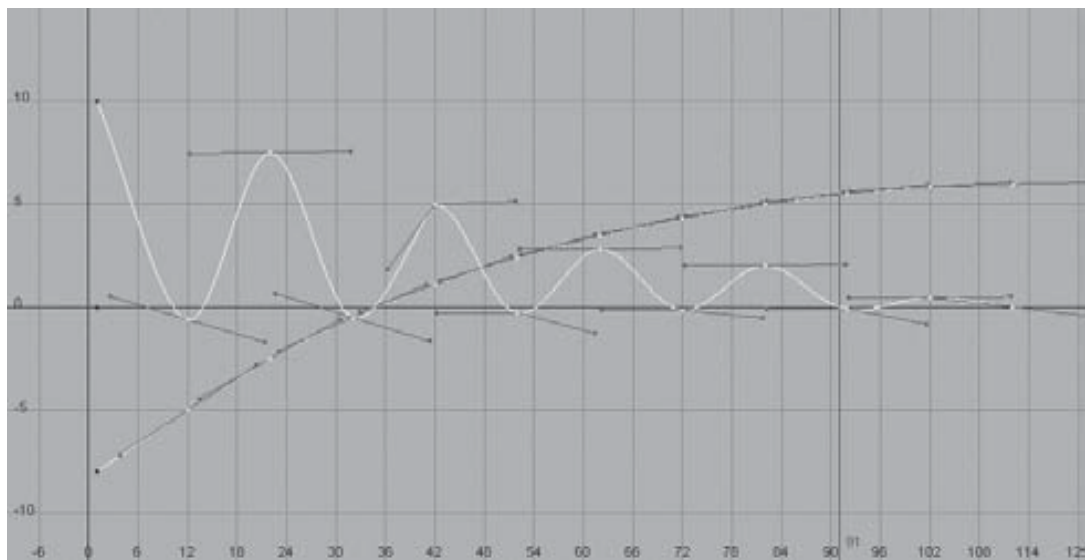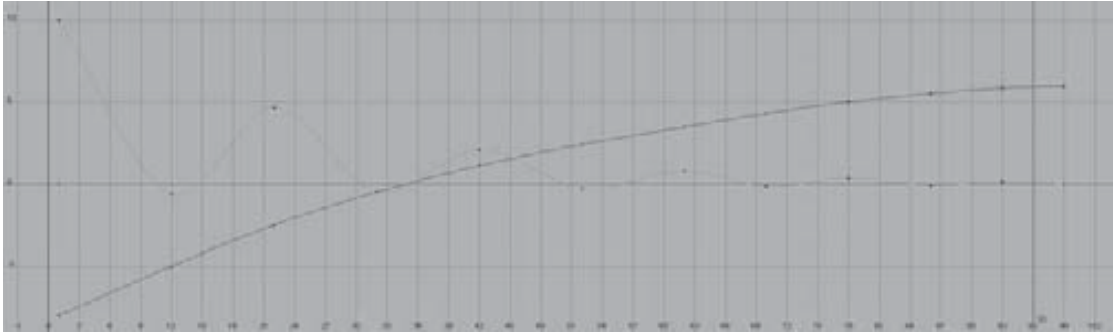
Figure 8.10

Move the keys to make the final short bounces quicker and the bounce height feel right.

To move a key in the Graph Editor, press W to open the Move tool, MMB+click, and drag the cursor in the Graph Editor window. Press the Shift key, and drag the cursor left and right or up and down to lock the movement to either horizontal or vertical to make it easier to control.

Understanding Timing

In animation, timing is all about getting the keyframes in the proper order. Judging the speed of an object in animation is critical to getting it to look right, and that comes down to timing. The more you animate, the better your timing will be, which is why the bouncing ball is such a popular exercise.

Load the file ball_v01.mb from the Bouncing_Ball project on the CD to get to this point.

When you play back the animation, it should look more natural. But it still looks fake, as if it's rising and falling on a wave as opposed to really bouncing. You need to edit the timing of the ball. The problem with the animation is that the ball eases in and out as it rises and falls. By default, setting a key in Maya sets the keyframes to have an ease-in and ease-out in their curves, meaning their curves are smooth like a NURBS curve.

Because of the smooth animation curve, the ball doesn't look natural in its timing. You need to accelerate the ball as it falls with a sharp valley in the curve, and you need to decelerate it as it rises with smooth peaks. Follow these steps:

1. In the Graph Editor, select the Translate Y entry in the left panel of the window to isolate your view to just that curve in the editor panel on the right. Select all the landing keyframes (the ones in the valleys of the curve) and change their *interpolation* from smooth to linear by clicking the Linear Tangents button ( ).

2. Likewise, select all the peak keyframes at the ball's rise, and change their tangents to flat by clicking the Flat Tangents icon ( ) to make the animation curve like the one shown in Figure 8.11.

3. When you play back the animation, the ball seems to be moving more realistically. If you need to, adjust the keys a bit more to get the timing to feel right to you, before you move on to squash and stretch and rotation.

Figure 8.11

The adjusted timing of the bounce
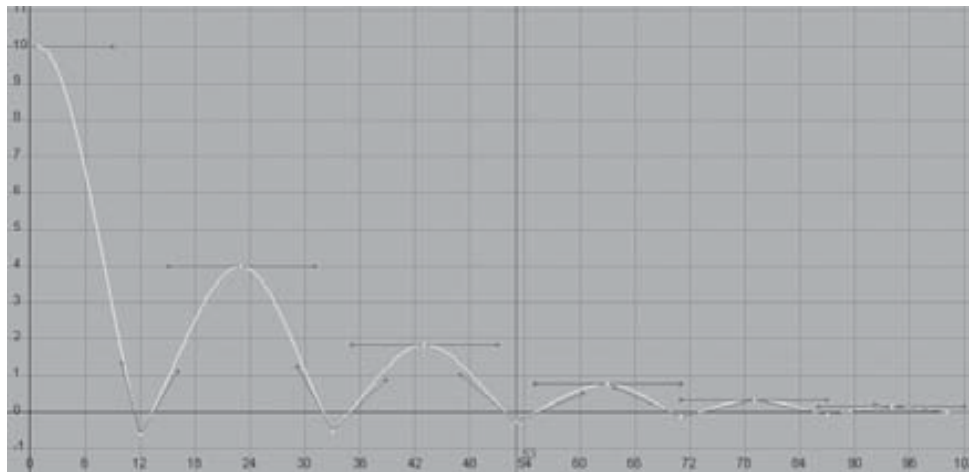
Squash and Stretch

The concept of *squash and stretch* has been an animation staple for as long as there has been animation. It's a way to convey the weight of an object by deforming it to react (usually in an exaggerated way) to gravity and motion.

In Maya, you use the Scale tool to squash and stretch your object—in this case, your ball, using only the scale node, not the sphere or translate node.

Load the file `ball_v02.mb` from the Bouncing_Ball project on the CD, and follow these steps:

1. Select the scale node, and select Modify → Center Pivot. This places the scale pivot point in the middle of the ball.

2. At frame 9, press R to set initial scale keyframes on the scale node of the ball, a couple of frames before the ball impacts the ground.

3. To initiate squash and stretch, go to frame 12, where the ball hits the floor the first time. With the scale node selected, press R to open the Scale tool; scale the ball down in the *Y*-axis until it no longer goes through the floor (about 0.6), as shown in the image on the left in Figure 8.12. Set a keyframe for scale by pressing Shift+R.

Figure 8.12

Squashing and Stretching the ball to react to bouncing on the floor
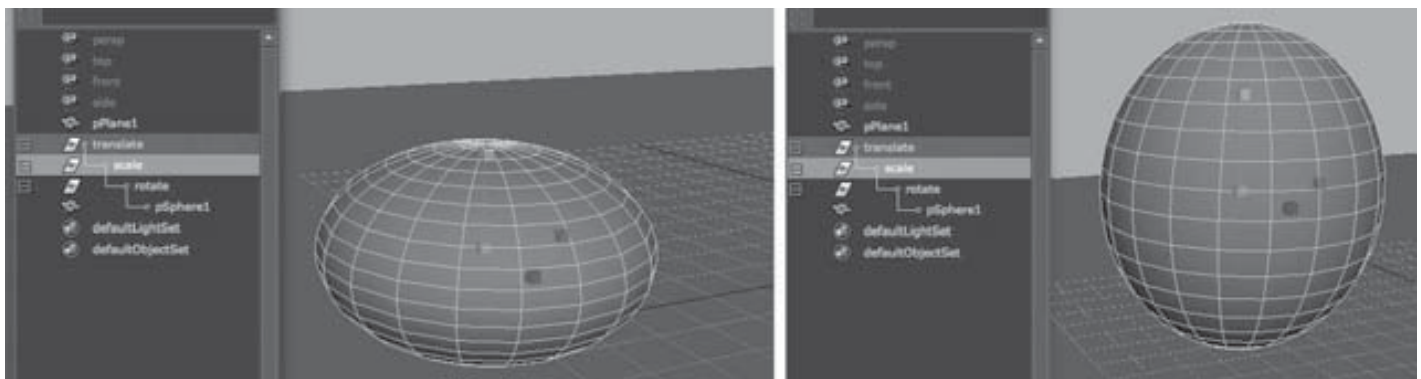
4. Move ahead in the animation about three frames to frame 15. Scale the ball up in the *Y*-axis slightly past normal to stretch it up (about 1.15) immediately after its bounce, as shown in the image on the right in Figure 8.12. Three frames later, at frame 18, set the *Y*-axis scale back to 1 to return the ball to its regular shape

5. Scrub your animation, and you should see the ball begin stretching even before it hits the ground. That's a bit too much exaggeration, so open the Graph Editor and move the *Y*-axis scale key from 9 to 11. Now, the ball squashes when it hits the floor and stretches as it bounces up.

6. Repeat this procedure for the remaining bounces, squashing the ball as it hits the floor and stretching it as it bounces up. Remember to decay the scale factor as the ball's bouncing decays to a stop, as when you decayed the height of the ball's bounce earlier. The final bounce or two should have very little squash and stretch, if any.

Load the file `ball_v03.mb` from the Bouncing_Ball project on the CD to get to this point. And now, let's rotate the ball as it bounces.

Rotation

Let's add some roll to the ball in these steps:

1. Select the ball's rotate node, and select Modify → Center Pivot to set that node's pivot at the center of the ball.

2. At frame 1, press Shift+E to set keys for rotation at (0,0,0).

3. Scrub to the end of your animation (frame 100 in this example), and set a value of -480 for Rotate Z in the Channel Box, as shown in Figure 8.13.

4. Open the Graph Editor to see the rotation curve on the ball's rotate node. It's a linear (straight) line angled down from 0 to -480. You need the rotation to slow to a stop at the end of the animation, so select the final keyframe and click (⬛) to make it a flat tangent.
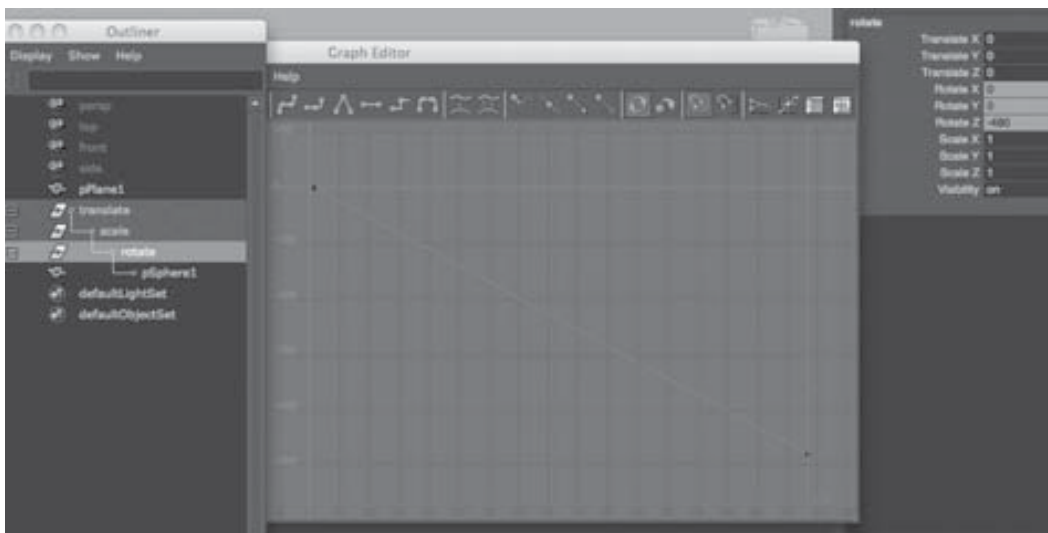

Figure 8.13

Setting a roll for the ball

Load the file `ball_v04.mb` from the Bouncing_Ball project on the CD to see an example of the finished bouncing ball. Although the bouncing of this ball looks okay, it could definitely use some finesse, a little timing change, and so on. Open the file, open the Graph Editor, and edit the file to get a feel for how the ball bounces and rolls. For example, it could continue rolling with no bouncing for another 20 frames or so.

Throwing an Axe

This next project will exercise your use of hierarchies and introduce you to creating and refining motion to achieve proper animation for a more complex scene than the bouncing ball. The workflow is simple but standard for properly setting up a scene for animation, also known as *rigging*, especially for more complex objects, as you'll see later in this and the next chapter when you rig the locomotive for animation. First, you'll model an axe and a target, and then you'll set up the grouping and pivots for how you want to animate. Then, you'll throw your axe!

