CHAPTER 12

Decisions, Decisions

This chapter covers the final block in the Common Palette. When you are finished with this chapter, you will have the ability to create some great little programs for your robots and give them the ability to move, talk, listen, bump, stop, wait, and a lot more.

Let's give your robots one more talent—the ability to make choices and select from multiple possible actions.

Left or Right? Door 1 or Door 2?

Let me give SPOT some pseudo-code for his next task:

> **Me:** SPOT, I want you to move forward three rotations and stop. If your Light sensor detects a light level over 30, turn left. Otherwise, turn right.

At this point, you already know how to program SPOT with a `MOVE` block that moves him forward three rotations. But how do you take the light value from the Light sensor and use it to help SPOT make a decision about turning left or right?

The answer is easy; you'll use the `SWITCH` block shown in Figure 12-1.

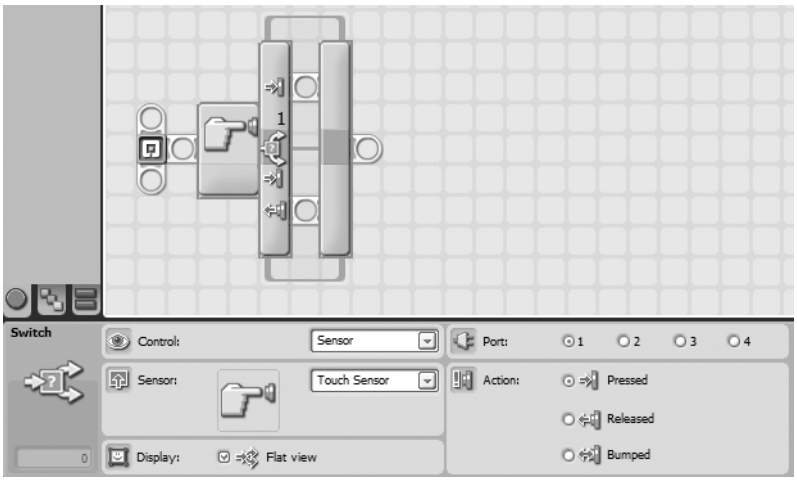

Figure 12-1. *The SWITCH block lets your bots make choices.*

The SWITCH block uses an input value to determine a path to take. This value can be a number, a bit of text, or a Logic value (True or False). And you're not just limited to two paths. You could configure a SWITCH block to handle the following pseudo-code:

Me: SPOT, pick a random number from 1 to 5.

Me: If the number is 1, turn left.

Me: If the number is 2, turn right.

Me: If the number is 3, spin 180 degrees.

Me: If the number is 4, spin 360 degrees.

Me: If the number is 5, keep moving forward.

In this example, I have SPOT pick a random number. This is done using the VARIABLE block that you'll learn about later in Chapter 18. But for now, let's just assume for the moment that SPOT can pick his own numbers. Now, since there are five potential numbers (1, 2, 3, 4, and 5), there are five potential actions that can be taken. Throughout the remainder of this chapter, I'm also going to use the term *path* instead of *action*, because the SWITCH block will allow your bots to choose from different paths available to them.

Depending on the path a robot selects, different actions will occur. One path can have your robot moving forward, checking its Ultrasonic sensor for an object in front. Selecting a different path might send the same robot in the reverse direction, waiting for its Touch sensor to be pressed and counting the number of rotations the motors spin. That's the great thing about the SWITCH block. Each potential path choice can have unique programming blocks that give your robots even more power. (And you can add another SWITCH block to a path, creating another set of paths for your robot to choose from!)

Now, before I show you how the SWITCH block works, I need to mention one special item in the SWITCH block's configuration panel. Take a look at Figure 12-2.

