

中文版

Authorware

6

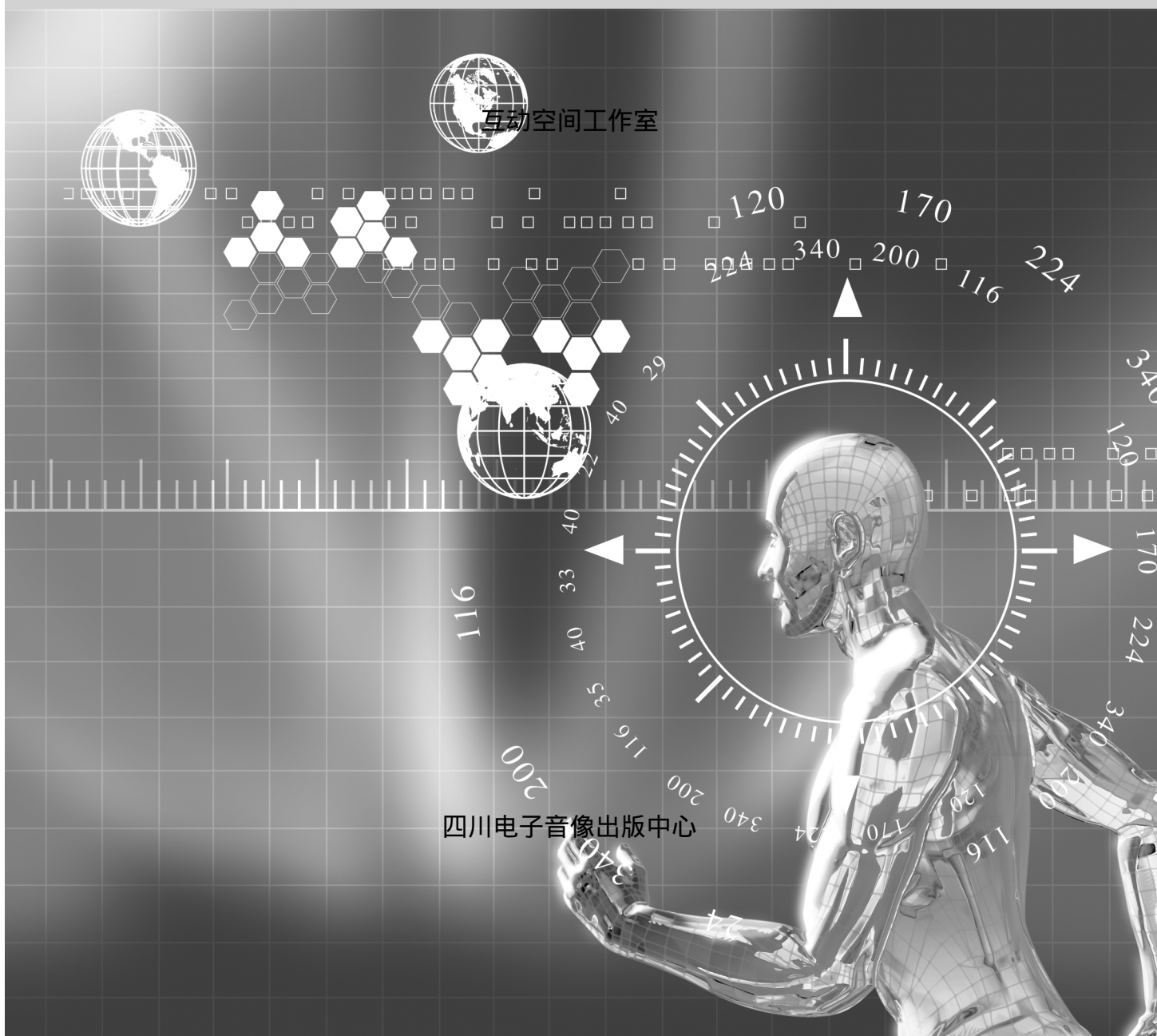
一本通

基础 技巧 实例



互动空间工作室

四川电子音像出版中心



内 容 提 要

Authorware 是功能强大的多媒体制作软件,被广泛应用于多媒体教学、课件制作和各种演示性光盘,在众多多媒体制作软件中, Authorware 以其简便易用、功能强大而倍受广大用户的青睐,作为一个优秀的多媒体制作软件, Authorware 不仅利用自身图标和函数来设计程序,而且还能够利用它的控件、接口、自定义函数等实现许多复杂的功能。

本书以 Authorware 6 为蓝本,全面、详细地介绍如何使用 Authorware 进行多媒体应用程序设计。全书共分 18 章,层次清楚、详略得当,且每讲解一个知识点后都精心设计了一个实用性很强的实例,本书可供从事多媒体创作及相关人员学习和参考、更适合作教材使用。

书 名	Authorware 6 中文版一本通			
文 本 著 作 者	互动空间工作室			
审校 /责任编辑	陈学韶			
C D 制 作 者	四川电子音像出版中心多媒体制作部			
出版/发 行 者	四川电子音像出版中心			
地 址	成都市桂花巷 21 号 (610015)			
经 销	各地新华书店、软件连锁店			
C D 生 产 者	东方光盘制造有限公司			
文 本 印 刷 者	成都墨池教育印刷总厂			
规 格 / 开 本	787 毫米 × 1092 毫米	16 开本	19.25 印张	456 千字
版 本 号	ISBN 7-900364-84-6/TP · 56			
定 价	29.00 元 (1 张光盘含配套手册)			

版权所有 盗版必究

举报电话:四川省版权局:(028) 86636481 技术服务电话:(028) 85577653

8.10.1 选择“重试限制”选项后的 响应类型属性对话框.....	161
8.10.2 在交互中创建限制次数响应.....	162
8.10.3 限制次数响应实例 ——猜数游戏.....	163
8.11 限制时间响应（时间限制）.....	165
8.11.1 选择“时间限制”选项后的 响应类型属性对话框.....	165
8.11.2 限制时间响应实例 ——抢答题.....	167
8.12 事件响应（事件）.....	169
8.12.1 选择“事件”选项后的 响应类型属性对话框.....	169
8.12.2 在交互中创建事件响应.....	171
8.12.3 事件响应实例——日历.....	171
第9章 【计算】图标.....	175
9.1 【计算】图标简介.....	175
9.1.1 使用【计算】图标.....	175
9.2 【计算】图标的操作.....	176
9.2.1 编辑【计算】图标的计算语句.....	176
9.2.2 改变计算编辑窗口中参数设置.....	177
9.3 【计算】图标实例——计算器.....	178
第10章 【数字电影】图标、【声音】图标....	181
10.1 图标简介.....	181
10.1.1 【数字电影】图标.....	182
10.1.2 【声音】图标.....	186
10.2 声音动画实例——小电影.....	188
第11章 开始旗、终止旗以及图标调色板.....	192
11.1 “开始旗”和“终止旗”简介.....	192
11.1.1 “开始旗”和“终止旗”介绍.....	192
11.1.2 “开始旗”和“终止旗”的使用....	193
11.2 图标调色板介绍及其使用.....	194
第12章 库的使用.....	195
12.1 库的简介.....	195
12.1.1 库的特点.....	195
12.1.2 库窗口.....	196
12.2 库的使用.....	196
12.2.1 创建一个新库.....	197

12.2.2 建立链接和断开链接.....	199
第13章 变量、表达式、函数的使用.....	201
13.1 变量简介.....	201
13.1.1 变量的类型.....	202
13.1.2 变量的赋值.....	203
13.2 系统变量.....	204
13.2.1 在应用程序中显示变量的值.....	204
13.2.2 常用系统变量介绍.....	205
13.3 自定义变量.....	207
13.3.1 定义自定义变量.....	207
13.4 表达式的使用.....	209
13.4.1 运算符.....	209
13.4.2 各种运算符的优先级和结合性.....	210
13.5 函数简介.....	211
13.6 系统函数.....	212
13.6.1 应用程序中使用系统函数.....	212
13.6.2 常用系统函数介绍.....	213
第14章 Authorware 的外部函数——UCD.....	221
14.1 UCD（用户代码文档）简介.....	221
14.1.1 装载UCD函数.....	222
14.1.2 常用UCD函数详解.....	223
14.1.3 其他一些常用UCD功能简介.....	227
14.2 使用UCD函数实例——MIDI播放器..	228
第15章 知识对象的使用.....	233
15.1 知识对象简介.....	233
15.1.1 知识对象窗口.....	233
15.1.2 使用知识对象.....	234
15.2 系统提供的知识对象简介.....	235
15.2.1 评估 ——与评定操作有关的知识对象.....	235
15.2.2 文件 ——与文件操作相关的知识对象.....	235
15.2.3 Icon Palette Settings ——与图标板设置相关的知识对象.....	236
15.2.4 界面构成 ——与界面部件相关的知识对象.....	236
15.2.5 Internet ——与网络相关的知识对象.....	237
15.2.6 Model Palette	