Why won't you throw the NURBS axe you've already created and textured? Because later in this chapter, you'll need it for an exercise on importing and replacing an object in Maya while keeping the animation intact.

The Preproduction Process

To begin the animation right away, you'll create a basic axe, focusing on the animation and the technique. Toward this end, connect to the Internet if you can, and look up axes and the art of axe throwing to get more familiar with the task at hand. You'll also need to create a simple bull's-eye target at which to throw your axe, so look for some references for a target as well.

Create a new project; choose File → Project → New. Place this project in the same folder or drive as your other projects, and call it Axe. Click the Use Defaults button to fill in the rest, and click Accept. Click the Animation Preferences button, and set the frames per second to 30fps. Later, you'll replace this simple axe with a finer NURBS axe model to learn how to replace objects and transfer animation properly in Maya.

Setting Up the Scene

To get started, model the axe and target from primitives, and set up their grouping and pivots. When your scene is set up properly, you'll animate. It's important to a healthy workflow that you make sure the scene is set up well before you begin animating.

Making the Axe

The axe will be made of two polygon primitives: a cylinder and a cube. Follow these steps:

1. Choose Create → Polygon Primitives → Cube ❒. Set Width Divisions to 4, and click Create.

2. Call this cube **axe_head**.

3. Choose Create → Polygon Primitives → Cylinder to create a cylinder to be the handle for your axe, and call it **handle**.

4. Scale the cylinder so that it's about one-half unit across and about 14 units tall.

Figure 8.14

Placing the axe's head on the handle

5. Move the cube to the top of the cylinder, leaving just a little of the tip showing, and scale it so that it's about 2.5 units high and 4 units wide in the front view. (See Figure 8.14.)

6. Scale the cube in the *Z*-axis so that it's just a little thicker than the handle.

7. To put a sharp edge on the axe, go into Component mode (F8), and select the four vertices on the very end of the cube:

    •   Press R to activate the Scale tool.

    •   Scale the vertices down in the Z-axis to a sharp edge, and scale them slightly up in the *Y*-axis.

    •   Select the next four vertices in from the edge. Scale them down in the *X*-axis about halfway, and scale them up slightly in the *Y*-axis. (See Figure 8.15.)

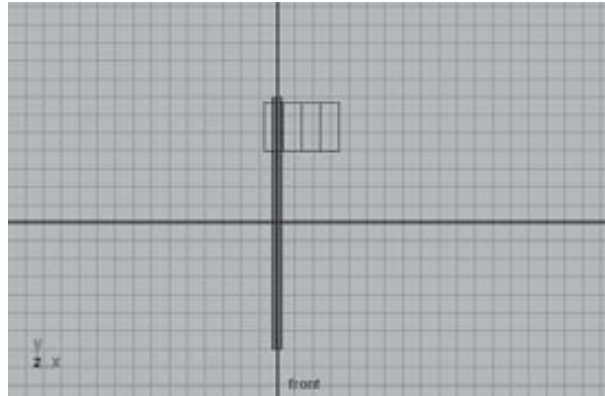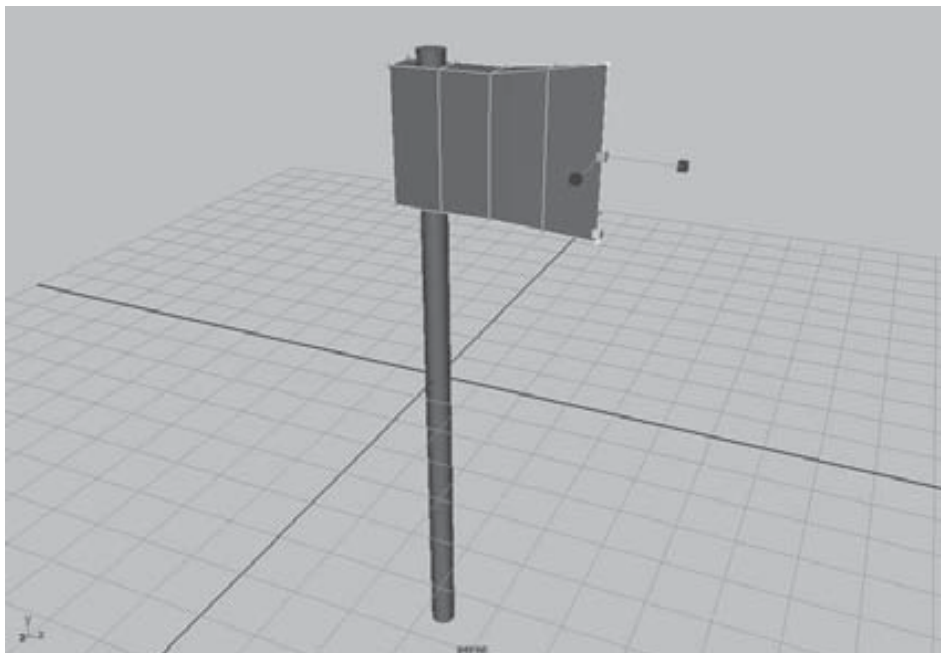8. Press F8 to get back into Object mode, and select both pieces.

Figure 8.15

Creating a sharp edge for the axe

Most animation work doesn't depend on precise measurements. The key is using proportions and relative sizes. You can almost always use Maya's generic units (which are set to centimeters by default). The scope of your project will determine if greater precision is necessary.

9. Choose Edit → Group to group the pieces into one hierarchy, and call it **axe**.

   Now, you need to identify the *center of balance* for this axe. Doing so will give you the place for your pivot point. Because the heaviest part of this axe is the head, the center of balance is toward the top of the handle. In addition, the axe head protrudes in only one direction, so the balance point is a unit or two away from the handle.
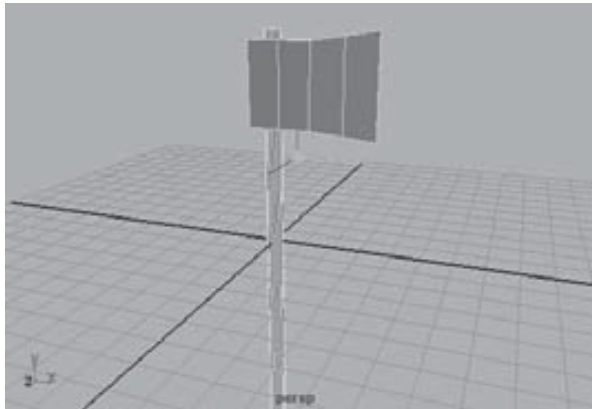
10. Press Insert to activate the Pivot manipulator, and move the pivot just under the axe head and about 1 unit from the handle, as in Figure 8.16.

    Your scene should now look like the file axe_v1.mb in the Scenes folder of the Axe project on the CD.

Figure 8.16

Find the axe's center of balance to determine the proper place for its pivot point.

Making the Target

Move the axe about 40 units in the negative *X*-axis away from the origin to make some room for the throw, and move it about 15 units up in the *Y*-axis. Now, you need to create a simple target for your axe to hit. Follow these steps:

1. Start with a polygonal cylinder.

2. Scale down the cylinder's height to make it squat, and rotate it just a bit past being perpendicular to the ground, facing somewhat toward the sky.

3. Scale the cylinder up about seven times its original size.

   You also need to create a simple stand upon which to rest the target. Follow these steps:

1. Choose Create → Polygon Primitives → Cube for the base.

2. Scale the cube so that the target fits on the base with some room to spare.

3. Choose Create → Polygon Primitives → Cylinder to create two cylinders, to make a cross brace for the back of the target.

4. Scale, rotate, and position the cylinders to fit behind the target and into the back of the base. Figure 8.17 shows the positions of the pieces to make up the target.

5. When you're happy with the target, group the four objects together and name the group **Target**.

Figure 8.17

Positioning the cross braces for the target

6. Move the target about 40 units in the positive *X*-axis away from the origin, and move it up in the *Y*-axis so its base is basically on the ground plane.

7. Press Insert to move the target's pivot point to the bottom of the base, right where the cross beams connect to it. Figure 8.18 shows the proper pivot placement.
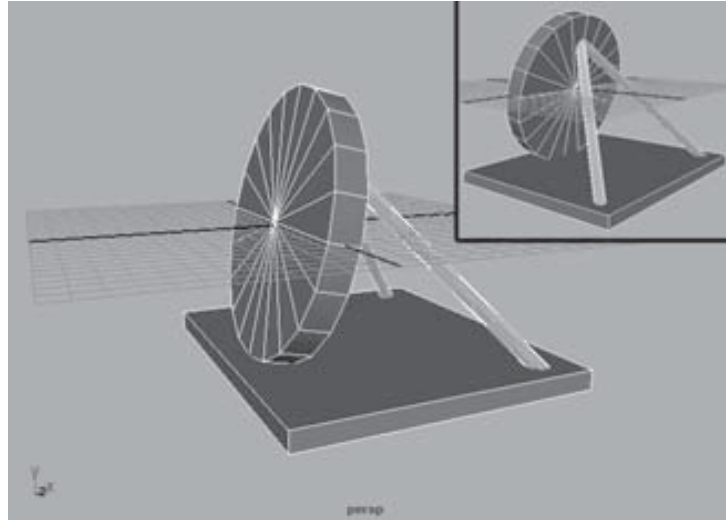
You place the pivot point at that location because the target will jerk up and back a little bit when the axe hits it. This is the best point for you to rotate around to lift the front of the base up a little to make it look like the axe is really hitting it. Save your file as your next version number. The scene is now ready for animation.
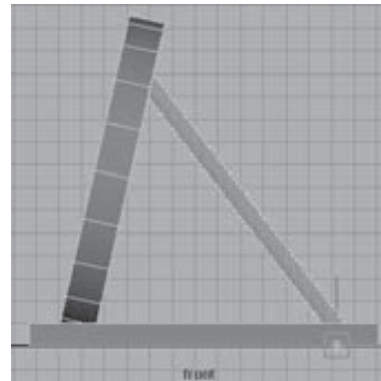

Figure 8.18

The proper placement of the target's pivot point

Preproduction: Keyframes and Motion Study

What separates good animation from bad animation is the feeling of weight that the audience infers from the animation. Being lifelong observers of natural physics, people instinctively understand how nature works in motion. You see an object in motion, how it moves, and how it affects its surroundings. From that, you can feel the essence of its motion, with its weight making a distinct, albeit subliminal, impression on you. As it pertains to animation, that essence is simply called *weight*, and its observation is called *motion study*.

Unfortunately, merely knowing how something should look while moving isn't all you need to animate it properly. Giving the axe believable weight is your primary job as an animator. You need to make the audience feel as if that CG axe is really moving. Modeling, texturing, and lighting are small parts of it, but motion can make or break the

animation. Creating a convincing scene takes compelling motion, and creating compelling motion requires close observation.

A good feeling of weight in animation depends on timing and follow-through, which require practice.

It's a good idea first to try out an action you want to animate. It may upset the cat if you grab a real axe and start throwing it around your house, but you can take a pen, remove its cap, and lob it across the room. Notice how it arcs through the air, how it spins around its center of balance, and how it hits its mark. Now put the cap on the pen, lob it again, and notice the subtle yet instrumental differences in motion caused by the cap's mass.

As an animator, this experimentation is part of your preproduction and motion study. It's important to have as thorough an understanding of your subject matter as possible. Just try not to take out anyone's eye with the pen.

According to some Internet research, the perfect axe throw should contain as few spins as possible. This is good information to know, because it will shape your animation and come in handy if you're ever cornered in a hatchet shop.

Animating the Axe: Keyframing Gross Animation

The next step is to keyframe the positions of the axe, starting with the *gross animation*—that is, the movement from one end of the axe's trajectory to the other.

Setting Initial Keyframes

You can start in your current scene or load the unanimated, premade axe and target from the CD (axe_v2.mb in the Scenes folder of the Axe project). Follow these steps:

1. Select the axe's top group node—not just the pieces. To make selecting groups such as this easier, display the object's selection handle. To do so, select the axe's top node, and choose Display → Transform Display → Selection Handles. Doing so displays a small cross, called a *selection handle* (⟡), at the axe's pivot point. You need only select this handle to select the top node of the axe.

   As you'll see later, in this chapter's catapult exercise, you can use selection handles to select the children of a group as well as the top node.

   Because this node is the parent node of the axe, the selection handle displays as a hollow cross at the node's current pivot point.

   You can turn on selection handles for practically any object in Maya—no matter where it is in a group's hierarchy—whether it's a child or a parent. If it isn't the top node, the selection handle appears as a regular cross (✛).

2. With the axe selected, go to frame 1, and set a keyframe for the rotation and translation.

3. Hold down the Shift key, press W for the axe's translation keyframe, and then press E for the axe's rotation keyframe. You don't need to set a scale keyframe on the axe by pressing Shift+R because you won't be changing its size. When you're finished, you have the initial keyframes for the axe at its start position.

Creating Anticipation

Instead of the axe just flying through the air toward the target, you'll animate the axe moving back first to create *anticipation*, as if an invisible arm were pulling the axe back before throwing it. Follow these steps:

1. Go to frame 15.

2. Move the axe back in the *X*-axis about 8 units, and rotate it counterclockwise about 45 degrees.

3. The Auto Keyframe feature sets keyframes for the position and new rotation at frame 15.

    Because you've only moved the axe back in the *X*-axis and made the rotation on the *Z*-axis, Auto Keyframe sets keyframes only for Translate X and Rotate Z. The other position and rotation axes aren't keyframed because their values didn't change.