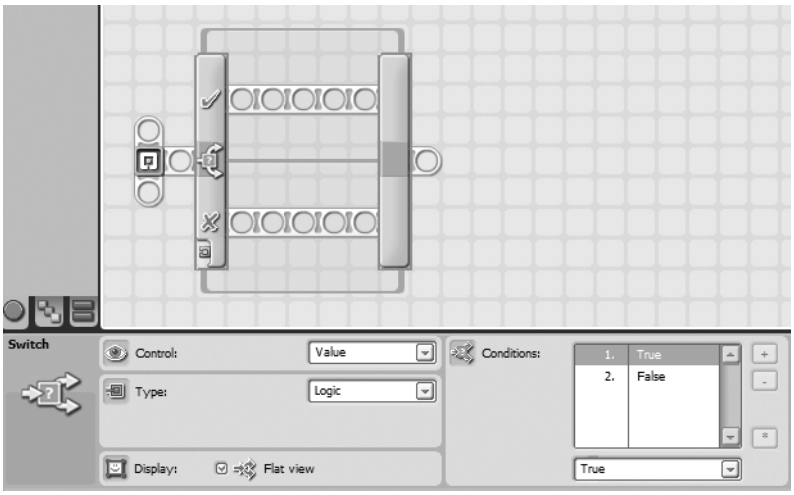

Figure 12-2. *Flat view for the SWITCH block*

When using the SWITCH block, you need to choose between Flat view and Tabbed view. When using Flat view, you need to leave the Flat view box checked (as shown in Figure 12-2). Flat view does have a limitation that you need to be aware of, however. When using it, you can program *only* two paths. Notice in Figure 12-2 that there are two paths: one labeled with a checkmark and the other with an "X". The checkmark path is also called the *default* path (more on this in a little bit).

Now, in Figure 12-3 I've unchecked the Flat view box, and you can now see that the SWITCH block has tabs along the top edge.

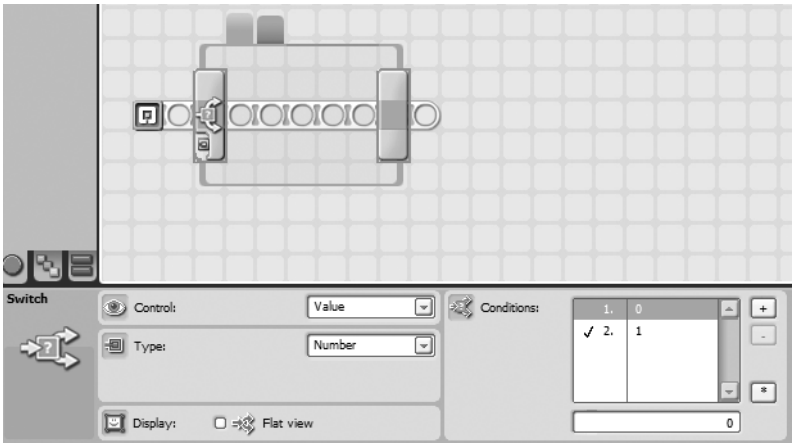

Figure 12-3. *Tabbed view for the SWITCH block*

With the Tabbed view, you must click a tab to see the programming blocks (if any) that have been placed inside it. This is a small price to pay for the ability to specify more than two options, however. Remember the earlier pseudo-code where SPOT picked a random number between 1 and 5? Figure 12-4 shows a SWITCH block with five tabs; each tab will now correspond to one of the potential actions I asked SPOT to perform.

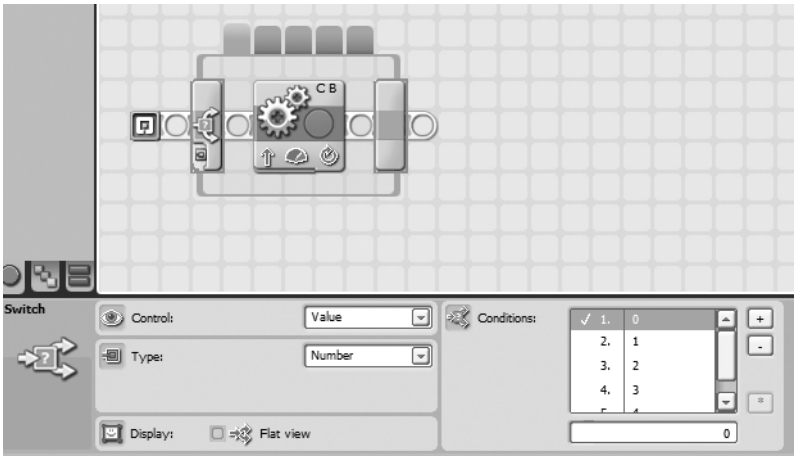

Figure 12-4. *This SWITCH block has five tabs for five different paths.*

Okay, now it's time to show you how to configure the `SWITCH` block, so you can use it to give your robot choices.

In Figure 12-5, I've placed a single `SWITCH` block that is using the Flat view. This means I only have two possible paths for my robot to take. The first path (with the checkmark) is on top and the second path (with the "X") is on the bottom.