——与图标模板相关的知识对象.....	237
15.2.7 新建文件	
——与创建新文件相关的知识对象.....	237
15.2.8 RTF 对象	
——与 RTF 对象相关的知识对象	237
15.2.9 指南	
——与教程相关的知识对象.....	238
15.3 知识对象应用实例	238
15.3.1 使用“查找 CD 驱动器”	
知识对象	238
15.3.2 使用“发送 Email”知识对象	240
第 16 章 Authorware 的高级功能	243
16.1 使用 OLE 对象	243
16.1.1 创建一个 OLE 对象.....	243
16.1.2 嵌入一个 OLE 对象.....	244
16.2 ActiveX 控件的使用.....	245
16.2.1 添加一个 ActiveX 控件.....	246
16.2.2 编辑 ActiveX 控件图标.....	249
16.2.3 使用 ActiveX 控件实例	
——媒体播放器.....	250
16.3 数据库的调用——ODBC	253
16.3.1 创建一个数据库	253
16.3.2 使用 ODBC 链接数据库	254
16.3.3 使用 Authorware 调用	
外部数据库.....	256
16.3.4 插入 Flash 动画.....	260
16.3.5 插入 QuickTime 电影	262
16.3.6 插入动态 Gif 文件	264
16.4 “影片欣赏”的制作.....	266
16.4.1 “影片欣赏”简介	266
16.4.2 “影片欣赏”的使用说明.....	266
16.5 设计“影片欣赏”程序.....	267
第 17 章 Authorware 其他高级功能	274
17.1 输出媒体	274
17.2 将.WAV 文件转换成.SWA 文件	275
17.3 查找 Xtras 文件.....	277
17.4 RTF 对象编辑器	278
第 18 章 Authorware 作品的发行与打包.....	280
18.1 应用程序的调试	280
18.1.1 使用“开始旗”和“终止旗”	280
18.1.2 使用控制面板	281
18.1.3 中断正在运行的程序.....	283
18.2 作品的打包和发行	283
18.2.1 发行作品时所包含的文件.....	283
18.2.2 外部文件的搜索路径.....	284
18.2.3 非网络作品的打包和发行.....	284
18.2.4 网络作品的打包和发行.....	295
18.2.5 编辑网络发行的 Map 文件	297
18.2.6 如何制作自动运行的光盘.....	299
18.2.7 如何改变可执行文件的图标.....	299

第1章 Authorware 6 简介

- Authorware 概述
- Authorware 6 新增功能
- 初识 Authorware 6 界面
- Authorware 的常见设置

在信息技术高速发展的今天，多媒体计算机开始进入到千家万户，成为人们的常用工具，多媒体软件的需求量也越来越大。多媒体的出现，改变了教育观念，使出版物的表现形式更加丰富多彩。

如同电影、电视作品的创作一样，多媒体作品的创作过程一般也包括选题立项、编写脚本、准备素材、制作合成等环节。而我们现在所做的就是用 Authorware 6 进行作品素材的制作合成，这也是通常意义上的多媒体制作。

多媒体创作工具主要包括四种类型：

- (1) 传统的编程语言多媒体创作工具，主要软件有：Visual C++，Visual Basic，Delphi 等。
- (2) 基于时间轴的多媒体创作工具，主要软件有：Action，Director 等。
- (3) 基于卡片和页面的多媒体创作工具，主要软件有：ToolBook，Hypercard 等。
- (4) 基于图标和流程线的多媒体创作工具，主要软件有：Authorware，IconAuthor 等。

在众多的多媒体开发软件中，Macromedia 公司的 Authorware 是深受广大用户和多媒体创作人员喜爱的优秀多媒体创作工具，大部分的多媒体光盘都是使用 Authorware 制作而成的。

在 Macromedia 公司于 1992 年推出 Authorware 2.0 版本以来，Authorware 一直是多媒体制作人员的首选多媒体开发软件。Authorware 4.0 曾经是非常流行的多媒体创作软件。但是随着网络的发展，Authorware 4.0 创作的多媒体作品的发行速度和效果受到了网络速度的影响。为此，Macromedia 公司在 1998 年底推出了具有创作网络多媒体学习软件功能的 Authorware 5 Attain。该版本加强了制作多媒体网络学习软件的功能，同时增加了知识对象（Knowledge Object）向导功能，创作人员可以利用该制作工具提供的向导应用程序来创建基于网络的多媒体学习软件，这样大大提高了网络学习软件的创作速度，降低了软件开发人员的劳动强度。在 Authorware 5 Attain 的基础上，Macromedia 公司于今年又推出了 Authorware 6。



1.1 Authorware 概述

Authorware 是一种面向对象并基于图标和流程线的多媒体创作平台,具有丰富的函数和程序控制功能,融入了编辑系统和编程语言的特色。它允许用户使用动画、图形、声音、视频等多媒体信息来创作一个具有交互功能的多媒体应用程序。

Authorware 6 是创建 Web 和在线学习的可视化多媒体制作工具,它可以为您提供如下功能:

- 创建强大交互功能、多媒体学习程序。
- 通过 Web、LAN 和 CD-ROM 为客户和职员提供信息。
- 跟踪学生的学习进度和学习效果。

1.1.1 Authorware 的特点

Authorware 作为一个优秀的多媒体应用程序创作软件,具有以下特点:

1. 为非程序员设计的

Authorware 是为非程序员设计的。一般的教师、培训人员和专家们都可以自由地设计自己的多媒体软件。Authorware 提供的基于图标和流程线设计思路,是每一个非程序员都可以理解和使用的,只要程序按照流程线顺序执行,即可实现相应的功能。Authorware 的图标提供了全面创作交互式应用程序的能力,这些图标不但可以帮助用户组织程序的总体结构,使整个程序更加具有逻辑性,而且 Authorware 也提供了一个可视化的设计环境和创作工具,使创作过程更加轻松。

2. 与其他多媒体软件的配合使用

由于 Authorware 是一个多媒体制作软件,它考虑的只是如何将一些多媒体元素组织起来,形成一个具有声音、图形、图像、动画等信息的程序。而这些多媒体元素,就需要使用外部软件制作好,或者是使用现成的元素库。而要设计一个功能齐全、令人震撼的多媒体作品,必须要同一些相关软件配合使用。如,使用 Photoshop 设计图形图像、使用 SoundEdit 编辑声音、使用 3D Studio MAX 制作动画等等。同时,要想制作出基于网络的多媒体培训软件,还可以使用 Flash 制作的矢量动画、加入 GIF 文件、使用 Apple QuickTime VR (虚拟现实) 文件,以及配合使用 Macromedia Dream Weaver 和 Macromedia Pathware。当然,如果用户对一些软件不是很熟,也可以使用现成的元素库,照样可以设计出十分漂亮、功能齐全的多媒体软件。

3. 模块化功能

Authorware 允许将整个应用程序分成几个独立的逻辑模块,由多个开发人员单独设计完成,大大提高了开发效率。

4. 多平台的支持

使用 Authorware 可以在一个操作系统下创作,也可以在另一个操作系统下继续开发,并不会影响设计效果。而且开发出的软件产品,可以通过一定的技术,使得可以在任何平台上发行。

5. 方便的 Internet 发布

Authorware 中的所有功能,包括动画、交互以及数据采集,都可以应用到 Internet 上。任何大小的文件都可以使用 Authorware 引擎预做,压缩率可以达到 50% 以上,同时使用多媒体网络媒体传输技术——Shockwave 技术,可以用安装的 Shockwave 插件的浏览器查看相关网页。

1.1.2 Authorware 6 的新增功能

Authorware 从 1.0 到 4.0 版本,一直是众多多媒体开发工具中的佼佼者,从 Authorware 5 Attain 开始,正式定位于多媒体教育培训软件的开发上,并增加了 CMI 计算机学习管理的强大功能,为开发基于网络发行的学习软件提供了强有力的支持。Authorware 6 相对与以前版本,主要增加了以下主要功能:



1. 单按钮发行和其他方面的提高 (One-button publishing and other enhancements)

在发行过程中,单按钮发行结合和自动化了所有步骤。它使只是单击一下按钮就可以将应用程序发布到 Web、CD-ROM 或者企业网络中成为可能。

(1) 可以在同一时刻不同方式地打包和发行产品。例如,你可以仅仅在一个步骤中就可以将产品打包为非运行时文件 (A6R 文件)、Web Player 文件 (AAM 文件) 和一个 Web 页面 (HTM 文件)。

