CHAPTER 6

■ ■ ■

StringBot: Planning

What's needed to solve this latest challenge is a bot that can move along a string, carry a small object, drop that object at a specific location, and return for another object to do it all over again. In this chapter, we're going to figure out how to make that happen.

Design and Planning

I'm sure I didn't fool you by reversing the words in the section title. Since you are no longer scared by the phrase "planning and design," I'm just a little worried you might become a little bored with it and try to skip it. So I'm going to shake it up a bit in this chapter . . . challenge you a little bit to come up with something truly unique. The jar isn't going to magically fill with pebbles, so you've got some work to do.

The StringBot

In Chapter 2, I gave you an advanced look at the ExploroBot before we began working on the Design Journal page. While I've got my version of the StringBot ready to go, try to keep from skipping ahead to Chapter 7 to look at it. I want you to start seeing your own StringBot in your head without my design influencing you.

We're going to use the Design Journal page again in this chapter. If you followed along in Chapter 2 carefully, this chapter will follow the same steps. Get a blank Design Journal page and a pen and let's start designing.

■ **Note** There are four blank Design Journal pages left in the back of this book (if you used one for Chapter 2). If you need more pages, feel free to make photocopies of the Design Journal page or visit the Apress web site to download the page in PDF format.

At the top of the Design Journal page, in the Robot Name box, go ahead and write **StringBot**. Feel free to create your own name for the bot; names such as TwineBot, MonkeyBot, or Gort are available. (You get Big Bonus Points if you recognize the "Gort" reference.) Once you've selected your bot's name, it's time to work on the Robot Description.

The Robot Description

Remember, the Robot Description doesn't need to be pages and pages of written text. Your goal is to keep it simple and uncluttered. As I said in Chapter 2, this isn't where you describe the bot as "lightweight, a mixture of parts, some motors, and maybe a sensor or two." This section is where you try to accurately describe the overall process the robot will follow.

Ask yourself the question, "What is this robot supposed to do?" and start writing inside the Robot Description box. Write "visually." What do I mean by this? Picture a box (a shoebox, for example) in your hands—the box doesn't have any sensors or motors or any other items yet, it's just a box. Imagine this box doing what you believe needs to be done to solve this challenge (refer to Chapter 5 if you need a reminder). Now compare what you wrote down to my Robot Description in Figure 6-1.

Figure 6-1. *The StringBot's Robot Description should be short and simple*

Did you cover the major requirements? Again, don't worry if your Robot Description doesn't match mine exactly. Although it wasn't mentioned in the challenge (Chapter 5), I had a thought that if the bot were moving too quickly on the string, it might overshoot the jar. So I added "If the StringBot can move quickly on the string, but slow down as it nears the jar (to keep from overshooting the container), this will reduce missed drops." You might not have added that to your Robot Description—maybe you have an idea to prevent that from happening. Great! Put it into your Robot Description.

The major items you need in your Robot Description are the bot's need to traverse the room on a string, backward and forward, and to be able to hold a small object and drop it into a container. Make sure that the bot returns for another object (it will take many visits to fill the jar) and that you have a good Robot Description. If so, you are ready to move on to the next step—the Task List.

The Task List

If you'll remember back to Chapter 2, the `Task List` takes your `Robot Description` and breaks it down into the bot's individual functions—move forward, stop, drop rock, etc. Look on your Design Journal page. As I stated before, if you have a good `Robot Description` written down, the `Task List` almost writes itself. Go back and review your `Robot Description` and start listing the individual tasks the StringBot will perform. I've made my list for the StringBot, and you can view it in Figure 6-2.
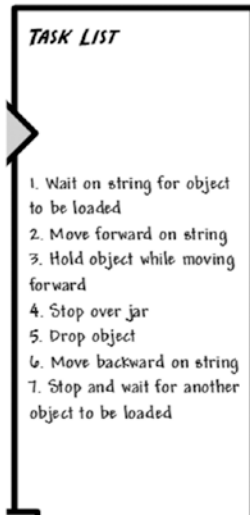

Figure 6-2. *The StringBot's task list is almost identical to the robot description*

The `Task List` is one of the most important sections of the Design Journal; each of the tasks you list will affect the construction of your bot. Since each item is an action item, each action must be paired up with a physical assembly that will either perform the action or assist with the performance of that action. Have you ever heard the phrase "form follows function?" Basically what it means is that the shape of an item (its form) is usually determined by what it will do (its function). Your bot is no different. In order for your StringBot to perform its duties, you must keep its main job in mind while you are designing it.