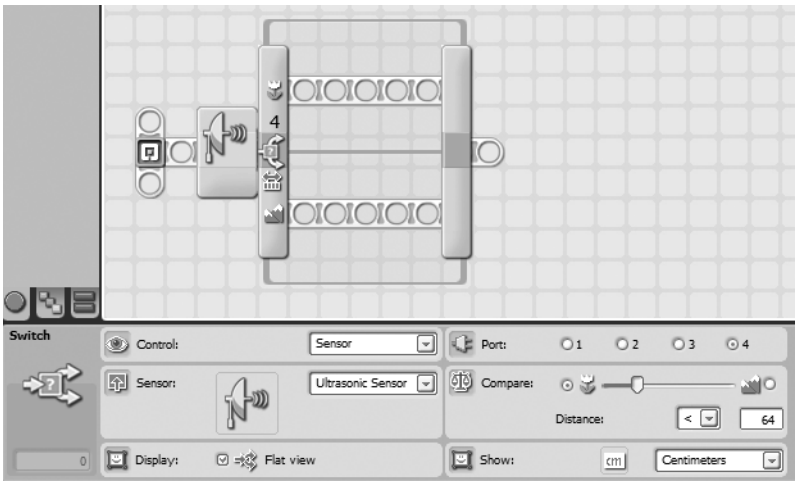

Figure 12-5. *This SWITCH block has two possible paths.*

This example also shows you the power of the `SWITCH` block. In the Control section, you have a pull-down menu that offers two options: Sensor and Value. Choosing the Sensor option will allow you to configure the `SWITCH` block to determine the correct path for your robot using the sensor and its trigger, which you select.

In this example, I've selected the Ultrasonic sensor. I've configured the Ultrasonic sensor to detect when an object or obstacle is detected less than 25 centimeters in front of the robot.

If this condition is met (True), the SWITCH block will execute any blocks found in the True path (the upper beam, with the checkmark). If the condition is not met (False), the SWITCH block executes any blocks found in the False path (the lower beam, with the "X" mark).

For the moment, let's assume that SPOT has his Ultrasonic sensor and Sound sensor mounted. I'm going to give SPOT the following pseudo-code:

> **Me**: SPOT, when your Ultrasonic sensor detects an object less than 25 centimeters in front, turn left if your Sound sensor detects a sound level greater than 20.

I've already shown you how to configure the first SWITCH block to use the Ultrasonic sensor. So let's assume that SPOT's Ultrasonic sensor detects an object less than 25 centimeters in front of him. This means that any blocks on the upper beam (True path) will be executed. From the pseudo-code, we know that if the first condition is met, we want SPOT to turn left *only* if his Sound sensor detects a sound greater than 20. How will we do this? Simple—we'll use another SWITCH block!

First, I drop another SWITCH block on the top beam and configure it as shown in Figure 12-6.

