Grad student by day, Blenderhead by night (well, sometimes grad student by night, too), Ryan uses Blender partly for figures in scientific presentations but mostly for playing around with character animation. Ryan participated in the first Blender Summer of Documentation by writing the Introduction to Character Animation tutorial. The full version of the tutorial (only part of which is in this book) can be found on the Blender wiki and covers the workflow of bringing a character to life from modeling and rigging all the way through animating and adding sound.
Introduction

In this tutorial, you'll use some of Blender’s animation tools to create an action: a wave of the arm. This very simple action will be blended later with a more complex animation.

The Action Editor is where you create individual actions: blinking the eyes, nodding the head, a walkcycle, and so on. Later, you can mix the actions in another window called the NLA Editor. While complex “acting” for the main characters in an animation should probably be done in a single Action, the NLA is excellent for building variety in characters that do not hold the main focus of scene.

In the “examples” folder on the included disk, find the file called “characteranimation.blend” and open it with Blender.

The file contains a fully rigged and skinned character. He’s a little goofy looking and rather dynamic. Let’s call him Hank.

Setting Up Your Workspace

When you first open the file with Hank, you are in the "1-Animation" screen. This is the default animation screen that comes with Blender, and as it works fairly well, you can use it with some minor adjustments.

The Ipo window on the right, which you’ve seen before in Chapter 3, won’t be needed right now. Replace it with an Action Editor window. An Ipo window can only show the keys for one object or bone at a time. When working with character animation, you need to see keys for many bones at once so you can easily adjust and align their timing.
Creating a Wave

RMB click on the armature to select it. The first thing you will notice is that you can’t see the armature when it is inside Hank’s mesh. How can you work with it if you can’t see it? One solution would be to just work in wireframe mode. That could be handicapping, though, as character animation relies on visual feedback from the character itself. The better the visualizations of a character’s poses are, the better the final animation will be.

With the armature selected, check out the Armature panel in the Edit buttons (F9). Enable the X-Ray option. Now, the armature is visible regardless of whether it is inside or outside the mesh.

To pose and animate an armature, you need to enter Pose Mode. This can be selected from the main modes pop-up menu on the 3D view header, or with Ctrl-Tab. As the frequently-used Edit mode is the Tab-key, this is a pretty easy one to remember.

When you enter pose mode, many of the bones of the armature turn gray, while some are yellow. The yellow bones have constraints on them, which you can learn about in Chapter 7.

Bone transformations work much like object transformations, using the same methods and hotkeys: manipulators, mouse gestures, and G/S/R. One difference you will notice is that sometimes asking for a translation (grab move, G-key), results in a bone rotating instead. Some bones, like the ones in the middle of Hank’s arms and spine, are parts of longer chains of bones. They are not free to translate in space. Instead of simply having translation controls for these bones do nothing, they trigger a rotation instead.

Before you begin animating, let’s make the job a little easier. In the Timeline window in the middle of the screen are the animation playback controls that were covered in Chapter 3. You’ll make use of another one of those controls now. Enable the button with the red dot, commonly seen in audio/video devices as the Record button.

You have just enabled automatic keyframing, meaning that any bone that moves or rotates will automatically have a keyframe set on the current frame. This will prevent the unfortunately common occurrence of setting a complex pose, then accidentally advancing the frame number and losing it.
How to Pose

For the wave, you want to raise the hand and arm into the air, tilting the hand outward a bit. By selecting each arm bone and applying rotations, you could achieve such a pose, but it would be difficult and rather unintuitive. If you like, try to use RMB selection and R-key rotation on the arm and hand bones to get something like this pose:

Very difficult, no? One thing you may have noticed when rotating the bones was that as soon as you moved them, they turned a bright blue. The blue color is an indication that a bone has at least one keyframe set. Because of the automatic keyframing you enabled a moment ago, each rotation resulted in a keyframe.

RMB select all of the now-blue bones and use Alt-R to clear any rotations you may have set.

Now RMB on the bone called "hand.l" to select it. It is the first bone of the left hand immediately following the two longer arm bones. On the Armature panel in the Edit buttons, enable the Auto IK option.

Use Grab mode (G-key) to move the hand. This time, translating the hand bone pulls the rest of the arm around with it, letting you create the pose much more intuitively. When you have the hand close to what looks like a decent pose for the beginning of a wave motion, disable Auto IK.

