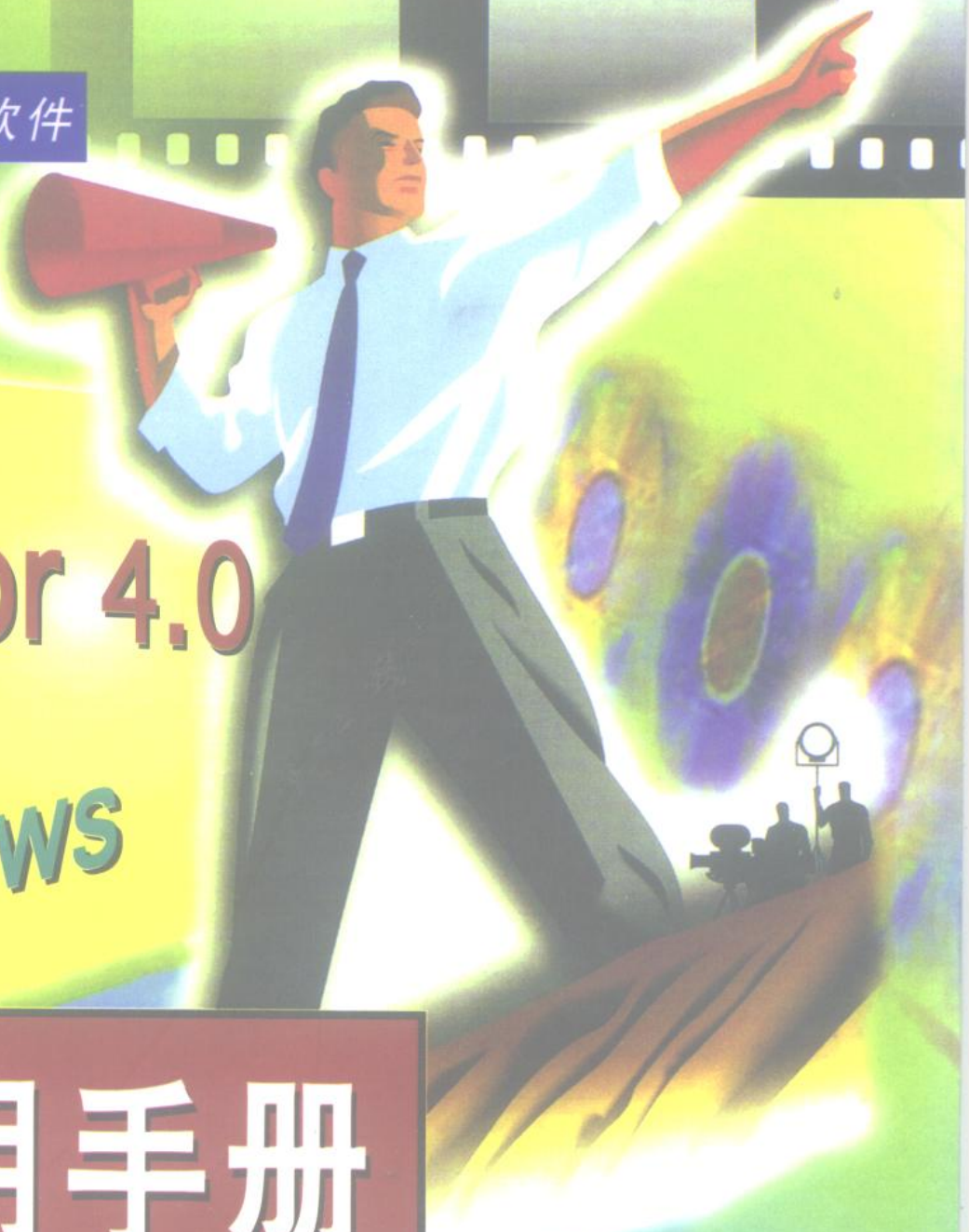
清华松岗系列丛书

多媒体工具软件



Director 4.0 FOR Windows

使用手册

曾安德 编著

清华大学出版社

Director 4.0 for Windows 使用手册
清华大学出版社

Director 4.0 for Windows
使 用 手 册

曾安德 编著

清 华 大 学 出 版 社

（京）新登字158号
北京市版权局著作权合同登记号：01-96-0929号

Director 4.0 for Windows 使用手册

曾安德 编著

本书中文繁体字版（原书名为Director 4.0 for Windows 进阶使用手册）由台湾松岗电脑图书资料股份有限公司出版，1995。本书中文简体字版经台湾松岗电脑图书资料股份有限公司授权由清华大学出版社与北京清华松岗电脑信息有限公司合作出版，1996。任何单位或个人未经出版者书面允许不得用任何手段复制或抄袭本书内容。