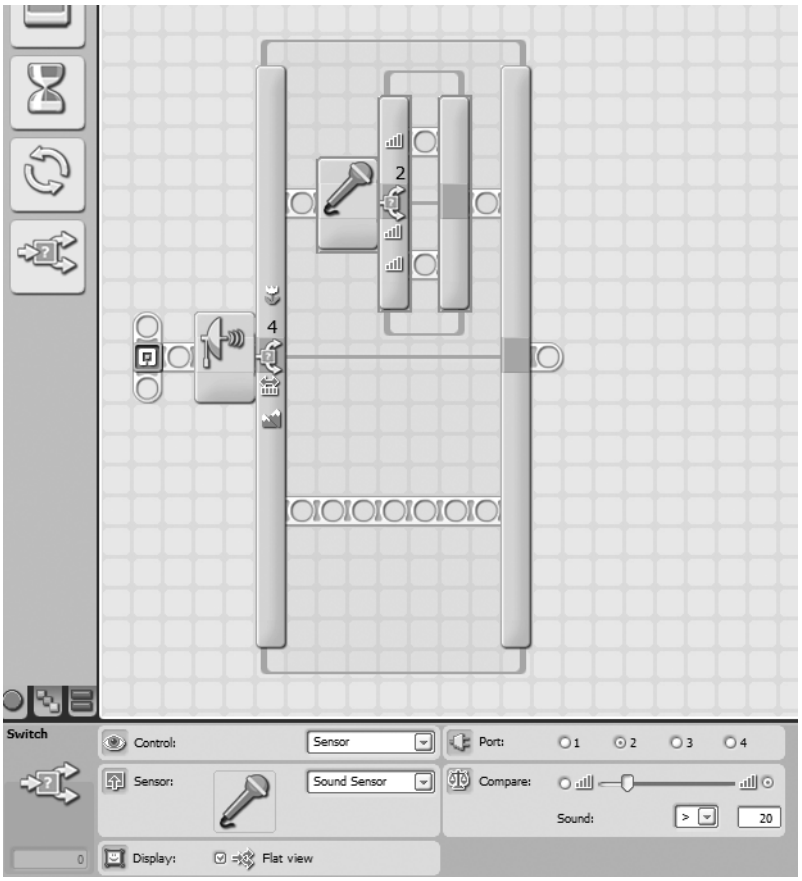

Figure 12-6. *This second SWITCH uses the Sound sensor.*

I configure the second SWITCH block to use the Sound sensor and to detect a sound greater than 20. If this happens, the True path (upper beam) in the second SWITCH block will execute any blocks found inside it. And that is where we'll place the MOVE block that allows SPOT to turn left (see Figure 12-7).

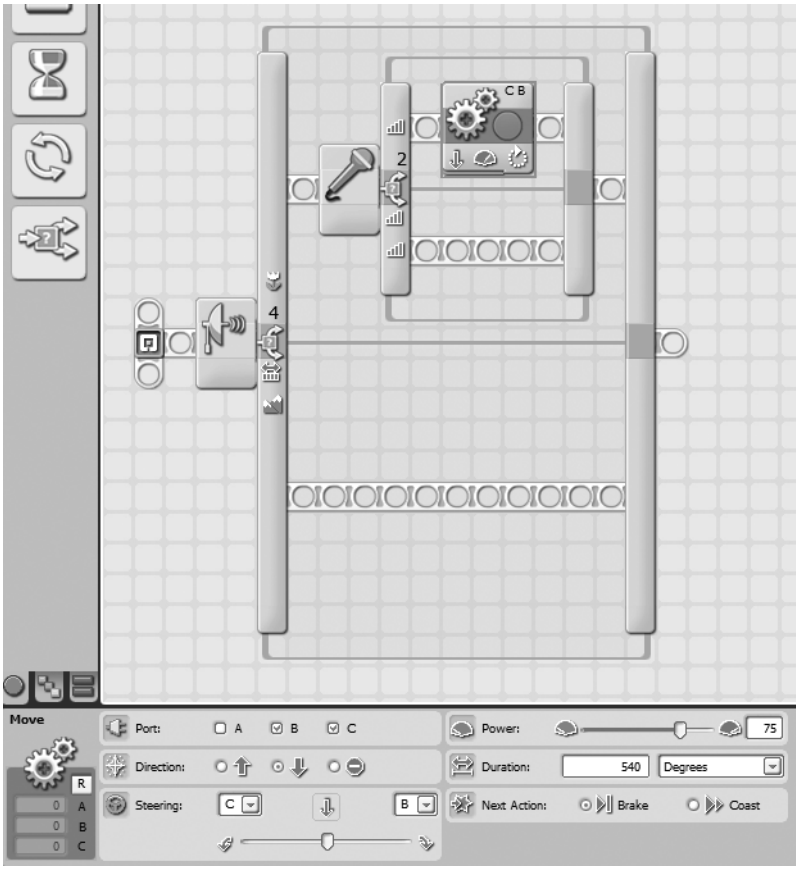

Figure 12-7. *SPOT will turn right if the first and second SWITCH block conditions are met.*

This is an example of embedded SWITCH blocks. You could keep going and place more SWITCH blocks inside other SWITCH blocks. This will give your robots some excellent decision-making control!

But what if you need to program your robot to test a greater number of conditions? Not all situations will have conditions that only have two options, right? So let's take another example for SPOT; look at Figure 12-8.