(2) 自定义发行方式,您可以重复使用设置好的发布设置。

(3) 自动识别和收集您发布产品中所需要的很多支持文件,如 Xtras 文件,DLL 文件和 UCD 文件等。

(4) 将你设置好的 WEB 文件 FTP 到远服务器上。

如果你需要发布多样的 Authorware 产品,你可以使用批发布特性在同一时间内将多个产品以不同方式发行,并且你可以跟踪发布的所有产品。

尽管单按钮发行可以帮助你定位文件中需要使用的 Xtras,并可以在作品发布过程中将 Xtras 放置到合适的文件夹中。你也可以在发布产品前使用新增加的查找 Xtras 命令来定位 Xtras。

2. 新的命令菜单 (New Commands Menu)

在 Authorware 6 中,您可以创建和添加您自己自定义的菜单命令,来扩充命令菜单。

3. RTF 编辑器和 RTF 知识对象 (Rich text format (RTF) editor and knowledge objects)

现在,您可以通过使用新的丰富文本编辑器创建出丰富的,可以重复使用内容,以满足于新的创作需求。在 Authorware 6 中,主要有两个 RTF 组件:

(1) RTF 对象编辑器:您可以使用 RTF 对象编辑器来设计屏幕或者其他任何 RTF 文件。在这里,您可以混合使用文本、图像、图形、Authorware 变量和表达式,以及热点文本。

(2) RTF 知识对象:您可以使用 RTF 知识对象以在 Authorware 展示窗口中显示外部的 RTF 文档。可以动态地链接到外部的使用 RTF 创作好的丰富文本文档,这样文本内容就可以不需要重新编辑和直接更新了。

另外,如果您导入了 RTFObj.u32,您还可以使用几个新 RTF 对象函数。

4. 接口的改进 (Interface improvements)

Authorware 接口已经重新修订以增强流程线、声音和影片图标的同步,以及改善屏幕功能。

(1) 如今您可以创作自定义图标并将其存储到图标板中。

(2) 如今可以很容易地判断出流程线中一个图标的内容是空的还是包含有内容。空内容的图标的颜色是灰色的。

(3) 新增了 10 种新颜色,以帮助您区别流程线和库中的图标。

(4) 为了帮助您更好地组织流程线,你现在可以在【窗口】菜单中显示一个地图菜单,或者在设计窗口中单击鼠标右键。

(5) 您可以调整函数对话框和变量对话框的大小和尺寸。

5. 其他接口的增强 (Other interface enhancements)

Authorware 接口还在如下方式中得到了增强:

(1) 当您创建一个新文件时,一般会显示一个新工程对话框,让您使用知识对象来创建一个新应用程序或者一个测验。您可以选中一个复选框来关闭该对话框。如果你想再次使用该工程对话框,可以通过选择【文件】→【新建】→【方案】菜单来创建新工程。

(2) 当您选择菜单【文件】→【打开】→【文件】来打开最近的文件时,可以显示最近文件的数量已经增加到 10 个。

(3) 在库窗口中增加了一个颜色栏,您可以使用该颜色调色板来为库中的图标进行着色,并且可以依照颜色将图标进行排序。

(4) 当您选择菜单【文件】→【打开】→【库】命令来打开一个库文件时，缺省文件类型是库文件（A6L 文件），而不是所有文件。

(5) 关闭窗口和关闭所有窗口已经移到【窗口】菜单下。

(6) 增加了新的键盘快捷键：Control-E 是显示响应属性对话框窗口；Ctrl+Alt+M 是显示保存模板菜单选项。

6. 新媒体特性 (New media features)

在 Authorware 6 中，除了新 RTF 特性外，您还可以用到另外两个新的媒体特性：

(1) 直接事件通过时间来同步显示媒体和其他基于时间媒体的事件。媒体同步允许 Authorware 的【声音】图标和【数字电影】图标可以通过在声音文件或者电影文件的媒体位置来触发任何事件。

(2) 在 Authorware 6 中，您可以将 MP3 加入到您的作品中。

7. 计算窗口功能的增强 (Calculation window enhancements)

Authorware 6 中，计算窗口做了几处改善。如今，您可以改变计算参数、插入符号和分隔线、或者消息框，而且可以查看函数或者变量中选字符的 ASCII 值。

成绩跟踪的增强 (Performance tracking enhancements)

(1) PWInt.x32 Xtra PWInt.x32 Xtra 做了几处改变。现在 PWInt.x32 Xtra 可以支持 IsCourseChanged、IsLibraryChanged、CMISetAICCVersion 和 CMIGetAICCVersion 函数。另外，在 Authorware 中，CMI 支持 the CMI support in Authorware no longer requires a version sent from the CMI system. ExitAU is now sent with CMIFinish.

(2) 问题评估知识对象：热区、热对象、真/假、单选和多选问题评估知识对象已经更新，它允许导入和导出存储在 XML 文件中的问题内容，该 XML 文件必须要符合 IMS Question Test Interoperability (QTI) 1.1 版本标准。它允许您可以将问题导出到一个 XML 文件中，或者从一个 XML 文件中导入问题。

8. Web 技术的增强 (Web technology enhancements)

现在，您可以在您的 Authorware 程序和支持标准的 Web 技术：

(1) 可以通过使用任何 ActiveX 控件所支持的扩展属性和方法来扩展您的应用程序。

(2) XMLParser Xtra 允许 Authorware 作品可以读入、解析和使用 XML 文档的内容。通过导入存储在基于 Web 标准的外部 XML 文件中的数据来创建动态的、数据驱动的应用程序。

(3) 您可以通过增强的数据跟踪能力、SCO Metadata Editor 来跟踪学生的活动。

(4) 通过 Authorware Web Player 从 Web 下载 Authorware 6 的内容，可以缩小内容的 40%。

9. 新的属性、函数和变量 (New properties, functions, and variables)

Authorware 6 提供了很多新的属性、函数和变量。现在大概有超过 100 个新的属性可以在 GetIconProperty 和 SetIconProperty 函数中使用。

10. 新的函数

Authorware 6 增加了如下所列的函数。

- GetExternalMedia (IconID@ "IconTitle"/ LibraryID)
- GetFunctionList (Category)
- GetLibraryInfo()
- GetPasteHand()
- GetSelectedIcons()
- GetVariableList (Category)
- GroupIcons()
- OpenFile ("filename")

- OpenLibrary ("filename")
- PackageFile("OutputFile",Runtime, ResolveLinksAtRuntime, PackLibsInternal, PackMediaInternal, UseDefaultNames, 【LibraryLocations】)
- PackageLibrary(LibraryID,"OutputFile", ReferencedOnly, UseDefaultName, PackMediaInternal)
- SaveLibrary(LibraryID, 【"New Filename"】)
- SetHotObject(iconID for response, iconID for object)
- SetMotionObject(iconID for response, iconID for object)
- SetTargetObject(iconID for response, iconID for object)
- UngroupIcons()

11 . 新的变量

Authorware 6 增加了如下所列的变量。

- LastObjectClickedID
- ObjectClickedID
- ObjectMovedID

1.2 初识 Authorware 6 开发环境

1.2.1 Authorware 6 的窗口结构

Authorware 6 是一个典型的 Windows 应用程序窗口，如图 1-1 所示。

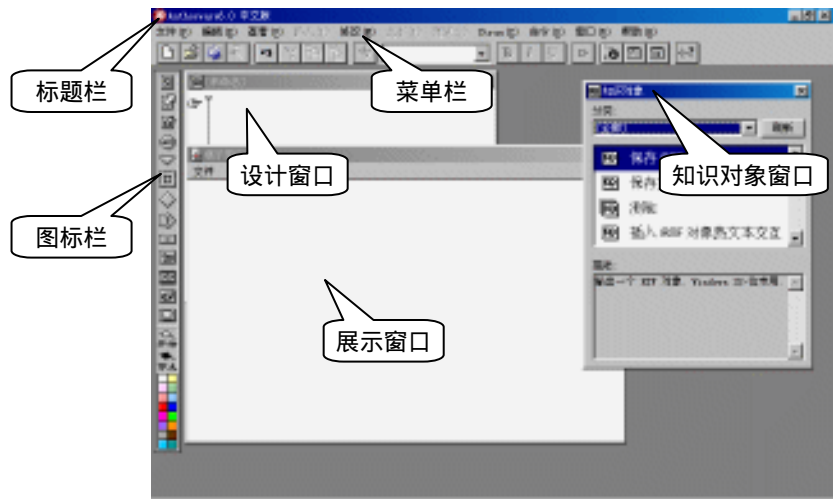


图 1-1 Authorware 6 窗口结构

窗口主要由七个部分组成，即标题栏、菜单栏、工具栏、图标栏、设计窗口、展示窗口和知识对象窗口。通过这七个部分，用户可以编辑出强大的多媒体作品。下面我们对这七个部分分别进行介绍。