本书以 Lingo 语言为核心，主要介绍了 Windows 环境下的 Lingo 多媒体设计。

本书特点在于以解决实际问题为实例，逐步讲述 Lingo 程序设计及具体操作步骤。全书共分为六章。其中第一章为 Lingo 的介绍；第二章到第五章为程序设计部分，内容包括电影播映控制的设计、程序设计、控制部分、FileIO 的控制等；第六章为指令介绍部分。

全书叙述清楚，语法指令解释详细。作为 Director 的进级篇，本书将成为多媒体用户的实用参考手册。

图书在版编目（CIP）数据

Director 4.0 for Windows 使用手册/曾安德编著. —北京：清华大学出版社，1996. 10
ISBN 7-302-02377-8

Ⅰ. D… Ⅱ. 曾… Ⅲ. 多媒体技术-软件工具，Director-手册 Ⅳ. TP311. 56-62

中国版本图书馆 CIP 数据核字（96）第 23213 号

出 版 者： 清华大学出版社（北京 清华大学校内，邮政编码：100084）
责任编辑： 黄四平
印 刷 者： 北京市清华园胶印厂
发 行 者： 新华书店总店北京科技发行所
开 本： 787×1092 1/16 印张：14.25 字数：337 千字
版 次： 1997 年 1 月第 1 版 1997 年 1 月第 1 次印刷
书 号： ISBN 7-302-02377-8/TP・1189
印 数： 0001—5000
定 价： 25.00 元

出 版 说 明

本书原版（中文繁体字版）是由台湾松岗电脑图书资料股份有限公司出版。由于海峡两岸计算机科技术语的译名不一致，因此在出版中文简体字版的时候，对正文中的术语进行了转译。但由于书中的屏幕显示图采用照相制版方式，故其中文字仍为繁体字，且专业术语亦未转译过来。为便于读者阅读查对，现将图中有关术语与文中所用译名对照列出如下：

繁体字版术语	简体字版术语
档案	文件
程式	程序
磁碟	磁盘
磁碟机	磁盘驱动器
档案管理员	文件管理员
字元	字符
位元组	字节

序　言

在撰写这本书的时候，已经完成了 Authorware Professional 的撰写。这两种截然不同的工具，各有它们优秀的地方，很难用简单的描述来说明它们的不同。

我想只有身临其境才有自己的感受。基本上这本书以介绍 Lingo 语言为主。从程序设计的角度来看，Lingo 确实提供了很多好用的指令，对于程序的语法要求又不高，应该很受欢迎才是。

唯一的缺点是，它的原始设计概念并不是在 PC 平台上开发出来的，所以您会发现它与 PC 上可以看到的语言有很大的不同，但这是很容易克服的。

经过这一段长时间的写作，我发现最好的学习方式是利用课程教授的方式，学生与老师一起做做实际的作品会比较有效，所以在本书中，我用了较少的篇辐来写范例，反而用了较多的篇幅来介绍个别的指令。

希望以后有机会我们能够在一个教室中见面，我很愿意将个人的经验传授给各位有兴趣的人。Director 的 Lingo 与基本操作并不是一两个月就可以学好的，所以一起学习是个很重要的过程。

我也会继续以教材写作的方式介绍这个工具，或者以 Title 的方式来介绍类似的技巧，如果您还有兴趣，我很愿意听到您的意见。

其实多媒体的制作是很花时间的，我们所能够提供的是如何让您节省一点时间，所以多媒体的制作工具（Multimedia Authoring Tools）其作用就是帮助用户节省制作时间，Authorware 也是提供相类似的功能。

现在先来看看书中的章节介绍吧！

第一章　Lingo 的介绍	简单介绍什么是 Lingo，以及它的功能
第二章　电影播映控制的设计	利用控制镜头的指令来学习如何制作一个简单的 Lingo 程序
第三章　程序设计的概念	介绍设计 Lingo 所要具备的一些概念，以及它的基本元素说明
第四章　控制部分	分类列出各组对象所对应的 Lingo 指令
第五章　FileIO 的控制	利用 FileIO 介绍 Director 如何与外部沟通
第六章　Lingo 指令	介绍所有 Lingo 中的指令