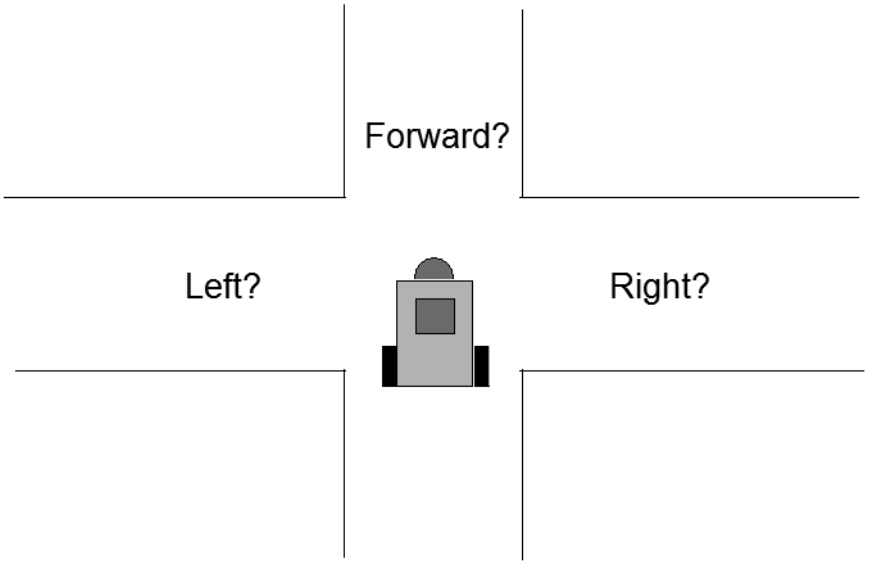

Figure 12-8. *SPOT has some decisions to make.*

Here's the pseudo-code:

Me: SPOT, when you come to the hallway intersection, pick a number from 1 to 3. If the number is 1, turn left. If the number is 2, turn right. And if the number is 3, move forward.

To do this bit of programming, recall that I told you you'll have to turn off the Flat view for a `SWITCH` block to use more than two conditions. That's the first requirement. The second requirement for configuring a `SWITCH` block for more than two paths is that the `SWITCH` block must be configured to use the Value option. This is found in the drop-down menu in the Control section and is shown in Figure 12-9.

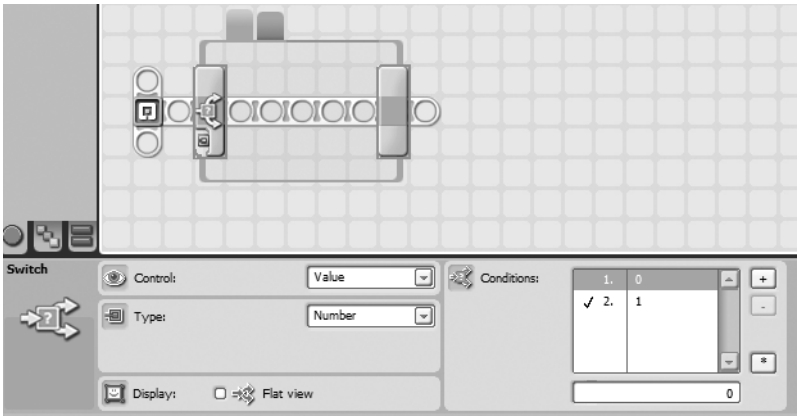

Figure 12-9. *Start with a SWITCH block with Flat view turned off and using a Value.*

The SWITCH block now has a small input data plug that will be used. This data plug can accept a Number data type, a Text data type, or a Logic data type, and you select the option from the drop-down menu in the Type section.

Using our pseudo-code, we can see that there are three possible conditions:

- Turn left if the number SPOT picks is 1.
- Turn right if the number SPOT picks is 2.
- If the number is 3, move forward.

For this example, I'm going to use the RANDOM block that is discussed in Chapter 14. For now, don't worry about how it works—just drag and drop a RANDOM block from the Complete Palette onto the beam, and place it in front of the SWITCH block as shown in Figure 12-10. (If you really want to know how the RANDOM block works, just jump ahead and read Chapter 14; I'll wait for you right here.)

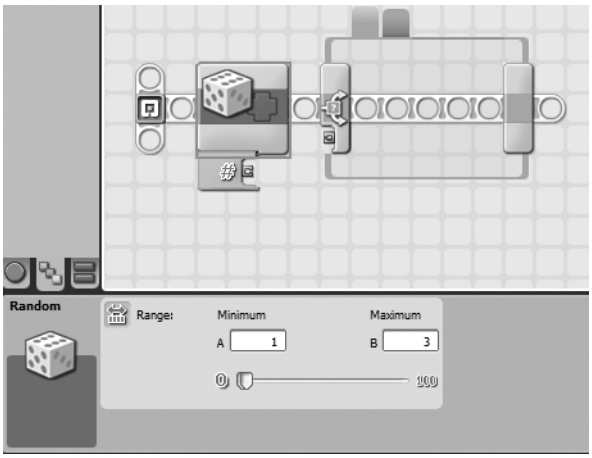

Figure 12-10. *A RANDOM block will generate a value of 1, 2, or 3.*

For the configuration panel of the RANDOM block, simply enter a value of 1 in the Minimum text field and a value of 3 in the Maximum text field, as shown in Figure 12-10.