Now, back to your `Task List`. Read over it again and look at your `Robot Description`. Did you catch everything? Did anything new come to mind?

In my `Task List`, you might have noticed that the first item, "Wait on string for object to be loaded," isn't mentioned in my `Robot Description`. That's okay! This task was an afterthought. While the idea is fairly common sense (what good is sending the bot down the string if it's not holding an object, right?), I decided to add it to my `Task List` anyway. I could have gone back and added this at the end of my `Robot Description`, but as long as I have it *somewhere* on my Design Journal page, I should be okay.

I want you to keep in mind that your Design Journal page is for you to keep track of *all* of your thoughts on this project. It doesn't have to be neat and clean—as a matter of fact, the best Design Journal page will have scribbles and notes and things crossed out—a real mess! Later, if you want to record your work for history, feel free to rewrite your Design Journal page all nice and clean.

The final point I want to make on the `Task List` is an item that is *not* listed. Look back at the last sentence of my `Robot Description`—"If the StringBot can move quickly on the string, but slow down as it nears the jar (to keep from overshooting the container), this will reduce missed drops." Do you see it on my `Task List`? Why isn't it there?

Well, it turns out that this really isn't a task. It's an observation that I want to remember later, when I start designing. It would fit perfectly in the `Mindstorm` box, but I didn't want to take the chances that I'd forget my thought later, so I put it down in the spot I was currently working—`Robot Description`. I mention this only as another example of how important it is to write down everything that comes to mind! That great thought you have now might slip away in 30 minutes when you're sketching or skateboarding or eating lunch. Remember: when the thought pops into your head, write it down! (I'll revisit that last sentence in my `Robot Description` later in the "Mindstorm" section.)

This `Task List` was shorter than the one for the ExploroBot (mainly because this bot goes back and forth with one task in mind—the ExploroBot moved forward different distances, which were each given their own task), but each item is just as important.

Limitations and Constraints

I want to remind you here that you should write down in the `Limitations/Constraints` box *only* limitations that come quickly to mind. Don't spend too long thinking over this section because, truthfully, until you start building, you really can't imagine all the limitations that you are going to run into. Just visualize the StringBot (or the shoebox) moving down that string and write down where you think you might run into trouble. Spend five to ten minutes (max) thinking about a bot that moves along a string to drop a pebble in a jar. What could go wrong? Well, take a look at Figure 6-3 to see what I came up with for the `Limitations/Constraints` box.
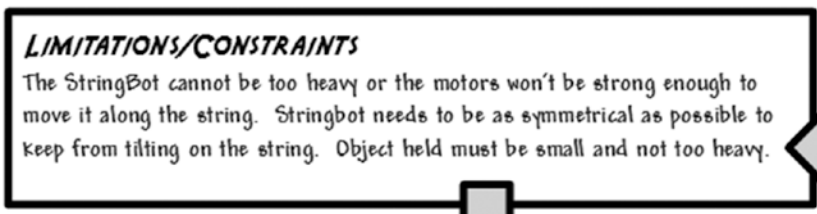

LIMITATIONS/CONSTRAINTS
The StringBot cannot be too heavy or the motors won't be strong enough to move it along the string. Stringbot needs to be as symmetrical as possible to keep from tilting on the string. Object held must be small and not too heavy.

Figure 6-3. *The StringBot does have some limitations to overcome*

One *huge* constraint that we really need to focus on for the StringBot is how it will move.

When it comes to the StringBot's size and weight, we need to keep it at a minimum. Think about a simple piece of string or twine tied between two poles (or pegs). If you put something extremely heavy on that string, the string will dip down. If it dips down too much, it might slide to the center of the string and stop moving completely. And even if the string doesn't dip down, another problem can occur. If a bot's motor spins too fast *or* if the motor is simply too powerful, the bot might just spin in place and not move at all! The EV3 parts are made of plastic. Like a rubber tire on a wet road, if you try to move plastic parts on a string too quickly, it can be difficult for the plastic parts to "grab" on and move. Figure 6-4 demonstrates these two problems.

