

第 3 篇

编程和破解

第 10 章 Scratch 编程

第 11 章 Python 编程

第 12 章 硬件破解

第 13 章 扩展电路板

第 10 章

Scratch 编程

到现在为止，你已经学了很多在树莓派上运行别人写的程序的技巧和方法，但树莓派的主要目标是让所有人（而不仅是资深专家）能够在树莓派上开发自己的应用程序。树莓派基金会致力于让树莓派成为各个年龄段教学的工具设备。

10.1 Scratch 介绍

Scratch 是由美国麻省理工学院媒体实验室的 Lifelong Kindergarten 小组在 2006 年开发的，是 Squeak 和 Smalltalk 语言的一个分支。Scratch 继承了它们的核心编程概念，让它能够很容易让所有人使用。冗长又让人疲劳的代码输入（对小孩来说是枯燥乏味的）被替换为一个个简单的类似拼图游戏一样的拖拽编程环境，然而它却非常鼓励编程的思想，拥有所有编程语言所使用的核心概念。

从官方网站上来看，Scratch 是针对 8 岁及以上孩子的，但年纪更小的程序员也可以轻易上手 Scratch，并得到很多的帮助和指导。Scratch 看起来很强大，其背后的环境更是丰富多彩。鼠标驱动的用户界面类似一种编程语言，包括了强大的多媒体功能。所以，有 250 万用户在 Scratch 的官方网站上共享软件项目是不足为奇的，它们多数都和游戏一样有趣。

在孩子们的游戏中悄悄灌输并鼓励他们学习开发自己的游戏是一个极好的方式。Scratch 友好的用户界面和出色的操控性使孩子不太可能因为陡峭的学习曲线而感到沮丧。而且，在 Scratch 中学习的编程概念为后边学习更为灵活的语言，如 Python（请参见本书第 11 章），提供了一个良好的基础。

作为一种编程语言，Scratch 提供的不仅仅是一个游戏框架。它还可以用于创建互动演示和漫画，以及通过使用附加的接口硬件与外部的传感器或马达进行交互，如 PicoBoard 和 LEGO WeDo 的机器人套件。

树莓派推荐的 Debian 发行版配备了最新版本的 Scratch 环境，如果你已经安装了本书中所有相关的开发工具和环境，就可以开始进一步学习了。如果你使用的是不同的操作系统，请访问 Scratch 官方网站 <http://scratch.mit.edu> 下载软件（你也可以找到适用于 Windows 和苹果的 OS X 版本）。

10.2 例 1: Hello World

每当开始学习一种新的编程语言时，一个约定俗成的方式是写一个简单的程序在屏幕上显示一行字，这就是所谓的 **Hello World** 程序，同时这也是走向自主开发程序道路的第一步。

不同于其他传统编程语言，**Scratch** 不希望用户刻意去记很多指令语句，比如 `print` 或者 `inkey$`。相反，几乎所有的工作都可以通过拖拽相应的代码块来实现所需的逻辑组织结构。

首先，双击桌面上的图标或者单击程序菜单中的图标来载入 **Scratch**。几秒后，**Scratch** 主界面就会出现（图 10-1）。如果界面不在中间或者比较小，单击最大化按钮（顶部标题栏右边三个按钮中中间的按钮）来放大界面。

Scratch 界面被分为多个窗格。左边的窗格是选项板，它包含了所有构成程序的不同代码块。程序用到的对象（也称元素）列表显示在右下角的窗格中，该窗口也包含了对应的舞台控制器，来控制对象出现的舞台。而舞台本身显示在界面右上角，它用来显示程序的运行效果。最后，窗口中间的窗格显示了程序的构成。

为了让用户很容易地开始编程，一个新的 **Scratch** 项目已经包含了一个空白舞台和一个元素。你需要为它制作一个程序，单击窗口右上角的绿色旗帜不会发生任何事情，因为 **Scratch** 并不知道你想做什么。

对于 **Hello World** 程序，你需要单击屏幕左边的颜色按钮来改变选项板块的模式。下边列表中间有一个写着“say Hello!”的模块，单击此模块然后将其拖动到 **Scripts** 窗口中。如果你一定要坚持流行了几十年的传统，你也可以单击程序段中的“say Hello!”，把文字修改为“Hello World!”（见图 10-2）。

如果你再次单击绿色旗帜，程序仍然什么都不做。这是因为，虽然 **Scratch** 知道它应该使猫说些什么，但是不知道什么时候说。该事件需要触发块，它可以在控制选项板中找到。