You need to adjust the hand a little so it tilts away from the body. Many Blender animators prefer to use the manipulators for bone rotations like this, as they give excellent visual feedback. If the Transformation Manipulator isn’t showing, enable it on the 3D header, or with Ctrl-Space. When the manipulator is visible, set it to Rotation mode.

With the manipulator set to the default Global mode on the header, it’s not very useful. Change it to Normal, which will cause the manipulator to align itself with the active bone. Now, LMB clicking on any of the manipulator’s orbits and moving the mouse will rotate the bone along that axis.

Although you don’t have to use the manipulators for bone rotations, they certainly can help. If you prefer, continue to use the R-key. When working with hotkeys and bones, you will probably find it most useful to rotate with the R-key R-key combination to enter trackball mode, or the R-key followed by XX, YY, or ZZ to enter local inverse and forward kinematics are covered in Chapter 7: Rigging and Skinning. If you’ve not worked through that chapter and do not plan to, it’s enough to say that IK, Inverse Kinematics, lets you pull an entire chain of bones by moving a target bone, instead of posing each bone individually.
Finishing the Wave

rotation mode with the Alternate Transformation Space set to “Normal” on the 3D header. However, the illustrations in the rest of this chapter will show the manipulator.

Work with the hand in Auto IK mode and by directly rotating it until it looks something like the earlier illustration.

Now, take a look at the Action Editor. It has several rows, or channels, and each has a name that corresponds to a bone in the active armature. When you select a bone in the 3D view, the corresponding channel in the Action Editor is selected. Just the same, RMB clicking a channel name in the Action Editor selects that bone in the 3D view. In the illustration, the lower_arm.l channel is selected.

Notice that some yellow diamonds have appeared within the Action Editor’s timeline. These represent the bones’ keyframes. They are aligned with the vertical green frame indicator that functions like the ones in the timeline and Ipo windows. Currently you’re on Frame 1, and the animation keys were all added there.

Finishing the Wave

Advance to Frame 5, either by LMB clicking in the Action Editor or Timeline, or by pressing the Right Arrow key four times.

The pose on Frame 5 was created by RMB selecting the middle arm bone and rotating it slightly on its X axis (the red manipulator orbit).

After rotating the bone, look at the Action Editor and note the following:

- Keys were automatically inserted for the arm bones that were moved. In this case, only the “lower_arm.l” bone was moved.
- No key was inserted for the other arm bones, since they weren't altered in this frame. It is true that they changed position, but they were not directly manipulated, and maintain their position and rotation relative to their parent bone further up the chain.
- The Frame 1 key for “lower_arm.l” is deselected (white) and the new Frame 5 key is now selected (yellow). The keys from Frame 1 for the other bones are still selected, as they didn’t receive new keys.

**Completing the Wave Action**

Advance to Frame 9.

You’ll perform the next bit of animation in the Action Editor itself. Like most Blender window types, the Action Editor uses a common set of selection, transformation, duplication and deletion tools.

- A-key to select/deselect all keys;
- RMB and Shift-RMB to build selections; and
- B-key followed by LMB drag to select an area.

You’re going to duplicate the keys from Frame 1 and move the duplicates to Frame 9, copying Frame 1’s pose to Frame 10. Doing so will cause the Wave action to start and end in the same position. To get a better view of what you’re doing, use the mouse’s scroll wheel and MMB-drag to zoom into and position the view of the keys.

Perform the following actions:
- Deselect all keyframes by pressing the A-key;
- Use the B-key to bounding-box select all the keys on Frame 1;
- Duplicate the selected keys with Shift-D; and
- This should seem familiar — the duplicate keys begin their life in Grab mode.

Move the duplicated keys to Frame 9. Don’t worry if you hit Frame 9 exactly or not.

With the new keys still selected, press Shift-S, which, as you may remember from Chapter 3, brings up the Snap menu. Choose “Current Frame” from the pop-up menu, and the keys are snapped to Frame 9. Of course, if you are in the habit of holding down the Ctrl-key while moving key markers, your keyframes will never fall between frames, and you won’t need to adjust them afterward like this.
Scrub back and forth between Frames 1 and 9 to make sure you are happy with your animation.

On the Action Editor’s header, change the name of this set of keyframes to something useful, like “Wave.”

You have now created your first character animation Action.

**Creating a Walkcycle**

A walkcycle is an Action that includes one full stride of a character walking, both with the left and right feet, that, when played over and over (cycled), gives the appearance that the character is walking.

Click the “X” next to the name of the Wave action on the header to unlink it. LMB click on the selector and choose “ADD NEW” to create a new, blank Action. If you wanted to work with the Wave action again, it could be accessed by selecting “Wave” from this same menu.