Figure 6-4. *Problems with a heavy bot on a string (left image) or a motor that spins so fast that the bot doesn't move on the string (right image)*

The next item on my `Limitations/Constraints` list is "StringBot needs to be as symmetrical as possible to keep from tilting on the string." If you didn't think of this one here, don't sweat it. It would have become *very* obvious once you started building. When it comes to your StringBot, what is on the left side of the string needs to be fairly symmetrical to what's on the right side. If one side has more components or motors, it will tend to pull the bot in that direction and can cause the bot to slip off the string (see Figure 6-5).
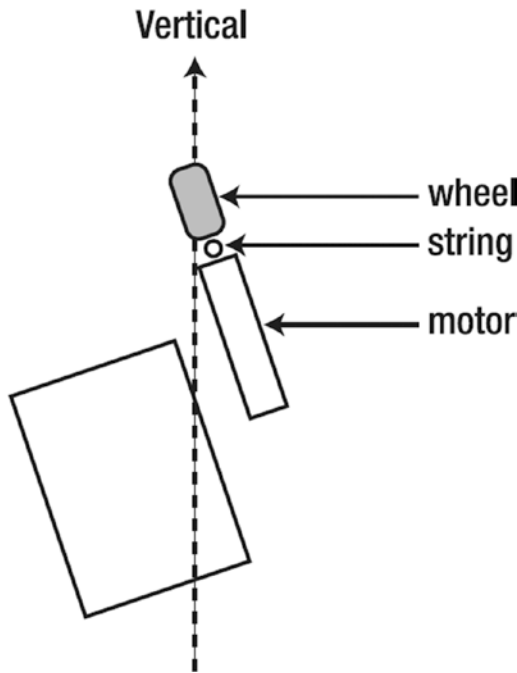

Figure 6-5. *A nonsymmetrical bot can lean*

The final constraint, "Object held must be small and not too heavy," is also a fairly important consideration. You can design the most lightweight bot on the planet, but if all you've got available for it to carry is a steel padlock or other heavy item, the bot isn't going anywhere.

The design of the bot should include some sort of "carrier" that has been built for a small item. Sure, it will take a lot of trips to fill a jar with smaller, lighter items, but the bot will also travel faster. A larger, heavier bot designed to carry a heavier item will take longer to make the round trip. Ultimately, it's your call. A heavier load will use up battery power faster because the motors will draw more electricity. But it could be argued that with a lighter object, it will take many more trips to the jar, which in turn will use up the batteries. (If you're really curious, try it both ways—design a HeavyStringBot and a LightStringBot and see which one solves the challenge faster.)

You might encounter more constraints as the project moves forward. Or maybe you've come up with a constraint or two not mentioned here. Perfect. Keep them in mind when designing your own version of the StringBot.

Mindstorm

I like this section. It's my favorite part of the process before I actually start grabbing and putting pieces together. When you are mindstorming, there are no right or wrong ideas. Just find a comfortable chair and start thinking about your bot. You've probably already got a ton of ideas floating around in your head; this is the time to put the best of them down on paper. And if you think you're not that creative, let me make another suggestion.

Sometimes I have difficulty "getting creative." I might be tired or just not in the mood to do some heavy thinking. If this happens to you, especially now that you're trying to get your StringBot design started, it can be frustrating. What I do when I find that my creative energies are not at full strength is to play a game called "Won't Work."

With "Won't Work," instead of focusing on solutions to the challenge, you're going to think about solutions that (ta-da) won't work. The idea is that if you're not feeling creative (typically considered a positive emotion), you can be anti-creative. (Okay, I made up that word, but it does work. Trust me.) By thinking about things that won't work, I typically begin to start finding things that *will* work . . . and then I'm thinking creatively. Here are a couple of examples of my "Won't Work" session:

- I'll never get the bot to cross the string by walking across like a human because that requires too much balance.

- I can't lower the bot into the jar (its weight might set off the trigger). If the weight isn't enough, I won't get the bot back for further attempts.

Your vision of the StringBot is probably going to be very different from mine. Whether you use the "Won't Work" trick or immediately start putting your thoughts down on the Design Journal page, you're beginning to finalize the design of your StringBot and how it will work.

