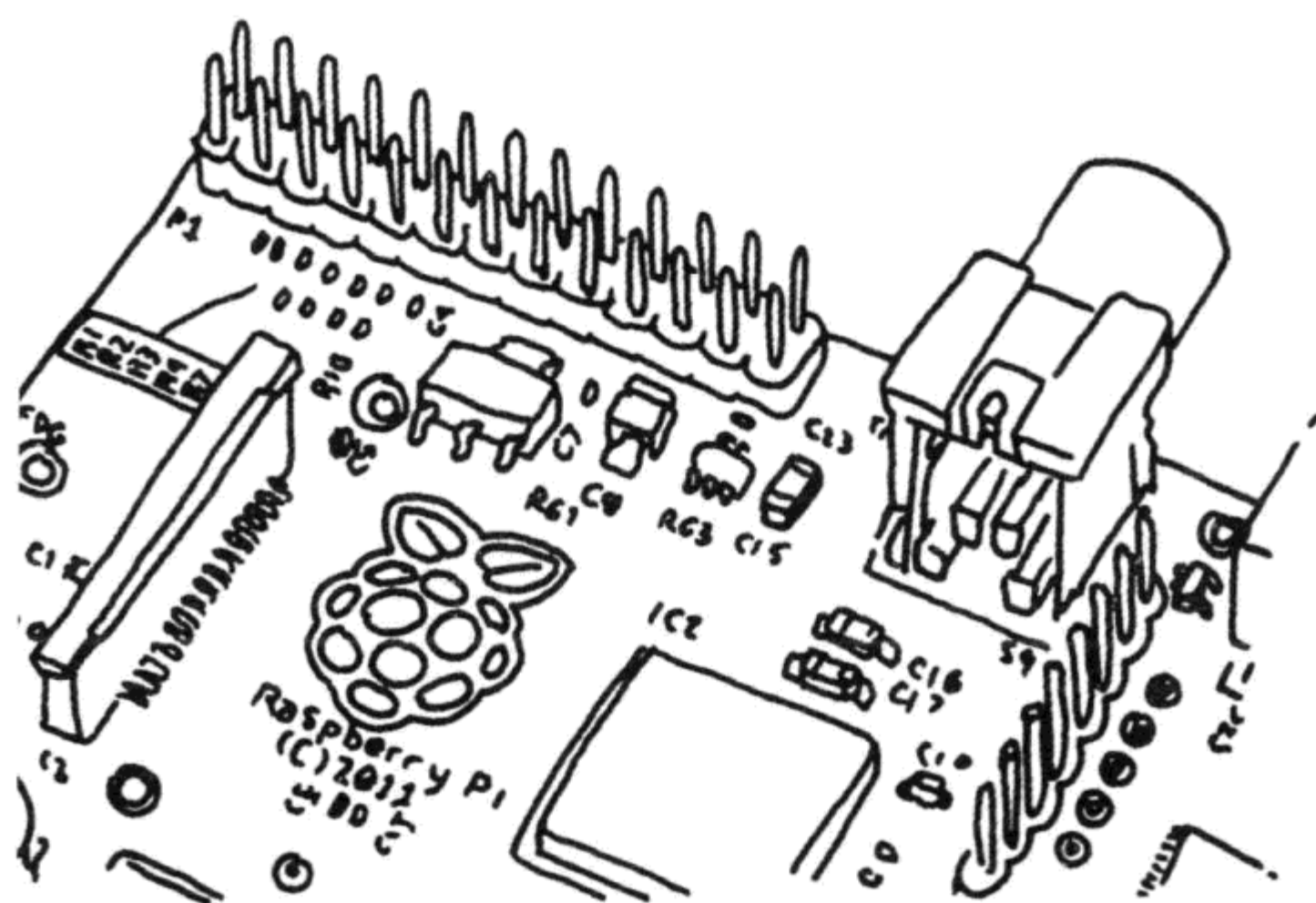


第 5 章

Pi上的Scratch

Scratch on the Pi





Scratch 由 MIT（麻省理工学院）媒体实验室的 Lifelong Kindergarten 小组开发，它采用一种创新的方式在青少年中开展程序设计教学。Scratch 程序由一系列彩色的代码积木组成，每块代码积木可以用于完成一项操作。用积木这种可视化的方式来编程，可以让一些初学者避免在初次接触到文本编程语言时产生困惑。

如果只是把 Scratch 看成一种简易的彩色编程语言难免有失偏颇，它其实提供了一个友好的开发环境，让开发者能更为便捷地完成一项任务。在程序运行过程中，当前运行到的代码积木会变得高亮，并且所有的代码积木都可以随时更换并实时看到更换后的效果，这对于初学者来说很有帮助。

编写 Scratch 程序的基本形式是操作角色在一个舞台上完成一系列动作。Scratch 有一个很大的用户社区，并且 Scratch 平台本身就提供了在社区中分享角色和代码的功能。

初识 Scratch

用 Scratch 编程非常容易，我们以一个名为“盒子中的猫”的程序作为开始。Scratch^①的主界面中包含了多个操作区域，图 5.1 中标明了每个区域的功能。你可以双击桌面上的 Scratch 图标，也

① 点击 Scratch 主界面左上角的“设定语言”（Select Language）地球图标可以选择 Scratch 界面上显示所用的语言。先点击地球图标，在下拉菜单中点击 More，再选择“简体中文”就可以把界面语言改成中文。如果先前已经把系统区域设置设为 zh_CN.UTF-8，则启动 Scratch 后会自动切换到中文界面。——译者注



可以点击屏幕左下角的桌面菜单并选择 Education → Scratch 来启动 Scratch。

Scratch 的程序由舞台上的角色组成，角色的行为由脚本控制，而脚本是由积木盒中的一块块积木组成的。在新建一个 Scratch 程序时，你会默认得到一只小猫作为你的角色。如果你不喜欢这只猫，可以在角色列表区域中的小猫角色上点击鼠标右键，并在菜单中选择“删除”（Delete）来删除它，然后再自己画一个喜欢的角色或点击角色列表区域上部的“来个令人惊喜的角色吧”（Get Surprise Sprite）按钮从 Scratch 社区设计的其他角色形象中随机选择一个角色。