如果您已经在 Director 上有了经验，我想不必再花时间学习其它的工具，或许可以说了解一下也无妨，但并不建议您转换工具，因为 Director 所提供的功能一定足够您使用。总

之，把一个工具学好就差不多了，至少这是我现在的感觉。

本书以进级为主题，如果您已经有了一些 Director 的经验，相信这本书可以帮助您更上一层楼。我觉得本书还有很多不够完善的地方，希望大家能够补充它的不足。

如果对 Director 还有一些疑问，或是对本书有任何意见，欢迎您与我联络。倘若读者的问题太多，最好能够以传真或是 Email 的方式联系，我会依自己的能力回答这些问题，但请务必将问题描述清楚一些，以便作答。

我的 Email 地址如下：

Email address：a1206@tptsl.seed.net.tw

曾安德

1995 年 9 月 21 日

目　录

第一章　Lingo 的介绍

1-1　Lingo 的应用范围

在基本的 Director 技巧中，学到的是如何利用它的界面来设计一个影片式的多媒体。而 Lingo 的加入，将增加影片的交互（Interactivity）、控制（Navigation）以及链接（Linking）三大方面的功能。

交互的加入主要是让您可以控制影片的跳跃方式，在基本的操作中，学到的是线性方式播映影片，所以无法让用户与影片之间实现交互，而交互的加入则可以使播映的顺序由 Lingo 来控制。

此时所设计的交互就可以赋予影片跳跃的镜头位置，从而使其具有非线性播映的功能。影片设计再复杂一些时，甚至可以实现超链接（Hyperlink）的功能。

交互的设计是多媒体的灵魂，虽然 Director 制作出来的多媒体很可能是用做动态简报而已，但是交互的加入则可以赋予这个作品生命力。对于用户而言，则有意外的收获，而并非只是接收传播的信息。

但是交互与界面的设计有着非常大的关系，所以您在制作脚本时，就必须针对这个功能做比较深入的讨论与设计，以免在制作之中或者之后发现交互的方式增加了用户的困扰。不过这与 Lingo 的功能无关，只是如果您知道交互的方式，就比较容易利用 Lingo 设计出所要的交互了。

Lingo 所提供的第二个主要功能是对象控制部分。针对各种不同的对象，或者是剧本视窗中的分镜做控制，例如您想控制声音的播映时，可以选取 SoundLevel 来控制声音的大小，配合按键或是按钮的设计，将声音的控制交给用户自己设定。

如果您想要设计一种功能，即当光标移动到某个对象上，它就会自动呈现反白。这时就可以用分镜控制的指令，利用卡司成员（Cast member）交换的方式来实现这种功能。

这些控制功能在 Lingo 的程序设计中，应用的部分最多，功能也最强。其实说得简单一点就是利用交互的跳跃，当跳跃时启动各个对象的控制指令，来显示各种不同的效果，配合影片的设计，自然就是一个精彩的多媒体了。

那么是否需要由外部程序如 C 所编写的函数来表现呢？答案是肯定的。特别是在 Windows的环境下，常常需要利用比较低级的程序来控制特别的对象（如 3D 的对象），结果速度会提高。这是 Lingo 可以做到的，但是我们希望由其它的程序来控制，以便提高执行的效率。

当需要由外部程序控制才能达到所要的功能时，就需要沟通（Communication）的功能

了。在 Lingo 中可以借助 XObject 的设计将其它程序所具有的功能，集成到自己的作品之中，如此便可以呈现所要的效果了。

以上三个部分是 Lingo 主要的应用所在，了解了 Lingo 的用途之后，剩下的就是如何操作以及编写了。我们将在第二章之后陆续地介绍如何实际设计 Lingo 的程序。

1-2 什么是程序（Script）

到底什么是程序（Script）呢？程序就是一堆 Lingo 指令的排列组合，以实现跳跃、控制、沟通等三大功能（这三个功能我们在上一节中谈过）。

所以在设计一个程序时，心中所想的应该是这些功能。程序的种类大致可分为三种，各有不同的启动方式与功能，当选取一种程序时，设计出来的就是针对这个类型的程序，自然它的功能也就应用在这个类型之上。