Take a look at Figure 6-6 and you'll see the ideas I've written down for the `Mindstorm` section of the Design Journal page.
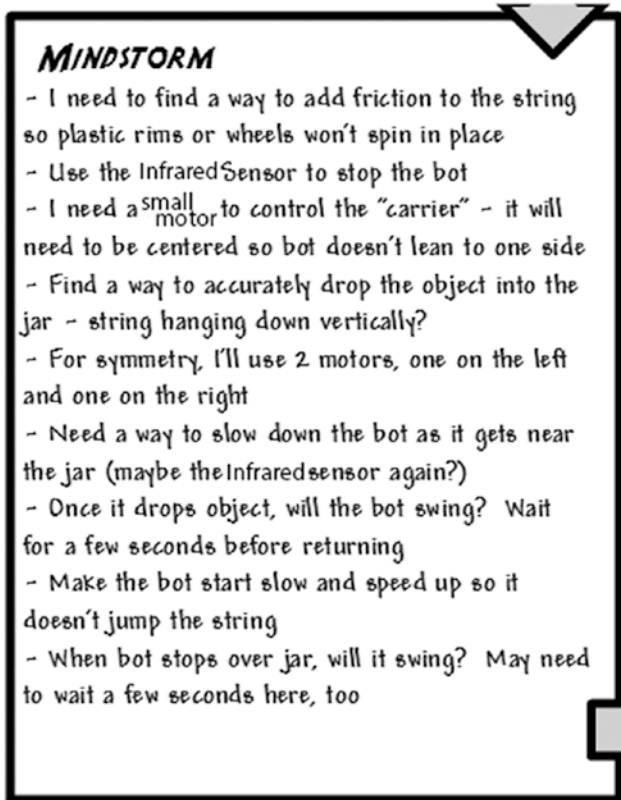
MINDSTORM

- I need to find a way to add friction to the string so plastic rims or wheels won't spin in place
- Use the Infrared Sensor to stop the bot
- I need a small motor to control the "carrier" – it will need to be centered so bot doesn't lean to one side
- Find a way to accurately drop the object into the jar – string hanging down vertically?
- For symmetry, I'll use 2 motors, one on the left and one on the right
- Need a way to slow down the bot as it gets near the jar (maybe the Infrared sensor again?)
- Once it drops object, will the bot swing? Wait for a few seconds before returning
- Make the bot start slow and speed up so it doesn't jump the string
- When bot stops over jar, will it swing? May need to wait a few seconds here, too

Figure 6-6. *The Mindstorm section will help you complete your StringBot design*

The first Mindstorm entry came to me fairly quickly—"I need to find a way to add friction to the string so plastic rims or wheels won't spin in place." I know from playing around with my MINDSTORMS EV3 kit that most of the pieces are made of very smooth plastic. My thoughts for the StringBot don't include hands and arms like a monkey. I plan on building a StringBot that moves along a string like a cable car (see Figure 6-7). I'm concerned that if I use one or two pulley wheels, the small wheels will slip on the string. When I begin to build, I'm going to try to incorporate one of the rubber wheels, because I think the rubber probably won't slip on the string.
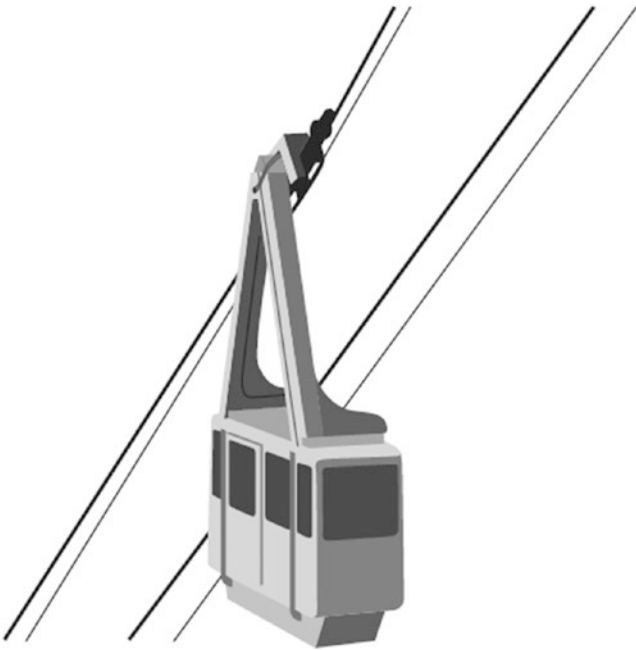
Figure 6-7. *My StringBot will work like a cable car. But will the small pulley wheels slip?*

I won't go through every item on my `Mindstorm` list, but there are a couple more items I feel are important for me to explain. The first is, "Find a way to accurately drop the object into the jar—string hanging down vertically?"

Most jars that I own have small openings at the top. I don't want my StringBot wasting time moving down the string and missing the drop. Maybe for your challenge setup, you'll use a large jar or box, but where's the fun in that? So it occurred to me that I better come up with a way for my bot to accurately drop its held object right into the jar. I'll obviously test my bot many times by observing where the object falls when the carrier releases it. What I think I'll do is tie a piece of loose string around my bot. Gravity will keep the string hanging straight down; when the string moves over the jar and is roughly in the center of the jar, I'll know to stop the StringBot and release the object.