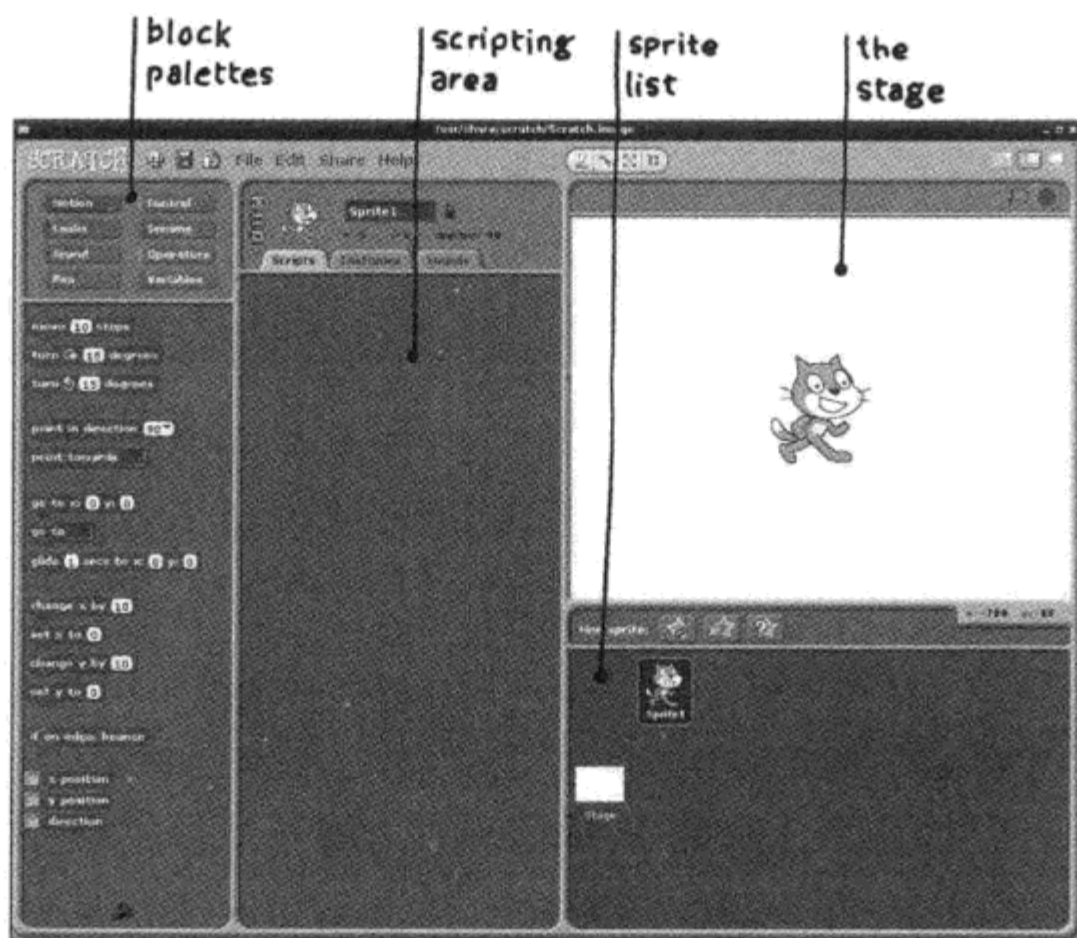


图5.1 Scratch的界面，程序的窗口被划分为多个区域，分别是积木盒、脚本与造型区、角色列表和舞台

脚本区中显示的是当前选中角色的控制脚本。在角色列表中点击猫的图标，由于这时还没有为这个角色编写脚本，所以你会看到一个空白的脚本区，可以在脚本区中的任意位置开始构建脚本。如



果你新添加了一个角色，也可以通过点击这个角色在脚本区中显示出与它相关的脚本。

通过把积木盒中的积木块拖动到脚本区，就可以来构建脚本。有些积木内部还可以包含其他积木——这从积木的形状上就可以看得出来。Scratch 中有 3 种类型的积木，如图 5.2 所示。



图5.2 Scratch中的3种积木：帽子型（左）、可堆叠型（中）和报告型（右）

帽子型积木

“当绿色小旗被点击”（when green flag clicked）积木就是帽子型积木的一种，它应该被安置在其他积木的最上方，等待相应事件的发生。

可堆叠型积木

上部有个凹槽、下部有个突起的积木块属于可堆叠型积木，它们可以被层叠组合在一起。所有的可堆叠型积木会按从上到下的顺序执行。

报告型积木

具有圆角或箭头型边框的积木属于报告型积木，它们可以安装在其他积木内部。报告型积木可以是一个变量，也可以用来获取鼠标坐标或其他信息。



要开始构建一个新的脚本，点击“控制”（Control）积木盒，把图 5.3 所示的积木拖到脚本区。

下一步是添加一个“重复执行”（forever）的可堆叠型积木，这块积木会不断地循环执行它内部的所有脚本，直到脚本的运行被中止为止，如图 5.4 所示。



图5.3

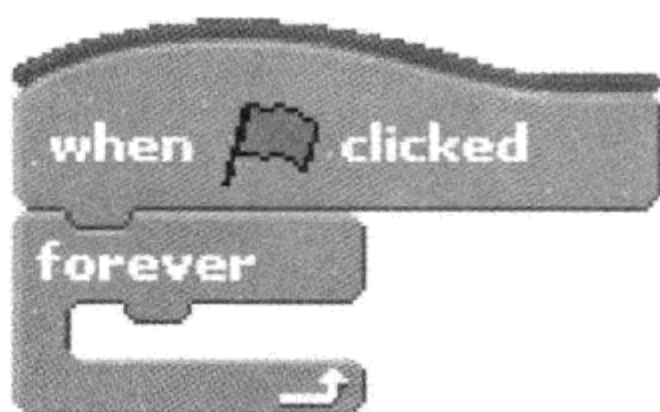


图5.4

你需要通过不断地练习才能熟悉各种积木的功能。当你选中一块积木时，也就同时选中了与它相接的后续所有积木。所以，要移除积木堆中的一块积木，你得先把这块积木后续的积木堆移开。你也可以在积木上点击鼠标右键来复制它们或获取相关的帮助信息。

再下一步，选择“动作”（Motion）积木盒，拖动一块“旋转”（turn）积木，并放置在“重复执行”（forever）积木中，如图 5.5 所示。

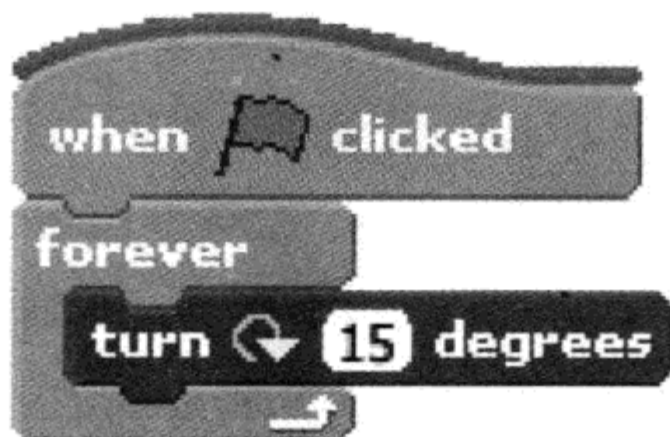


图5.5



现在，你可以点击舞台右上角的绿色旗子图标来运行这个脚本了。Scratch 有一个优点——在脚本的执行过程中你也可以对脚本做出修改，并且这些修改会立即在舞台上展现出来。这对于调试脚本而言十分方便。



当你点击绿色旗子图标时，Scratch 会向你的项目中的所有脚本发送开始运行的信号（如果点击的是停止图标，则会发送停止信号）。

如果你注意观察“旋转”（turn）积木，你会发现旋转角度的数值是在一个圆角矩形框中显示的，你可以直接修改这个数值，也可以用一块形状相同的积木来替换这个数字。下面我们来做一点修改，让角色随机运动起来。

首先，从“数字和逻辑运算”（Operators）积木盒中找出“随机选一个数”（pick random）积木，用它替换掉“旋转”（turn）积木中的默认值。替换后，修改“随机选一个数”（pick random）积木中的值，使它可以产生一个 -10 ~ 10 的随机数，如图 5.6 所示。