1.2.2 标题栏

与其他 Windows 应用程序窗口一样，标题栏位于窗口的最上方，标题栏的左侧是应用程序图标和应用程序名称，右侧的三个按钮是最大化、最小化和关闭按钮。单击标题栏左侧的应用程序图标，弹出

下拉菜单，如图 1-2 所示。这也与其他 Windows 应用程序窗口一样。



图 1-2 标题栏中的下拉菜单

1.2.3 菜单栏

Authorware 6 的菜单栏共有 10 个菜单：文件，编辑，查看，修改，文本，调试，Xtras，命令，窗口，帮助等。单击每个菜单都会弹出一个下拉菜单，菜单中的每一个菜单选项称为一个菜单命令。如图 1-3 所示是 Authorware 6 的菜单栏。



图 1-3 Authorware 6 的菜单栏

在使用 Authorware 菜单时，有一些特殊含义的标记符代表不同的含义，如图 1-4 所示。

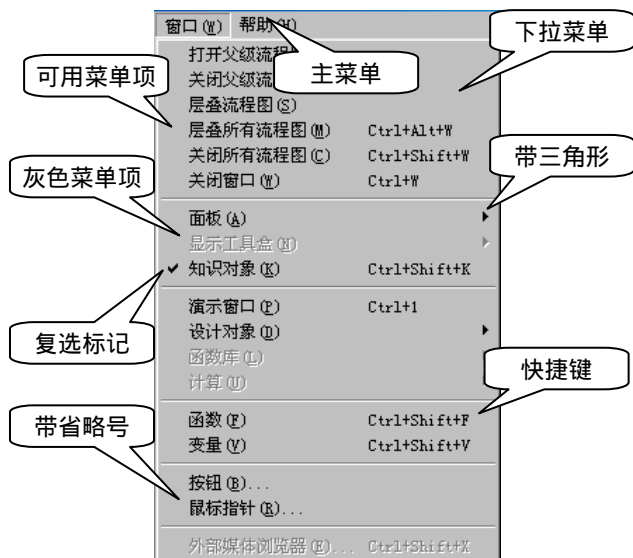


图 1-4 菜单的不同选项

带有不同的标记符的菜单选项代表不同的意义：

- 黑色菜单选项
表示该菜单命令是可用的。
- 灰色菜单选项
表示该菜单命令是不可用的。
- 带有复选标记的菜单选项

表示该菜单是一种选择式命令，但出现勾号时表示该命令正在使用，没有勾号时表示该命令没有被使用。进行选择该命令，可以实现选择和没有被选择的切换。

- 带有省略号的菜单选项

表示选择该菜单命令后会弹出一个相关的对话框,在对话框中可以进行更加复杂的选项设置和信息确认。

- 带有三角形的菜单选项

表示该菜单命令还有下一级的子菜单命令。

在有些菜单选项的右边有一些诸如“Ctrl+2”式的字符,表示是该菜单命令的快捷键。如图 1-3 所示,“Ctrl+2”是打开控制面板(Control Panel)的快捷键。在 Authorware 的使用中,多多使用快捷键可

1.2.4 工 具 栏

在菜单栏下面,Authorware 6 提供了工具栏。工具栏中的工具按钮都与某一项菜单命令等效。工具栏的作用是为了加快访问一些常用菜单命令的速度。工具按钮上的图标形象地说明了该工具按钮的作用,此外还可以将鼠标指针放到工具按钮上面,则在该按钮的下方会显示出一个小方框,里面是该工具按钮的名称。这在 Authorware 6 中称为屏幕提示。由于工具栏是一些比较常用的菜单命令的快捷执行方式,所以熟练掌握工具栏,可以提高开发效率。

Authorware 6 的工具栏中各个工具按钮如图 1-5 所示。



图 1-5 Authorware 6 工具栏

工具栏中的工具按钮的含义为：

 “新建文件(New)”按钮：用于新建一个 Authorware 文件。单击该按钮,Authorware 将新建一个名为“Untitled”的设计窗口。该按钮的功能等价于菜单中

 “打开文件(Open)”按钮：用于打开一个 Authorware 文件。

 “保存文件(Save All)”按钮：用于保存编辑过的 Authorware 文件。

 “导入文件(Import)”按钮：用于在 Authorware 文件中引入外部的图形图像、文字、声音、动画或者 OLE 对象等。

 “撤消(Undo)”命令按钮：用于撤消用户上一次的操作。

 “剪切(Cut)”命令按钮：用于将用户选定的对象剪切到剪贴板上。

 “复制(Copy)”命令按钮：用于将用户选定的对象复制到剪贴板上。

 “粘贴(Paste)”命令按钮：将剪贴板上的内容(用户剪切或者复制的对象)粘贴到指定位置。

 “查找(Find)”命令按钮：用于在文件中查找用户指定的文本。

 “文本风格(Text Style)”下拉框：从该下拉框中选择已经定义好的文本风格

使用到当前的文本中。



“粗体 (Bold)”命令按钮：将选中的文本字体加粗显示。



“斜体 (Italic)”命令按钮：将选中的文本字体斜体显示。



“下划线 (Underline)”命令按钮：将选中的文本字体加上下划线。



“执行程序 (Restart)”命令按钮：执行当前的 Authorware 文件。如果流程线上有 Start 开始旗，则从开始旗处开始执行程序。



“控制面板 (Control Panel)”命令按钮：单击该按钮，系统会弹出控制面板。使用控制面板可以调试程序。



“函数窗口 (Functions Window)”命令按钮：单击该按钮，系统弹出函数窗口。



“变量窗口 (Variables Window)”命令按钮：单击该按钮，系统弹出变量窗口。



“帮助 (Help)”命令按钮：单击该按钮，鼠标指针变成  形状，用鼠标在有疑问的地方单击，Authorware 就会弹出相应的帮助信息。

1.2.5 图 标 栏

在 Authorware 窗口的左边是 Authorware 编程的主要工具——图标栏。图标栏是 Authorware 进行多媒体创作的基本工具，每一个图标都有不同的功能。

在图标板下面还有三个图标：开始旗、终止旗和图标调色板。各种图标的功能和具体使用将在后面的章节给予介绍，下面我们简单给出各种图标的简要功能，详见表 1-1 所示。

表1-1 各种图标的名称和功能

图标按钮	图标按钮名称	功 能
	【显示】图标 Display Icon	显示文字、图形、图像等。这些文字或者图形图像可以由外部导入，也可以用绘图工具箱中的工具自己绘制
	【移动】图标 Motion Icon	将显示的对象在指定的时间内或者按指定的路径从一处移到另一处。只有可显示的图标内对象才可以移动
	【擦除】图标 Erase Icon	类似于橡皮擦，按一定的方式擦除图标中的文字、图片、声音、动画等
	【等待】图标 Wait Icon	用于中止程序的执行以等待用户与程序交互，或者等待一段时间

	【导航】图标 Navigate Icon	实现程序内的跳转，其目的地只能是【框架】结构下的某一页，建立超级链接，实现超媒体导航
	【框架】图标 Framework Icon	提供一组定向控制按钮，与【导航】图标互相配合实现超级链接
	【决策】图标 Decision Icon	用来设置一种判定逻辑结构，附属于【决策】图标下的其他图标被称为路径。当遇到【决策】图标时，Authorware 根据条件的不同自动选择执行路径
	【交互】图标 Interaction Icon	用来创建一种交互作用分支结构。一个交互作用分支结构可以由一个交互项和附属于它的其他图标组成。Authorware 提供了 11 种交互方式
	【计算】图标 Calculation Icon	用于执行算术运算、各种函数的运算以及执行一定的代码运算
	【群组】图标 Map Icon	利用【群组】图标将一组图标合成一个简单的【群组】图标，形成下一级流程窗口，缩短流程线，优化流程线结构
	【数字电影】图标 Digital Movie Icon	又称为动画图标，该图标可以播放*.AVI，*.MOV，*.DIR，*.FLC，*.FLI 等常见的数字电影动画格式文件
	【声音】图标 Sound Icon	可以播放*.WAV，*.AIF，*.VOX，*.SND 等常见格式的声音文件，并且可以对播放方式进行控制
	【视频】图标 Video Icon	在多媒体应用程序中控制视频设备的播放
	开始旗 Start Flag	用于程序的调试，设置程序运行的起始点。将开始旗放到流程线上，当用菜单[Control]♦[Restart from Flag]命令或者单击按钮执行程序时，Authorware 会从开始旗处执行
	终止旗 Stop Flag	用于程序的调试，设置程序运行的终止点。将终止旗放到流程线上，当执行程序遇到该标志时，Authorware 会自动停止执行程序
	图标调色板 Icon Color	为图标着色，让程序开发者方便地区分各类图标，它对程序的运行没有影响

