**Exercises**

1. Create a new sprite that looks like a button.
2. Brainstorm ideas for what you want the button to do. The action should be something that one of the three existing sprites will do. Describe your idea here. You will create a new feature of the Scratch project by making this button work properly.
3. Right-click the new sprite in the sprite list and change the name of the sprite to describe what it does.
4. When you duplicated an existing sprite to make your new sprite, the new sprite inherited a copy of the existing sprite’s scripts. You can see the script(s) in the script area. That script includes a broadcast block. Select the dropdown arrow for the broadcast message, and select **new**. Name the new broadcast message following the target.action() convention.
5. Create a new script in the target sprite’s script area. Begin the script with the when I receiveblock.
6. Strategize, code, and test, iterating (repeating as necessary) until the new feature works as you want it to.

**Conclusion Questions**

1. Scratch broadcasts a message to all sprites. This is just how Scratch is set up as a programming language. We are using messages so that they are targeted to a particular sprite. In a large project with hundreds of sprites and scripts, explain why it might help to follow our convention that only one sprite will respond to a message and to follow our naming convention for messages.

2. Based on what you learned during this activity, explain what you think **events** and **handlers** are.

3. As computer programs get big and complicated, it is essential to manage the complexity. An important strategy for managing complexity is called **encapsulation**. With encapsulation some details of one part of a program are hidden from other parts of the program. In cars, for example, the details of the accelerator pedal’s actions are encapsulated under the hood. As the user of the accelerator pedal, the driver only gets to press the pedal and is not encouraged to tinker with the internal mechanisms that make the pedal do what the engineer intended.

a. What are some details of the playG() method that are encapsulated by the handler?

b. A software developer can make it easier for other developers to tinker with the details of the program she is writing by exposing the details. We’ll get into what this means later. For now, just consider: What details mentioned in part a might be useful to expose? In other words, what variations on playG() do you think would be convenient to allow?