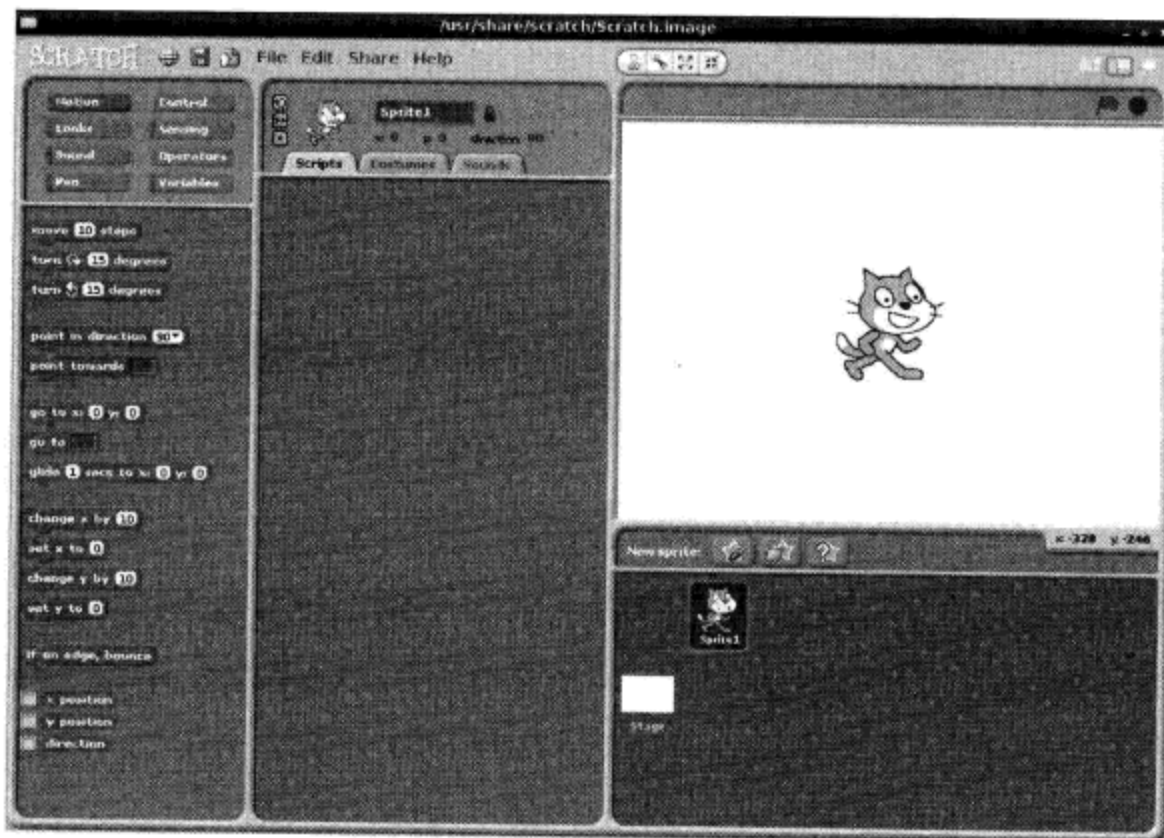


图 10-1 运行在树莓派上的 Scratch 主界面

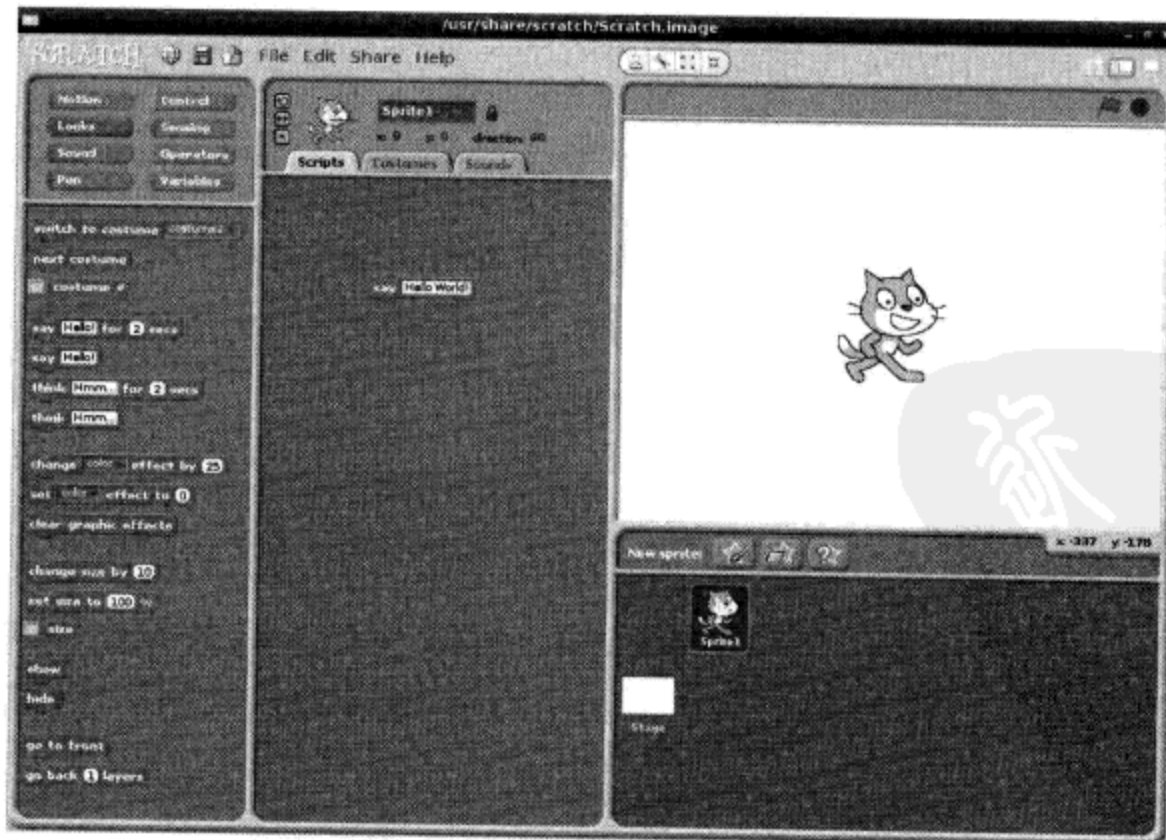


图 10-2 在 Scratch 程序中放置第一个程序块

现在单击“Control”按钮，然后拖动将最上边的条目（标有 when[旗帜]clicked）

放置在刚才提到的紫色的 say 方块上边（如图 10-3 所示）。如果你把它们放的足够近时，它会自动加入到现有的方块中，就像一个拼图。



图 10-3 在 Scratch 程序外形块中加入控制块

这种多个块连接在一起的方式，是 Scratch 的核心。如果你看看你刚才放置的控制块，你会看到它上边没有连接孔。这意味着，你不可以上面放置另一个块，因为这个控制块被设计为直接触发一系列其他块，它必须出现在一个程序堆栈的开始。同时，在 say 程序块下边有一个连接器，该连接器安装到其他块的顶部，这表明下方还可以放置更多的块。

现在已经有两个程序块了，我们单击屏幕右上角的绿色旗帜图标。这时候，会从猫的嘴里出现一个语言气泡（如图 10-4 所示），那么 Hello World 程序就完成了。

在开始学习下面的例子之前，使用菜单上的“File”按钮来保存你的工作。Scratch 不能同时打开多个项目，所以如果你创建一个新的空白文件，现有的文件将被关闭以腾出空间。不过不要担心，如果你忘了保存而直接去创建一个新的文件，且现有文件中有未保存的更改，Scratch 会提示你在关闭前保存现有的更改。