这三个种类稍后即做比较详尽的介绍，本节主要介绍一个简单的程序范例，利用这个范例，引导您进入程序设计之中。

其实 Lingo 程序的设计是非常简单的，只要稍微花一点功夫，就可以了解。但是应用的设计是需要想象力的，也就是说程序的设计与想象力是密切相关的。

现在就来设计一个简单的程序，假设所要设计的是一个循环，在影片执行到某一个镜头时，希望它在这个镜头上停止，当然程序不是停下来不动，而是镜头停止，然后借助交互来指定它跳到特定的位置上播映。

现在我们碰到的困难是循环的设计，在基本的设计中无法达到这个功能，所以必须要通过 Lingo 程序的设计，下面是步骤的说明。

步 骤

1. 将光标移到特定镜头的 Script Channel 上按一下选取它。
   通常这个镜头是要做循环的镜头，因为我们要做的设计是镜头的循环，这是一种 Frame Script，所控制的是镜头的播映。
2. 按一下剧本视窗上方的程序设计按钮，请参考图 1.1。会看到一个 Script Window（程序视窗）出现，这个视窗与文字视窗非常类似，此处所写的就是一小段程序。

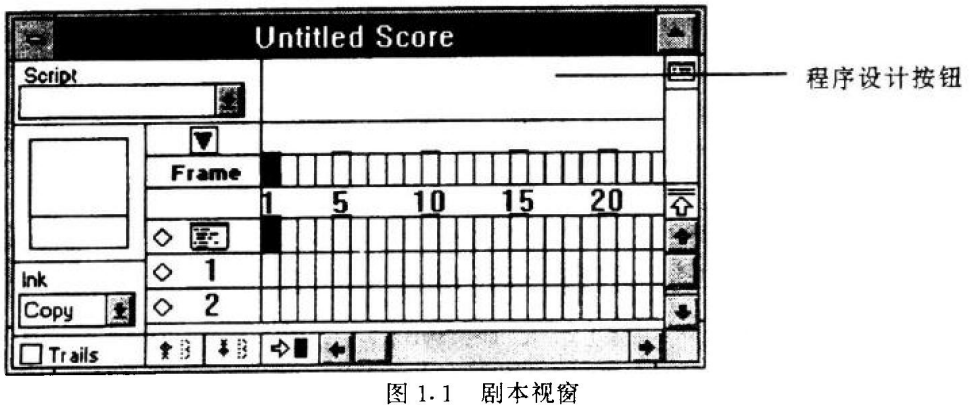


图 1.1 剧本视窗

在视窗的最上方标题行是这个程序的名称，由系统自动赋予。您可以由这个名称发现程序的类型。此处您所看到的应该是 Score Script # 的一个名称，这表示程序的应用主要是控制剧本视窗中的某一个部分，此处所要控制的是一个镜头。请参考图 1.2。

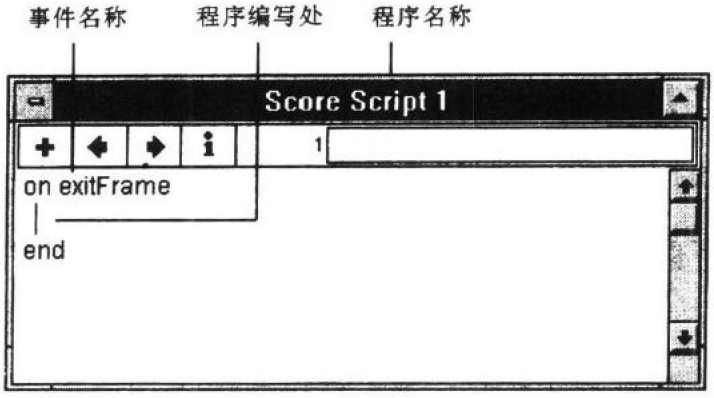


图 1.2　程序视窗

接下来在视窗中会发现，系统已经写好了两行语句，在此处看到的是 on ExitFrame 以及 end。在一般情形下，系统都会自动地写好这两行语句，第一行代表启动事件的条件，最后一行则是程序的结束。中间的部分则由您来完成。而这一个范例的意思则是："当离开这一个镜头的时候，请执行程序。"