4. Scrub through the animation, and notice how the axe moves back in anticipation.

> Auto Keyframe inserts a keyframe at the current time for the selected object's changed attributes only.

5. Go to frame 40, and move the axe so that its blade cuts into the center of the target.

    Notice that you have to move the axe in the *X*- and *Y*-axes, whereas before you only had to move it back in the *X*-axis to create anticipation. This is because the axis of motion for the axe rotates along with the axe. This is called the *Local axis.* The Local axis for any given object shifts according to the object's orientation. Because you angled the axe back about 45 degrees, its Local axis rotated back the same amount.

The file axe_v3.mb in the Scenes folder of the Axe project on the CD will catch you up to this point in the animation.

This last step reveals a problem with the animation. If you scrub your animation now, you'll notice that the axe's movement back is different from before, setting a keyframe at frame 40.

This is because of the Auto Keyframe feature. At frame 1, you set an initial keyframe for all axes of translation and rotation. Then, at frame 15, you moved the axe back in the *X*-axis only (in addition to rotating it in the *Z*-axis only).

> **TROUBLESHOOTING AND AGGRAVATION**
>
> Why should you intentionally go through steps that create a problem, such as the axe's movement back? Understanding how to troubleshoot is the biggest challenge in learning a CG program. A good CG artist needs to know how to diagnose issues with their scene and be able to find a way to fix them. Your first forays into CG may be highly frustrating—riddled with simple troubles and issues that you just don't understand. When you can't figure out why things went wrong, you may turn red with aggravation and want to walk away. This is where you start molding yourself as a CG artist. Instead of giving up, ponder the steps you've taken, and see if you can spot where your CG has taken a weird turn. You'll probably find yourself in such spots several times as you study this book. Instead of throwing the baby out with the bathwater, stay patient and try, try again. You'll learn more from your mistakes and missteps in the tutorials in this book than you will if you follow everything to the letter.

Auto Keyframe set a keyframe for Translate X at frame 15. At frame 40, you moved the axe in *both* the *X* and *Y* axes to strike the target. Auto Keyframe set a keyframe at 40 for Translate X and Translate Y. Because the last keyframe for Translate Y was set at 1 and not at 15 as in the case of Translate X, there is now a bobble in the *Y* position of the axe between frames 1 and 15.

With the axe selected, open the Graph Editor (choose Window → Animation Editors → Graph Editor) to see what's happening. As you saw in the Bouncing_Ball exercise, using the Graph Editor is crucial, and the more practice you get with it, the better.

When you open the Graph Editor for this scene, you should see red, green, and blue line segments running up and down and left and right. You'll probably have to zoom your view to something more intelligible. By using the Alt key (or the Alt/Option key on a Mac) and mouse-button combinations, you can navigate the Graph Editor much as you can any of the modeling windows.

The hotkeys A and F also work in the Graph Editor. Click anywhere in the Graph Editor window to make sure it's the active window, and press A to zoom all your curves into view. Your window should look something like Figure 8.19.

Figure 8.19

The Graph Editor displays the axe's animation curves.
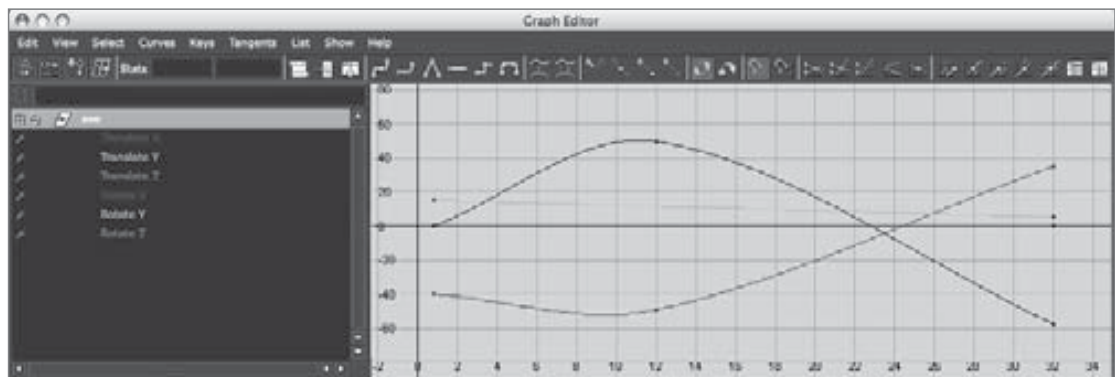
The curves in the Graph Editor represent the values of the axe's position and rotation at any given time. The three axes are in their representative red, green, or blue color, and the specific attributes are listed much as they are in the Outliner in the left column. Selecting an object or an attribute on the left displays its curves on the right.

You should also notice that the curves are all at different scales. The three rotate curves range in value from about –45 to 45, the Translate Y curve ranges from about 15 to 5, and Translate Z looks flat in the Graph Editor. It's tough to edit a curve with low values and still be able to see the timings of a larger value curve.

You can select the specific attribute and zoom in on its curve to see it better, or you can *normalize* the curves so that you can see them all in one view, with all their values in check. Click the Enable Normalized Curve Display icon in the top icon bar of the Graph Editor ( ). Doing so *normalizes* the view of all the curves within a scale of –1 to 1 to allow you to see the relative movement of all the curves at once.

Figure 8.20 shows the Graph Editor from Figure 8.19 after the curves have been normalized. Keep in mind that this doesn't change the animation in the slightest. All it does is allow you to see all the curves and their relative motion. You can denormalize the view by pressing the Disable Normalized Curve Display icon in the Graph Editor ( ). Normalizing your view is particularly helpful in busy scenes when you want to adjust the smallest scale of values alongside the largest scale of values without having to zoom in and out of the Graph Editor constantly to see the appropriate curves.
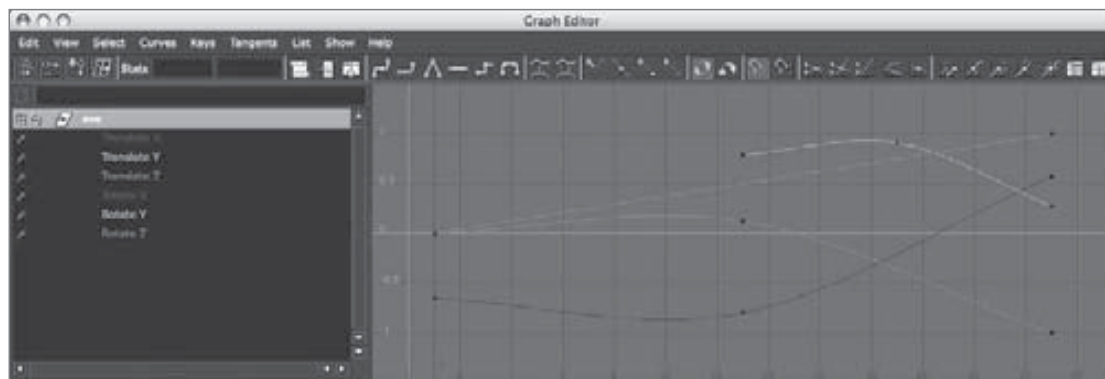

Figure 8.20

The normalized view in the Graph Editor lets you see all the curves of an animation together in the same scale.

Notice that the Scale attributes on the axe aren't shown in this window; only animated attributes appear here.

Also notice that the curve for Translate Y has keyframes only at frames 1 and 40. The animation dips in the first 15 frames because there is no keyframe at frame 15 as there is for Translate Z. That dip wasn't there before you set the end keyframe at frame 40.

Continue the exercise by fixing this issue:

1. Move the first keyframe of Translate Y from frame 1 to frame 15 to fix the dip:
   - Press W to activate the Move tool in Maya, or click the Move Nearest Picked Key Tool icon ( ) in the Graph Editor.

- Click the Time Snap On/Off icon ( ) to toggle it on.
- Select the offending Translate Y keyframe at frame 1, and MMB+click and drag it to the right until it's at frame 15.

Scrub your animation, and the backward movement looks as it did before. You might have prevented this problem by manually setting your keyframes for the axe instead of using Auto Keyframe. The more you work with Maya, the more valuable you'll find Auto Keyframe (although plenty of people get by without it just fine).

The axe now needs an arc on its way to the target.

2. Go to the middle of the axe's flight, frame 27.

3. Move the axe up in the *Y*-axis a bit using the green handle of the Tool manipulator.

   If the axe is slightly rotated in frame, Auto Keyframe can set a key for both Translate Y and Translate X, although you were perhaps expecting only a key in Translate Y. Because the Move tool is on the axe's Local axis, and because the axe was slightly rotated at frame 27, there is a change in the *Y* and *X* positions in the World axis, which is the axis represented in the Graph Editor.

4. Select the Translate X key at frame 27, if one was created, and press Delete to delete it.

5. Now you'll add a full spin to the axe to give the animation more reality and life. You can spin it in one of two ways:

   - Go to frame 40, select the axe, and rotate it clockwise a full 360 degrees positive. Auto Keyframe enters a new rotation value at frame 40, overwriting the old value. You should see the Rotate Z curve angle down steeply as soon as you let go of the Rotate manipulator.
   - In the Graph Editor, make sure you're at frame 40, grab the last keyframe on the Rotate Z curve, and MMB+click and drag it down, probably past the lower limit of the window. If you keep the middle mouse button pressed as you move the mouse, the keyframe keeps moving as you move the mouse, even if the keyframe has left the visible bounds of the Graph Editor.

If you hold down Shift as you MMB+click and drag the keyframe to move it in the Graph Editor, the keyframe will move in only one axis (up or down, left or right).

By moving the keyframe down, you change the Rotate Z value to a lower number, which spins the axe clockwise. Before you try that, though, move your Graph Editor window so you can see the axe in the Perspective window. As you move the Rotate Z keyframe down in the Graph Editor, you see the axe rotate interactively. Move the keyframe down until the axe does a full spin.

6. Play back the animation by clicking the Play button in the playback controls. If your animation looks blazingly fast, Maya's playback speed is probably set to Play Every Frame. Open the Animation Preferences window by clicking its icon (⬛), and set Playback Speed to Real-Time (30fps).

> Changing the playback speed of an animation through the Animation Preferences window doesn't alter the timing of your animation. It only changes the speed at which Maya plays the animation back to you in its windows. To change the playback speed, choose Window → Setting/Preferences → Preferences to open the Preferences window, choose Settings → Working Units, and select the proper setting.

Now, when you play back the animation, it should look slow. Maya is playing the scene back in real time, as long as the options in the Animation Preferences window are set to play back properly at 30fps. Even at 30fps, the scene should play back slowly, and this means the animation of the axe timing is too slow.

7. All you need to do is tinker in the Graph Editor a bit to get the right timing. For a good result in timing, move the first set of keyframes from 15 to 13. Then, grab the Translate Y keyframe at frame 27 and move it to 19. Finally, grab the keyframes at frame 40 and move them all back to frame 25. Play back the scene.

Adding Follow-Through

Load the `axe_v4.mb` file from the Axe project on the CD, or continue with your own file.

The axe is missing weight. Nothing much in this scene indicates that this is a heavy axe. You can add some finesse to the scene using follow-through and secondary motion to give more weight to the scene.

In the axe scene, follow-through motion is the axe blade driving farther into the target a little beyond its initial impact. Secondary motion is the recoil in the target as the momentum of the axe transfers into it. As you increase the amount of follow-through and secondary motion, you increase the axe's implied weight. You must, however, walk a fine line; you don't want to go too far with follow-through or secondary motion. The Graph Editor is great for adding these nuances to the axe. Follow these steps:

1. Select the axe in the scene using its selection handle, and open the Graph Editor.

2. Because you'll add three frames to the end of this animation for follow-through, go to frame 28 (25 being the end of the current animation).

3. In the Perspective window, rotate the axe another 1.5 degrees in the *Z*-axis.

4. Rotating the axe in step 3 moves the axe's blade down a bit in the *Y*-axis. To bring the axe back up close to where it was before the extra rotation, move the axe up slightly using the Translate Y manipulator handle. This also digs the axe into the target a

little more. You'll see a keyframe for Translate Y and most probably for Translate X, as well as Rotate Z.

If you play back the animation, the follow-through doesn't look good. The axe hits the target and then digs into it as if the action were done in two separate moves by two different animators who never talked to each other. You need to smooth out the transition from the axe strike and its follow-through in the Graph Editor.

5. Highlight the Rotate Z attribute in the Graph Editor to get rid of the other curves in the window. Figure 8.18 shows the Rotate Z curve of the axe after the follow-through animation is added.

6. Focus on the last three frames of the curve, and zoom into that range only. The curve, as it is now, dips down past where it should and recoils back up a small amount.

When you set keyframes, you create animation curves in the Graph Editor for the axe. These curves are Bézier splines, which stay as smooth as possible from beginning to end. When you set the new keyframe, rotating the axe about 1.5 more degrees for follow-through, the animation curve responds by creating a dip, as shown in Figure 8.21, to keep the whole curve as smooth as possible.