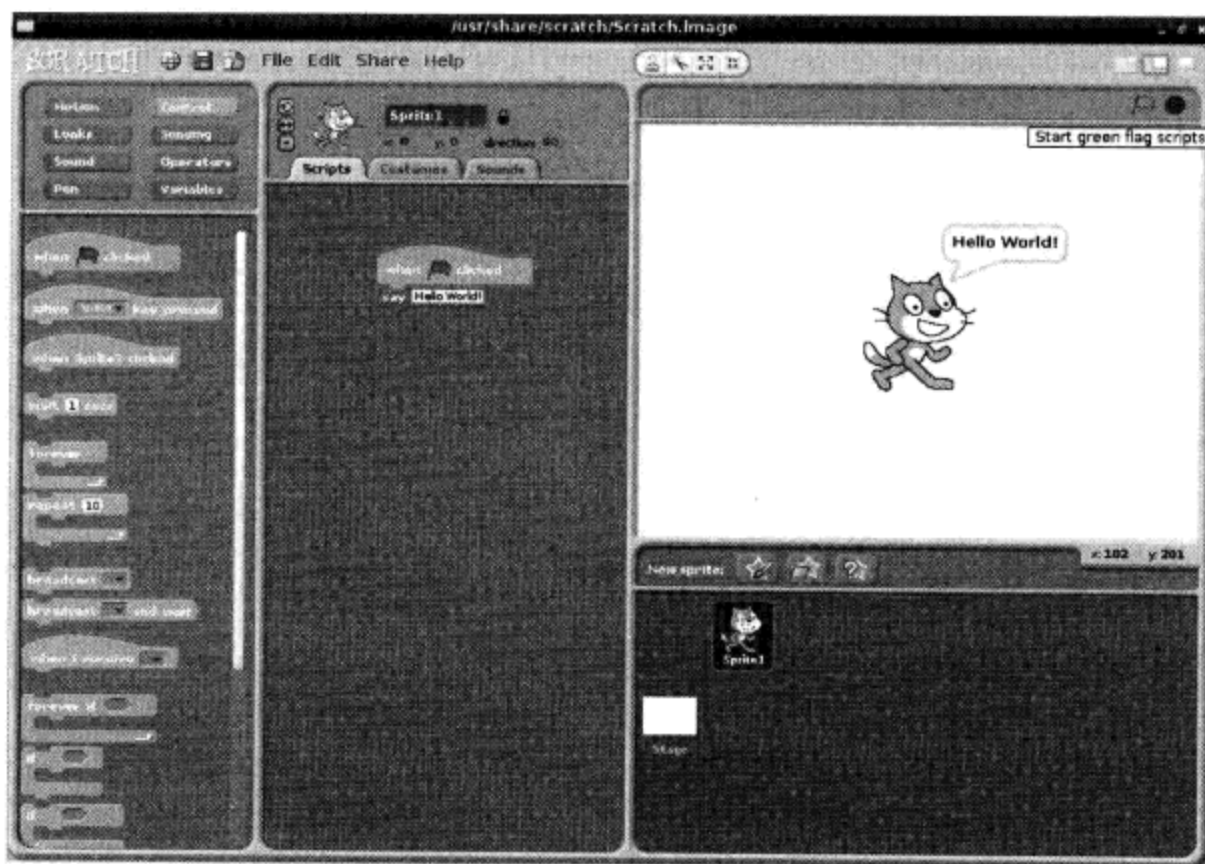


图 10-4 在 Scratch 中执行 Hello World 程序

10.3 例 2：动画和声音

Hello World 是一个非常传统的例子，它不是特别有趣。它也没有显示出 Scratch 的真正力量，Scratch 有强大的多媒体功能和元素处理系统。本系统特别适合简单的动画制作，它可以作为一个互动游戏的基础。

首先，重新加载 Scratch 程序或从“文件”菜单上选择“新建”启动一个新的项目。正如任何新启动的项目，Scratch 会提供一个默认的元素（你可以控制这个元素）。

要控制 Scratch 的一个简单的动画，你可以使用选项板的 Motion 部分。当你启动一个新的项目时，这是默认的选项板。拖动标记有 move 10 steps 的块到 Script 区域。正如它的名字所示，此块用于控制元素朝着它所面对的方向移动 10 步。在 Scratch 中，元素开始默认的面对方向是右，所以，元素将会向右移动 10 步。

10 步并不是一个很大的值，因此单击 10 将其更改为 30。那么这个块就变成了 move 30 steps。然而，一个向右移动的猫并不是很有趣，我们切换到 Sound

块选项板并拖动 play sound meow 块到 Scripts 窗口并和下边的 move 块连接。为了让猫在这个位置上保持一段时间，我们从控制选项板中加入一个 wait 1 second 块。如果没有这个，元素会迅速从其起始位置移动到目标位置。

为了让程序运行多次而不让猫从舞台上消失，在 play sound 块下边增加一个 move 10 steps 块，修改值为 move -30 steps。Scratch 中可以像这样使用负数，如果 30 让元素向右移动，那么 -30 会使元素向相反方向移动相同距离。

最后，从 Control 选项块中选择并添加 when [旗帜图标] clicked 块到 Scripts 区域最上边来完成程序，如图 10-5 所示。单击屏幕右上角的绿色旗帜图标来运行程序，不过要确保你有音箱或者耳机连接到树莓派以获得全部效果。