Set the frame counter to Frame 1.

In the 3D view, use the A-key to select all the bones in the armature, and use Alt-R and Alt-G to remove all rotations and translations, returning the armature to its rest position. Notice that even though the “Record” button is still pressed, keys were not added when you cleared rotation and location. Automatic key insertion does not recognize clearing location or rotation as actual movement.

**The Contact Pose**

In a walkcycle, the contact pose is the point when the leading foot just touches the ground in front of the character. It’s generally the first pose to animate in a walkcycle.

In the 3D view, switch to a side view (Numpad-3). Make sure Auto IK is off. The legs are set to already use IK, and Auto IK will cause them to malfunction.

Move the lowest bone in the spine, “spine1,” down a little along the Z axis so the legs bend a bit.

An armature can have many different Actions, but only one active Action, which is displayed in the Action Editor. This active Action is the one that will receive any new keys you insert, and whose keys you can directly edit.
RMB select “leg.l.” You may have to MMB rotate the view to see and select it accurately, and return to side view after making the selection. Notice that you are not selecting one of the actual leg bones, but the bone that extends below the left foot. This bone is the IK target of the left leg. Move this bone back and up to match the illustration.

With “leg.l” still selected, rotate it counterclockwise so the toe of the foot passes through the “floor.” In this case, it might be simpler to use the R-key, as the rotation you want corresponds exactly to the side view.

RMB select the toe bone, called “toe.l,” and rotate it clockwise so that it appears to be bent up to meet the floor. Be aware that you may have to alternate between tweaking the foot and toe positions to get this just right.

RMB select the controller bone for the right leg: “leg.r.” Move it forward and rotate it clockwise to try to match the illustration.

The goal here is to get the heel of the mesh’s right foot to appear to contact the ground at the same level as the toe of the left foot. The right foot should not be so far out in front of the body that the knee becomes completely straight which can cause the foot bones to detach from the leg.

Now pose the arms (illustrations on the next page). Once again, an easy way to do this is to turn on Auto IK and drag the hands into position, bringing the arms along for the ride. Afterward, disable Auto IK and finely adjust the rotations of the arm and hand bones with the rotation manipulator set to Normal mode.

You may have to rotate or change the view several times to get the arms to go where you want them to. Keep in mind that when people walk, the legs and arms have opposing motion: right leg forward means right arm back. It might take a while to get things just right, but be patient — learning to create poses does not happen in an instant. If you can do so without feeling silly, you should try walking naturally around your work area, observing how your arms and hands swing and twist to give yourself a reference. Of course, if you have a video camera available and record reference motion to play back frame by frame, it can be an even bigger help.
Here are front, side, and top views of the posed arms:

Making a good walkcycle requires more than just arm and foot positioning.

Spend some time on the spine. When walking, the leading hand twists the top of the spine toward it, and the trailing arm twists the lower spine toward it. Be aware, though, that by rotating the lower spine bones, the upper spine, arms, neck, and head will rotate as well. You may have to compensate by rotating the upper spine bones back the opposite way. Rotations of the spine during a walkcycle should be subtle: they add hints to the overall motion. Over-rotating will produce animation that looks wild and unnatural.

Flipping the Pose

Select all bones in the armature with the A-key. Click the “Copy Pose” button at the bottom of the 3D Window. This copies any existing keys (location, rotation, and scale) from all selected bones.

You may have to use MMB Drag to pan the header of the 3D Window back and forth in order to find these buttons.
Press the Up Arrow key to advance ten frames to Frame 11. Click the “Paste Flipped Pose” button.

This pastes the pose as a mirror-image of the pose you just copied.

Advance to Frame 21, and this time press the normal Paste Pose button (not Paste Flipped Pose). The effect is similar to when you used Shift-D to duplicate and reposition the keys in the Action Editor in the “Wave” action.

In the Action Editor or the Timeline, scrub the frame slider back and forth between Frames 1 and 21 to see the character walk. From the first contact pose to the opposite contact pose and back to the first again takes 21 frames in this case. Everything else in the walkcycle is just filling in keys between those frames.

Recoil Pose

You made a rough cut of a walkcycle with just one pose and a couple of mouse clicks! When played back as animation, though, it’s clearly inadequate. You can improve the walk by adding poses in between the ones you just created.