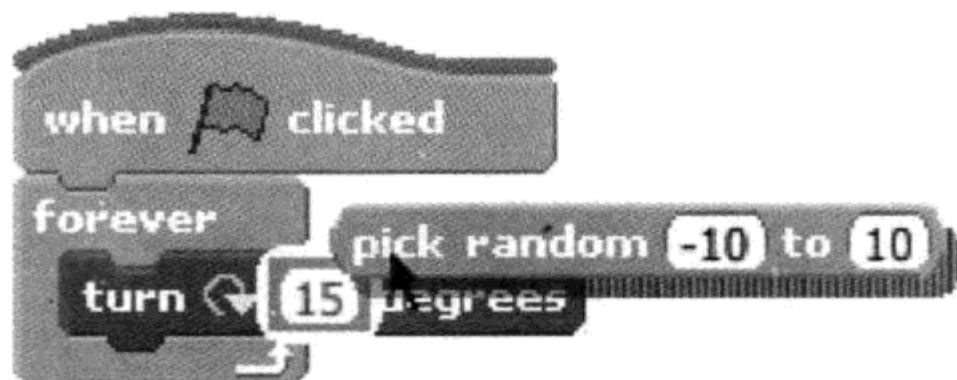


图5.6

从“动作”（Motion）积木盒中找出“移动”（move）积木，这个积木会控制角色以当前的角度向前移动指定个数像素的距离。



当前的角度会在脚本区上面的蓝色区域中以数字形式显示出来。当你把这块积木放置到你的脚本中，你马上会看到角色开始了随机移动，如图 5.7 所示。

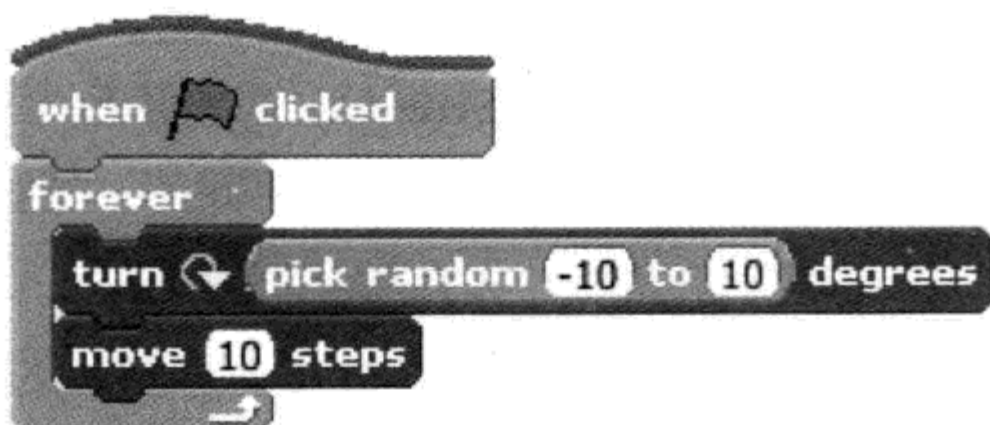


图5.7

最后，为了不让角色跑出舞台边框，添加一块“动作”（Motion）积木盒中的“碰到边缘就反弹”（if on edge, bounce）积木，如图 5.8 所示。

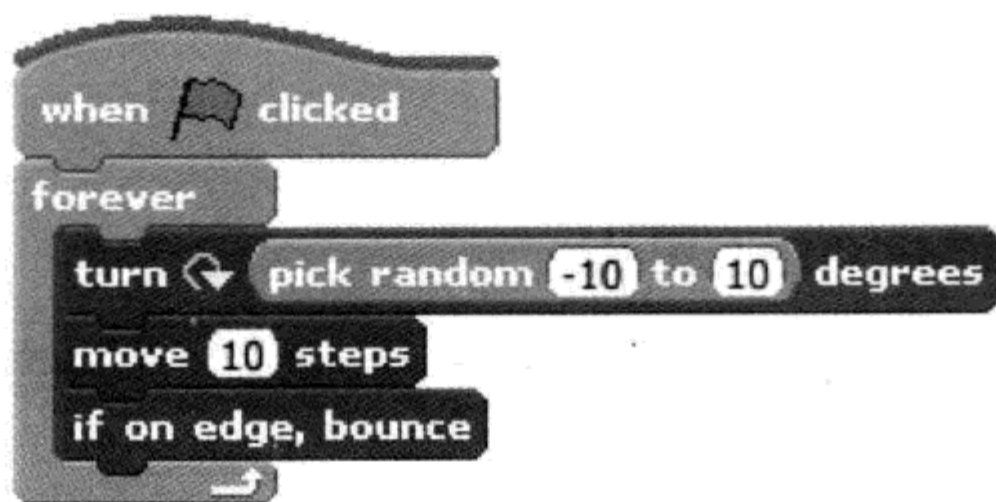


图5.8

以上就是编写 Scratch 脚本所需的全部基础知识！

舞 台

舞台是指界面右上角的区域，程序中的角色在舞台区域完成脚



本中所指定的动作和交互。与角色一样，也可以用脚本来改变舞台的外观和行为。可以通过脚本区的“多个背景”（Backgrounds）页来绘制舞台的背景图片。

Scratch 中使用的坐标系统与其他常见的多媒体环境（如 Pygame）中的坐标系统不一样，Scratch 中的坐标系统采用了数学

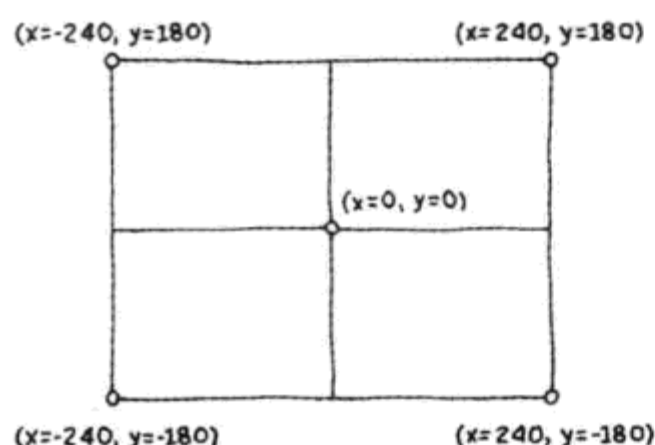


图5.9 Scratch的坐标系统，原点
设定在舞台的正中央

中常用的方式，即坐标 $(0,0)$ 位置对应舞台的中心位置。如图 5.9 所示，舞台的可视区域是 $(-240,180)$ 和 $(240,180)$ 为对角顶点的矩形区域。你可以用程序控制角色移动到舞台的可视区域以外的位置。当选中一个角色时，可以在它的脚本区域顶部看到它所处的当前位置，

在舞台的底部可以看到当前的光标位置。

有关角色的两点知识

在脚本区，还有两个页面可以使用：造型（Costumes）和声音（Sound）。

一个角色的造型包含了多幅图像，这些图像可以用于在制作这个角色相关的动画时切换使用，用于表现一个角色的不同状态（如一艘爆炸的飞船）。例如，创建一个新的角色，它的图片是一只睁开的眼睛，我们可以通过创建造型来让这只眼睛眨一下。点击“绘制新角色”（Paint New Sprite）按钮——状态区中“新增角色”（New Sprint）右侧的图标，并按图 5.10 绘制，完成后点击“确认”（OK）。