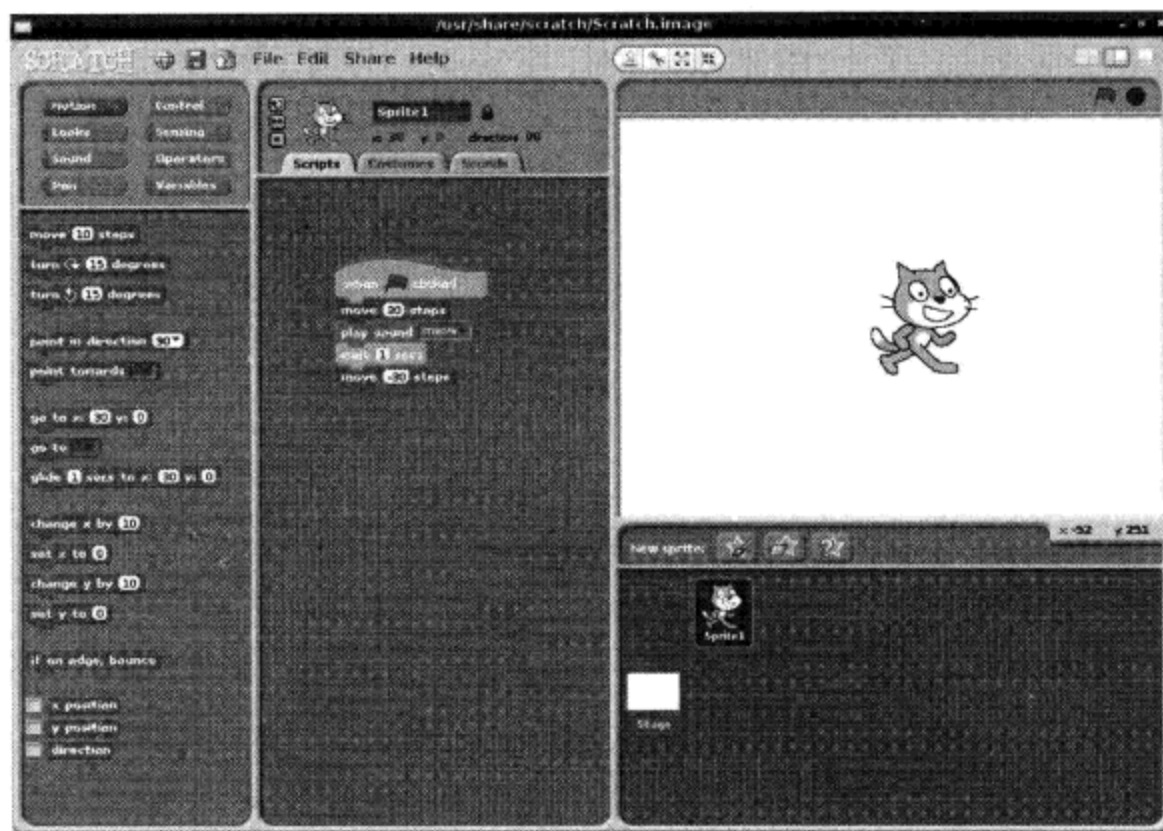


图 10-5 Scratch 中执行完整的动画程序

这个简单的动画程序能够用很多方式扩展。使用 Scratch 窗口右边舞台下方的 New Sprite 选项来添加更多的元素，它们可以独立地移动和播放声音。添加例 1 中的 say 块（或者 think 块，创建一个小的想象气泡而不是语言气泡）可以制造漫画的动画效果。

更重要的是，这个简单的例子说明了一个重要的编程概念。尽管只有 5 个块，它包含了元素的正负方向移动、声音播放和延迟的概念。为了介绍另一个概念（冒

着你因重复的噪声而抓狂的风险), 我们试着从 Control 选项板添加一个 `forever` 块 (如图 10-6 所示)。这会在程序中添加一个循环, 使它一直循环运行程序块, 或者直到你听够了这噪声后主动单击舞台上方的红色的 `stop` 按钮。

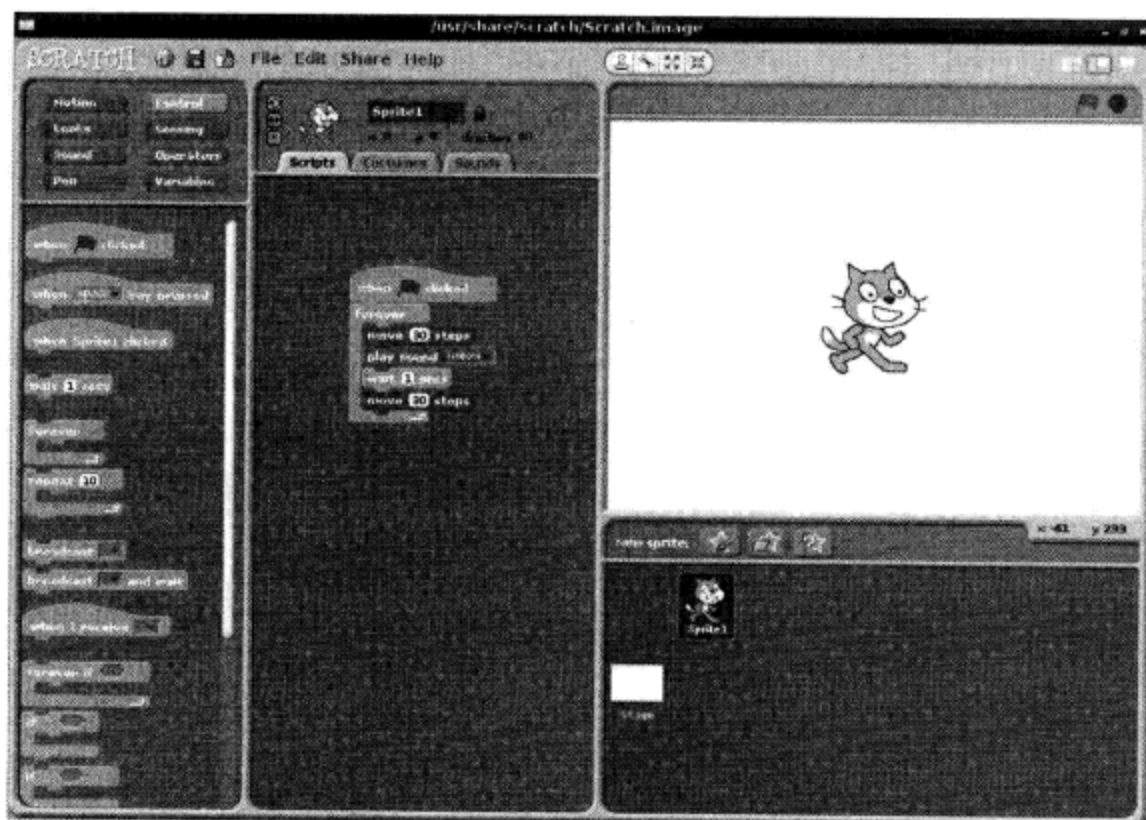


图 10-6 在 Scratch 动画中添加无限循环

10.4 例 3: 一个简单的游戏

使用 Scratch 可以制作简单的动画, 但该软件还允许用户从键盘输入来进行交互。通过把一些简单的动画控制和先前描述的程序结合, 你可以创建一个简单的游戏。同时, 我们引入元素碰撞、`if` 语句和输入的概念。