The recoil pose is where the front foot takes the weight of the body. The forward foot flattens, and the body sinks down a little. Here’s how to create it:

Move to Frame 2. RMB select the right leg controller “leg.r.” Clear its rotation with Alt-R. You need to set a key with this bone un-rotated, but if you recall from before, clearing a rotation does not set a new key. To do that, press the I-key and select “LocRot” from the keying menu that pops up.

Grab the bone at the base of the spine (“spine1”) and move it down a little. A new way to approach a small move like this is to press the G-key, and then press the Down Arrow key a couple of times. Pressing Enter confirms the move.

Now, select only the “spine1” and “leg.r” bones. This may be easier to do by RMB and Shift-RMB clicking on their names in the Action Editor. Use the Copy Pose button, and paste the flipped pose 10 frames later into Frame 12.

This is one reason to strictly follow the naming instructions when creating a rig in Chapter 7. Because they end in “.l” and “.r,” Blender recognizes that the bones are on the opposite side of the body and does the calculations to automatically insert the mirrored pose for you.
The Passing Pose

Move to Frame 6 and adjust “leg.r” (the leg controller) and “toe.r” so that the foot is flat on the ground. If you are having trouble telling exactly where the ground should be, move to Frame 1 and LMB in the 3D view to set the 3D cursor at the very bottom of the forward foot. When you move back to Frame 6, use the 3D cursor as a guide. Adjusting the right foot and toe rotations will most likely only involve clearing the rotation with Alt-R, then setting a rotation key by pressing the I-key and choosing “LocRot.”

Move “spine1” up until the right leg is mostly straight.

Select only the bones moved in this step and copy and paste the flipped pose 10 frames later into Frame 16.

The High Point Pose

Move to Frame 8. Move “spine1” upward a little, so that the right leg straightens completely and just begins to stretch. Fix that stretch by moving “leg.r” until it re-attaches to the bottom of the leg bones. Rotate “leg.r” so the toe goes through the floor just a bit. Rotate “toe.r” so it is level with the floor again.

Grab the controller for the left leg, “leg.l,” and move it forward and up.

After selecting the four bones that were changed in this step —  leg.r, toe.r, leg.l and spine1 — copy the pose and paste the flipped version on Frame 18.

Looping the Animation for Playback

In the Timeline window, set the Start: frame to 1 and the End: frame to 21. This will restrict animation playback to only that frame range, letting you watch your work in a continuous loop. In the 3D view, press Alt-A to start the playback. Pressing the Esc-key stops playback.

When looping an animation like this, there will be a slight “catch” as the frame counter returns to 1. This is normal behavior. Don’t be fooled into thinking that there is a glitch in your animation. In the next section on the NLA Editor, you will be able to get a better view of how well your Action cycles.
On the Action Editor’s header, name this new Action something useful like “Walkcycle.” Also, in the timeline view or the Scene buttons, change the overall animation end frame value back to something higher, like 250, so we no longer loop from frame 1 to 21 on playback.

**The NLA Editor**

You now have two separate actions: “Wave” and “Walkcycle”. There’s another window in Blender called the NLA Editor (Non-Linear Animation) where you can combine the two actions.

Change the Action Editor window to an NLA Editor. This will be your primary workspace.

You can also change the far left window to a new Action Editor, just to make some things clearer in a bit.

This is the NLA Editor. It may not look like much, but it’s quite powerful. Right now there are only two rows: “hankbones” and “Walkcycle.” “hankbones” refers to the armature you have been working with, although any objects that have Actions associated with them would show up here as well. “Walkcycle” refers to the current action selected in the Action Editor. Just like the Action Editor, the diamonds in the right-hand portion of the window indicate the location of keyframes along the timeline. As opposed to the Action Editor, though, keyframe markers in the NLA view are present if a key exists at that frame for any channel of the associated Action.

If you still have an Action Editor window available, use the header Action block selector to switch to the “Wave” Action. If you can’t see the Action block selector on the header, you may have to MMB drag on the header to show it. In the NLA Editor, note how the row under Armature has changed, with three diamonds representing the three keyed frames from the “Wave” action.

### Converting an Action into an NLA Strip

Switch back to the “Walkcycle” action by selecting it in the Action Editor. With the mouse over the NLA Editor, press the C-key to convert this Action into an NLA strip.
By converting the Walkcycle Action into an NLA strip, you’ve created a “window” to the walkcycle, letting you see the Action through the NLA Editor. This “window,” the Action strip, can be manipulated without affecting the original action. Strips like this can be scaled and moved along the timeline, blended, stacked and set to repeat. You can create any number of strips that show you the same Action. A single strip represents the entire “Walkcycle” Action. You haven’t changed the original “Walkcycle” Action in any way, and none of the transformations you can do in the NLA Editor will affect the original.