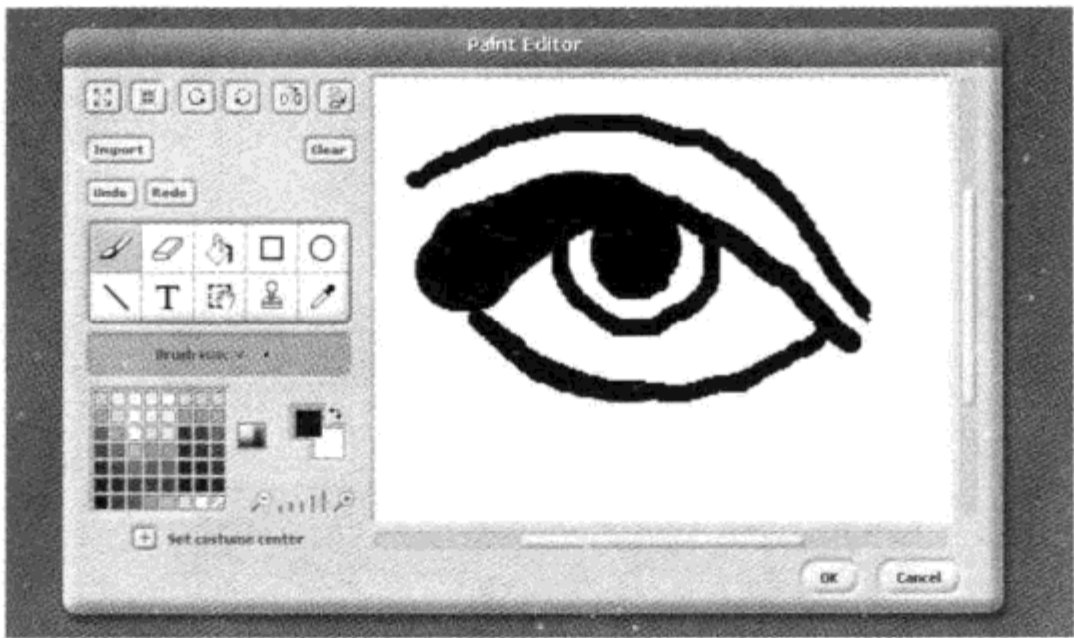


图5.10

选择“造型”（Costumes）页（这里会显示出你刚才绘制的图片），把这个造型的名字改为 open。点击“复制”（Copy）按钮创建一个新的造型，点击“编辑”（Edit），擦除眼珠并重新画上一个闭上的眼皮，然后点击“确认”（OK）并把这个新的造型名字改为 shut，如图 5.11 所示。

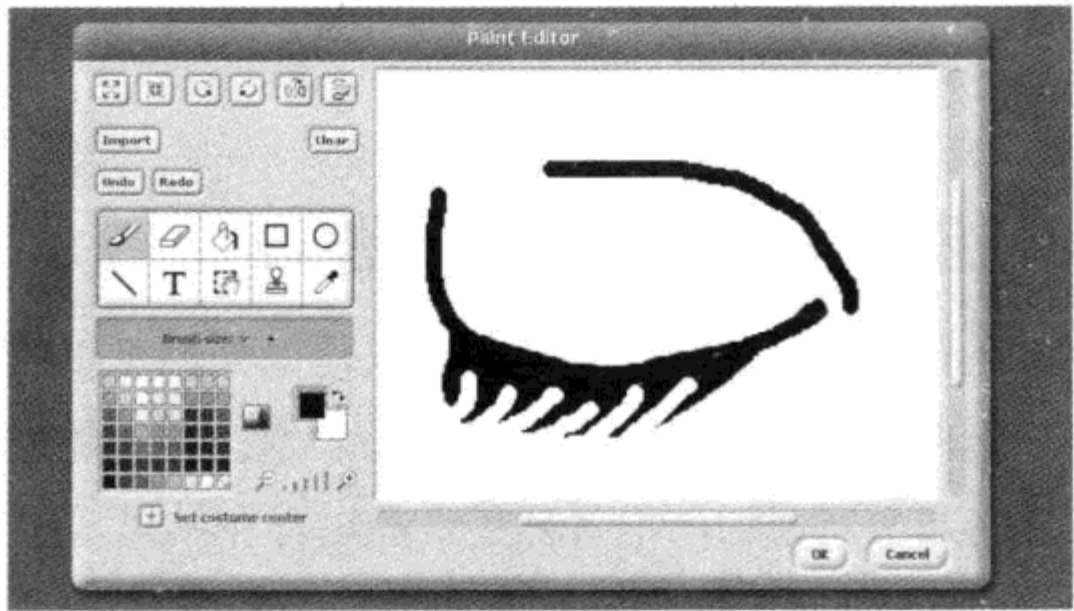


图5.11

然后，选择“脚本”（Script）页面，并把积木拖到脚本区组成图 5.12 所示的脚本，这个脚本会控制眼睛每秒钟眨一次。

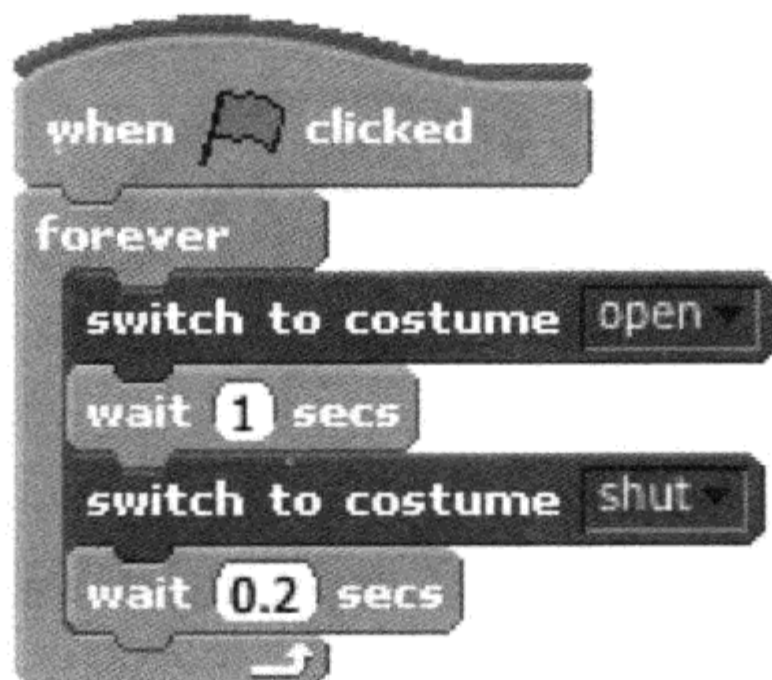


图5.12

最后，你可以用“声音”（Sound）标签为角色添加声音效果。点击“录音”按钮就可以打开 Scratch 内置的录音机（Sound Recorder）程序。你需要在 Raspberry Pi 使用外置 USB 声卡和麦克风才可以录音。为这个声音起一个名字，然后就可以通过“播放声音”（play sound）或“播放声音直到播放完毕”（play sound until done）积木来播放这段声音。

更复杂的例子：星际入侵者游戏

在星际入侵者游戏中，玩家需要对入侵的外星飞船射击，如果不能成功地阻挡外星飞船撞击到玩家的加农炮，游戏就结束了。我们会在这个游戏中展现如何把各种元素都组合在一起。

先点击“文件（File）→新建（New）”，然后点击“文件（File）→另存为（Save As）”保存文件。下面可以开始创建角色。先按住 Control 键点击或右键点击角色列表中的小猫角色，从菜单选择“删除”（Delete）把它删掉，然后创建 5 个角色，你可以自己画这些角色的图片，也可以加载现成的图片文件。我画了如图 5.13 的 5 个