Figure 8.21

The normalized Rotate Z curve of the axe after the follow-through animation
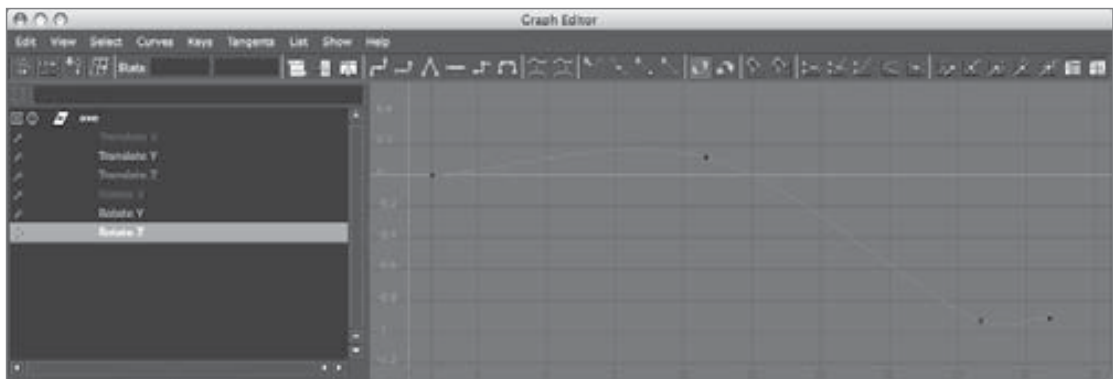
SECONDARY MOTION AND FOLLOW-THROUGH

Secondary motion in animation comprises all the little things in a scene that move because something else in the scene is moving. For example, when a superhero jumps from a tall building and his or her cape flutters in the wind, the cape's undulation is secondary motion.

Follow-through is the action in animation that immediately follows an object's or a character's main action. For example, after the superhero lands from their jump, his or her knees buckle a little, and the superhero bends at the waist, essentially squashing down a bit. That squashing motion is follow-through. The more follow-throughs, the more cartoon-like the animation appears.

The axe needs to hit the target with force and dig its way in, slowly coming to a stop. You need to adjust the curvature of the keyframes at frame 25 by using the keyframe's tangents. *Tangents* are handles that change the amount of curvature influence of a point on a b-spline (Bézier spline). Selecting the keyframe in question reveals its tangents. (See Figure 8.22.)

7. Select the Out tangent for the Rotate Z attribute's key at frame 25, and MMB+click and drag it up to get rid of the dip. Notice that the tangency for the In tangent also changes.
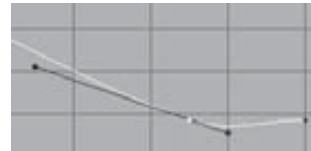
Figure 8.22

The tangent handles of a keyframe. The handle to the left of the keyframe is the In tangent, and the handle to the right is its Out tangent.

8. Press Z to undo your change. You need to break the tangent handles so that one doesn't disturb the other.

9. Select the Out handle, and click the Break Tangents icon ( ) to break the tangent.

10. Move the handle up to get rid of the dip so that the curve segment from frame 25 to frame 28 is a straight line, angled down. Figure 8.23 is zoomed into this segment of the curve after it's been fixed.

   Now, to get the axe to stop slowly as it digs into the target, you need to curve that end segment of the Rotate Z curve to flatten it out.
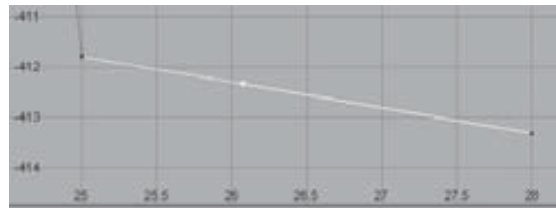

Figure 8.23

Zoomed into the end segment of the Rotate Z animation curve after the dip is fixed

11. Grab the last frame to reveal its handles. You can manually move the In handle to make it horizontal, or you can click the Flat Tangents icon ( ) on the left side of the icon bar, under the menus in the Graph Editor.

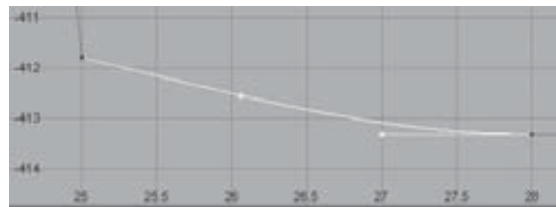   The curve's final segment for Rotate Z should now look like Figure 8.24.

Figure 8.24

Zoomed into the end segment of the Rotate Z animation curve. Notice how the curve now smoothly comes to a stop by flattening out.

12. Adjust the keyframe tangents similarly for the axe's Translate Y and Translate X curves. (See Figure 8.25.)

13. Play back the animation, and you should see the axe impact the target and sink into it a bit for its follow-through.

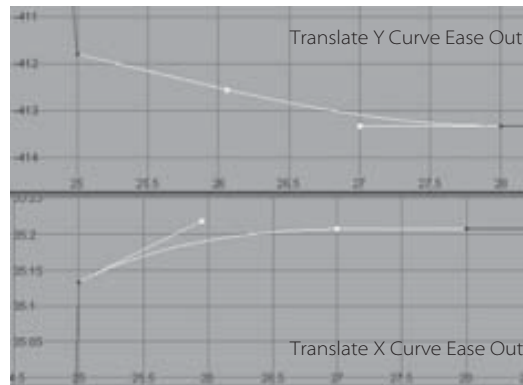   Now, you need to polish things up more.

Translate Y Curve Ease Out

Translate X Curve Ease Out

Figure 8.25

Smoothed translate curves to ease out the motion

Adding Secondary Motion

Load axe_v5.mb from the Axe project on the CD, or continue with your own scene file.

For secondary motion, you'll move the target in reaction to the impact from the axe's momentum.

An object in motion has momentum. Momentum is calculated by multiplying the mass of an object by its velocity. So, the heavier and faster an object is, the more momentum it has. When two objects collide, some or all momentum transfers from one object to the other.

In a game of pool, for example, when the moving cue ball collides with a stationary eight ball, the cue ball transfers some of its momentum into the eight ball, setting it in motion; the cue ball uses the rest of its own momentum to ricochet off in another direction.

In the axe scene's impact, the axe lodges in the target, and its momentum is almost fully transferred to the target. But because the target is much more massive than the axe, the target moves only slightly in reaction. The more you make the target recoil, the heavier the axe will seem.

First, group the axe's parent node under the target's parent node. The axe will be left behind to float in midair if you animate the target's parent node without grouping the axe under it. By grouping the axe under the target, you'll move the target to recoil while keeping the axe lodged in it. The animation on the axe won't change when you group the axe and target under a new node.

Grab the parent node of the target (called Target), and reset its attributes to 0 by freezing its transforms as you did with the ball. This sets the target node's Translate and Rotate attributes back to 0 and its Scale attributes back to 1:

1. To freeze the transforms, select the target node, and then choose Modify → Freeze Transformations.

2. In the Outliner, MMB+click and drag the axe node to the target node to group it under. You can also MMB+click and drag the nodes in the Hypergraph to group the axe node under the target node.

3. Go to frame 25, the moment of impact, and set the position and rotation keyframes on Target.

4. Go to frame 28, rotate the target node in the *Z*-axis about 2.5 degrees, and move it up and back slightly in the *Y* and *X* axes, as shown in Figure 8.26.
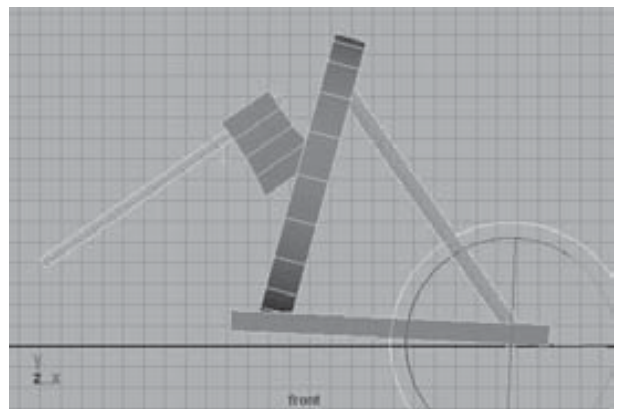

Figure 8.26

The front panel display of the target reacting to the impact of the axe

5. Go to frame 31. Rotate the target node back to 0 in the Z-axis, move it down to 0 in the *Y*-axis, and move it back a bit more in the *X*-axis.

6. Go to frame 35 and repeat step 4, but only move it half as much in Rotate Z, Translate X, and Translate Y.

7. Go to frame 40 and repeat step 5, but move Target back only slightly in the *X*-axis.

> If you don't freeze the transforms on the target's parent node before grouping the axe under it, the axe's animation will change and yield undesirable results.

The preceding steps should give you an animation similar to axe_v6.mb in the Axe project on the CD.

Motion Trails

You can see a moving object's trajectory, or *motion trail*—that is, its path of motion. Follow these steps:

1. Select the axe through its selection handle, and then choose Animate → Create Motion Trail ❑. (See Figure 8.27.)

2. Select Line for Draw Style, and make sure the Show Frame Numbers check box is checked.

3. Click Apply, and then click Close to close the window.

Figure 8.27
The Motion Trail Options window

The motion trail is useful for fine-tuning motion. Editing the animation curves in the Graph Editor and watching the motion trail adjust in the work panels shows you the precise trajectory of the axe throughout its movement. Play back your animation a few times to get a good sense of how the scene looks.
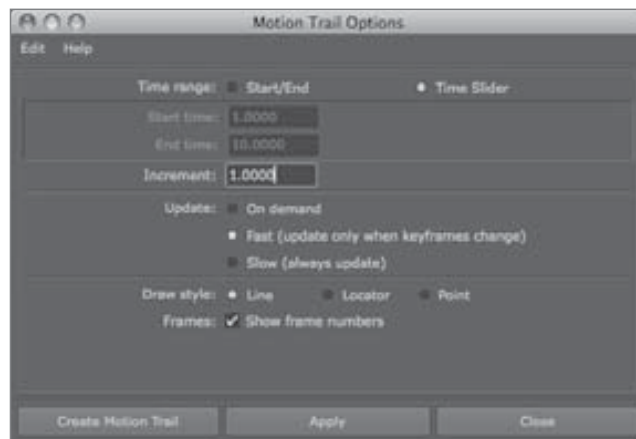
4. Select the axe, and open the Graph Editor.

5. Try adding more arc to the axe in the middle of its trajectory to the target.

6. In the Graph Editor, focus on the Translate Y curve, and select the keyframe at frame 19.

7. Move the keyframe up about 2 units, and watch the motion trail adjust to show you the higher arc.

8. Replay the animation with the higher arc in the middle.

Notice that the axe seems a little more solid than before. The extra height in the trajectory really helps give the axe more substance. Figure 8.28 shows the axe and its motion trail after more height is added to its arc.

Figure 8.28

The axe and its motion trail
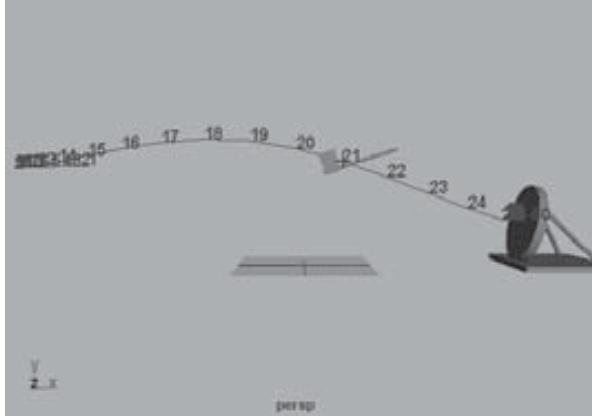
You can toggle the frame-number display on/off and change the display type of the motion trail from curve to points or locators through the motion trail's Attribute Editor. To get rid of the motion trail, select it and press Delete.
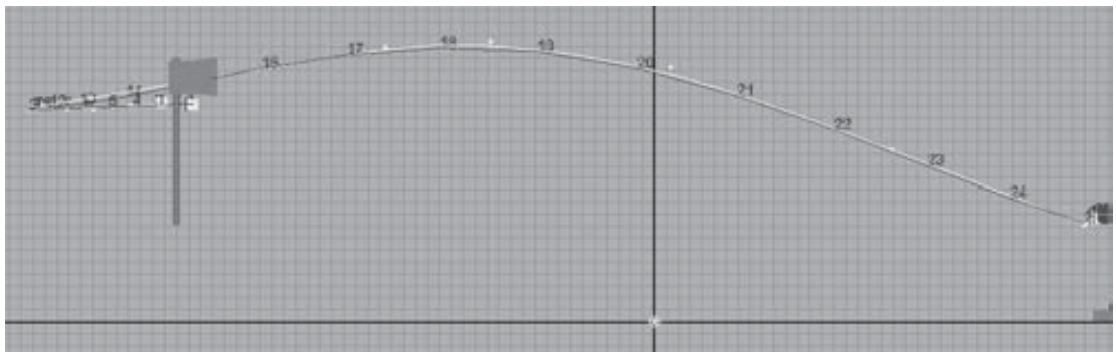
Path Animation

As an alternative to keyframing the position of the axe, you can animate it on a path. *Path animation* allows you to assign an object to move along the course of a curve, called a *path*.

Load axe_v7.mb from the Axe project on the CD. This is the finished axe animation. You'll delete most of your hard work by removing the translation animation on the axe, but you'll keep the rotation and everything else. You'll replace the translation keyframes you set up with a motion path instead. Follow these steps:

1. In the front window, and with the motion trails turned on, trace the motion trail with a CV curve (choose Create → CV Curve Tool) from the beginning of the trail to the end, as shown in Figure 8.29. Make sure the CV Curve tool is set to make cubic (3) curves.

Figure 8.29

Trace the motion trail.

