

第31章 Shockwave短程序

几乎任何Director影片都可以做成 Shockwave短程序(applet)。前面四章和下一章里的大多数例子都可以放在 Web或局域网上，供用户用浏览器观看。

但是，有几类程序只在 Web上才有作用。浏览页和浏览条等浏览工具是为用户浏览 Web网站而专门制作的。此外还有 Web广告，用 Director可以把它们做得非常生动。还有一些程序要从 Web上的其他信息源提取信息。

31.1 创建浏览页

几乎每个Web网站都有主页。尽管这可以是一个简单的 HTML页面，但我们也可以用更复杂的东西，如 Shockwave短程序。

Macromedia的网站就用 Shockwave Flash 创建了一个交互式的主页。Director的 Shockwave 更擅长于制作交互式的页面。

尽管我们能够用本书的前面的章节所介绍和各种技巧制作交互式的影片，但那些影片与 Shockwave浏览页的最大区别在于后者能进行 Web浏览。

Web浏览是用 gotoNetPage实现的。该命令把当前 HTML页面替换成另一个页面。它使我们能够把 Shockwave按钮变成像标准的 HTML A HREF标签一样的链接。

图31-1的 Director舞台上有一个典型的 Shockwave浏览页。它包含6个按钮以及其他一些图形。顶部的三个色条从左向右运动。



图31-1 一个简单的Director影片，它可以被用作Web网站的首页

每个按钮所附带的行为都非常简单。它只有一个参数：该按钮所使用的网址。它甚至不像第14章“创建行为”里所讨论的复杂的按钮行为那样使用“按下”状态。

property pURL

on getPropertyDescriptionList me

```
list = []  
addProp list, #pURL, [#comment: "URL", #format: #string, #default: ""]  
return list  
end  
  
on mouseUp me  
gotoNetPage(pUrl)  
end
```

这个行为的目的是要强调创建一个 Web浏览按钮都需要哪些功能。这些功能其实就是指定一个URL，然后用gotoNetPage命令跳到那个网址。

我建议你使用一些复杂的行为，如第 14章介绍的按钮行为。可以照搬照抄那个行为，只不过在Lingo to Execute的参数那里放置一个gotoNetPage("info.html") 命令就可以了。

当影片制作完毕后，可以把它嵌入一个 HTML网页，并放在你的网站上。有关的具体方法请参见第36章“送货”里的36.2节“制作Shockwave影片”。图31-2是在Netscape Navigator里看到的嵌入的影片。

图31-1和图31-2里的例子只是一鳞半爪。由于 Director有那么丰富的功能，如果只显示几个按钮，做一点简易的动画，根本就不值得一提。我们可以做些更有趣的东西，如第 15章“图形界面元素”里的掠过行为。或者，你也许愿意添加一些声音和带行为的角色，并使它们对鼠标的不同位置有不同的反应。在 Director里实现这些都很容易，但用HTML制作就比较困难了。



图31-2 Shockwave浏览影片已被嵌入一个普通的白色Web网页

参见第15章里的15.1节“创建显示掠过”。

参见第22章“使用Shockwave和因特网”。

参见第36章里的36.2节“制作Shockwave影片”。

31.2 创建浏览条

很多网页使用分块结构。分块结构即把浏览器的窗口分割成几个部分，每一部分包含一

个独立的HTML页面。

分块结构的使用方法之一是设立一个中心区域，其中显示各种信息；再设立一个小窗口，其中含有一些按钮，可以用来控制大窗口的内容。尽管我们可以在小窗口内放置一个普通的HTML，但也可以为它制作Shockwave影片。

图31-3就是这样一个影片。这是一个很窄很小的画面，出现在主窗口的左侧。

这个小程序里的按钮只比前面那个程序里的按钮稍微复杂一点。这次，它们需要指定gotoNetPage命令所影响的区域。这是通过第二个参数实现的。

```
property pURL, pTarget
```

```
on getPropertyDescriptionList me
  list = []
  addProp list, #pURL,
    [#comment: "URL", #format: #string, #default: " "]
  addProp list, #pTarget,
    [#comment: "Target", #format: #string, #default: "main"]
  return list
end
```

```
on mouseUp
  gotoNetPage(pURL, pTarget)
end
```

对于每一个按钮，pTarget属性都应被设置为目标区域的名称。这个名称是由父代页面的HTML编码确定的。下面是一个例子：

```
<HTML><HEAD>
<TITLE>Navigation Bar Demo</TITLE>
</HEAD>

<FRAMESET Cols="100,*">
<FRAME Name="bar" SRC="navbar.html" Scrolling="Auto">
<FRAME Name="main" SRC="info.html" Scrolling="Auto">
</FRAMESET>

</HTML>
```