1.2.6 设计窗口

Authorware 几乎所有的程序设计都是在设计窗口中进行的，如图 1-6 所示。



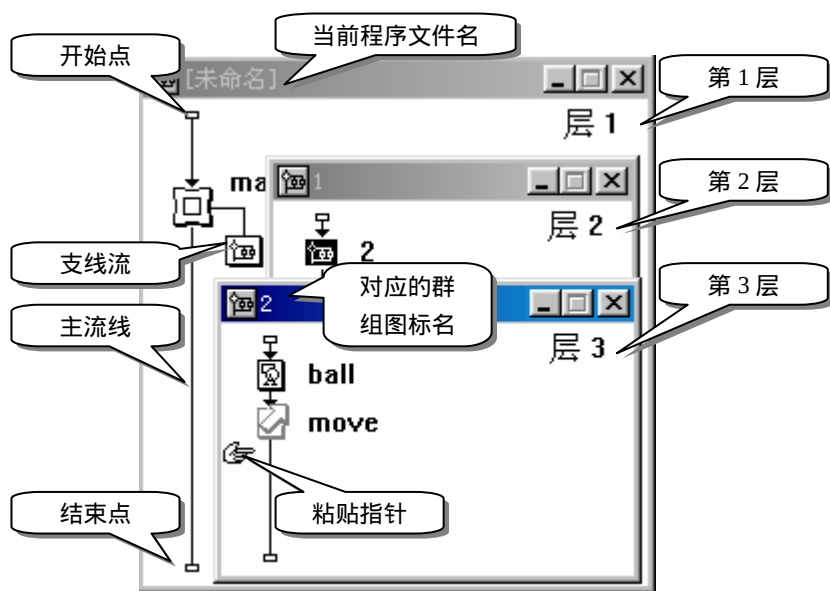


图 1-6 设计窗口

在设计窗口中的各个概念：

- 流程线：在设计窗口中一条被两个小矩形封闭的、贯穿上下的直线。Authorware 按照从上到下执行程序。
- 主流线：流程线上从上面的小矩形到下面的小矩形中的一条直线。
- 支流线：在设计窗口中除了主流线外的其他直线，主要包括【框架】图标、【交互】图标等下面的分支。
- 开始点：在流程线最上面的小矩形称为开始点，它是 Authorware 程序的开始。
- 结束点：在流程线最下面的小矩形称为结束点，它是 Authorware 程序的终止。Authorware 程序将沿着流程线从开始点一直运行到结束点。
- 粘贴指针：在设计窗口中呈现小手形状，表示复制粘贴图标时，需要将图标粘贴的位置。粘贴指针有两个作用：
 - (1) 指示下一步设计图标要放置在设计窗口中的位置；
 - (2) 指示可以在当前位置上按照什么流向来放置设计按钮。
- 窗口所处的层：在设计窗口的右上角有“层 1”、“层 2”等字样，表示当前窗口所处的层。设计窗口的主窗口为第 1 层，打开设计窗口中的【群组】图标，可以打开二级窗口，其窗口的标题栏上是该【群组】图标的名称，窗口的右上角有“层 2”字样，表示该窗口是第 2 层；同样，如果第 2 层窗口中也有【群组】图标，双击打开【群组】图标，即可打开第 3 层窗口；……

1.2.7 展示窗口

展示窗口是多媒体程序的基础，用户的各种媒体和交互式操作都是在这个窗口上进行的，如图 1-7 所示。

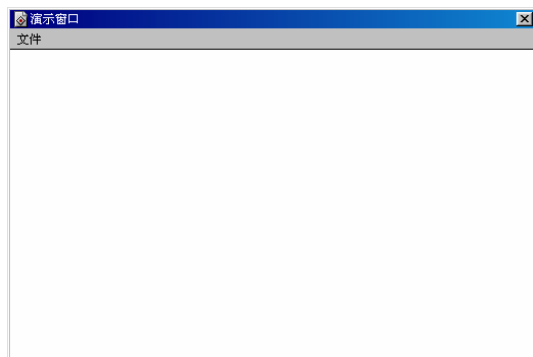


图 1-7 展示窗口

展示窗口又称为演示窗口，它是演示程序和运行程序时的显示界面。双击设计窗口中的【显示】图标、【移动】图标、【擦除】图标等显示对象，都可以打开展示窗口。在缺省情况下，展示窗口上有一个“文件”菜单，单击“文件”菜单弹出下拉菜单，在其下拉菜单中有一个“退出”命令（快捷键为“Ctrl+Q”），如图 1-7 所示，单击该命令可以退出展示窗口回到设计窗口；也可以单击展示窗口右上角的“关闭”按钮关闭展示窗口。

1.2.8 知识对象窗口

知识对象窗口提供了所有的知识对象，可供程序调用，如图 1-8 所示。

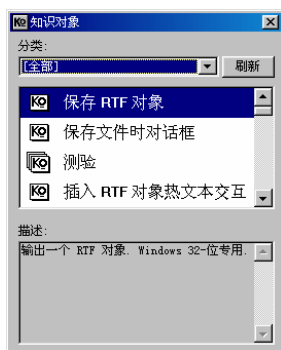


图 1-8 知识对象窗口

知识对象是 Authorware 6 所提供的一系列完成特定功能的功能模块，当用户使用 Authorware 6 所提供的这些知识对象时，知识对象会自动启动一个向导程序来引导用户完成当前知识对象的设置，等将每一步的设置都完成后，知识对象自动生成相应操作和功能。关于知识对象将在后面相应章节进行讲解。

1.3 Authorware 的常见设置

在使用 Authorware 之前，最好先对 Authorware 的一些使用条件进行设置，使得各种条件满足用户的需要。

1.3.1 文件属性设置

选择菜单[修改]→[文件]→[属性]命令，打开“文件属性”对话框，如图 1-9 所示。



图 1-9 文件属性对话框

在文件属性对话框的最左边记录了文件的大小，图标数目、变量数目以及所占的存储空间大小。文件属性对话框的最上面的文本框中是该文件的标题栏名称。

在文件属性对话框的中间有三个标签页，分别是：

1. “属性”标签页

在“属性”标签页中，设置主要针对的是展示窗口的各种属性。它可以控制展示窗口的大小展示窗口的显示样式等。

(1) 在“颜色”选项中有两个按钮： “背景颜色”按钮和 “色彩浓度关键色”按钮。

按钮显示的是展示窗口中的背景色，通过单击该按钮，弹出“颜色”对话框，如图 1-10 所示。

在“颜色”对话框中选择不同的颜色，可以将展示窗口中的背景色改变成不同的颜色。单击“颜色”对话框左边的颜色样例中满意的颜色框，在“颜色”对话框右上角显示选择的颜色样例。选择合适的颜色后，单击“确定”按钮确定，此时展示窗口的背景色已经改变为选择的颜色。背景色的缺省值是白色。

如果用户使用视频重叠卡，而且使用的视频重叠卡支持色彩浓度关键色，则可以使用色彩浓度关键色。使用色彩浓度关键色格式化的图形无论在何种条件下，Authorware 都可以正确地将这些对象显示在屏幕上。单击按钮设置色彩浓度关键色，打开“色彩浓度关键色”对话框，如图 1-11 所示。

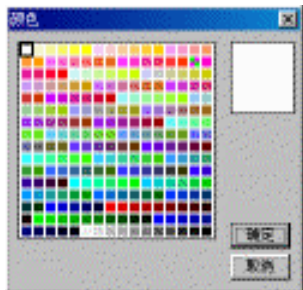


图 1-10 “颜色”对话框

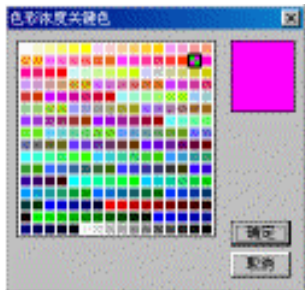


图 1-11 “色彩浓度关键色”对话框

在“色彩浓度关键色”对话框中的颜色设置同“颜色”对话框中的设置相同。

(2) “大小”下拉框中的选项控制的展示窗口的大小。“大小”下拉框中有以下选项如图 1-12 所示。



图 1-12 “Size”下拉框

- “根据变量”选项：当用户选择该选项时，用户可以根据自己的需要随意调整展示窗口的大小和位置。调整窗口的方法是将鼠标移到展示窗口的边框或者角上，此时鼠标指针变为双向箭头，按住鼠标左键拖动即可改变窗口的大小；将鼠标指针放到展示窗口的标题栏上，按住鼠标左键拖动即可移动窗口的位置。