角色，并把它们扫描成 PNG 图片。

创建完角色后，它们会被添加到舞台上。如果它们的大小不是很合适，可以在舞台顶部的工具栏中选择“放大角色”（Grow Sprit）或“缩小角色”（Shrink Sprite）来调整它们的大小。你还需要对这些角色进行重命名，可以按图 5.13 中的名字来给它们命名。

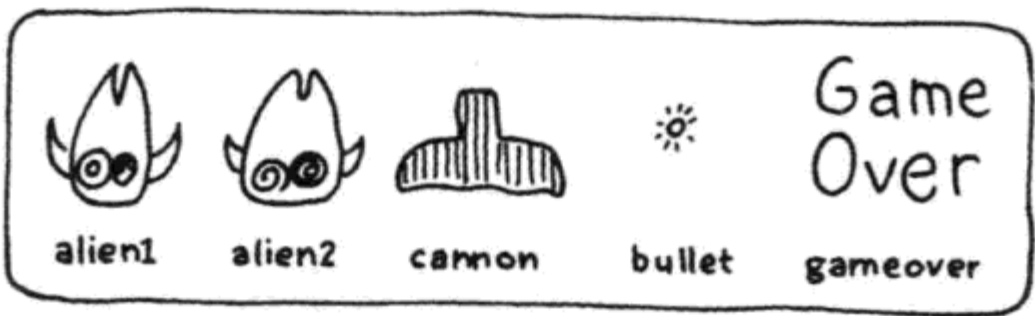


图5.13 星际入侵者中的5个角色

外星飞船和加农炮都有两种状态：正常状态和爆炸状态。因此，你需要为这几个角色添加额外的造型，如图 5.14 所示。

需要为每个角色分别编写脚本，用来控制它们在游戏中的行为。为了清晰起见，每一个动作都需要定义一个独立的脚本，这些脚本会被同时运行。

附录 B 中列出了为每个角色定义的完整脚本。

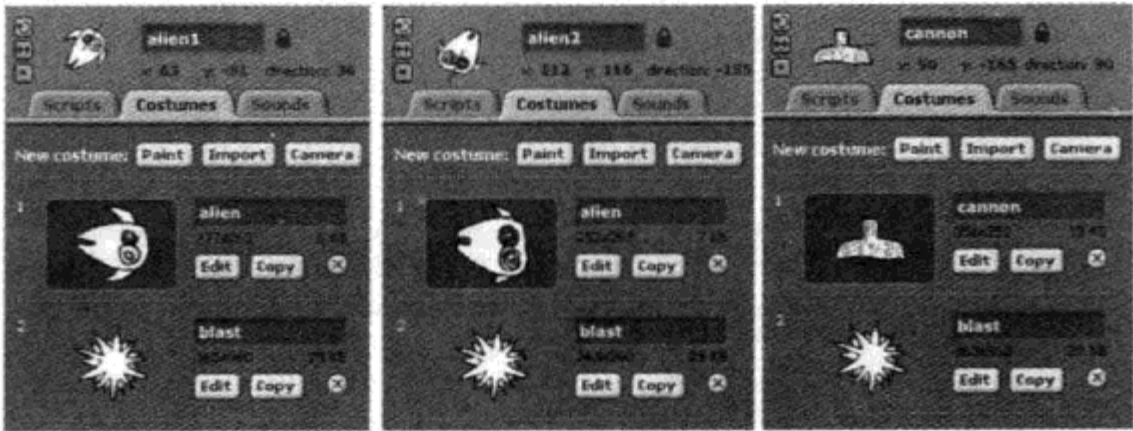


图5.14 为每个角色创建两个造型：一个用于正常状态，一个用于爆炸状态。按本图中的名字给这些造型命名



我们从第一个外星飞船开始，选中这个角色，然后把图 5.15 所示的积木拖入脚本区。



图5.15

这段脚本把相关角色移动到它的起始位置，确保它是可见的，并且处于没有爆炸的模式。

然后，拖动图 5.16 所示的积木并在脚本区中把它们组合好。

点击绿色小旗后你会发现，现在角色的行为与我们在本章开始时所创建的“盒子中的猫”的程序相当一致。由于现在舞台上还有 3 个角色共存，所以你需要处理角色之间发生碰撞时的行为。当两艘外星飞船即将发生碰撞时，可以让它们各自旋转 90° （图 5.17），就可以避免两个角色重叠在一起了。