Shockwave小程序位于navbar.html页面里。另一个区域叫做main，其首页是info.html，即当用户点击小程序里的Info按钮时出现的页面。

图31-4是Netscape Navigator里显示的分块结构。info.html页面显示出来了。若不是因为这只是个简单的例子，其中应有丰富的内容。

同前面介绍的浏览页小程序一样，这个浏览条只是一个例子。要使它成为一个名副其实的Shockwave而不仅仅是一个带链接的HTML页面，应该向其中添加更多动画或交互性的内容。



图31-3 一个很窄的浏览条小程序，用来控制浏览器的另一区域中的HTML页面

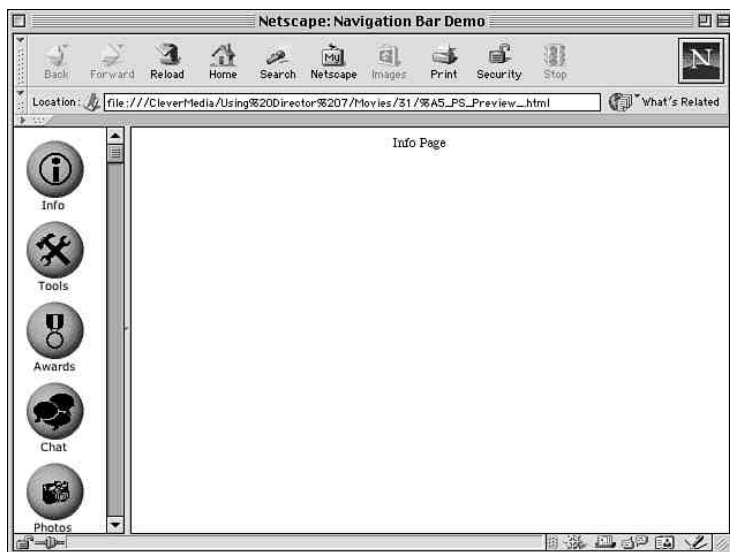


图31-4 浏览器窗口被分为两块。左边的页面里包含一个Shockwave短程序，它控制右边的页面

参见第22章里的22.1节“用Lingo控制Web浏览器”。

参见第36章里的36.2节“制作Shockwave影片”。

31.3 创建广告

现在，广告在Web上随处可见。起初，它们只是一些静止的图像，但现在，Web广告通常是动画GIF。

但有时，广告是Java短程序、动态HTML或Shockwave。尽管用Shockwave Flash影片作为广告比Director Shockwave更常见，但后者仍旧时常被用作广告影片。这是可行的，因为JavaScript可以用来判断用户是否有Shockwave，如果没有，则显示简单的广告，如动画GIF等。

在Director里制作广告是一项相对简单的任务，但也不一定。由于Web广告大都只是动画，于是我们就倾向于保持现状，做一些简单的动画。但实际上我们能做的还有很多很多。

这其中之一是用Lingo和行为制作动画GIF或Flash所无法制作的东西，如第18章“控制位图”里的3D立方体剧本。这个剧本通过操纵6个方形位图的quad属性而生成了3D立方体。这个立方体根据鼠标的位置旋转和改变角度。

我们可以把同样的剧本放到一个广告短程序里。由于它对鼠标的位置有所反应，因此用简单的动画GIF或Flash是无法实现的。可以摆出的各种角度也是用其他方法难以实现的，因为我们的立方体几乎可以以无穷多种方式出现。

图31-5就是这样一个Web短程序。它的尺寸是486×60像素，即Web广告的常用尺寸。

我们对这个3D立方体所做的唯一修改是把它半径由60像素改为20像素，并把它居中放置于一个新的位置。影片本身有几帧，而不是只有1帧，这样，三个色条可以从左向右移动，而不需要额外的Lingo。

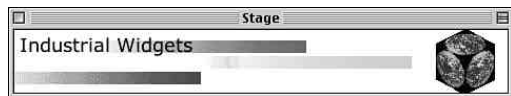


图31-5 一个Shockwave短程序可以被用作一则广告。
这个广告的右侧有一个交互式的D立方体

参见第18章里的18.1.5节“quad属性”。

31.4 处理和显示信息

一个经常容易被忽略的 Director 的功能是 getNetText 系列命令的威力。用这些命令可以从因特网上得到任一 Web 网页，并处理它的原始 HTML。用同样的方法也可以得到一个纯文本文件。