所以应该填入的程序是"再回到这一个镜头重播"，这个程序相当于 go to the frame，所以输入的程序应该如图 1.3 所示。

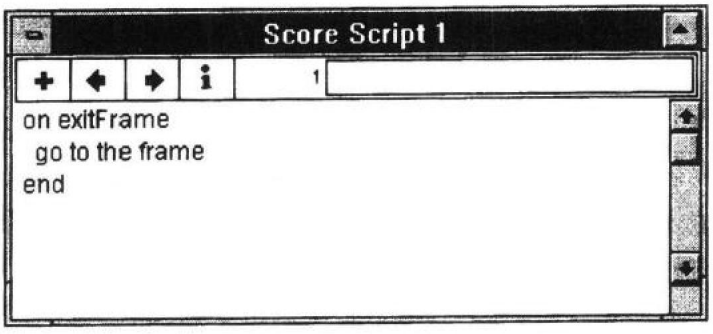


图 1.3　在视窗内输入程序

3. 按一下最右方数字键旁的 Enter 键。

这个键并不是常按的 Enter 键，请特别留意。

以上就是回到本镜头重播的程序，有时候程序不是一行就可以解决，也不是一个事件就可以解决的，但是输入方式在道理上是相同的，只要掌握这一点就可以了。

如果在其中输入程序要隔行时，只要按一下 Enter 键即可，所以 Enter 键在此处并不代

表程序的结束，而是隔行的输入键，当您要关闭程序视窗时，必须要按的是数字盘上的 Enter 键。

除了用这个方式关闭程序的视窗外，也可以利用控制菜单框来关闭视窗，在控制菜单框上按两下即可，如图 1.4 所示。

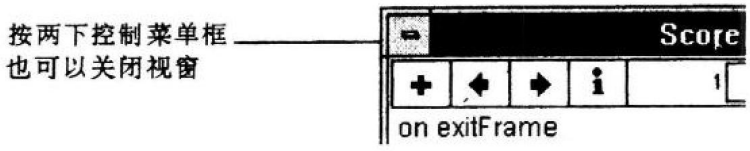


图 1.4　关闭视窗

1-3　Handler 是什么

启动程序称之为“事件（Event）”，事件代表在电影播映的过程中，来自系统或用户的响应。上一节中介绍的 on ExitFrame 事件，就代表一个离开镜头时自动启发的事件（如图 1.5）。

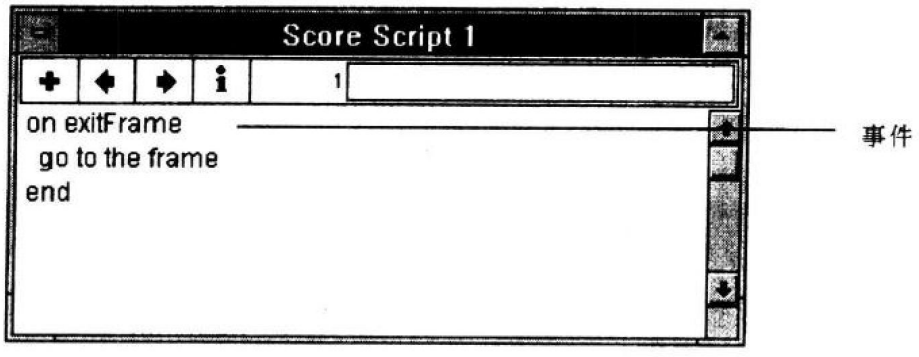


图 1.5　程序引发的事件

当电影播映完这个镜头正要播映下一个镜头时，就会先执行完一段程序，才进入下一个镜头播映，这就是一个“事件（Event）”。

事件发生之后，ExitFrame 被当作一个信息交给系统，系统就会明白这个事件所代表的意思以及启动的时间，接着找到该执行的程序（Script）来执行它。

这时 go to the frame 会告诉电影不要进入下一个镜头，应回到本镜头再播放，所以播映头会回到这个镜头再行播放，从而形成了一个无限的循环，当然所看到的结果就是电影播映到这个镜头的时候，便停在这个镜头上了（图 1.6）。

这样的事件、信息、程序的关系您是否了解了一点？事件的设计就是整个程序设计的精华所在，您也会常常看到其它的事件，而这些事件都是比较常见的。

图 1.6　执行循环播映

如 MouseDown 表示用户按下鼠标键的事件，MouseUp 则表示用户放开鼠标按键的事件，但是这些都是 Lingo 会自动判断的事件。

有时候我们也需要自己设计一个事件，这时就需要由自己来确定信息名称，这个信息名称即为 Handler，如下面的例子：

```
On MouseDown
    Fly
end
```

当看到这样的设计时，表示只要鼠标键被按下时，就会启动一个名为 Fly 的程序，当您寻找所有的程序时，就会发现有一个程序名为 On Fly，请看：

```
On Fly
    go to frame 100
end
```