如果选择该选项，在最终发行软件时，展示窗口的大小以最后一次设置的窗口大小为准。

- “512 × 342（Mac 9”）”选项：选择该选项时，展示窗口在水平方向上显示 512 个像素点，在垂直方向上显示 342 个像素点，其窗口的大小相当于 Machintosh 的 9 英寸屏幕上的标准 Windows 窗口大小。
- “640 × 480（VGA Mac13”）”选项：表示用户要在配有 VGA 显示器的计算机上运行程序。在标准的 VGA 显示器上显示出的展示窗口水平方向上显示 640 个像素点，垂直方向上显示 480 个像素点，其窗口的大小相当于 Machintosh 的 13 英寸屏幕上的标准 Windows 窗口大小。
- “800 × 600（SVGA）”选项：表示用户要在配有 SVGA 显示器的计算机上运行程序。在标准的 SVGA 显示器上显示出的展示窗口水平方向上显示 800 个像素点，垂直方向上显示 600 个像素点。
- “Use Full Screen”选项：选择该选项时，表示在运行程序时，Authorware 会根据用户显示器的分辨率自动地把展示窗口占据整个屏幕，即全屏显示。

其他选项：用户可以根据自己的需要调整展示窗口的大小，在此就不多述了。

（3）“选项”选项有 8 个命令选项，如图 1-13 所示。

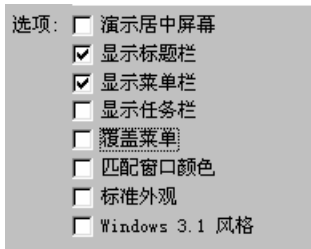


图 1-13 “选项”选项

“选项”选项用来定制展示窗口的显示样式。在“选项”选项中有 8 个命令选项，用户可以自己组合以满足需要。在缺省的情况下，Authorware 的展示窗口有一个菜单栏，菜单栏只含有一个“文件”菜单项，“文件”菜单下只有一个“退出”命令，展示窗口的背景色为白色。

“选项”选项有 8 个命令选项，含义如下：

- “演示居中屏幕”复选框：选择该选项，不管屏幕有多大，展示窗口都显示在屏幕上居中对齐。
- “显示标题栏”复选框：选择该选项，展示窗口中显示出标题栏。
- “显示菜单栏”复选框：选择该选项，展示窗口中显示出菜单栏，该菜单栏只包含一个“退出”

命令的“文件”菜单。如果您不想显示该含有“退出”命令的菜单，那您必须提供一种用户可以退出的方式。用户总是可以按下快捷键“Ctrl+Q”退出展示窗口。

用户自己创建的一般菜单，包括交互图标和菜单响应，如果没有选择该选项也不会显示出来。

- “显示任务栏”复选框：该选项用来控制 Windows 的任务栏是否显示在展示窗口中。Windows 的任务栏一般显示在屏幕的底部，也可以显示在顶部和两侧。如果用户的任务栏是打开的，它就会占据 Authorware 程序的一部分。当选择该选项，任务栏不出现在展示窗口中，此时“标题栏”复选框灰显，表示该选项不可用。

要显示/隐藏任务栏有以下条件：

- 只有 32 位版本的 Authorware Run-time 应用程序（Runa5w32.exe）才可以控制用户的任务栏。
- 如果要隐藏任务栏，必须同时隐藏标题栏。当打开任务栏时，Authorware 会自动将标题栏显示出来。
- 如果用户将任务栏设置为自动隐藏，则该选项无效。
- 如果展示窗口比用户的显示器的屏幕小，那么无论怎么控制隐藏任务栏，它都会依照用户的窗口设置显示出任务栏。
- “覆盖菜单”复选框：选择该选项，展示窗口中的标题栏会重叠在菜单栏上，使菜单栏暂时不可见。
- “匹配窗口颜色”复选框：选择该选项，用户可以为自己的窗口设置窗口颜色，在程序运行时，用户选择的窗口颜色将覆盖创作时选择的颜色。
- “标准外观”复选框：选择该选项时，运行时的展示窗口为 Windows 的标准外观。
- “Windows 3.1 风格”复选框：选择该选项后，系统弹出一个提示框，如图 1-14 所示。

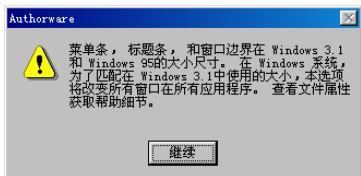


图 1-14 “Windows 3.1 风格”提示框

该提示框说明，Windows 95 与 Windows 3.1 的菜单栏、标题栏和窗口边框都有所不同。如果选择该选项，Authorware 会自动将应用程序中的所有窗口样式改变为 Windows 3.1 样式。单击提示框中的“继续”按钮，可以发现已经选择了“Windows 3.1 风格”复选框。如果要取消该选项，在此单击“Windows 3.1 风格”复选框即可。

2. “交互作用”标签页

单击文件属性对话框中的“交互作用”标签页，即可打开该标签页，如图 1-15 所示。



图 1-15 文件属性对话框的“交互作用”标签页

在“交互作用”标签页中，设置主要针对的是展示窗口的交互特性，包括等待按钮的样式、按钮上的标签、过渡类型以及 Windows 路径和文件名方式等。

- “等待按钮”预览区

在“等待按钮”预览区显示的是等待按钮的形状。如果想改变按钮的风格，可以单击后面的按钮，打开“按钮”对话框，如图 1-16 所示。



图 1-16 “按钮”对话框

在该对话框中，左半部分是按钮的预览窗口，右半部分是该按钮的描述文字。可以通过单击按钮，打开按钮编辑器，对按钮进行编辑。也可以单击按钮对已有的按钮进行编辑。关于按钮的一些操作，我们将在【交互】图标一章讲解。

为了使用等待按钮，必须在【显示】图标或者【交互】图标对话框中的选择显示按钮选项。当用户选择该选项时，当运行程序遇到该图标，Authorware 会显示一个按钮，按钮上有一定的文字标签。

- “标签”文本框

在“标签”文本框中的文字是显示在等待按钮之上的文字。用户可以更改“标签”文本框中的文字，以更改等待按钮的显示文字。

等待按钮会自动调整大小以适应标签上的文字。

- “返回”选项

用“返回”选项表明用户是返回程序开头还是程序离开的地方。“返回”有两个选项：

“继续执行”选项：从程序的开头运行。当 Authorware 重新运行时，它会将所有的自定义变量和系统变量重新赋初值。默认情况下，Authorware 是选择该选项。

“重新开始”选项：从用户跳转的地方重新运行。Authorware 会记录程序跳转处所有的变量值，并将这些信息存储在用户的一个记录文件里。

Authorware 会为 Authorware 的每一块每一个标明创建一个用户记录文件，记录文件的扩展名为 REC。Authorware 会将用户的记录文件保存在“A5w_data”文件夹里（在 Windows 文件夹里）或者保存在用户指定的文件夹里。

- “过渡”选项

使用“过渡”选项，可以选择一个过渡类型。当用户返回这段程序时，会按照过渡类型过渡。关于过渡类型，请参考【显示】图标一章和附录 A。

- “搜索路径”文本框

在“搜索路径”文本框中，键入 Authorware 运行该程序时需要的外部文件地址。

当用户程序链接了一个库或者导入了一个数字电影，Authorware 会记录下外部文件的地址。当要使用该文件时，Authorware 会首先查看该地址。如果在原来地址中找不到该文件时，它会检查 Authorware 所在的文件夹或路径。Authorware 也会使用“搜索路径”文本框中的信息来决定何处查找库文件、数字电影和其他外部存储内容。

如果有多个搜索路径，需要用分号分开，而不要使用空格。比如，如果您希望让 Authorware 首先在 C 盘的根目录下查找外部文件，然后在 D 盘 Authorware 目录下的 Graphic 目录下查找，您应该在“搜索路径”文本框中输入如下格式文字：“C:\;D:\Authorware 5\Graphic”

- “窗口路径”下拉框

选择“窗口路径”中的选项，指明您希望系统函数和变量返回的网络路径的类型。

“窗口路径”下拉框有两个选项：DOS（Drive Based）和 UNC（Universal Naming Convention，通用命名约定）。

到底选择哪个选项，取决于用户程序运行的网络类型。如果用户在只能认识 UNC 路径名（如：运行 Windows NT 的网络）的网络上运行程序，就该选择“UNC”选项；这种命名允许用户指定服务器名称而不必担心盘符的约定。如果用户在不能识别 UNC 命名的网络上运行程序，就请选择“DOS”选项。