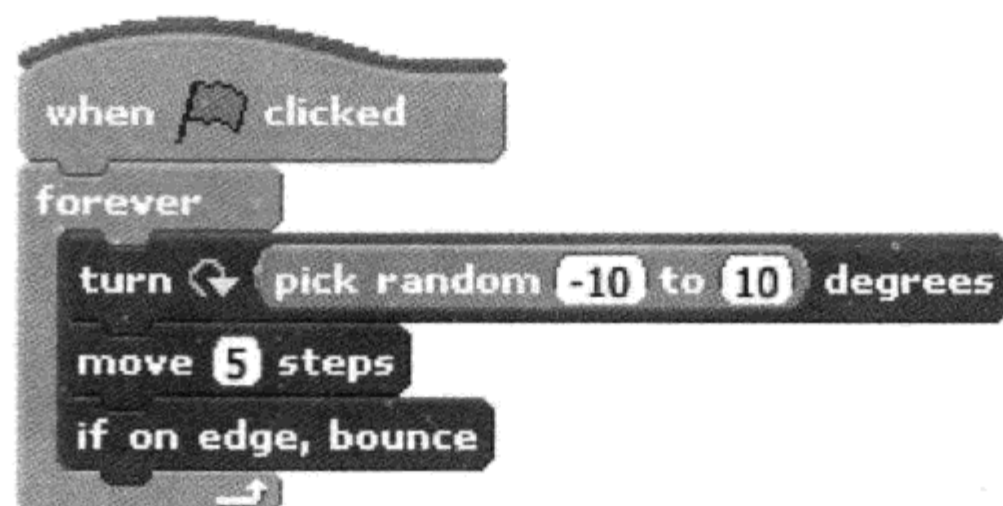


图5.16

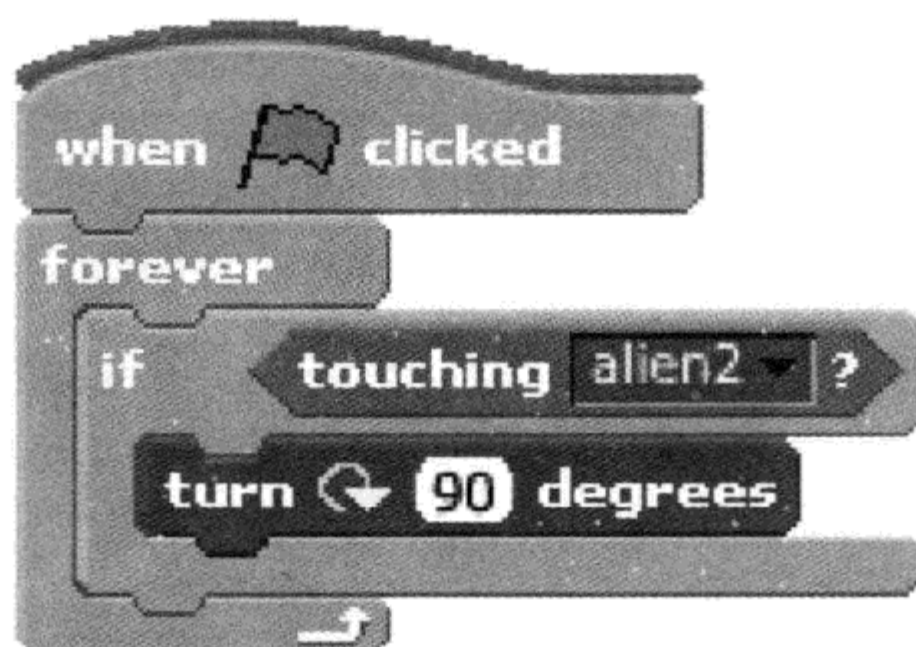


图5.17

如果子弹与外星飞船发生碰撞，我们需要让飞船发生爆炸，如图 5.18 所示。

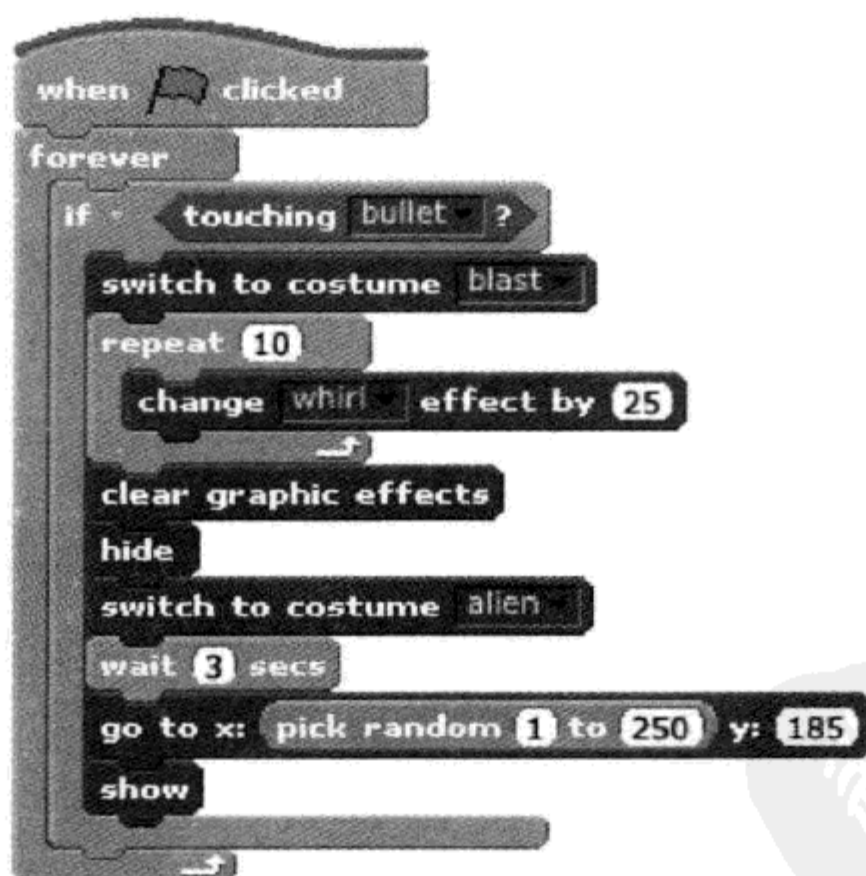


图5.18

爆炸通过把造型改变成 explosion 来实现，并且添加一小段时间的特效，然后把这个角色隐藏起来，这样它就从舞台上消失了。



几秒钟之后，让外星飞船重新出现在舞台上部的一个随机位置上。

这就是我们为外星飞船设定的全部脚本，如果你有多个角色需要使用类似的脚本，可以通过在积木上点击鼠标右键并选择“复制”（Duplicate）来复制脚本。重复这个操作，并把复制出来的每一段脚本拖到相应的外星飞船上。然后选择第二艘外星飞船，并在脚本中修改它的起始位置和碰撞处理行为，如图 5.19 所示。

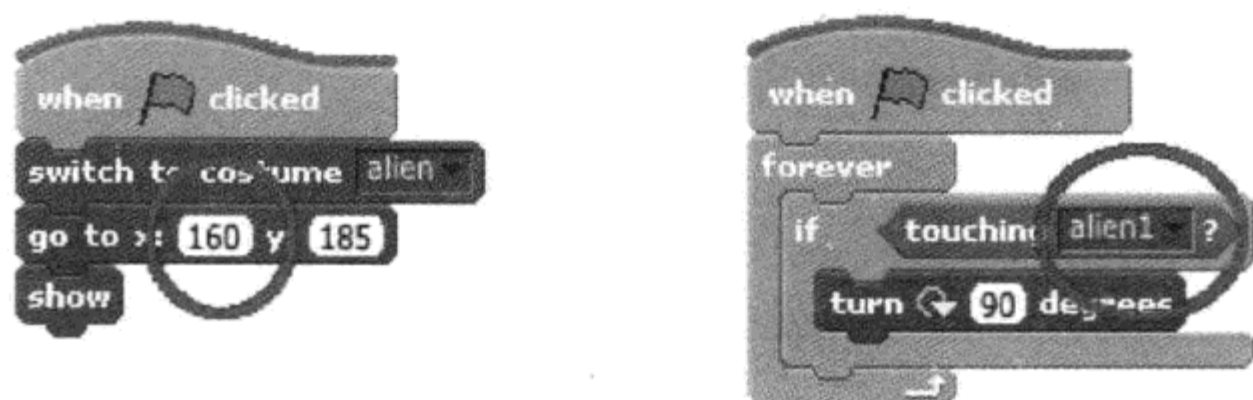


图5.19

接下来，选择加农炮角色。首先把它移动到一个合适的位置，并确保它处于非爆炸的状态，如图 5.20 所示。



图5.20

加农炮只能用键盘上的方向键来控制左右移动，图 5.21 是处理键盘按键的脚本。



图5.21

使用“侦测”（Sensor）积木盒中的“碰到”（touching）积木，可以判断外星飞船是否撞到了加农炮。如果撞到了，就把加农炮设置为爆炸状态，具体的做法与让飞船爆炸一致。加农炮爆炸后，会向所有的角色发送一个游戏结束的消息。这个消息会被 GameOver 角色捕获，并停止脚本的运行，如图 5.22 所示。