在这个例子中, 启动一个新的 Scratch 项目 (如果你还没有保存先前的例子, 记得先进行保存), 拖动 `move 10 steps` 块到 `Scripts` 窗口。这一次, 我们不使用单击标志图标方式来开始执行代码块, 在 `Control` 块选项板中, 拖动一个 `when space key pressed` 块到 `move` 块的上边。

正如名字所示, `when space key pressed` 块会等待用户的输入 (在此为空格键), 用此输入来触发程序块的执行。如果你按下了空格键, 程序会立即执行,

舞台上的元素会根据指令向右移动 10 步。

when space key pressed 块可以定制，这也是不同于 when [旗帜图标] clicked 块的一个方面。单击单词 space 旁边的下拉箭头，会看到所有该块可以监听的按键列表，选择其中的 right arrow 来把该块变为 when right arrow key pressed 块。

但是，一个只能让玩家向一个方向移动元素的游戏没什么意思，所以，拖拽一个新的 when space key pressed 块到 Scripts 区域。它并不能连接到现有的代码块中（你只能有一个触发块），像之前一样，单击 space 旁边的下拉箭头来定制该块，把它变为 when left arrow key pressed 块。最后，把选项板转换为 Motion 模式，拖拽一个 move 10 steps 到 when left arrow key pressed 块下方，同时把 10 改成-10。

现在，如果你按下左箭头或者右箭头，你会发现猫会根据你的输入进行移动（如图 10-7 所示）。按下左箭头，猫会向左移动 10 步（虽然就 Scratch 来说，它是向右移动了-10 步）；按下右箭头，猫会向右移动 10 步。

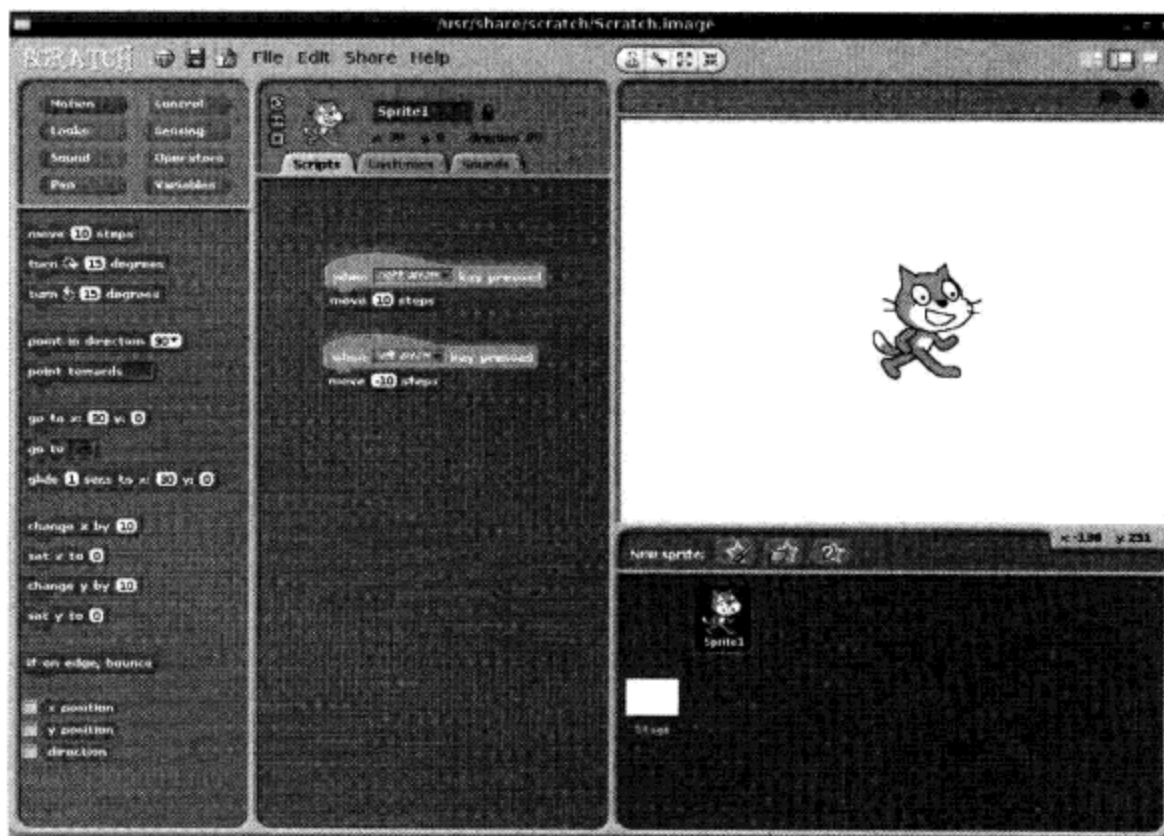


图 10-7 在 Scratch 中通过输入来控制元素移动

既然元素已经能够被玩家移动，是时候让该元素做点其他的了。由于这是一个非常简单的游戏，目标应该很简单，即收集食物。单击 Choose New Sprite From File 键，在 Scratch 窗口右下角 Sprite 选项板的右边三个按钮中间的那个。

一个对话框会出现，让你选择一个元素。双击 Things 文件夹，再双击 Cheesy-Puffs 元素（如图 10-8 所示）。这会在 Sprite 选项板中添加一个新的可控制的对象元素。

