So far we've controlled the behavior of the different objects in our games using events and actions. These actions let the instances of the object perform tasks when certain events occur in the game. In this chapter we are going to define those tasks in an alternative way: by using programs. Programs define tasks through lines of text called code that use functions instead of actions. This extends the scope of Game Maker considerably as there are only about 150 different actions but close to a thousand functions. These functions give you much more control than actions, allowing you to define precisely how tasks should be performed in different circumstances.

The text in a program needs to be structured in a very particular way so that Game Maker can understand what you mean. Communicating with Game Maker in this way is like learning a new language with its own special vocabulary and grammar. The programming language Game Maker uses is called GML, which stands for Game Maker Language. If you have written programs before in languages like Java or C++, then you will notice that GML is rather similar. However, every programming language has its own peculiarities, so you will need to watch out for the differences.

Before we go into more detail about GML, you need to know how to tell Game Maker to execute a program using a script resource. Script resources are similar to other resources like sprites and backgrounds in the way they are created and accessed through the resource list. You create a script resource by choosing Create Script from the Resources menu, and then typing your program into the text editor that appears. Once you've created your script, you can include an Execute Script action to call the script in a normal event just as you would for any other action. We'll see how this works in more detail next.

**Note** You can also use an Execute Code action for including GML. Dragging this action into an event will cause the editor to pop up so that you can type the program directly into the event. Although this method is sometimes easier, it is often better to use scripts because they can be reused more easily.

To help you come to grips with GML, in this chapter we're not going to create a whole game but just small examples. Unfortunately, because of its size, we cannot cover all of GML, but we will discuss many key aspects. You can always refer to the Game Maker documentation for a complete overview.