这个事件的意思就是，当有任何一个地方调用这个事件时，就会直接跳到第 100 个镜头播映。虽然 Fly 不是系统内定的信息，但是经过指定之后，系统会记下来，只要有人调用它，同样会执行它的程序，但是离开这个电影之后，系统就会忘了这个事件。

这表示除了系统内定的事件之外，还有一些其它的事件可以由您自己来设定，就好像是编写一个函数一样，当某一个启动的条件符合它时，就会自动执行。

所有的程序可以同时并存在电影之中，因为一个电影中可能设计有许多不同的事件，到底这些事件有没有类型的区分，以及它们的设计方式如何，这就是下一节所要讨论的内容。

1-4　不同类型的程序

知道了什么是 Handler 之后，这一节要介绍的是各种不同的程序。在 Director 中程序可分为四种：

1. Movie Script
2. Cast Script
3. Frame Script
4. Sprite Script

四种不同的程序各有其功能，开启程序视窗时，都可以看到它们，只是制作的地方不同，所以各有不同的功能。下面我们将介绍如何制作这四种不同的程序以及它们的功能。

Movie Script

Movie Script 是属于电影的程序，也就是电影文件一开启时，它就已经被执行了。那么在这个程序中应该放置哪些内容呢？例如：将所有变量的说明放置在 Movie Script 之中，电影一执行，变量也跟着被说明，因此其它程序引用到变量时，就不会产生混淆的情形了(图 1.7)。

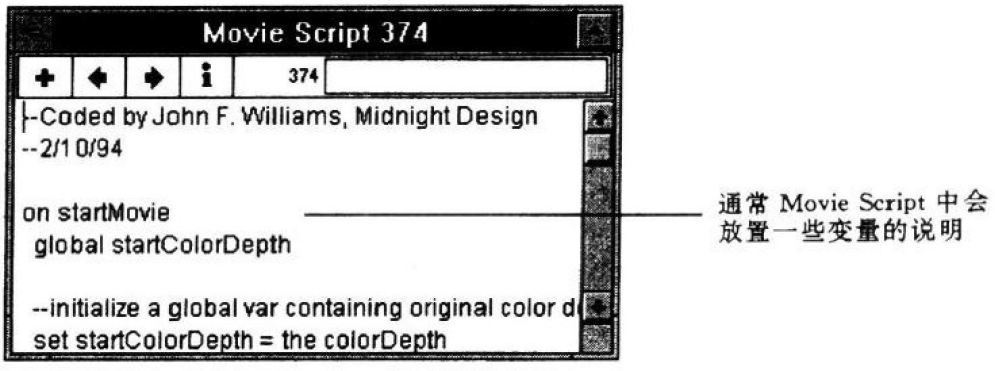


图 1.7　变量说明

另外一个功能是放置一些随时可以启动的事件，这些事件可以随时被调用。例如：时间计数的设计，在许多情形下，有必要启动一个事件，如果用户在固定时间内没有响应的话，这个事件就可以放置在电影的程序之中。

或者是当用户按某一个特殊键就会启动的事件，如 Esc 键或是 F1 键的设计，将这些事件放置在电影的程序之中，不论电影播映到何处，只要用户按一下这个键，就可以调用它来做特定的事情。

Cast Script

Cast Script 卡司成员程序主要的用意是，指定这个卡司成员为交互式成员，当用户或者是电影本身启动此成员时，就会执行其中的程序。如果您设定的事件是 MouseDown 或是 MouseUp，都表示此成员会变成可按式的成员，当用户在这个成员的上方按一下时，就会执行它的程序了。