2. Take note of the frames on which you set important keyframes for the axe's position (for example, when it recoils and releases at frame 10 and when it hits the target at frame 25). Select the axe's top node (axe), and delete its translation animation. Select all three Translate attributes in the Channel Box, and right-click to display the short-cut menu. Choose Break Connections to delete the animation from those channels. The axe now spins around, and it and the target recoil at the moment of impact, but the axe doesn't actually move.

3. Keep the motion trail in the scene for now to help with the timing you've already created.

4. Select the top axe node, and then Shift+select the path curve. In the Animation menu set, choose Animate → Motion Paths → Attach to Motion Path ❏.

5. In the Options window, turn off the Follow check box.

   The Follow feature orients the object on the path so that its front always points in the direction of travel. Because the axe moves backward in anticipation before it's thrown forward, Follow would cause it to turn around twice, so turn this option off.

   Now the axe will follow the curve end to end from frame 1 to frame 60. Of course, you have to adjust the timing to fit it, as before.

6. Select the motion trail, and move it down in the window to get it out of the way; but keep it lined up vertically with the path curve so you can figure out the timing again.

   The file axe_path_v1.mb in the Axe project will bring you up to this point.

7. Select the top axe node, and open the Graph Editor to see the axe's curves. The rotation curve is still intact, although the translation curves are missing. Click motion-Path1 in the Channel Box to highlight it; the motionPath1.U Value curve that took the place of the translation curves appears in the Graph Editor. On the left side of the Graph Editor, select the motionPath1.U Value curve to display only that. Zoom into it. (Press A to view all.)

8. The curve is an even, linear curve from 1 to 60. You need the axe to hit at frame 25, so move the end of the curve to frame 25 from frame 60.

9. Retime the backward movement. Scrub the animation until the axe moves all the way back (frame 4). Using the Insert Keys tool (▨), insert a keyframe at frame 4. Select the animation curve, click the tool, and MMB+click the curve to create a key on it. You can MMB+click and drag the cursor to place the key precisely at frame 10 before releasing the mouse button.

10. Move this new keyframe to frame 10 (the frame where backward movement originally ended).

11. Scrub the animation, and the timing is just about right. You'll have to adjust the tangents a bit to make the axe move more like before, but the movement is essentially there with path animation.

The file `axe_path_v2.mb` in the Axe project will bring you up to this point.

Path animation is extremely useful for a number of tasks, but particularly for animating an object along a particular course. By adjusting the resulting animation curve in the Graph Editor, you can readjust the timing of the path animation easily.

A good path-animation exercise is to create an atom and draw CV curves around the nucleus for the paths of the electrons. Then, animate all the electrons orbiting the nucleus with the paths. Also, try reanimating the Solar_System exercise with paths instead of the keyframes you set on the rotations.

> **GHOSTING**
>
> To see the position of an animated object a few frames before and after its current position, you can enable ghosting in Maya. For example, select the animated axe in your scene and, in the Animation menu set, choose Animate → Ghost Selected. Maya will display the axe's three frames before and after the current time. To turn off ghosting, choose Animate → Unghost Selected.

Axe Project Summation

In the Axe example, you furthered your use of layered animation by beginning with the gross animation to cover the basic movements of the axe. After those timings were set, you completed most of the remaining work in the Graph Editor by moving keyframes here and there to add detail to the motion. You added more keyframes to create follow-through and secondary movement to insinuate weight into the axe and target.

Without secondary movement in the target, the axe would seem to weigh nothing. With too much movement, however, the axe would seem too heavy, and the scene wouldn't look right. Subtle nuances can make stunning differences in the simplest of animations. You also went back into the animation and replaced the animation method entirely with path animation. This illustrates the multiple ways to accomplish a task in Maya; finding your own comfort zone with a workflow is one of the goals in learning Maya.

Replacing an Object

Aside from the need to model objects and texture them, there is the task of animation setup. Check your pivots, your geometry, and your grouping to make sure your scene will hold up when you animate it.

It's also common practice in setup to animate a *proxy* object—a simple stand-in model that you later replace. The next exercise will show you how to replace the axe you already animated with the fully textured NURBS axe from the previous chapter and how to copy an animation from one object to another.

Replacing the Axe

Load your completed, keyframed axe-animation scene (not the one using path animation), or switch to the Axe project and load the scene file `axe_v7.mb` from the CD. Now, follow these steps:

1. Choose File → Import.

2. Locate and import the `Axe_Replace.ma` scene file from the `Scenes` folder of the Axe project on the CD. The new axe appears at the origin in your scene.

Transferring Animation

To assume all the properties and actions of the original axe requires some setup. Follow these steps:

1. Move the pivot on the new axe to the same relative position as the pivot on the original animated axe (up toward the top and a little out front of the handle, just under the blade). This ensures that the new axe has the same spin as the old axe. If you don't make sure the pivots line up, the animation won't look right when transferred to the new axe.

2. Rotate the new axe's top node 180 degrees in *Y* to get it to face the right direction.

3. Use grid snap to place the top node of the new axe at the origin.

4. Freeze transforms on the new axe group to reset all its attributes.

5. Choose Display → Transform Display → Selection Handles to turn on the selection handle of the new axe.

6. Go to frame 1. Select the original axe, open the Graph Editor, and choose Edit → Copy.

7. Select the new axe to display its curves in the Graph Editor. It has no curves to display yet. With the axe node selected in the Graph Editor, choose Edit → Paste. As shown in Figure 8.30, the new axe is slightly offset from the original axe.

> When you copy and paste curves in the Graph Editor, make sure you're on the first frame of the animation. Pasting curves places them at the current frame. Because the animation of the original axe started at frame 1, make sure you're at frame 1 when you paste the curves to the new axe.

8. Move the new axe to match the original. Because it's already animated, move it using the curves in the Graph Editor as opposed to moving it in the viewport. With the new axe selected, in the Graph Editor, select the Translate Y curve; move it up to match the height of the original axe.

9. Select the Translate X curve, and move it to match the axe's *X* position, as shown in Figure 8.31. Line up the axe handle as a guide.
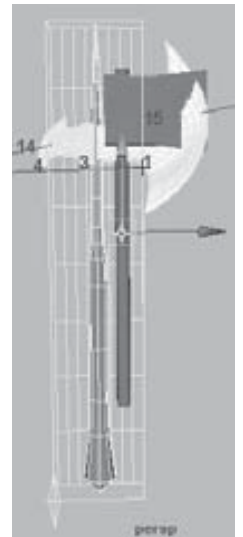
Figure 8.30

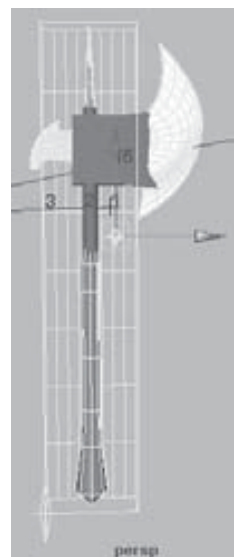**The new axe placed next to the original**

Figure 8.31

Lining up the two axes

Scrub the animation, and notice that the new axe has the same animation except at the end when it hits the target. It doesn't have the same follow-through as the original axe. Remember that you grouped the original axe under the target node for follow-through animation. Place the new axe under this node as well.

The file axe_v8.mb in the Axe project on the CD has the new axe imported and all the animation copied. It will get you caught up to this point.

10. After you scrub the animation and make sure the new axe animates properly, select the original axe's top node and delete it.

Animating Flying Text

It's inevitable. Sooner or later, someone will ask you to animate a flying logo or flying text for something or other. As late-1980s as that may sound, animating flying text—at least, the way you'll do it here—can teach you a thing or two about path animation and lattice deformers. You can use the following steps to animate pretty much anything that has to twist, wind, and bend along a path; this technique isn't just for text.

First, you'll need to create the text. To do so, follow these steps:

1. In a new scene, select Create → Text ❑. In the option box, enter your text, and select a font to use (see Figure 8.32). In this case, stick with Times New Roman. Set the Type to Bevel.

> Setting the Type attribute to Poly for your text creates curve outlines for the text and planar faces for the letters, for a flat-text effect. Setting the Type Creation option to Curves gives you just the curve outlines. Finally, using Trim to create your text makes the letters out of flat planar NURBS surfaces. However, no surface history is created with text. To allow you to edit the text later, you must re-create the text and/or font type as needed.

2. Leave the rest of the creation methods at their defaults. Doing so creates the text as beveled faces to make the 3D text, as shown in Figure 8.33.

3. When you have the text, you need to create a curve for it to animate along. Using either the CV or EP Curve tool, create a winding curve like a roller coaster for the logo, as shown in Figure 8.34.

4. Just as you did with the axe exercise, you'll assign the text object to this curve. Set your frame range to 1–100. Select the text, Shift+select the curve, and, in the Animation menu set, choose Animate → Motion Paths → Attach to Motion Path ❑. Set the Front Axis to *X* and the Up Axis to *Y*, and select the Bank check box, as shown in Figure 8.35.

Depending on how you create your curve and text, you may need to experiment with the Front Axis and Up Axis attributes to get the text to fly the way you want. Notice that history is created with the path animation, so you can adjust the axes' attributes after the fact to see how they work on the curve you have.
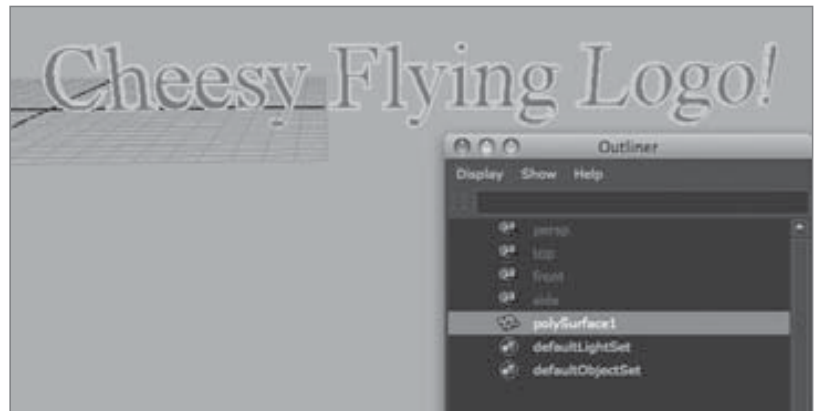
Figure 8.32
The Text Curves Options window

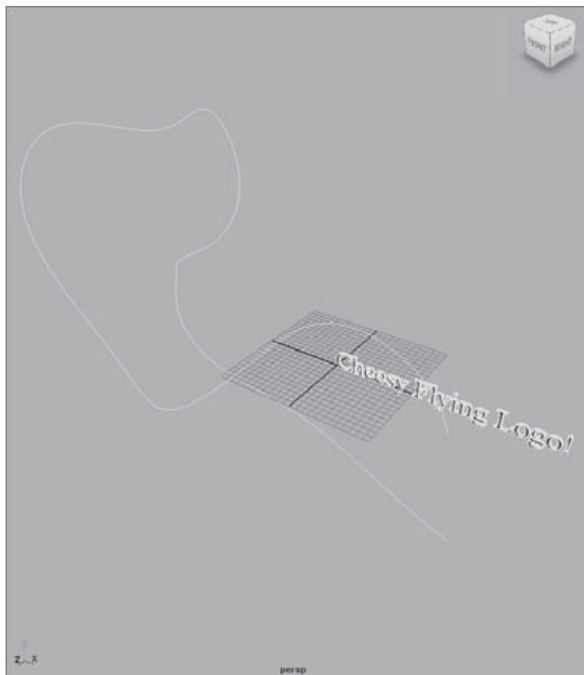Figure 8.33
Creating the 3D text for the flying logo

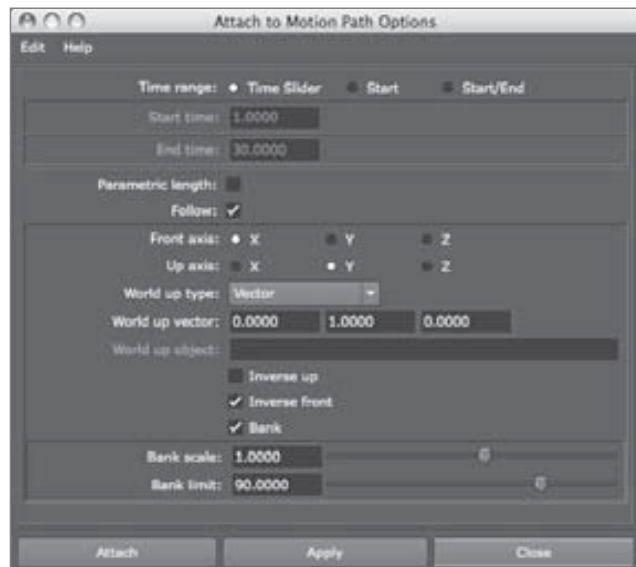

Figure 8.34
Create a curve path for the text to follow.