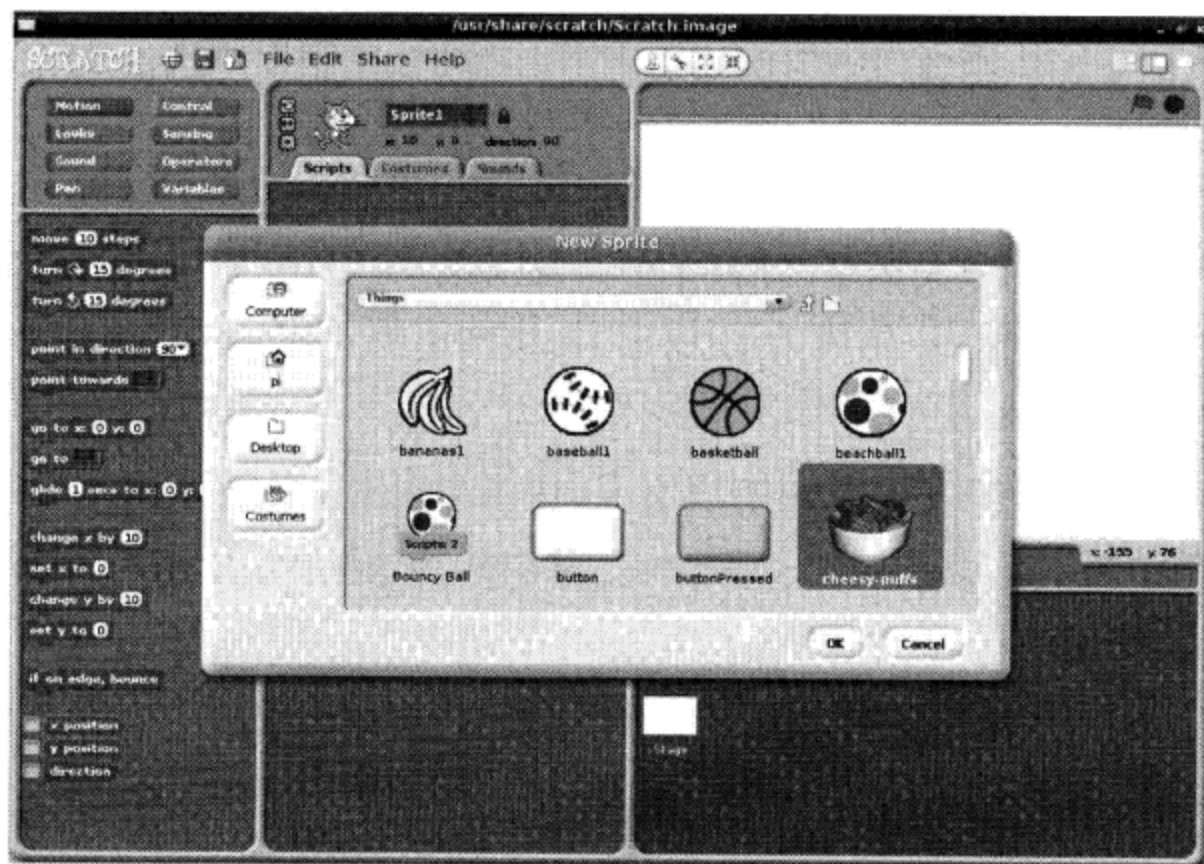


图 10-8 在 Scratch 游戏项目中添加 Cheesy-Puffs 元素

Scratch 语言本质上是多线程的，部分是面向对象的。这说明程序中的

小提示 每个对象包括元素都有自己的代码块，每个代码块同时并独立于其他代码块运行。如果使用得当，这些特性有利于创建一些很复杂的程序。

默认情况下，任何加入 Scratch 项目的新元素将出现在舞台中央。那么就会掩盖了现有猫元素。单击新的元素，并将其拖动到猫的右边来解决此问题。

这碗 Cheesy-Puffs 会移动，但是对于我们这个像人一样长着两条腿的猫来说还是太大了。单击 Shrink Sprite 按钮，它在舞台区域的右上角，看起来像是四个箭头指向中心。如果你不确定哪一个，把鼠标悬浮在图标上边，会显示出一个

简短的功能描述。

单击 Shrink Sprite 按钮（或者 Grow Sprite 按钮，它和 Shrink Sprite 功能相反）把鼠标指针变为一个类似按钮的图标。用这个新的鼠标指针，单击 Cheesy-Puffs 元素将其缩小。一直单击的话会一直缩小元素。一旦到达合理的大小，单击舞台区域之外的任何地方把鼠标指针还原。然后你可以把碗拖拽到靠近舞台右边界的位置。

现在，试着用箭头按键来移动猫到 Cheesy-Puffs。你可以看到，当两个元素相遇时，什么都不会发生。这是因为程序没有任何关于两个元素相遇时要执行的指令（也就是我们提到的元素碰撞），所以不会发生任何事。我们可以用一个新的块来完善它，即 Sensing 块。

保持 Cheesy-Puffs 元素为选中状态（它的图像应该显示在 Scripts 面板顶部，如果没有，双击舞台上的元素），单击 Sensing 按钮把选项板变为 Sensing 模式。在 Sensing 选项板中，拖拽 touching ? 块到 Scripts 区域。

类似 when space key pressed 块控制猫元素的移动，touching ? 块同样能够定制。单击问号旁边的下拉箭头，选择 Sprite 1（猫元素）。这个块将会在两个元素相遇时激活。

你可以通过单击 Scripts 面板中对应图片旁边的格子来命名一个元素。

小提示 为元素起一个有意义的名字（如猫、奶酪等）会帮助你更容易地跟踪程序中元素事件的发生。

观察 touching Sprite 1? 的形状，你可以发现，它上边没有类似拼图一样的接口，形状就像一颗钻石（在流程图中也用同样的形状表示决策点）。这不是偶然的，为了方便操作，大多数传感块需要嵌入在一个控制块中。

切换块选项板到控制模式，找到 if 块（形状类似一个压扁的粗糙的字母 C）。注意 if 块有一个钻石形状的缺口（和 touching Sprite 1? 的形状一样）。拖拽 if 块到 Scripts 面板，然后拖拽 touching Sprite 1? 块到它的钻石缺口上。结果是两个颜色的块上出现 if touching Sprite 1? 字样。

这表示有一个 if 条件在程序中。达到此点时，当且仅当该条件满足，位于其

限定内的所有代码将被执行。在这种情况下，该条件就是 Sprite2 和 Sprite1 相遇。配合使用 Operators 块的 and、or 和 not 逻辑，可以实现一些比较复杂的程序。