由于因特网上有丰富的信息，因此这些命令的用途广泛。例如，我们可以了解股市行情，并把它显示在股市收报机界面里。我们还可处理 HTML 页面的文本，删除其中的标记，并查到我们所需要的信息。

搜索信息的一个很好的例子是能够从 National Weather Service(NWS)读取信息的 Shockwave 影片。NWS 网站提供简单的天气情况和天气预报信息，网址是 <http://iwin.nws.noaa.gov>。仔细查看这个网站，我们会发现天气预报是按州分的，因此科罗拉多州的预报位于 <http://iwin.nws.noaa.gov/iwin/co/state.html>，加利福尼亚州的预报位于 <http://iwin.nws.noaa.gov/iwin/ca/state.html>。“co”和“ca”是州名的两个字母的缩写。图 31-6 是一页天气预报。

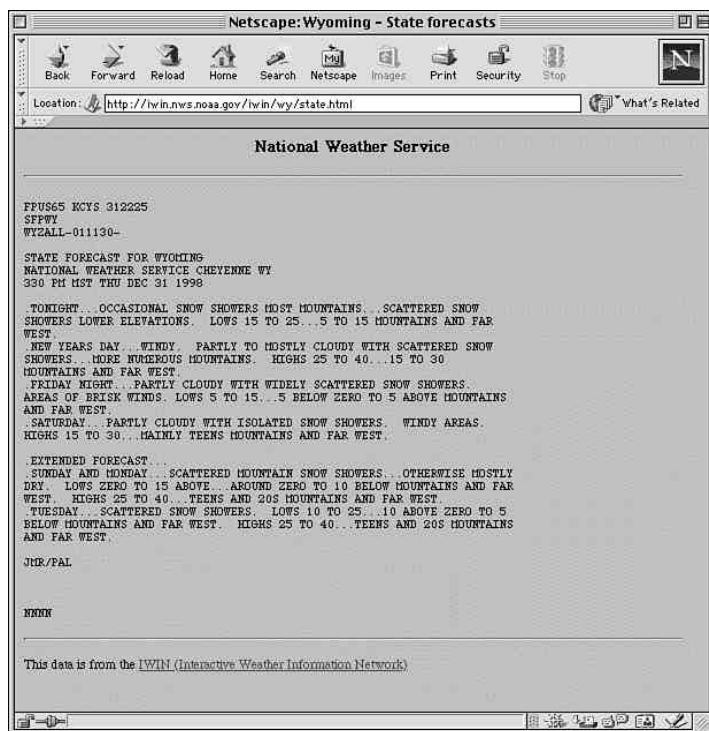


图31-6 浏览器的窗口里显示了National Weather Service的天气预报



图31-7 这个短程序搜索天气预报的文本，并以股市收报机的形式呈现出来

图31-7是一个短程序，它以股市收报机的形式显示天气预报。那些按钮使用户能够指定该短程序应该搜索某个特定的预报。

以下是图31-7里的按钮所使用的行为。每个按钮有两种状态，即按下状态和抬起状态。按下状态将一直保持，直至另一个按钮被按下。因此，这一组按钮的特性就像是单选按钮。

```
property pStateCode, pUpMember, pDownMember, pNetID

on getPropertyDescriptionList me
    list = [:]
    addProp list, #pStateCode, [#comment: "State Code", #format: #string,
#default: "XX"]
    return list
end

on beginSprite me
    pUpMember = sprite(me.spriteNum).member
    pDownMember = member(pUpMember.number + 1)
    gNetID = 0
end

on mouseDown me
    sendAllSprites(#reset)
    sprite(me.spriteNum).member = pDownMember
    pNetID = getNetText
    ("http://iwin.nws.noaa.gov/iwin/"&pStateCode&"/state.html")
    sendSprite(sprite 12, #newData, "Loading a new forecast... ")
end

on exitFrame me
    if pNetID > 0 then
        if netDone(pNetID) then
            text = netTextResult(pNetID)
            text = processReport(text)
            pNetID = 0
            sendSprite(sprite 12, #newData, text)
        end if
    end if
end

on reset me
    sprite(me.spriteNum).member = pUpMember
    pNetID = 0
end
```

在这个按钮行为里，当按钮被按下时，它用 `getNetText` 命令开始一次网络操作。它查看每一帧，看是否收到了信息。如果已经收到了文件，按钮行为则把它送给角色 12——即收报机——所附带的行为。在网络操作开始后，该按钮行为还立即向那个角色发一个消息。