只有运行在 Windows 95、Windows 98 或 Windows NT 上的 32 位版本的 Authorware，它的“Windows Paths”设置才与路径系统变量有关。在 16 位版本的 Authorware Run-Time 应用程序中，无论如何设置，返回路径名的变量和函数总是返回基于盘符（DOS，Drive-based Pathnames）的路径名。

系统函数不受“Windows Paths”设置和路径类型变量的影响，只有 3 个系统变量有影响。3 个系统变量是：FileLocation，OrigWorkingDirectory 和 RecordsLocation。

- “窗口名称”下拉框

运用“窗口名称”下拉框中的选项来选择文件名格式。

“窗口名称”下拉框中有两个选项：“DOS（8.3）”选项和“长文件名”选项。如果您希望系统函数和变量返回 DOS 文件类型文件名，即文件名中最多只允许有 8 个字符，其中有 3 个字符是扩展名，选择“DOS（8.3）”选项；如果您希望系统函数和变量返回长文件名，即最多允许有 255 个字符的文件名，选择“长文件名”选项。

如果用户的程序只运行在 Windows 95，Windows 98 和 Windows NT 上，您就选择“长文件名”选项；如果用户程序可能会运行在 Windows 3.1 上，或者您不知道用户运行的 Windows 版本，您最好选择标准的 DOS 命名规则，选取“DOS（8.3）”选项。

只有运行在 Windows 95，Windows 98 或 Windows NT 上的 32 位版本的 Authorware，它的“窗口路”设置才与文件名类型系统变量有关。在 16 位版本的 Authorware Run-Time 应用程序中，无论如何设置，返回文件名的变量和函数总是返回 DOS 8.3 类型的文件名。

受到窗口名称设置和文件名类型影响的系统变量只有文件名称，系统函数只有 Catalog。

3. “管理教学”标签页

单击文件属性对话框中的“CMI”标签页，即可打开该标签页，如图 1-17 所示。

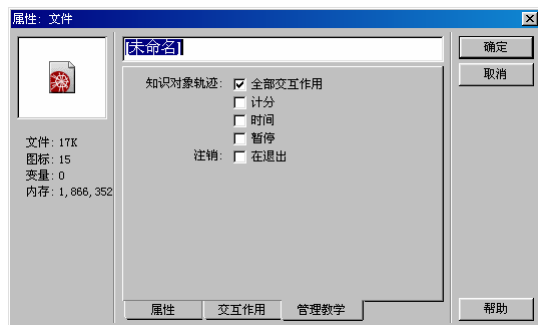


图 1-17 文件属性对话框中的“管理教学”标签页

在“管理教学”标签页中的设置主要是针对计算机管理教学的一些设定。

- “知识对象轨迹”选项：该选项是用来设置使用知识跟踪的跟踪内容。
 “全部交互作用”选项：选择该选项，可以跟踪所有的交互。
 “计分”复选框：选择该选项，可以跟踪学生的成绩。
 “时间”复选框：选择该选项，可以跟踪学生的使用时间。
 “暂停”复选框：选择该选项，可以跟踪学生的使用时间是否超时了。
- “注销”选项：设置退出登录的一些信息。
 “在退出”复选框：选择该选项，离开应用程序即退出登录。

1.3.2 在展示窗口中使用网格

如果希望展示窗口中的物体能够精确定位，可以在使用展示窗口中使用网格。

选择菜单[查看]→[显示网格]命令，可以在展示窗口中显示出网格，如图 1-18 所示。

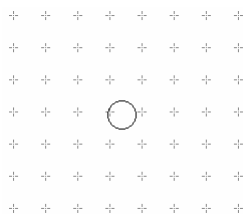


图 1-18 在展示窗口中显示网格

在展示窗口中，可以移动其中的图形对象，使之沿着网格对齐排列。

如果选择菜单[查看]→[对齐网格]命令，可以实现展示窗口中的显示对象自动对齐网格。

第2章 【显示】图标

- 【显示】图标简介
- 导入外部图形
- 绘图工具箱
- 图片遮盖模式

2.1 【显示】图标简介

在 Authorware 中，【显示】图标是使用频率最高的设计图标。画面是否有创意是多媒体产品是否吸引人的关键，而多媒体产品中的动画和图形画面是否吸引人是最重要的因素之一。各种各样的图形处理软件如：Photoshop，CorelDraw，Painter 等都可以制作出非常漂亮的图形图片，而 Authorware 不可能如同图形处理软件一样，可以处理各种各样的图形文字，但它提供了专门处理图形的工具图标——【显示】图标。它的作用是导入由图形处理软件处理过的图形图片，各种文字等等，然后设置导入的图形图片的样式即可实现作品中加入图形。当然，【显示】图标也可以用自带的绘图工具箱绘制简单的矢量图形元素，比如椭圆、矩形、直线等，还可以对绘制的图形进行填充，对文字属性进行定义等操作。

【显示】图标主要作用有以下几种。

- 导入外部图形文件。
- 绘制图形。
- 引入或直接输入文本。
- 引入 OLE 对象。
- 显示变量值。

1. 加入【显示】图标

在图标栏上将【显示】图标拖动到设计窗口的流程线上的相对位置，即可将【显示】图标加入到程

序中，如图 2-1 所示。

2. 【显示】图标命名

在【显示】图标后面的文字是【显示】图标的名称，要给【显示】图标命名，可以单击一下【显示】图标，此时【显示】图标后面的文字由蓝色底色覆盖，按下“Backspace（退格）”键将默认名称删除，然后输入新名字即可。将上面的【显示】图标名称命名为“背景”，如图 2-2 所示。

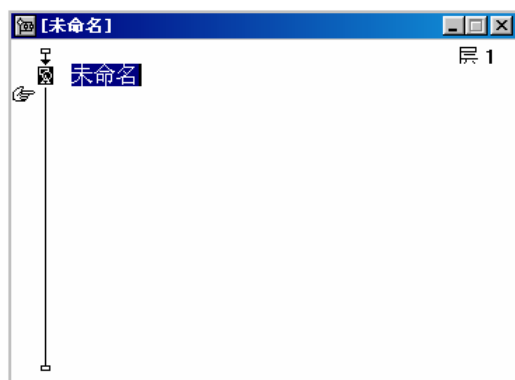


图 2-1 在流程线上添加【显示】图标

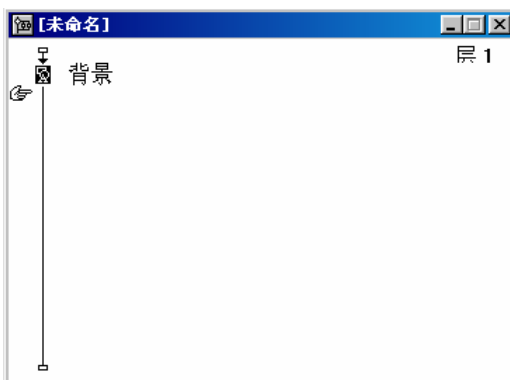


图 2-2 命名【显示】图标

3. 【显示】图标的编辑

双击鼠标左键，打开【显示】图标，此时屏幕上显示出展示窗口。当然您也可以首先单击【显示】图标选取它，然后选择菜单【窗口】→【属性】命令打开展示窗口，如图 2-3 所示。

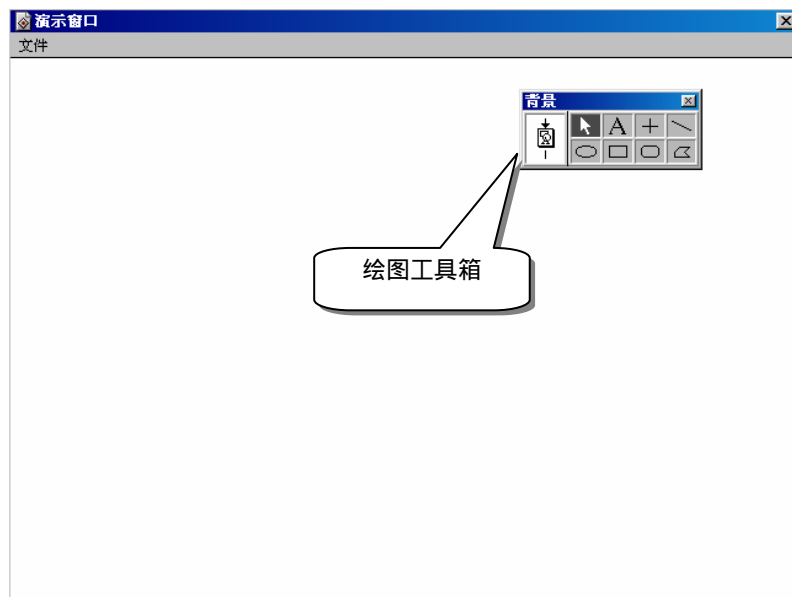


图 2-3 展示窗口

在展示窗口中，可以对【显示】图标中的内容进行编辑。

你也可以使用快捷键“Ctrl+1”打开展示窗口。如果有多个【显示】图标，为了参考上面【显示】图标中的对象，可以先双击打开上面的【显示】图标然后关闭，再分别按下<Shift>键双击打开其他【显示】图标，则上面图标中的显示对象也会同时显示在展示窗口中。