在 Looks 选项板中，拖拽 say Hello! For 2 secs 块到 if touching Sprite1? 块中间。修改文字为 Don't eat me!，再添加一个 wait 1 secs 控制块，修改值为 2。在顶部添加一个 when space key pressed 块，修改值为 when right arrow key pressed。最后，在 Looks 选项板中拖拽一个 hide 块到循环的最底部来结束程序，如图 10-9 所示。

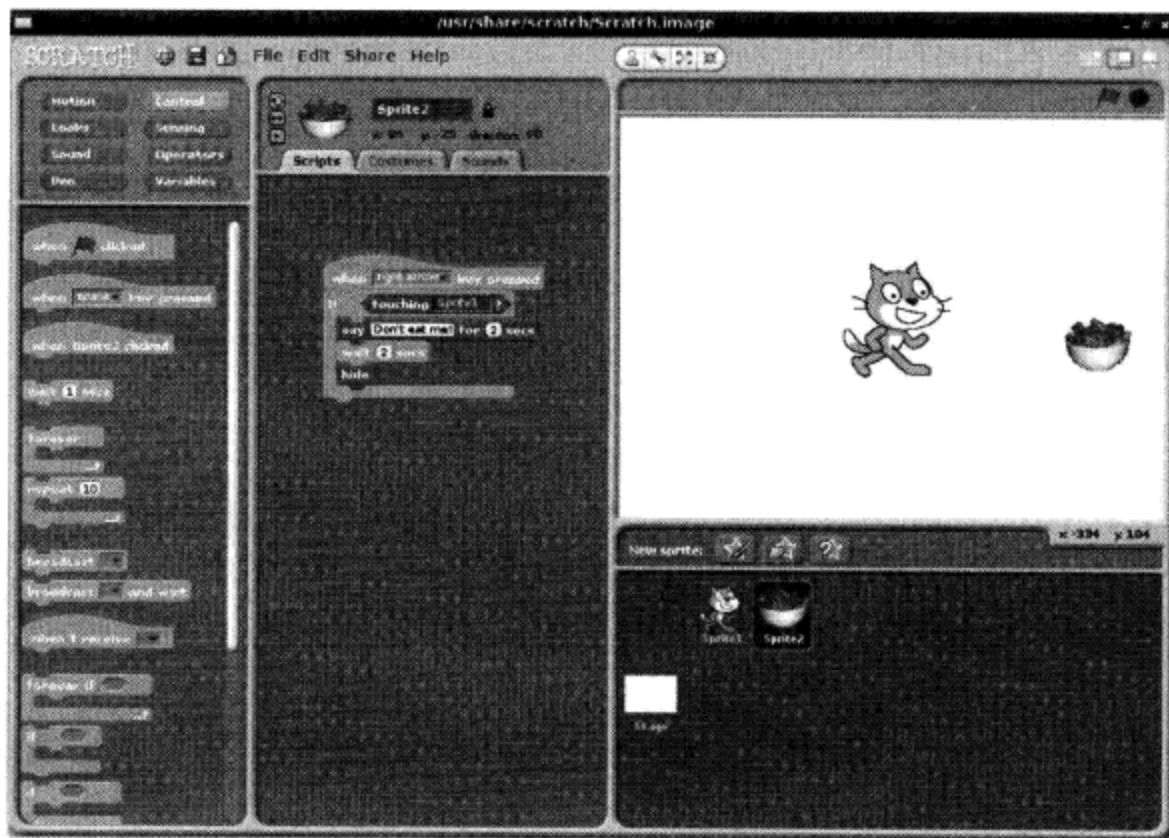


图 10-9 用 Scratch 的 if 块来控制 Cheesy-Puffs

布尔逻辑

布尔逻辑的命名是为了纪念 George Boolean，布尔逻辑或者布尔代数是理解计算机工作原理的核心。在 Scratch 中，布尔逻辑用 3 种方法来体现：and、or 和 not。

and 操作需要两个输入（例如在 Scratch 中的 Sensing 块）。如果两个输入都是真，那么输出为真。如果有一个为假或者都为假，输出为假。例如你可以利用该操作符来判断一个元素是否同时碰到另外两个元素。

对于 or 操作符, 如果输入中至少有一个为真, 那么输出为真。这是一个很好的代码重用方法。如果有很多元素对某个目标元素是致命的, 我们只需要一个代码块和 or 操作符就能达到任意一个元素碰到目标元素时立刻触发程序的目的。

最后, not 操作符是一个反向器。无论输入是什么, 输出总是输入的相反值。如果输入是假, 输出为真; 如果输入是真, 输出为假。

双击舞台上的猫元素来编辑它的脚本。Cheesy-Puffs 的脚本会消失, 但不用担心, 它仍然被保存着, 只有当你编辑它的时候才会出现。

拖拽一个 if 块和另一个 touching ? 块, 这一次修改 Sensing 块为 if touching Sprite2?。在该块中, 添加一个 wait 1 secs 控制块, 修改值为 2, 再添加一个 say Hello! 块和 wait 2 secs 块, 修改值为 Yum-yum-yum!。最后, 从底向上拖拽整个程序块, 使得底部和 move 10 steps 下边的 when right arrow key pressed 连接。最终的程序如图 10-10 所示。