With the mouse over the 3D view, press Alt-A to play the current animation. Note that the character takes two steps and then stops at frame 21, because, well… that’s the end of the walkcycle. This is about to change.

In the NLA Editor, press the N-key. As in other window types, the N-key brings up a transform properties panel. In the NLA Editor, this panel is used to work with strip settings.

Change the value of the Repeat control to 2. Notice that the NLA strip representing the walkcycle now has a faint line down its center, dividing it into two sections. Play the animation again with Alt-A in the 3D view. The character takes four steps now, because the walkcycle action plays twice. However, the walkcycle is almost too fast to see, because those four steps are being taken in only 21 frames. Twice the number of steps in the same amount of frames equals twice the speed.

In the properties panel, change the Strip End control to 41. Play the animation again. There are still four steps, but they are now taking place over the course of forty-two frames, which gives a better result.

By changing the values in the strip properties panel, you can adjust the speed and number of steps of a walkcycle. Strips can also be scaled directly within the NLA Editor with the S-key.

Set the walkcycle to a Repeat value of 5. Make sure the frame counter is on Frame 1. Now, instead of typing numbers into the panel controls, press the S-key to begin scaling.

With the mouse over the 3D view, press Alt-A to play the current animation. Note that the character takes two steps and then stops at frame 21, because, well… that’s the end of the walkcycle. This is about to change.

In the NLA Editor, press the N-key. As in other window types, the N-key brings up a transform properties panel. In the NLA Editor, this panel is used to work with strip settings.

Change the value of the Repeat control to 2. Notice that the NLA strip representing the walkcycle now has a faint line down its center, dividing it into two sections. Play the animation again with Alt-A in the 3D view. The character takes four steps now, because the walkcycle action plays twice. However, the walkcycle is almost too fast to see, because those four steps are being taken in only 21 frames. Twice the number of steps in the same amount of frames equals twice the speed.

In the properties panel, change the Strip End control to 41. Play the animation again. There are still four steps, but they are now taking place over the course of forty-two frames, which gives a better result.

By changing the values in the strip properties panel, you can adjust the speed and number of steps of a walkcycle. Strips can also be scaled directly within the NLA Editor with the S-key.

Set the walkcycle to a Repeat value of 5. Make sure the frame counter is on Frame 1. Now, instead of typing numbers into the panel controls, press the S-key to begin scaling.
Scale the strip until its end point is near Frame 120. Now you have 120 frames of walking!

**Mixing Actions in the NLA Editor**

With the mouse over the main workspace of the NLA Editor, press Shift-A. Shift-A brings up a selector of all the actions that are available to add to the selected object. Choose “Wave.” Adding an Action Strip can also be done through the Strip menu on the NLA Editor’s header.

When it is added to the NLA like this, the new “Wave” strip is automatically selected, and its properties are shown in the Transform Properties panel. Using the panel, change the “Wave” Action’s Repeat value to 4. Change the “Strip End” value on the panel to 50.

Press the G-key and slide the strip along the timeline until its beginning (its left most edge) is around Frame 23. This is one of the reasons that the NLA Editor is so powerful: once actions are defined in the Action Editor, you can add, scale, move and even duplicate them along the timeline as a single entity.

Press Alt-A in the 3D Window to view the animation. Not bad, eh? The wave and the walkcycle happen simultaneously. Use the LMB to scrub the timeline over the end of the Wave strip, though. When the strip ends and the hand comes back down, it’s a pretty abrupt motion.

With the Wave action still selected, change the “Blendin:” and “Blendout:” values on the panel to 7. The NLA strip reflects this change by putting “blending” ramps at the beginning and end of the strip. Now, LMB over the beginning and ending of the wave strip again. This time, the animation blends much more smoothly. Press Alt-A in the 3D window to see it play in time.

**Changing the Stacking Order of NLA Strips**

The stacking order of the strips in the NLA Editor is significant. In the example, the top-most strip is the “Walkcycle” action, and under it is the “Wave” action. Strips on the bottom override strips above them.

In other words, the “Walkcycle” action has keys for all the arm bones. The “Wave” action also has keys for the left arm bones. As the “Wave” strip is below the “Walkcycle” strip, it overrides any conflicting keys.