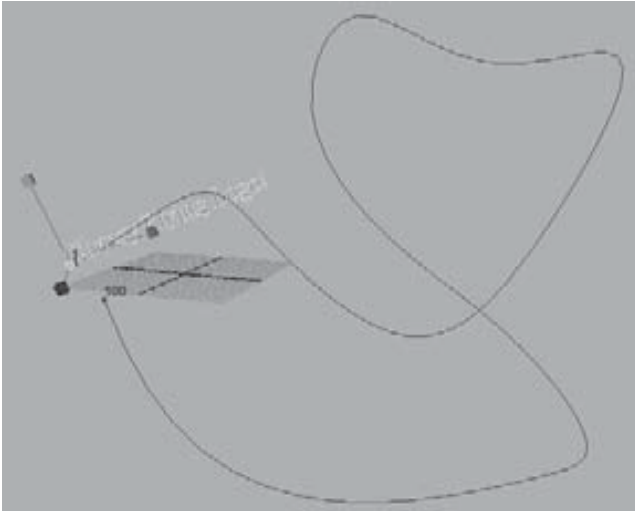

Figure 8.35
The Attach to Motion Path Options window

Figure 8.36
The text is on the path.
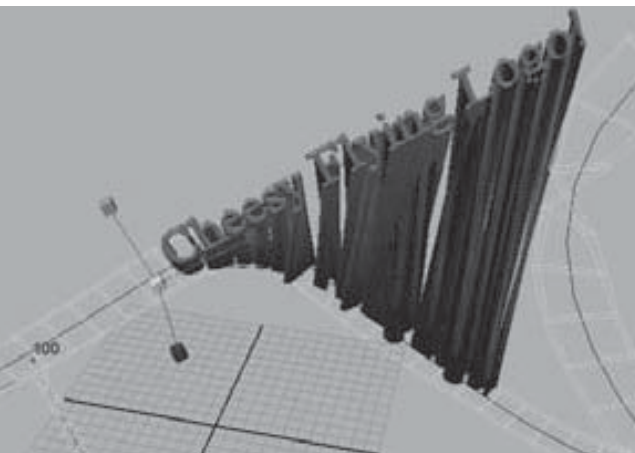

Figure 8.37
The Flow Path Object Options window

Figure 8.38
The geometry doesn't fit the lattice well.

5. Orbit the camera around to the other side, and you can see the text on the path, as shown in Figure 8.36. Notice the U Value attribute: this is the position of the text along the curve from 0 to 1. Scrub the animation, and the text should glide along the curve.

6. The text isn't bending along with the curve at all yet. To accomplish this, you need to add a lattice that bends the text to the curvature of the path. Select the text object, and choose Animate → Motion Paths → Flow Path Object ❑.

7. In the option box for the Flow Path Object, shown in Figure 8.37, set Divisions: Front to 120, Up to 2, and Side to 2, and select Curve for Lattice Around. Make sure the Local Effect box is unchecked. Doing so creates a lattice that follows the curve, giving it 120 segments along the path. This lattice deforms the text as it travels along the path.

8. Scrub your animation, and you see a fairly strange result: parts of the text explode out from the lattice, as shown in Figure 8.38.

9. The geometry is going outside the influence of the lattice, and this is causing the strange behavior. To fix the situation, select the lattice and base node, and scale up the lattice *and* its base node together to create a larger size of influence around the path (see Figure 8.39).

Scrub your animation to check the frame range and how well the text flies through the lattice. When the lattice and its base are large enough to handle the text along all of the path's corners and turns, voilà—Cheesy Flying Logo! (See Figure 8.40.)
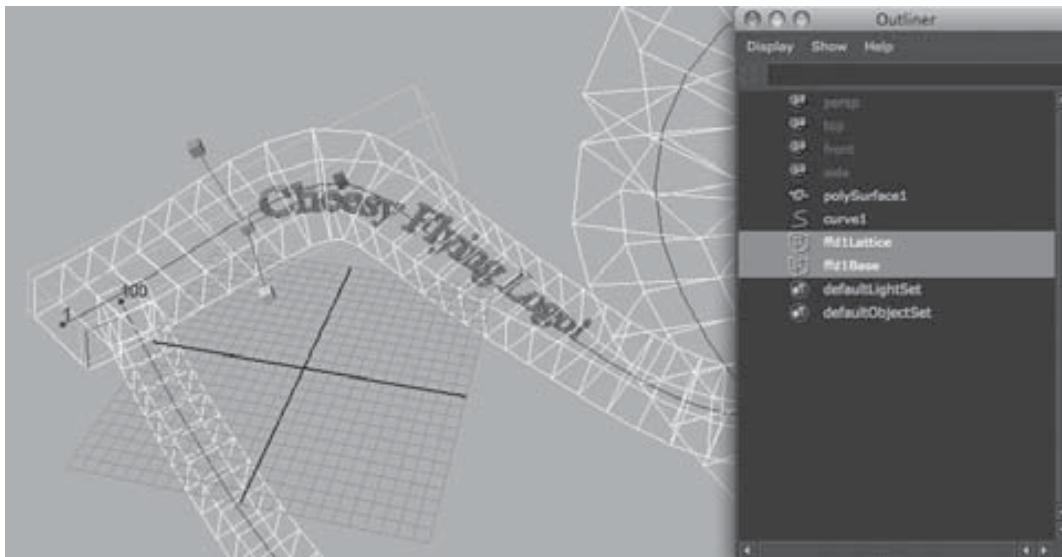
Figure 8.39

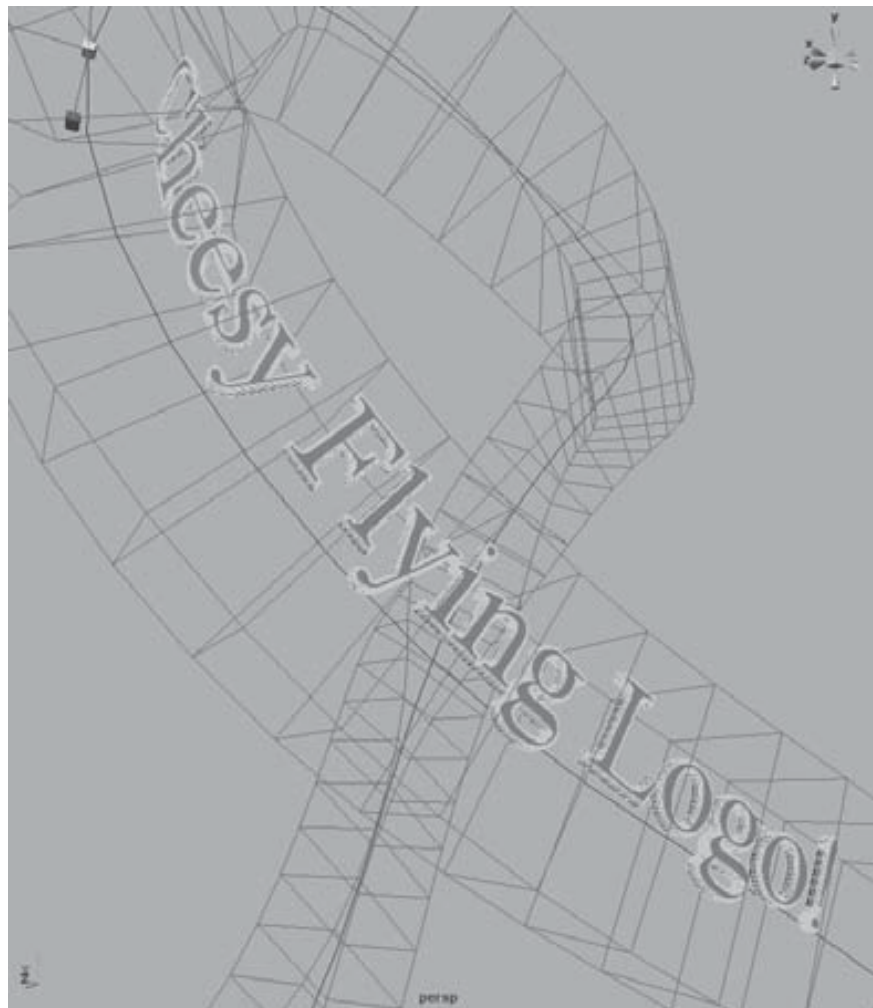**Scale the lattice and its base node to accommodate the flying text.**

Figure 8.40

Cheesy Flying Logo! makes a nice turn.

Rigging the Locomotive, Part 1

In this section, you'll return to the locomotive you modeled in Chapter 4, "Beginning Polygonal Modeling," and put your new animation skills to use.

Load the file `locomotive_model_v4.mb`, which is the completed model of the locomotive from the Polygon Modeling exercise in Chapter 4. You can find it in the `Scenes` folder of the Locomotive project on the CD.

The Scene Setup

It's important to keep animation in mind as you build a model. In particular, making a good scene hierarchy is crucial to getting a smooth animation workflow going. That's

Figure 8.41

The locomotive's Outliner view

why, when you built the locomotive in Chapter 4, you used the Outliner to make sure the objects in your scene were grouped logically. Now you're going back in to make sure you have good organization before beginning the task of rigging this thing for animation.
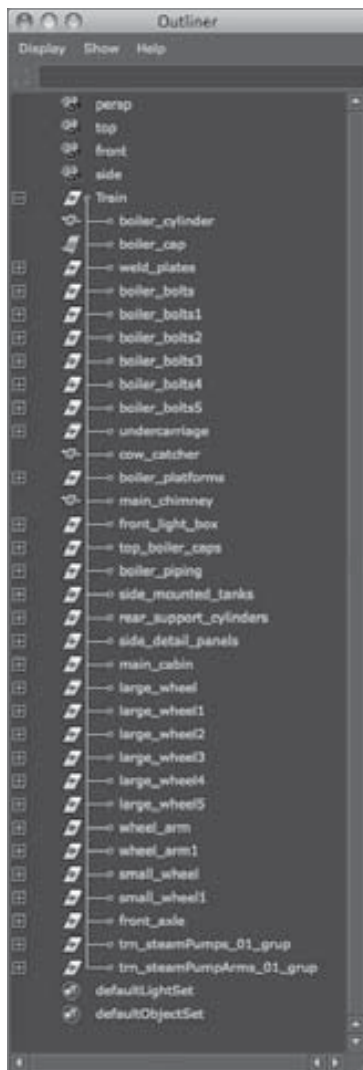
Figure 8.41 shows the organization of the locomotive as it stands from Chapter 4. Later, you'll use a more finely detailed and prettier locomotive model to rig that is already grouped properly, using the intentions outlined here. This scene is in fairly good shape, but you have to identify the parts of the locomotive that you need to rig, determine how they will move, and decide what is the best hierarchy for the model from there.

The major moving parts of the engine are the wheels and the steam pump drive mechanism. The wheels, of course, need to rotate, as wheels do, and the steam pump drives the wheel arms back and forth, which is what drives the main wheels to rotate. Aside from animating some steam pumping out of parts of the engine, which we'll cover in Chapter 12, "Maya Dynamics and Effects," this is the basic rigging for the locomotive. First, you need to make sure the hierarchies are settled well and the pivot points are in their proper places.

The `locomotive_model_v4.mb` scene file has a well-organized hierarchy, but it needs some help to make rigging easier. Be sure all the individual wheel arms have their pivots at the base, as shown in Figure 8.42. Use Insert (or the Home key on a Mac—MacBook users also need to press the fn [function] key to access the Home function using

the arrow key) to move the pivots as needed. You'll rig these wheel arms using IK bones in the next chapter.

Be sure all the elements that make up a wheel are grouped and that the group's pivot point is centered for all the wheel groups, as shown in Figure 8.43.

Finally, for this simple rig, make sure the steam pump arm elements are properly grouped and that the pivot is placed as shown in Figure 8.44.

If your locomotive is in organizational disarray, identify all the moving parts first, and then begin grouping them logically. Place pivots appropriately, and you're all set. The scene file locomotive_anim_v1.mb in the Locomotive project on the CD has the locomotive scene file with the pivots and grouping already finished, as discussed earlier.
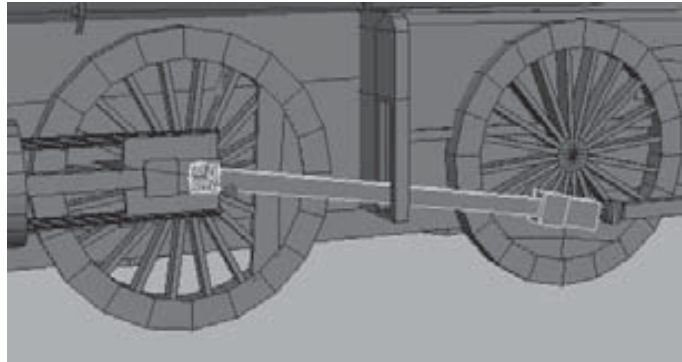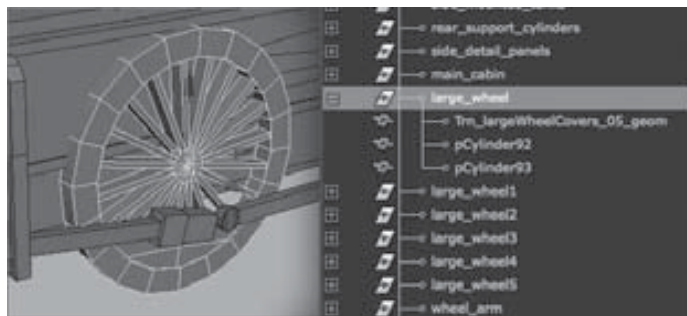

Figure 8.42

Check the pivot points.