Drag a data wire from the RANDOM block to the SWITCH block (see Figure 12-11).

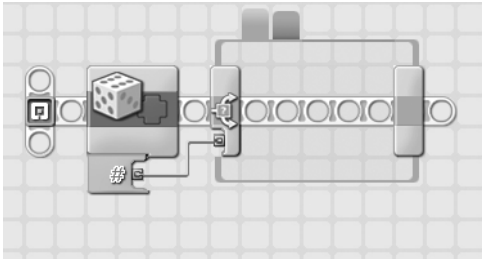

Figure 12-11. *Connect the RANDOM block to the SWITCH block with a data wire.*

Next, we need to configure the SWITCH block to accept more than two conditions. To do this, click the SWITCH block again (if it isn't already selected), and look at the configuration panel. On the left side of the configuration panel, you'll see the Conditions section (shown in Figure 12-12).

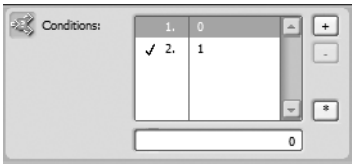

Figure 12-12. *The Conditions section of the SWITCH block*

Let me explain what you are looking at. This is a list consisting of path numbers. Each path number has a default value that the SWITCH block will check. In this example, you see the following:

1. 0
2. 1

The 1 and 2 on the left are the possible conditions, and these match the number of tabs you see along the top edge of the SWITCH block. The 0 and 1 values in the right column are the default Number values that the SWITCH block will use to pick a path. So, for example, if the Number value coming into the SWITCH block's data plug is 0, then the first condition will be selected, and any blocks found under the first tab will be executed.

Here's one more example to make sure you understand this concept. In Figure 12-13, I've got three tabs for the SWITCH block.

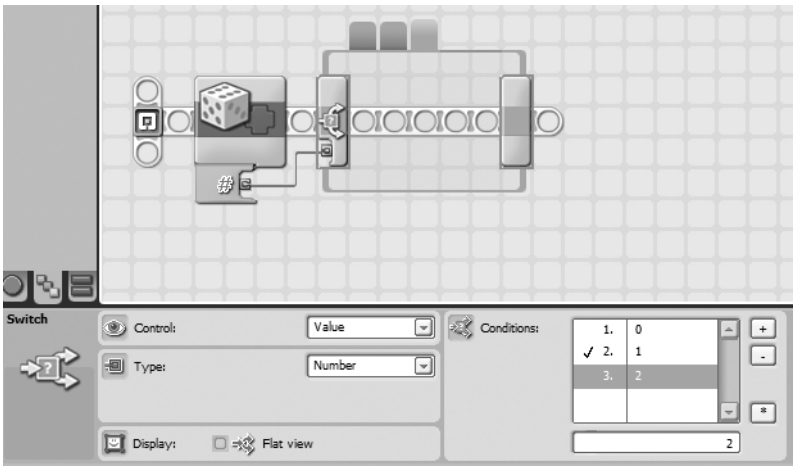

Figure 12-13. *Three possible paths for the SWITCH block*

Notice that there is now a new row in the Condition section:

1. 0
2. 1
3. 2

There are now three tabs on the SWITCH block. Tab 3 will have its blocks executed if the SWITCH block detects a Number value of 2.

Here we have a small problem: SPOT will pick only 1, 2, or 3. But the only options shown are 0, 1, and 2. How can we change this?

Easy—if you click one of the conditions, you can change its value in the text box just below the Condition section. First, click the condition whose value you want to change (see Figure 12-14).

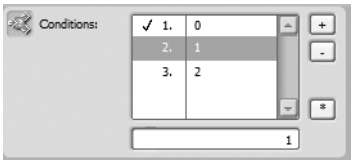

Figure 12-14. *Click a Condition to change its value.*

Next, you must click on the small asterisk (*) button shown in Figure 12-15; this will place a checkmark next to the value of the condition that you want to change (see Figure 12-16).