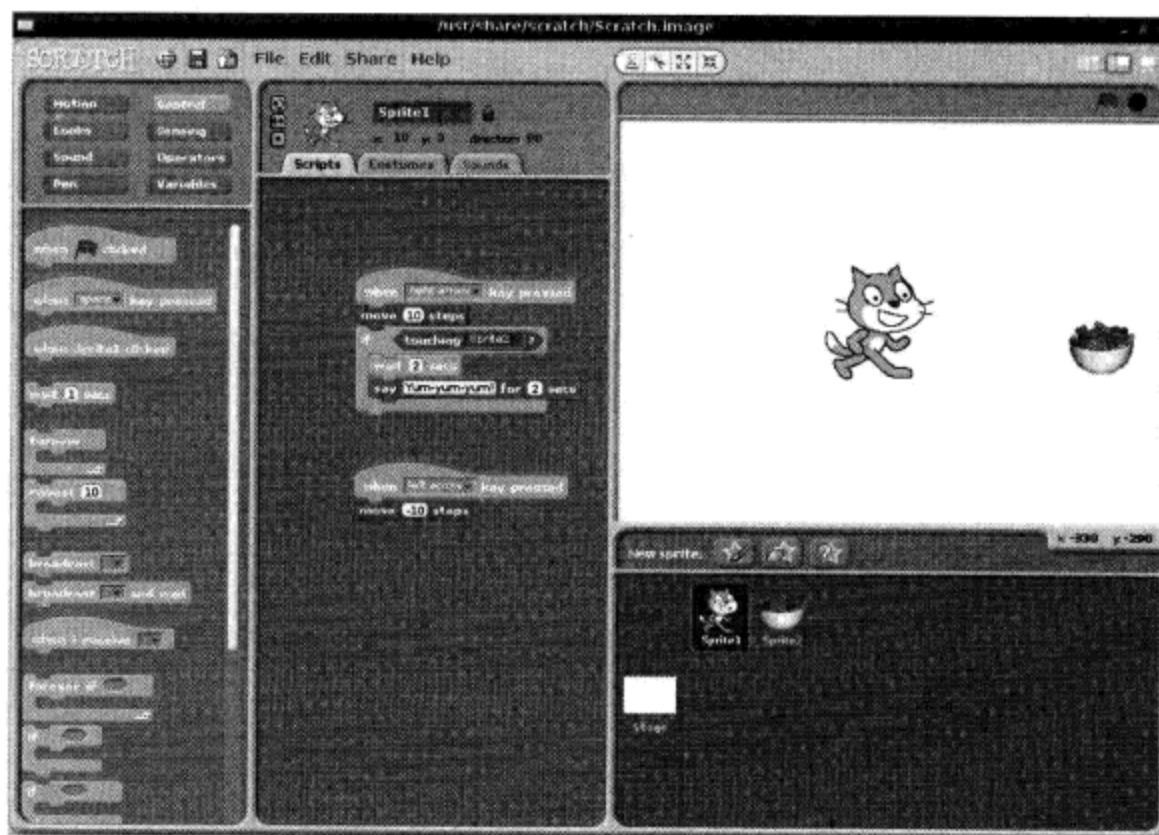


图 10-10 简单的 Scratch 游戏程序完整代码

如果你用右箭头键移动猫到奶酪旁，游戏就开始了。当猫到达奶酪旁，对话会出现同时奶酪会消失。

虽然这个例子有效地引入了一些重要的编程概念，但它不是最好的游戏编程方法。Scratch 包含一个消息广播系统，可以让一个对象的代码与另一个对象的代码交互，它使你能够创建更简洁的碰撞效果，而不依赖于精心策划的定时处理。

为了对广播方法进行实验，试着用控制选项板中的 broadcast 和 when I receive 块。对象中广播的消息会触发某一个包含 when I receive 标识的对象，这意味着你可以用它来把多个对象连接起来。

10.5 机器人和传感器

除了纯粹的基于软件的输入和输出，也可以把 Scratch 用于外部硬件传感器和机器人系统。由于其多线程的性质和功能强大的进程间消息传递系统，Scratch 可以用于创建一个令人惊讶的先进的机器人引擎，一个简单的机器人可以利用树莓派作为其核心。

Python 是一个更强大但明显更复杂的编程语言，与 Python 不同，Scratch 不能利用树莓派的 GPIO 端口与外部硬件通信。所以，它与物理世界的互动依赖于一些额外的设备。

10.5.1 PicoBoard 传感器

PicoBoard 是一个由 SparkFun 电子公司设计的附加模块，用于连接运行 Scratch 的电脑和滑动变量器、光线传感器、按钮、声音传感器和四对用户监控外部电气设备的鳄鱼夹。

目前有两种 PicoBoard。the original model，用于串口通信和 USB 连接。另一个由树莓派支持。只需把 USB 接口接到一个 USB 集线器上来连接树莓派，然后

重启 Scratch，你就可以用 Scratch 界面来调用树莓派的众多函数了。

PicoBoard 详情请访问 <http://www.sparkfun.com/products/10311>。

10.5.2 机器人与 LEGO

除了加载传感器模块，也可以通过 Scratch 来控制外部机器人系统。LEGO Education WeDo 机器人被设计为能够将电脑与电动机、距离传感器和速度传感器连接，这个设计的所有方面都可以在 Scratch 的程序块中找到。

WeDo 的套件对于儿童来说是一个很好的机器人入门教程。机器人部件夹在一起就像普通的 LEGO，而且没有任何因接触不良导致触电的风险。WeDo 软件套件兼容正规的 LEGO 和 LEGO 技术套件，这意味着它可以快速轻松地建立大项目。

LEGO Education WeDo 套件与 Scratch 的详情请访问 <http://info.scratch.mit.edu/WeDo>。

10.6 进一步阅读

虽然本章简要地介绍了 Scratch，然而这是非常简略的。但是对于年轻人来说，还是有些冗长，他们往往在大量彩色图片的辅助下能够更快地学习。

Scratch 官方网站上的 Support 板块 (<http://info.scratch.mit.edu/support>) 包含了起步手册的 PDF 格式文件链接。该手册使用丰富多彩、儿童友好的方式介绍 Scratch，是一个很好的学习工具。这样的学习经验可以进一步通过结合 Scratch Cards 加强，Scratch Cards 是可下载的 flash 卡片，包含了 Scratch 中每种类型块的描述解释。

麻省理工学院还为 Scratch 开设了论坛，让软件爱好者共同学习和分享解决方案。会员是免费的，网站在 Midori Web 浏览器（包含在树莓派的 Debian 发行版中）下工作正常。论坛地址是 <http://scratch.mit.edu/forums>。

然而，最简单的推进 Scratch 学习的方式就是玩。Scratch 的名称就来自唱盘。当 DJ 旋转记录盘，盘针会发出摩擦声。正如 DJ 混音现有的歌曲到一些新的东西中那样，Scratch 鼓励爱好者在官方网站提交自己的作品供他人下载、检查和修改。Scratch 网站目前拥有超过 250 万程序，这使其成为一个完美的学习如何从头开始的源，同时用于创建项目并与他人分享自己的想法。最近共享的项目列表可以访问 <http://scratch.mit.edu/latest/shared>。