Figure 8.43

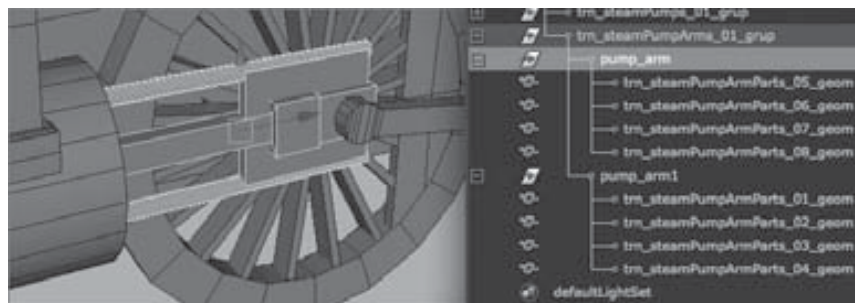**Check the hierarchy and pivot placement for the wheel.**

Figure 8.44

Place the pivot for the pump arm.

If the pivots and groupings are off, you'll notice as soon as you begin to animate; things just won't rotate on the correct axis, and pieces won't follow properly. The wheels, for example, may wobble around their axle.

For more practice in grouping and hierarchies, you can load locomotive_anim_v1.mb from the CD, ungroup everything in the scene, and piece it all back together. To ungroup everything, select the top node of the locomotive, and choose Edit → Ungroup. Doing so ungroups the major parts of the locomotive. With those groups selected, ungroup again to flush out individual geometry. You can also load locomotive_anim_v1_B.mb from the CD; it has all the major groupings removed. Then, regroup and repivot everything. This will be great practice and boatloads of fun!

Selection Handles

Using selection handles makes selection easier and work flow faster, so turn on selection handles for each of the groups you're animating. Select the wheel groups in the Outliner, and then choose Display → Transform Display → Selection Handles. Select the wheel arm

groups, and turn on their selection handles as well. Figure 8.45 shows the selection handles enabled for the wheel, pump arm, and wheel arm.

How selection handles work depends on Maya's selection order. *Selection order*, which you can customize, sets the priority of one type of object over another when you try to select in a work window.
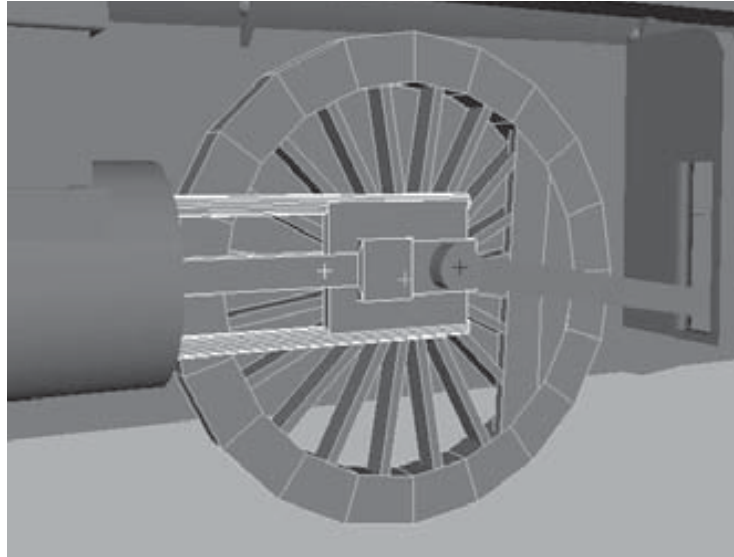
After the selection handles for the locomotive's wheels and drive arms are turned on, only the handles will be selected (and not the whole locomotive) when you make a marquee selection that covers the entire locomotive. Handles have a high selection priority by default, so they're selected above anything else.

Animating the Locomotive

Animating most of the locomotive is straightforward. Simple rotations will make the wheels turn. Translating the locomotive's top node will move the entire object. This, however, leaves out the drive mechanism with the steam pump and the wheel and pump arms. You'll rig these to animate automatically in the next chapter, using IK handles and connections.

Animating a Catapult

As an exercise in animating a system of parts of a model, before you continue to animate the locomotive in the next chapter, you'll now animate a catapult. This catapult model is straightforward to animate. You'll turn its winch to bend back the catapult arm, which shoots the projectiles, and then you'll fire and watch the arm fly up.

First, let's get acquainted with the scene file and make sure its pivots and hierarchies are set up properly. The scene file `catapult_anim_v1.mb` in the Catapult_Anim project on the CD has everything in order, although it's always good to make sure. Figure 8.46 shows the catapult with its winch selected and ready to animate.
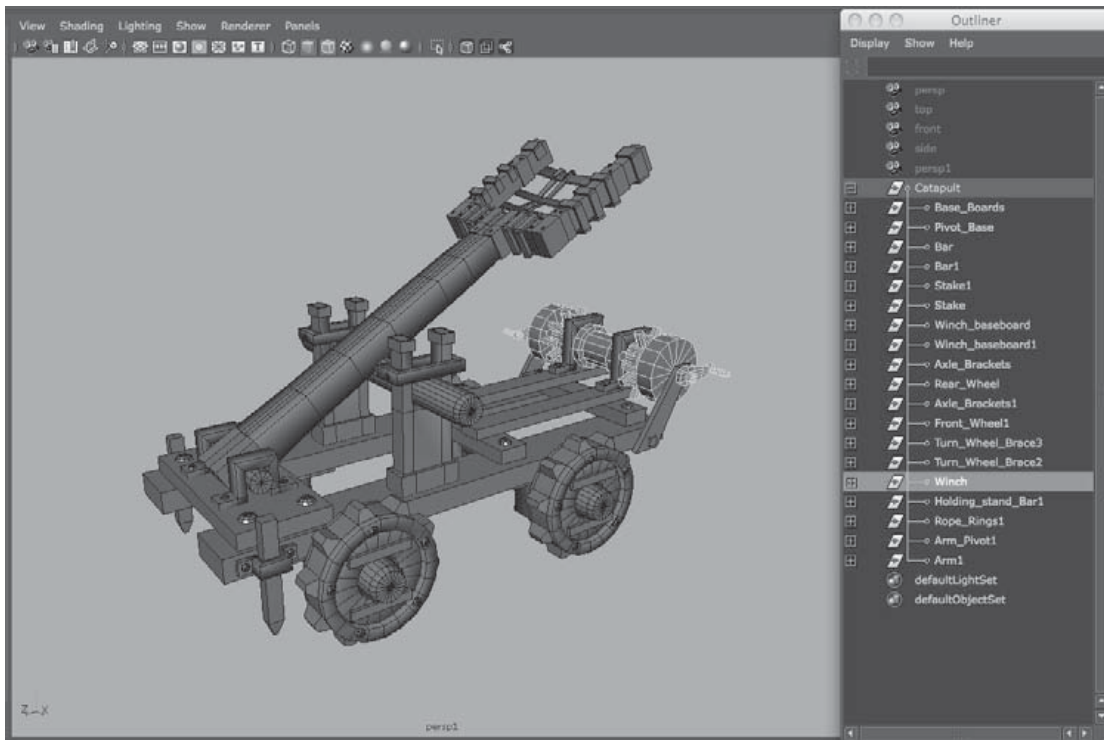
Figure 8.46

The catapult's winch is ready to animate.

Get a timing put down for the winch first, and use that to pull back the arm to fire. Follow these steps:

1. Select the Winch group with its selection handle. At frame 1, set a keyframe for rotation. If the selection handle isn't turned on, select Winch from the Outliner and turn on the selection handle by choosing Display → Transform Display → Selection Handles. To keep it clean as you go along, instead of pressing Shift+E to set a key for all three axes of rotation, select only the Rotate X attribute in the Channel Box, right-click to open the shortcut menu, and choose Key Selected. There only needs to be rotation in *X* for the winch.

2. Jump to frame 60.

3. Rotate the winch backward a few times, or enter **−400** or so for the Rotate X attribute.

4. Open the Graph Editor, ease in the curve a bit, and ease out the curve a lot so that the rotation starts casually but grinds to a stop as the arm becomes more difficult to pull back.

   Obviously, you're missing the rope between the winch and the arm. Because animating a rope is a fairly advanced task, the catapult is animated without its rope; but the principle of an imaginary rope pulling the arm down to create tension in the arm drives the animation.

5. To accentuate the more difficult winding at the end, add a key to the *X*-axis rotation through the Graph Editor. To do so, select the curve, and click the Insert Keys Tool icon ( ) in the upper-left corner of the Graph Editor. Your cursor changes to a cross.

6. MMB+click frame 42 to add a keyframe already on the curve at frame 42. You can drag the key back and forth on the curve to place it directly at frame 42. It may help to turn on key snapping first. (See Figure 8.47.)

7. Move that keyframe down to create a stronger ease-out for the winch. Be careful not to let the curve dip down so that the winch switches directions. Adjust the handles to smooth the curve. You can also add a little recoil to the winch by inserting a new keyframe through the Graph Editor at frame 70. (See Figure 8.48.)

Figure 8.47

Insert a keyframe at frame 42.

Figure 8.48

Creating a greater ease-out and adding a little recoil at the end
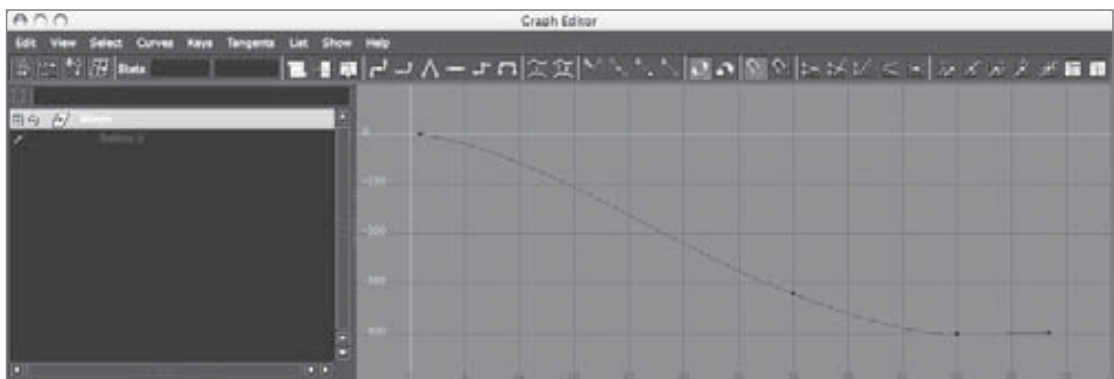
Animating with Deformers

It's time to animate the arm coiling back, using the winch's timing as it's driving the arm. Because the catapult's arm is supported by a brace, and the whole idea of a catapult is based on tension, you have to bend the arm back as the winch pulls it.

You'll use a nonlinear deformer, just as you did in the axe head exercise in Chapter 5, "Modeling with NURBS, Subdivisions, and Deformers"; but this time, you'll animate the deformer to create the bending of the catapult arm. Follow these steps:

1. Switch to the Animation menu set. Select the Arm1 group, and choose Create Deformers → Nonlinear → Bend to create a Bend deformer perpendicular to the arm. Select the deformer, and rotate it to line it up with the arm, as shown in Figure 8.49.
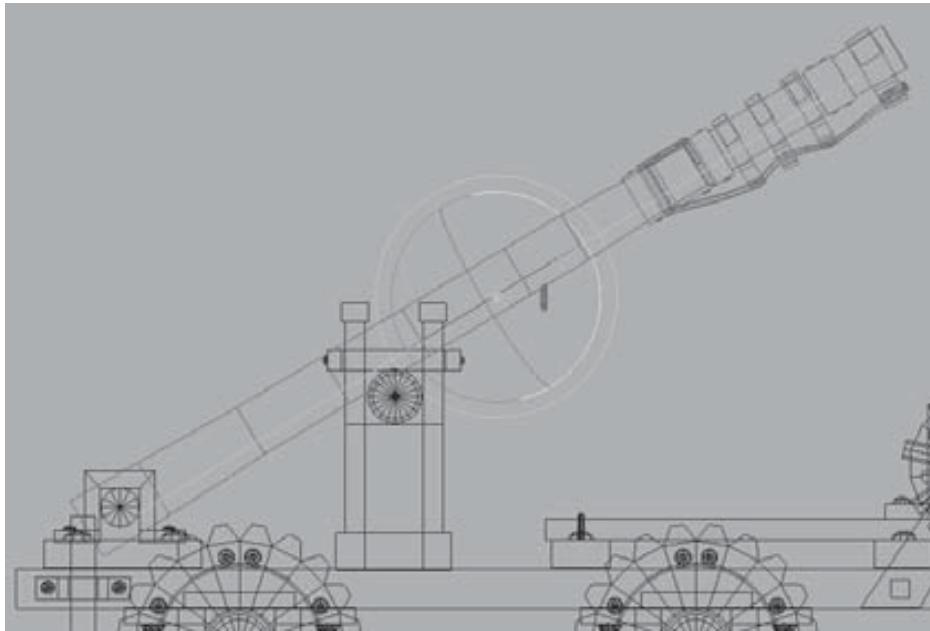

Figure 8.49

Align the Bend deformer with the catapult arm.