所以当卡司成员随身携带着程序时，就表示它已经变成了交互式的。不管它的位置如

何，也就是说不论它的镜头位置如何，也不论它的场次号码为何，都是可以启动的成员，如图 1.8 所示。

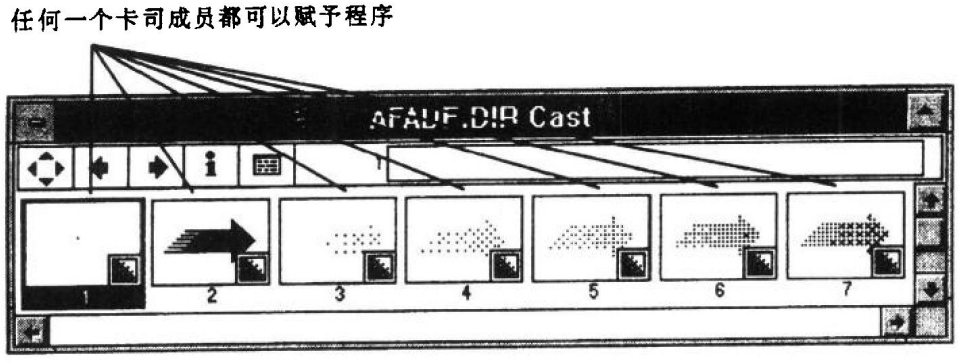


图 1.8　卡司成员

那么成员在连续的分镜上有好多个，是不是每一个成员都可以执行程序呢？答案是肯定的。只要指定成员的程序之后，不论它在任何位置上都是可以启动的，也不论它出现多少次，这是卡司成员程序的特性。

附带着程序的卡司成员有没有不同？如果仔细观察的话，可以发现在卡司成员视窗的左下角有个 L 形的符号，这与右下角的图示不同，只有在内含程序的成员中才有，请看图 1.9。

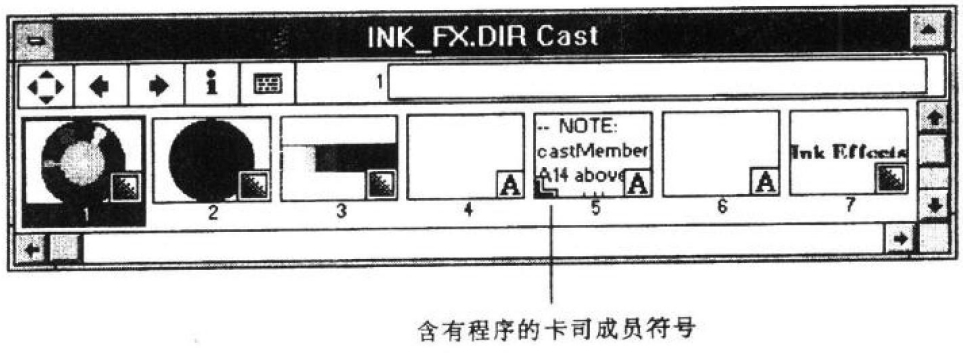


图 1.9　卡司成员的不同符号

所以当看到卡司成员上有一个 L 形的符号时，就可以知道它是一个带有程序的卡司成员。

Frame Script

第三种程序是镜头的程序，它所控制的是镜头的播映，或是到此镜头所要做的事情，上一节谈到的循环播映就是一个属于镜头的程序。如图 1.10 所示。

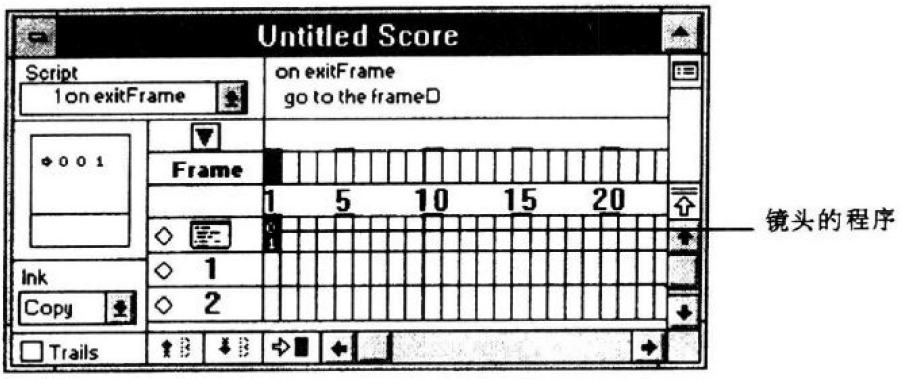


图 1.10　控制镜头的播映

镜头的程序只有到此镜头时才会启动，也就是说当电影播映时，播映头来到这个镜头时，就会启动这个镜头的程序，去做一些事情，然后电影再继续往下播映。

有时候需要在看到镜头之前就执行程序，这样的事件可以使用 on EnterFrame，这表示程序会在进入镜头之前执行，然后才播映镜头中的内容。而 on ExitFrame 则表示在镜头显示之后，才能执行程序。这二种执行的方式只是时间不同，您可以视需要选择适合的方式。

Sprite Script

最后一个要谈到的是分镜程序(Sprite Script)，这个分镜程序所控制的是分镜上的卡司成员，因为可能在连续的分镜上使用到同一个卡司成员，但是只希望在最后一个分镜上设计程序，这就表示只有这个位置上的卡司成员才有程序，请参考图 1.11。