Two other items in my `Mindstorm` box mention making the bot pause after certain actions. I believe (because I haven't tested it) that the bot will swing a little when it comes to a stop *and* after it drops the object. I want the bot to stabilize before it drops the object, and then again before it begins its return trip. If the bot is swinging, the dropped object might miss the jar. Also, a swinging bot might jump the string, and then it won't be able to return.

Your main objective here is to simply have some fun and write down some of your initial thoughts on what you'd like to do with your bot design. You may have to take a completely different direction after some testing. You may find that you exhaust your supply of a particular component. What you write down is not going to lock you in to a particular design. You can change the design anytime—even start completely over. Print out another Design Journal page and try a different design. It's supposed to be fun, so make it fun. Go crazy with your ideas—the crazier, the better!

The final `Mindstorm` item I want to quickly discuss is, "Make the bot start slow and speed up so it doesn't jump the string." I added this item because, in my earlier EV3 experiments, I've seen that when the motors start spinning quickly, sometimes the power of the motors can make a bot twist or "jump" and can slightly alter its direction. I offer this only as a suggestion—sometimes it's best to slowly increase the power of your motors while testing your bot to determine the best speed. Keep that in mind when you begin testing your StringBot.

Well, we're almost done with the Design Journal page. It's time to take everything you've collected—`Robot Description`, `Task List`, `Limitations/Constraints`, and `Mindstorm` information—and start with some rough sketches to help you when you begin construction.

Sketches

Even though we're not in the same room, I'll know if you're snickering. I told you before that I'm not an artist, so keep that in mind when you begin to look at my sketches of the StringBot.

Luckily, I know a little bit about what I'm looking for in terms of the shape and size of the StringBot. One of my constraints mentioned keeping the StringBot symmetrical. I know that if the bot isn't symmetric, it will lean to one side . . . and if it comes off the string, that will be bad, right?

In my `Mindstorm` box, I indicated that I want to use two motors, one on the left and one on the right. If I do this, I'll have to place that third motor near the middle of the bot to keep it from leaning. Since I'm going to use the idea of a cable car, I want the pulley wheels to be on top. And since I know my StringBot won't work without the Intelligent Brick, I've got to decide whether to have the Brick mounted horizontally, with the buttons parallel to the ground, or mounted vertically, so I can view the LCD screen as the bot moves away from me.

Okay, so it's time to draw. I tend to draw using basic shapes like rectangles, squares, and circles, so my sketches will consist of components drawn using their most basic shapes (the motor, for example, is two circles connected by a small rectangle). Figure 6-8 shows my initial sketches for the StringBot—no laughing!
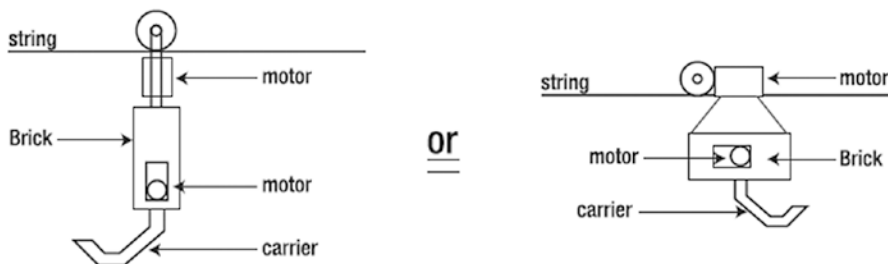

Figure 6-8. *I use basic shapes to define the StringBot in the Sketches area*

I've got quite a few concepts to test out when I start building. I might find that the vertical mounting of the Brick doesn't balance as well on the string as mounting the Brick horizontally. Or I might find that the weight of the third motor (for the carrier) isn't properly balanced and causes the bot to lean. Ultimately, if the StringBot moves to the end of the string, drops the object, and returns, I don't care what it looks like. (Okay, maybe I do a little—everyone loves a nice-looking bot.)

Chapter 7 shows you how to assemble my version of the StringBot. Compare it to my sketches and maybe you'll see the final version hidden in those rough drawings. Feel free to build my version. Or, if you're really proud of your design, go ahead and build YOUR StringBot. Remember, if your bot can fill a jar with dropped objects, you shouldn't have any trouble completing this challenge. Time to build the StringBot!