2. With the Bend deformer selected, look in the Channel Box for bend1 under the Inputs section, and click it to expand its attributes. Try entering **0.5** for Curvature. More than likely, the catapult arm will bend sideways. Rotate the deformer so that the arm is bending back and down instead. (See Figure 8.50.)
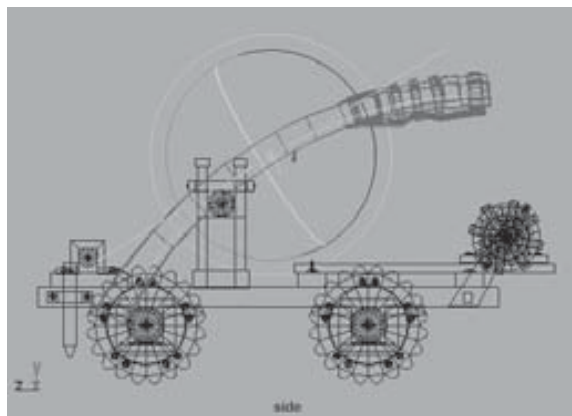

Figure 8.50

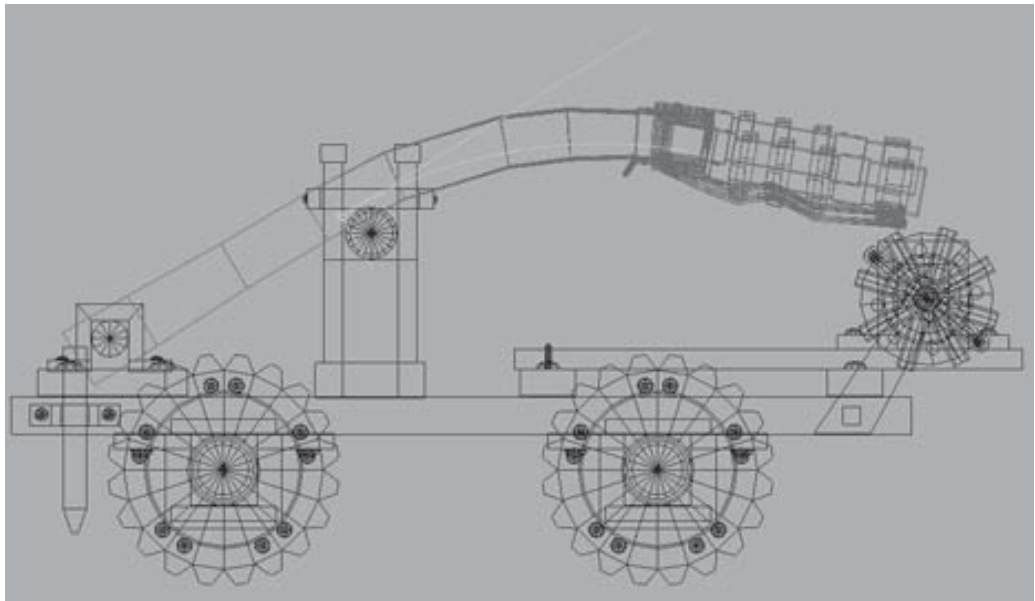**Orient the Bend deformer to bend the arm back and down.**

3. You don't want the arm's base to bend back, just the basket side.
You want it to bend at the brace point, not in the middle where it is now. Move the deformer down the length of the arm until the middle lines up with the arm's support brace.

4. To prevent the bottom of the arm from bending, change the Low Bound attribute to 0. To keep the basket from bending, set the High Bound attribute to 0.9.

The Low Bound and High Bound attributes control how far up and down the deformer the object is affected. The Envelope attribute for a deformer governs how much the object is affected overall, with 0 not affecting the geometry at all.

5. Instead of trying to match the speed, ease in and out of the winch and set the gross keyframes for the arm pulling back first. Reset Curvature to 0, and set a key for Curvature at frame 1. (Select bend1's Curvature in the Channel Box, right-click, and choose Key Selected from the shortcut menu.)

6. Go to frame 60, and set Curvature to 0.8. If Auto Key is turned on, this sets a keyframe; otherwise, set a key manually. (See Figure 8.51.)

Figure 8.51

Bend the arm back at frame 60 and set a keyframe.

7. If you play back the animation, notice that the way the winch winds back and the way the arm bends don't match. In the Graph Editor, you can adjust the animation curve on the Bend deformer to match the winch's curve.

8. Insert a key on the Curvature curve at frame 42, and move it up to match the curvature you created for the winch.

9. Insert a new key at frame 70, and make the arm bend back up slightly as the winch recoils. Set Curvature to about 0.79 from 0.8. (See Figure 8.52.)

10. Go to frame 90, and set a key again at Curvature of 0. Set a key at 0.82 for frame 97 to create anticipation, and then keyframe at frame 103 to release the arm and fire the imaginary payload with a Curvature of −0.8.
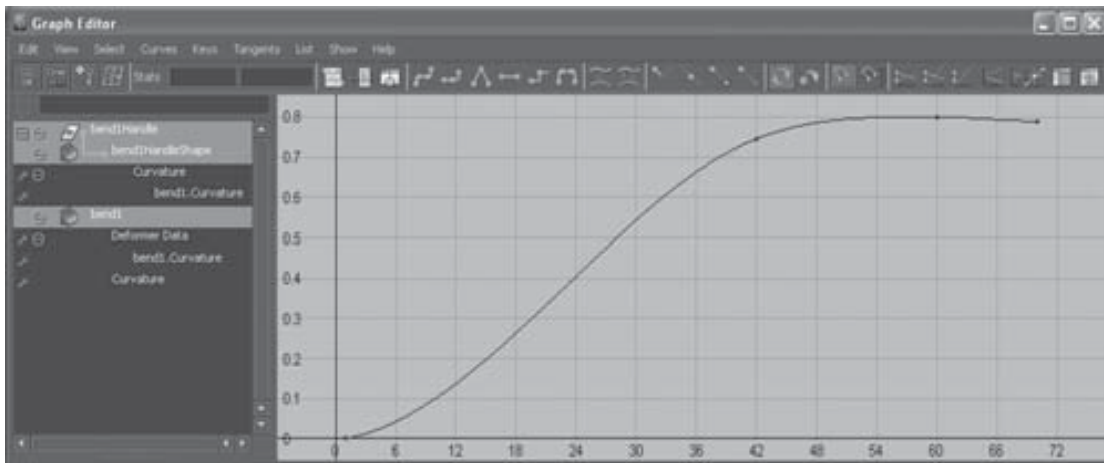
Figure 8.52

Try to match the relative curvature of the winch's animation curve with the Bend deformer's animation curve.

11. Add some rotation to the arm for dramatic effect. At about frame 100, during the release, the arm is almost straight. Select the Arm1 group, and set a rotation key on the *X*-axis. Go to frame 105, and rotate the arm 45 degrees to the left in the *X*-axis. If the starting rotation of the arm is at 30 (as it is in the sample file), set an *X*-axis rotation key of 75 at frame 105.

12. Notice that the arm is bending strangely now that it's being rotated. It's moving off the deformer, so its influence is changing for the worse. To fix this, go back to frame 100, and group the deformer node (called *bend1Handle*) under the Arm1 group, as shown in Figure 8.53. Now it rotates along with the arm, adding its own bending influence.

13. Work on setting keyframes on the deformer and the arm's rotation so that the arm falls back down onto the support brace and quivers until it becomes straight again. The animation curve for the Bend deformer should look like Figure 8.54. The rotation of the arm should look like Figure 8.55. Remember to make the tangents flat on the keys where the arm bounces off the brace linear and the peaks, like the ball's bounce from earlier in the chapter.
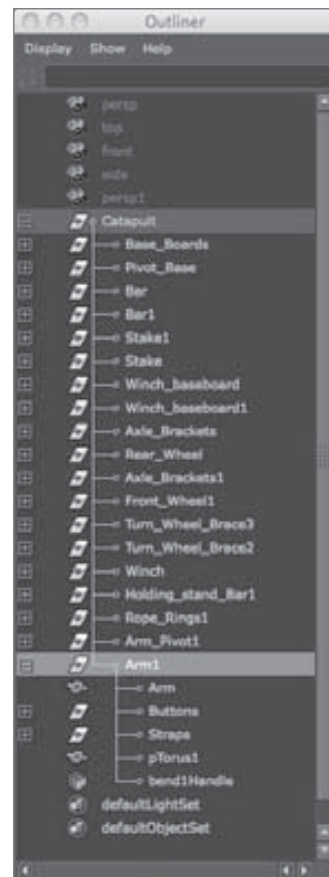

Figure 8.53

Group the deformer node under the Arm1 group node.

The file `catapult_anim_v2.mb` will give you a good reference to check out the timing of the arm bend and rotation.

Figure 8.54

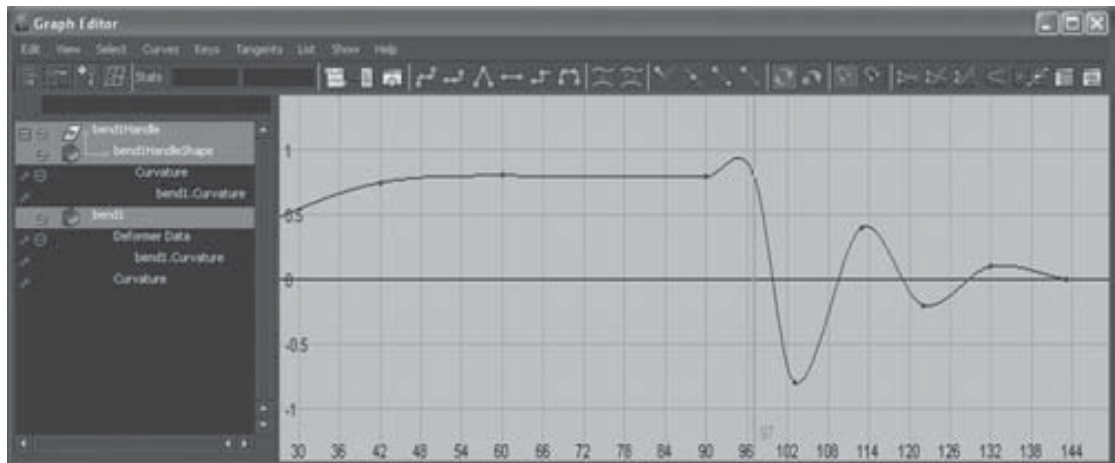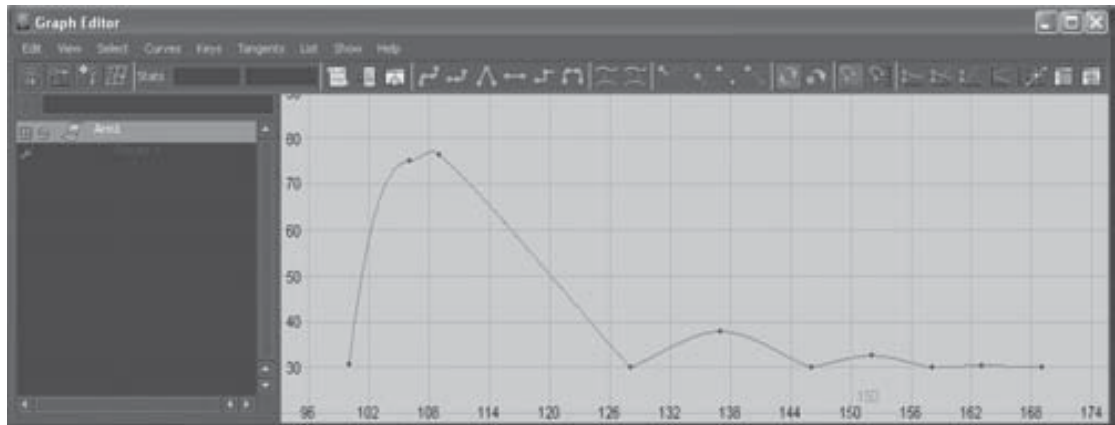**The animation curve for the arm's vibration back and forth as it comes to a rest**

Figure 8.55

The animation curve for the arm's rotation as it heaves up and falls back down, coming to an easy rest on the brace

Without getting into a lot more detail, try your hand at animating the catapult on your own. Here are some items you can animate to make this a complete animation:

- Spin the winch around as the arm releases, as if its rope is being yanked away from it.

- Animate the entire catapult rocking forward and backward as the arm releases, similar to the way a car rocks when you jump onto the hood.

- Move the catapult forward on a road, spinning its wheels as best you can to match the distance it travels.

- Design and build your own catapult, and animate it along the same lines.

Summary

In this chapter, you began to learn the fundamentals of animating a scene. Starting with a bouncing ball, you learned how to work in the Graph Editor to set up and adjust timing as well as how to add squash and stretch to the animation. The next exercise, throwing an axe, showed you how to set up a scene for animation, expanded on your experience

in creating timing in the Graph Editor, and showed you how to add anticipation, follow-through, and secondary motion to your scene. You then learned how to adjust animation using motion trails and how to animate the axe throw using path animation. You went on to learn how to replace a proxy object that is already animated with a different finished model and how to transfer the animation. Going back to the locomotive from Chapter 4, you began to set up the scene for rigging in the next chapter by setting proper pivots and hierarchy. Finally, you used a catapult to animate with deformers and further your experience in the Graph Editor.

Animating a complex system, such as a catapult or a locomotive, involves creating layers of animation based on facets of the mechanics of the system's movement. With the catapult, you tackled the individual parts separately and then worked to unify the animations. You'll use rigging concepts in the next chapter to automate some of that process for the locomotive.

The same is true of the Bouncing_Ball and Axe_Throwing exercises. The different needs of the animation were addressed one by one, starting with the gross animation and ending with finishing touches to add weight. Finally, the art of timing brought the entire effort into a cohesive whole.

Even when animation is already applied, it's simple to change *how* the animation is accomplished, as you did with path animation, or even to replace the animated object entirely.

Animation is the art of observation, interpretation, and implementation. Learning to see how things move, deciphering why they move as they do, and then applying all that to your Maya scene is what animation is all about.