制作收报机的方法有很多种。可以让角色从右向左移动，也可以直接替换每帧里的文本演员。由于这些天气预告的内容都比较长，每帧都重新放置文本，并给文本演员足够的字符，使它们正好充满屏幕里的一行，可能是最好的方法。

```
property pText, pChar, pSpaces
```

```
on beginSprite me
```

```

pSpaces = " "
repeat with i = 1 to 70
  put SPACE after pSpaces
end repeat

pText = pSpaces&"Click on a state button below to select a forecast. "
pChar = 1
end

on newData me, text
  pText = pSpaces&text
  pChar = 1
end

on exitFrame me
  text = pText.char[pChar..pChar+70]
  sprite(me.spriteNum).member.text = text
  pChar = pChar + 1
  if pChar > pText.length then pChar = 1
end

```

这个收报机行为为每块文本分配 70 个字符的位置。它迫使文本开始为空，然后第 1 个字符从右边移动进来。基本上每一帧显示一个不同的子字符串，它显示第 1 至第 70 个字符、第 2 至第 71 个字符、第 3 至第 72 个字符，等等。

在这个行为开始时，它在收报机里放置一些默认的文本。当信息显示完后，它又重新启动收报机。为了解决这个问题，可以让它开始先显示某个默认州的预报，或什么都不显示。也可以让它在显示完信息后，显示一些默认信息，提醒用户选择某个州。

请注意，收报机调用了名为 on processReport 的处理程序，它的作用是先修改 HTML 文本，再把它送往收报机。这个处理程序是必要的，因为 Web 网页的文本里肯定有一些标记和回车符，要在收报机里正确地显示信息，必须删除这些标签和回车符。它们实际上是被替换成了空格，这样可以避免把不同信息粘在一起。

```

on processReport text
  text = removeTags(text)
  text = removeReturns(text)
  return text
end

on removeTags text
  inTag = FALSE
  newText = " "
  repeat with i = 1 to text.length
    if text.char[i] = "<" then
      inTag = TRUE
      put " " after newText
    else if text.char[i] = ">" then
      inTag = FALSE
    else if inTag = FALSE then
      put text.char[i] after newText
    end if
  end repeat
  return newText
end

```



```
on removeReturns text
repeat while TRUE
  c = offset(RETURN, text)
  if c < 1 then exit repeat
  put " " into text.char[c]
end repeat
return text
end
```

on processReport处理程序的实际功能取决于所要读取的 HTML页面的具体内容。如果下一页的信息是以表格的形式出现的，我们必须写一个复杂的处理程序来提取其中的信息。

这个程序只是getNetText系列命令的功能的一个例子。不一定非要用收报机显示信息不可。可以直接显示页面信息，可以用图形显示信息，如股价的曲线图。我们甚至还可以让getNetText从多个页面里读取信息，最后编出一个报告。

参见第22章里的22.2节“从因特网上获得文本”。

参见第36章里的36.2节“制作Shockwave影片”。

31.5 Shockwave短程序的故障排除

如果你使用的是Director 6和Director 6 Shockwave，带有指定的目标帧的gotoNetPage命令在Microsoft Internet Explorer上的功能不正常。Explorer有时会创建一个新窗口，而不是沿目标窗口或目标帧。当 gotoNetPage在on exitFrame处理程序中被调用，而不是在 on mouseUp等处理程序中被调用时，就会发生这种现象。现在还不能肯定 Director 7是否已改正了这个程序错误。

要快速地测试Shockwave影片，可以选择File | Preview in Browser。这将在你的计算机里的缺省浏览器上测试影片。还需要在另外三种浏览器里进行测试，即需要在 Windows和Mac的 Netscape和Internet Explorer上测试。还应在不同版本的浏览器上进行设置。

Mac版的Microsoft Internet Explorer 4.0有一些问题，导致Shockwave影片不能正常进行。创作者们应考虑仅支持4.01以及上的Mac版本。

31.6 你知道吗

不一定非要把Shockwave影片嵌入HTML页面不可。可以像把浏览器指向 GIF或JPEG那样，把浏览器指向Shockwave影片。影片将从窗口的左上角开始填充窗口，但不会被缩放。

尽管可以用getNetText命令从服务器里的.html或.txt文件里搜索信息，但也可以用完整的类似CGI的路径名获得信息。可以用postNetText访问仅对post有反应的程序。

用Lingo函数netError可以检测getNetText命令是否发生了错误以及发生了什么错误。

使用getNetText、netDone和netTextResult命令时，也可以不必像本书的例子那样非在变量里存储一个网络ID不可。如果没有存储，程序将假定你将沿用最近一次网络操作的地址。如果每次只有一种网络操作，这种方法是没问题的。