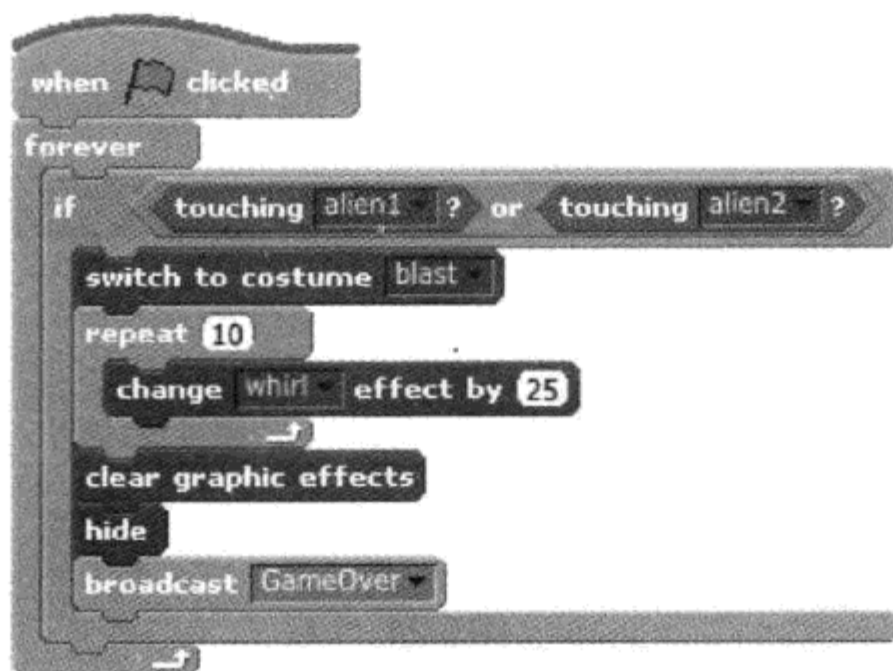


图5.22



下面来看子弹角色，为子弹角色添加图 5.23 所示的脚本。在按下空格键之前，子弹是不可见的。

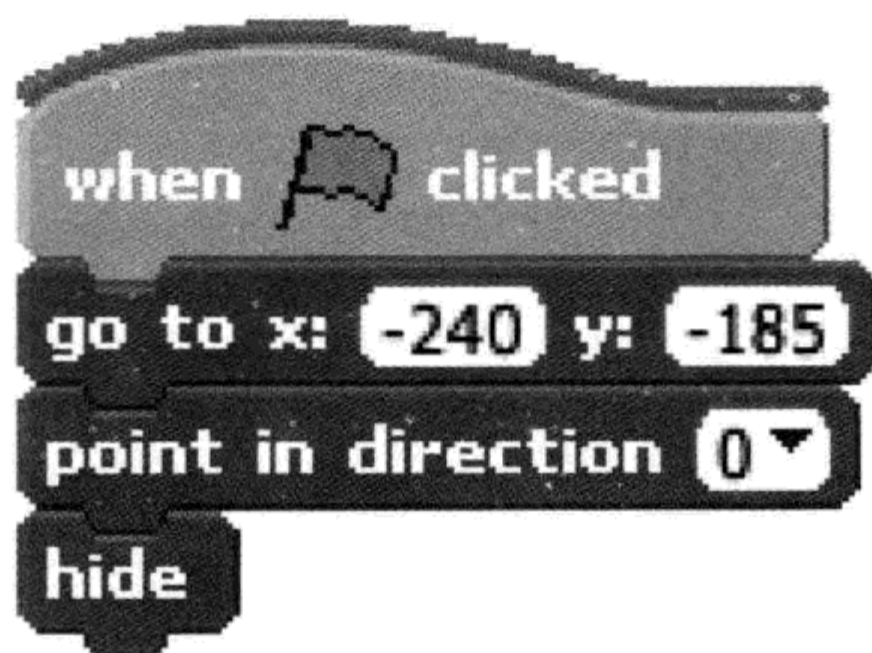


图5.23

每次按下空格键时，子弹会移动到加农炮的当前位置，变为可见并开始向舞台顶端移动，直到遇到舞台边界为止。

图 5.24 就是处理空格键按下消息的脚本。

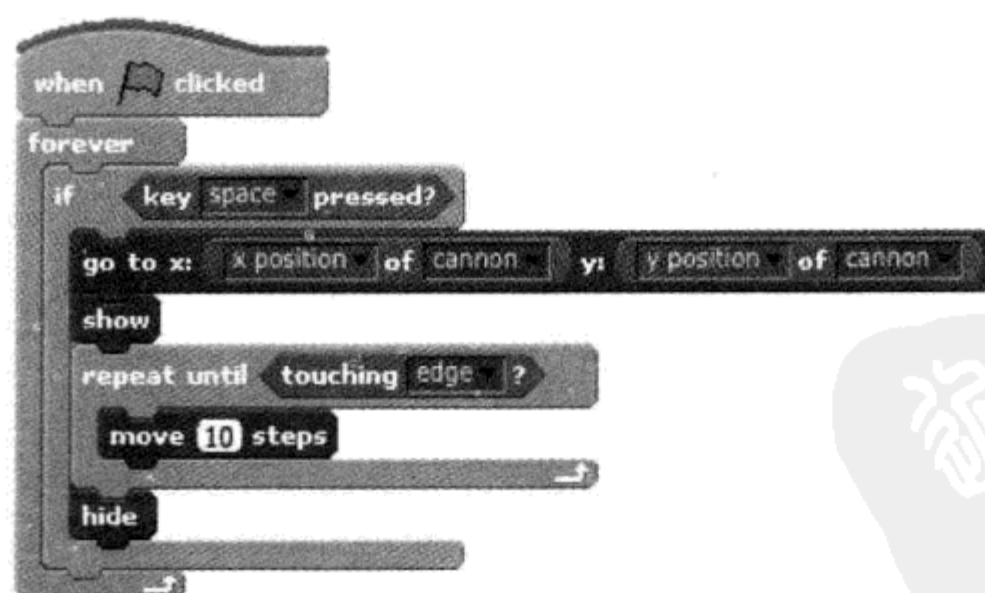


图5.24

最后一个角色是游戏结束角色。这个角色行为很简单：在接



收到 GameOver 消息以前，保持不可见的状态；接收到 GameOver 消息后，把它显示出来，并停止运行脚本，如图 5.25 所示。

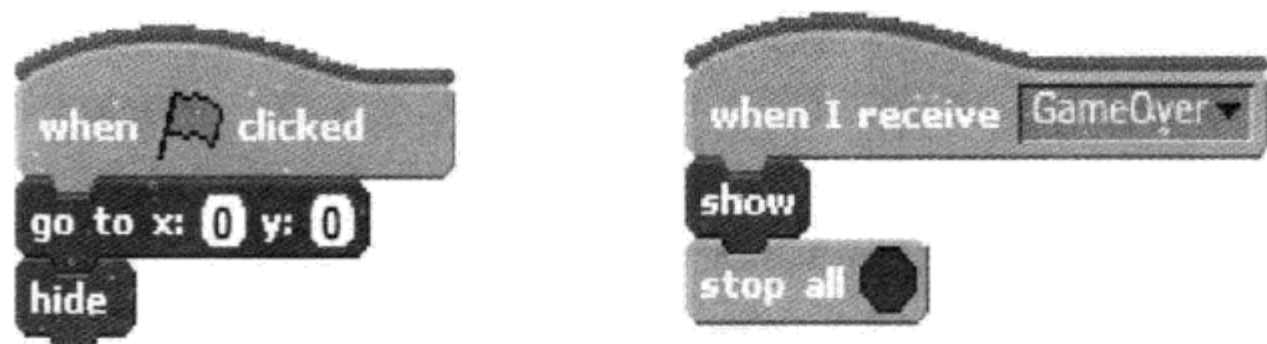


图5.25

这就是整个游戏的实现过程。请参考附录 B 中完整的游戏设计。

Scratch 与现实世界

如果你注意观察“侦测”（Sensor）积木盒，你会发现两块有意思的积木：“传感器”（sensor）和“传感器的值”（sensor value）。这些积木是用于与一块名为 PicoBoard（图 5.26）的外接传感器板配合使用的。PicoBoard 上有一块微处理器芯片，可以把传感器获取到的值通过 USB 连接返回给 Scratch。