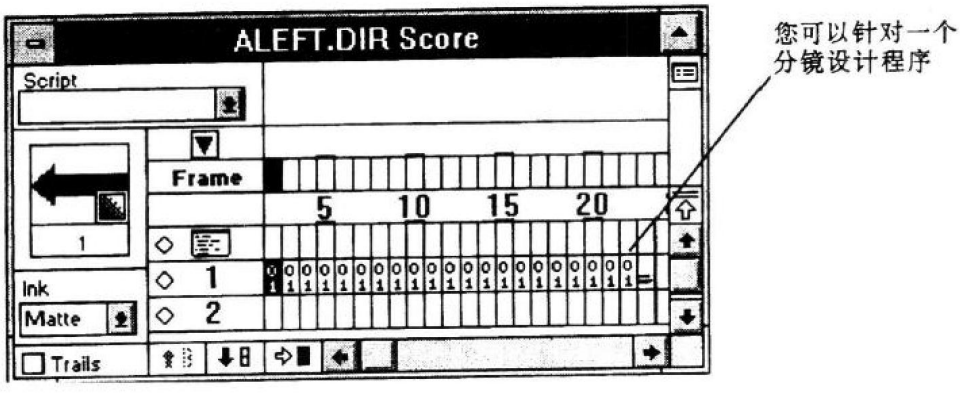


图 1.11　分镜程序

为什么不用 Cast Script 取代 Sprite Script 呢？因为这个卡司成员也可能在电影的其它位置上出现，但那里并不希望这个成员带有程序。如果使用卡司成员的程序就不太理想，因

为只要赋予成员程序，不论它在任何位置上都会执行程序。

这样看来 Sprite Script 就会显得比较理想，因为它只针对这一个分镜上的卡司成员设定程序，当离开这个分镜时，卡司成员就会忘记程序。

分镜程序除了针对个别的分镜设计之外，也可以针对多个分镜设计，最明显的是针对整个场（channel）的设计，如果这样的程序经过设计之后，当电影播映经过每一个分镜时，都可以启动这个分镜的程序。

四种程序如何应用呢？视情形需要，不过笔者还是列了几个简单的原则，以供参考：

1. Movie Script：适合于程序的说明，以及子程序放置。
2. Cast Script：当需要某一个特定的图形或按钮时，例如离开、返回主画面的按钮。
3. Frame Script：画面进入或离开时所做的动作。
4. Sprite Script：当某一个分镜中图形需要指令时，如暂停、继续播映等按钮。

1-5 如何监控程序的执行

一个程序设计者最注重的应该是如何监控程序的执行，因为只有设计者能够掌握住系统的状态变化，他才可能了解自己的程序到底哪里出了问题及如何修正。

所以在一个可以设计程序的软件中，一定要有视窗来监控系统的状态，在 Director 之中，这个视窗称之为 Message 视窗。

Message 视窗是一个专门为监控系统状态所设置的视窗，它的功能除了可以监控之外，还可以查询目前程序使用到的变量是哪个。

开启 Message 视窗的方式主要是启动 Window 菜单中的 Message 指令，或是使用 Ctrl-M 这个快捷键，屏幕如图 1.12 所示。

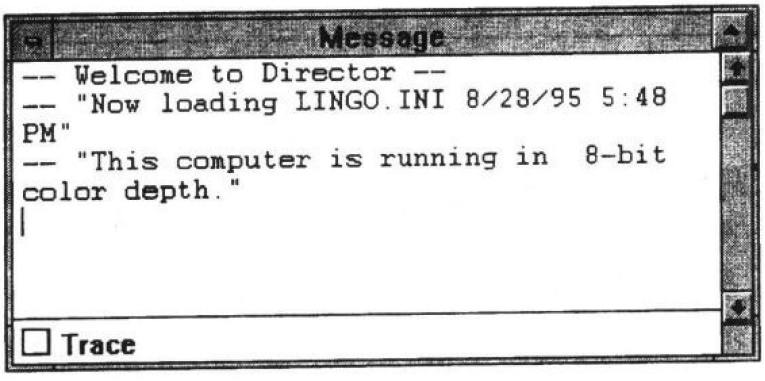


图 1.12 Message 视窗

在这个视窗中最主要的选项在左下角，有一个 Trace 检查框，选取它之后，就可以看到电影在执行时所发生的变化。