To change the stacking order of a selected strip, press Ctrl-PgUp and Ctrl-PgDn. Try this: RMB select the “Wave” strip and move it up one row with Ctrl-PgUp.
Play the Animation

The wave no longer happens. This is because the “Walkcycle” keys for the left arm bones override the “Wave” keys for the same bones.

Change the stacking order of the “Wave” strip so it is under the “Walkcycle” strip again (select “Wave” and use Ctrl-PgDn) and everything is back in working order.

You may be wondering why the walkcycle has Hank essentially treading water. This is the traditional method of producing walking animation. A walk is keyframed “in place;” like you have just done, then matched with a whole-body forward motion later. While you can still use this technique in Blender, there is a better way. Before you finish the tutorial, we’ll show you how to do it.

Offset Bone

Toggle the NLA Strip/Shark Attack icon so that the NLA is disabled, and the Action in the Action Editor will be used. In your Action Editor window, make sure that “Walkcycle” is selected. The last bit of setup is to LMB click in the upper channel that reads “Walkcycle” in the NLA. This tells Blender to use the timing of the original Action, as opposed to the timing dictated by the strip’s length and repeat settings.

Set the frame counter to Frame 1, and make sure that the Record button is selected in the Timeline window so that any transforms you make are automatically keyed. With that done, you’re ready to revisit the Walkcycle Action.

RMB select the bone sticking out of Hank’s back called “master.” This bone can be used to move the entire armature at once, which is exactly what you’re going to do. This bone will make Hank move forward during his walkcycle, and then provide the NLA with a reference when putting together repetitions of the Walkcycle Action.

In the 3D view, go into a side view, make sure that Hank’s armature is in Pose Mode, and RMB select the “master” bone. Press the I-key to insert a keyframe, and choose “Loc” from the menu that pops up.

LMB click in the 3D view to place the 3D cursor at the base of the heel of the forward foot. The 3D cursor will be your reference point. Use the Left Arrow key to advance one frame. See how the foot moves to the right of the cursor a bit? With the master bone selected, press the G-key to Grab and move it. Moving the
master bone moves the entire character, and your goal is to get the base of the heel back into the same relationship it had with the 3D cursor on the previous frame.

Advance forward one frame at a time, watching as the forward foot moves to the back. Stop advancing frames as soon as the heel comes away from the floor. At this point, you’ve gone one frame too far. Use the Left Arrow key to go back one frame, the last frame on which the foot is completely flat on the floor. Using the Grab tool again, move the master bone so that the heel of that same foot moves forward until it is once again on the 3D cursor.

At this point, Hank’s weight will shift to the toes of this foot. So, LMB click to reposition the 3D cursor at the place where his toes meet the ground. As it is the point of Hank’s body that bears his weight against the ground, the toe is the new reference point.

Advance to Frame 11, which is where you have the other foot finally meeting the ground. Move Hank forward using the master bone until the toe of the rear foot hits the center of the 3D cursor.

You can scrub through the first half of the Walkcycle action to see Hank move forward. When you’re done with that, return to Frame 11.

The procedure for the second half of the walkcycle is exactly the same as the first:

- Set the 3D cursor to the location of the heel of the forward, weight-bearing, foot.
- Advance one frame, and adjust the master bone location so the heel stays in place with the 3D cursor.
- Advance to the frame just before that heel leaves the ground, and adjust the master bone and armature location again.
- Change the 3D cursor location to the toe of the weight-bearing foot.
- Advance to the last frame of the Action, Frame 21, and move Hank forward one last time so the trailing toe matches the 3D cursor’s location.

When you play the Action back now, Hank should walk forward for an entire stride, and his feet should stay planted on the ground reasonably well as he moves.
This is not the ideal way to use the Offset Bone feature. If you were starting a walkcycle from scratch, knowing you wanted to use the Offset Bone, you would keyframe your character to move forward from the very beginning, with a “master” bone that did not control the feet. This would allow them to be truly anchored in their location when touching the ground.

If Hank is moving forward for you reasonably well, then it’s time to return to the NLA Editor. Change Hank’s NLA setting back to using NLA strips with the toggle icon. Make sure that the Walkcycle strip is selected, and that it still has a repeat value set (it was 5.0 in the previous example). In the “OffsBone” control immediately below the Repeat value, enter “master” — the name of the master bone you were just keyframing.

Now, if everything happened correctly, playing the animation in the 3D view should show Hank walking forward continuously! You can change how far he goes by adjusting the Repeat value.

If you like, you can add the Wave again as an NLA strip.

Hi Hank!