PicoBoard 上有一个按钮、一个滑杆、一个声音传感器和 4 个可以连接其他任何模拟传感器的接口。可以使用鳄鱼夹来连接各种传感器，读取模拟信号值。

PicoBoard 与 Scratch 之间通过一种专有的 PicoBoard 协议进行数据通信。S4A（Scratch for Arduino）项目（<http://seaside.citilab.eu/scratch/arduino>）为 Arduino 实现了相同的协议，所以你也可以用 Arduino 把 Scratch 与现实世界相接。S4A 需要一个专用版本的 Scratch 才能支持，这个特殊版本的 Scratch 在 S4A 的主页上可以下载。

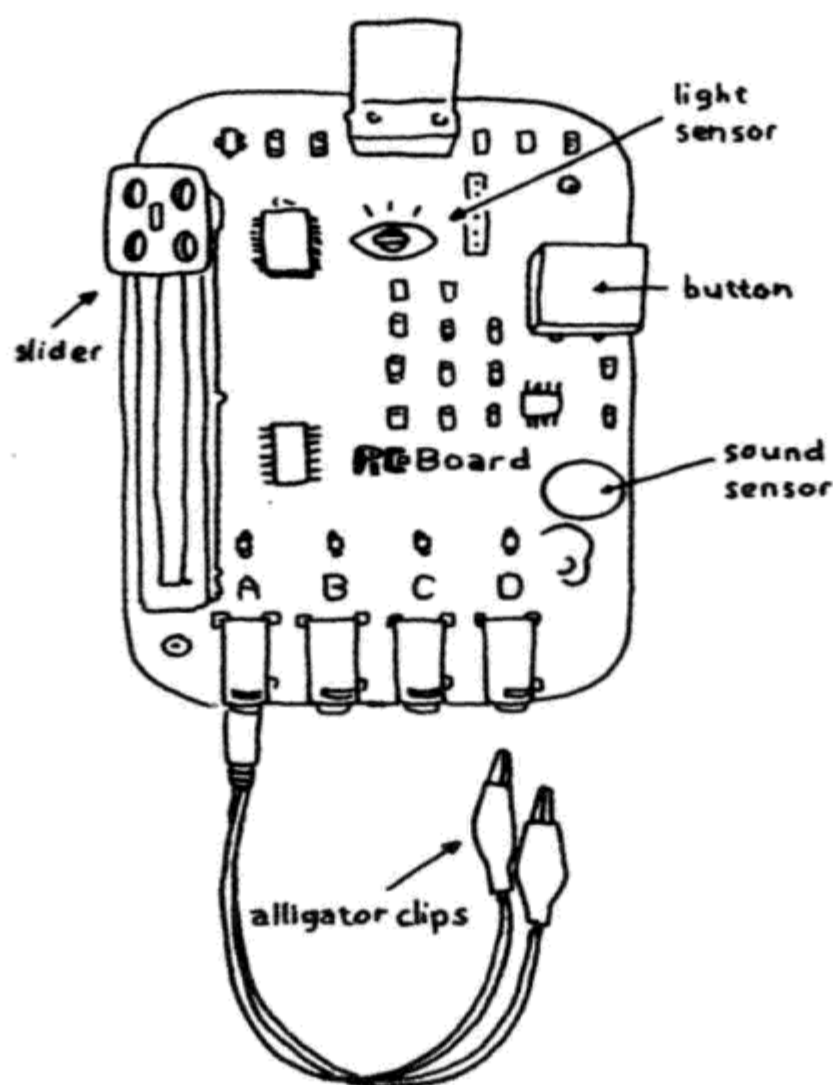


图5.26 PicoBoard是一块专门用于Scratch的传感器附件

分享你的程序

Scratch 有一个很有意思的特性就是内置了一个社区功能。除了先前提到的“来个令人惊喜的角色吧”（Get Surprise Sprite）工具（用于随机从其他 Scratch 用户那里获取角色），它还提供了一个分享的功能，可以把你自己的 Scratch 项目打包并上传到 MIT 的 Scratch 项目页面（<http://scratch.mit.edu/>）。最新的数据显示，现在有超过 120 万的 Scratch 用户，分享了超过 280 万个 Scratch 项目。

之所以会有这么多项目被分享出来，是因为分享一个项目非常简单。在 scratch.mit.edu（<http://scratch.mit.edu/>）注册账号后，只需点击“分享”（Share）菜单中的“将此作品在网络上分享”（Share This Project



Online），输入一些必要的信息后（图 5.27）你的项目就被上传到网站上了。每个上传到网站上的项目不能超过 10M，所以你可能需要把项目中的图片和声音进行压缩，参见“编辑”（Edit）菜单中的选项。通过 Scratch 项目主页，你可以了解到这个平台的各种玩法。

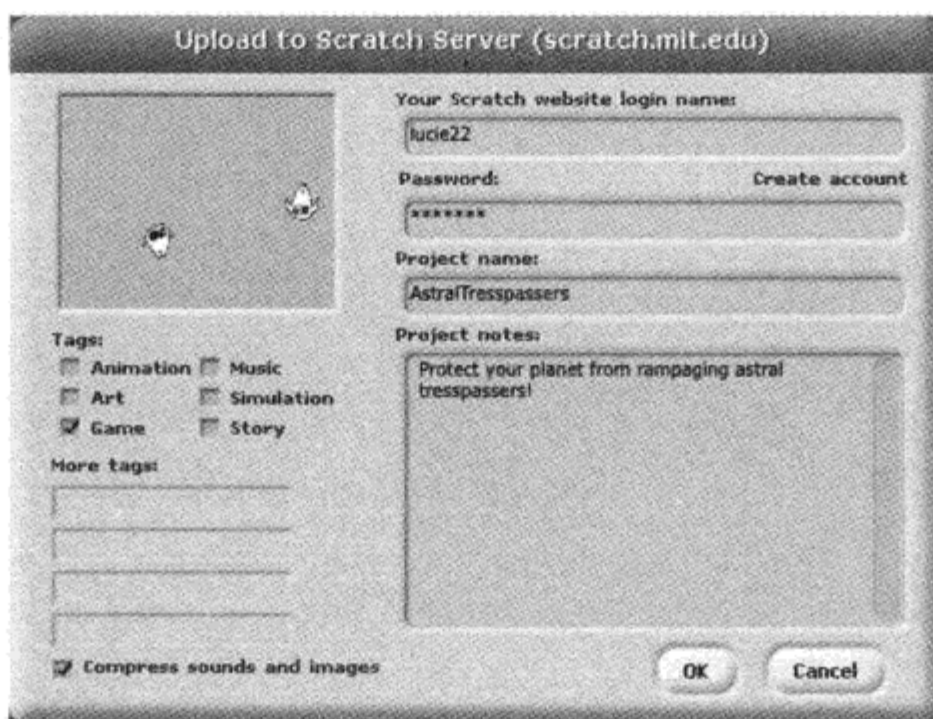


图5.27 当你完成了一个作品，可以用Scratch内置的功能把它分享到MIT Scratch项目的页面上。不过遗憾的是这个网页上使用了Flash动画，所以在Raspberry Pi上无法直接访问

进一步学习

Scratch 的 Wiki 页面

(http://wiki.scratch.mit.edu/wiki/Scratch_Wiki)

MIT 官方的 Scratch 参考资料。

MIT 的 Scratch 页面

(<http://scratch.mit.edu/>)

Scratch 的社区网站，上面有成千上万的项目和注册会员。