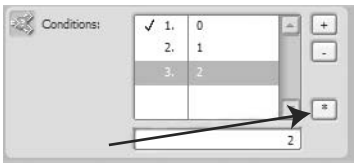

Figure 12-15. *Click the small asterisk (*) button to select a condition to change.*

Next, change the Number value in the text box just below the Condition section (see Figure 12-16).

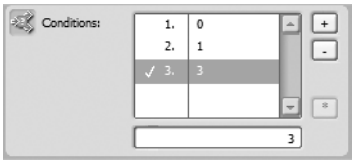

Figure 12-16. *Change the Number value.*

In this example, I've changed the Number value for the third condition from 2 to 3. I'll perform these same actions for conditions 1 and 2, changing their values, respectively, to 1 and 2. This is shown in Figure 12-17.

■ **Caution** Please be aware that you *cannot* have two or more conditions with the same Number value. This is because the SWITCH block would get confused and not know which path to take. Make sense? If condition 1 has a Number value of 1 and condition 2 has a Number value of 1, which path would the SWITCH block take if the RANDOM block sends a value of 1? Fortunately, the SWITCH block is smart and will not allow you to make this mistake.

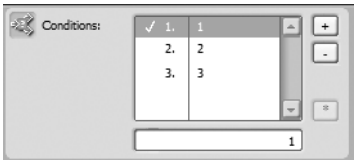

Figure 12-17. *Three possible paths are now available to SPOT.*

And now I can drop in a collection of MOVE blocks that will allow SPOT to turn left, turn right, or move forward.

If SPOT picks 1, then he turns left. So, I'll click the first tab (see Figure 12-18) and drop in a MOVE block that will allow SPOT to turn left.

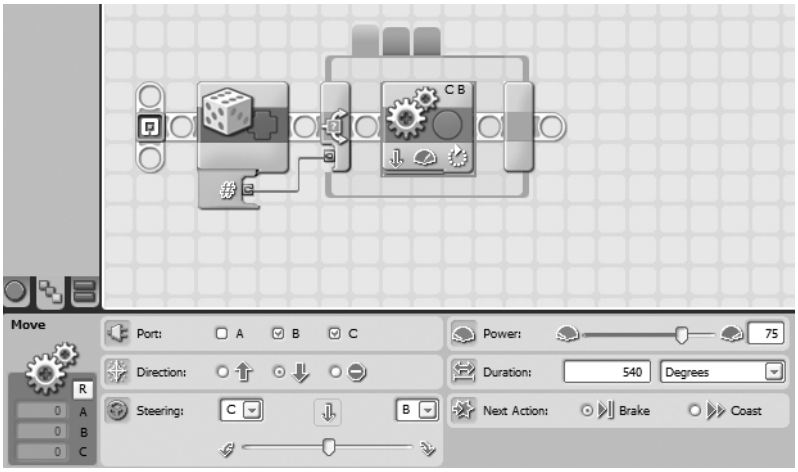

Figure 12-18. *The first tab has a MOVE block that executes if SPOT picks 1.*

I'll do the same thing for the second and third tabs. When the RANDOM block generates a number from 1 to 3, that number is passed to the SWITCH block. The SWITCH block takes this Number value and compares it to the values in its Condition section. If the RANDOM block sends

a 3, the SWITCH blocks takes that value, notices that it equals the 3 in the third condition, and then executes any blocks found on the third tab. Easy!

One final warning, though—what happens if the RANDOM block goes crazy and sends a value of 4 to the SWITCH block? There is no condition that has a matching value of 4, so what will happen? Well, remember when you selected a condition and clicked the * button and a checkmark appeared next to the condition? That checkmark also specifies the *default condition*. Look back at Figure 12-16. The checkmark is next to the third tab. This means that if a value of 4 is provided by the RANDOM block and 4 isn't a possible value in the Condition section, the default path will be chosen. So the blocks in the third tab will execute for a value of 4, 5, 100, or any other number except for 1 and 2.

The SWITCH block is a very useful block for giving your robots some powerful decision-making skills. The SWITCH block can use Logic values (True or False), Numbers, Text, and sensors to determine which paths are chosen by your robot to execute. Keep this in mind when you need to give your robots the ability to make different decisions based on different types of input.

As your programming skills progress, you'll find the SWITCH block one of your favorite tools to use.

Next, we'll be looking into some specialty blocks.