对于流程线上的所有图标，都可以通过双击图标将其打开，进入编辑状态。

4. 绘图工具箱

一般情况下，当打开展示窗口时，同时会弹出绘图工具箱。在绘图工具箱的标题栏上显示的是该【显示】图标的标题，在该绘图工具箱左边的图标显示框中显示的是当前设计图标的图标，如图 2-4 所示。

【显示】图标的编辑，都可以通过绘图工具箱中的各种工具进行设计。



图 2-4 绘图工具箱

关于绘图工具箱的使用，我们将在后一节讲解。

5. 【显示】图标属性对话框

单击【显示】图标选中它，然后选择菜单【修改】→【图标】→【属性】命令或者直接按下快捷键“Ctrl+i”，打开“属性：显示图标”对话框，如图 2-5 所示。

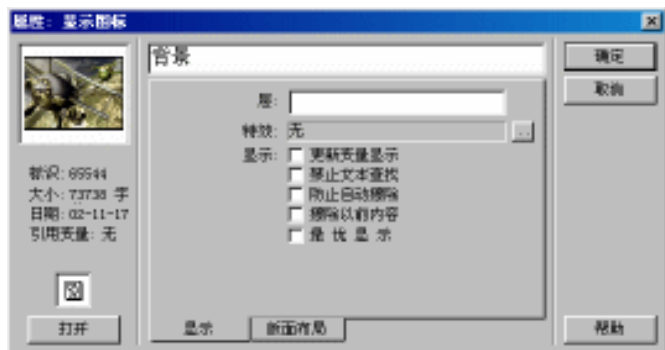


图 2-5 【显示】图标属性对话框

在【显示】图标属性对话框的左侧是该【显示】图标的 ID 号、大小、创建日期等信息；单击“打开”按钮将返回到展示窗口中，以对图像进行编辑。

在【显示】图标属性对话框的中间包括两个标签页：

(1) “显示”标签页

- “层”文本框：该文本框中的数字，表示该【显示】图标所处的图层。用户可以自己在文本框中填入数字。
- “特效”：该选项控制显示【显示】图标中内容时的过渡效果。单击后面的  按钮，可打开“特效”对话框。可从“特效”对话框中选择过渡效果。
- “显示”：共有 5 个选项；
 - “更新变量显示”复选框：选择该选项，可以将在展示窗口中使用的显示变量或者表达式随时更新。
 - “禁止文本查找”复选框：如果选择该选项，在全文检索的文本中，将不包括该【显示】图标中的文本。
 - “防止自动擦除”复选框：选择该选项，系统将不会自动擦除该【显示】图标中的内容。
 - “擦除以前内容”复选框：如果选择该选项，在显示该【显示】图标的内容时，将会将前面显示的内容擦除掉。

“最优显示”复选框：如果选择该选项，将该【显示】图标的内容覆盖其他显示内容，使该【显示】图标的内容显示在最上层。

(2) “版面布局”标签页



图 2-6 【显示】图标属性对话框的“版面布局”标签页

- “位置”下拉框：主要控制【显示】图标在展示窗口中的位置。在该下拉框中有四个选项：

“不改变”：当选择该选项时，【显示】图标中的内容总是显示在当初放置的位置，用户不可以移动。此时，无论选择“可移动性”下拉框中的哪个选项，下面的“出发点”单选按钮、“初始化”单选按钮和“结束点”单选按钮都是灰显的，表示不可以定义显示对象的位置。

“在屏幕上”：当选择该选项时，【显示】图标中的内容可以在整个屏幕移动，但是不可以移动到屏幕之外。此时无论选择“可移动性”下拉框中的哪个选项，下面的“初始值”单选按钮被激活，用户可以在“初始值”单选按钮后面的 X、Y 文本框中输入数值来定义显示对象在屏幕中的起始位置。

“沿特定路径”：如果选择该选项，对话框变为如图 2-7 所示样式。



图 2-7 【显示】图标属性对话框的“沿特定路径”选项

此时下面的三个选项发生了改变，变成显示对象起点到终点的一条直线上的某一点上的位置。“出发点”和“结束点”分别定义了直线的起点和终点，“初始值”决定了显示对象在直线上的初始位置。

“在某个区域中”：该选项定义了显示对象显示的范围。显示范围由下面的“出发点”、“结束点”和“初始值”决定。

- “可移动性”下拉框：该下拉框用来设置显示对象的移动信息，它包括四个选项：

“不能移动”：选择该选项，显示对象不可以在屏幕上移动。该选项为 Authorware 的默认选项。

“在屏幕上”：选择该选项，显示对象可以在屏幕内任意移动，但不可移动到屏幕外。

“在指定区域”：选择该选项，显示对象可以在指定区域内任意移动。移动范围由下面的“出发点”、“结束点”和“初始值”决定。

“任何地方”：选择该选项，显示对象可以随处移动，甚至可以移动到屏幕外。

在下面的“出发点”、“结束点”和“初始值”中可以手动输入【显示】图标的起始位置、终点位置和【显示】图标的初始位置。

2.2 导入外部图形文件

在【显示】图标中有两种方法可以显示图形对象：

- (1) 使用外部导入的编辑好的图形文件。
- (2) 使用绘图工具箱中的工具进行简单的图形对象绘制。

现在我们先讲解第一种方法，稍后讲解第二种方法。

2.2.1 导入外部文件

Authorware 6 可以导入大部分格式的图形文件，包括 WMF 文件、PICT 文件、GIF 文件、JPEG 文件、PNG 文件、Photoshop 文件、TGA 文件、TIFF 文件、EMF 文件、BMP 文件等，还可以导入外部的 TXT 文件和 RTF 文件。

导入图形文件一般操作步骤为：

- (1) 双击【显示】图标，打开展示窗口；

(2) 选择菜单【文件】→【导入】命令或者直接单击工具栏中的  按钮，打开“导入哪个文件”对话框，如图 2-8 所示。



图 2-8 “导入哪个文件”对话框

(3) 在如图 2-9 所示的“导入哪个文件”对话框中找到想要导入的图形文件，按下“导入”按钮执行导入操作。



图 2-9 有预览窗口的“导入哪个文件”对话框

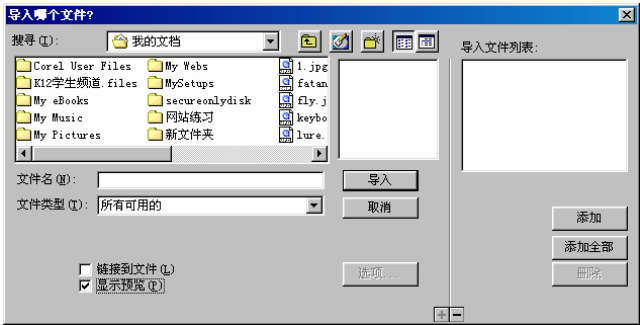


图 2-10 可导入多个图形文件的“导入哪个文件”对话框

2.2.2 “图像属性”对话框

双击导入的图形对象或选择菜单【修改】→【图像属性】命令，打开图片属性对话框，如图 2-11 所示。

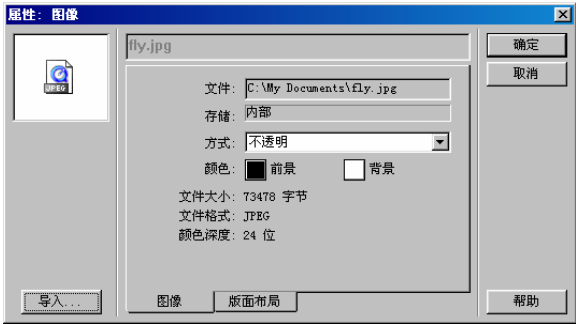


图 2-11 图片属性对话框

单击图片属性对话框左下角的“导入”按钮，可以打开“导入哪个文件”对话框，让用户重新选择导入的图形文件。

图片属性对话框包含两个标签页：

1. “图像”标签页

- “文件”：显示的是导入的外部图形文件的绝对地址。

- “存储”：该图形文件在应用程序中的存储方式，主要有两种存储方式：“内部”方式和“外部”方式。“内部”方式是将其图形文件作为内部文件，所以即使删除了外部的该图形文件，也不会影响程序；“外部”方式是建立与外部文件的连接，并不将该图形文件放入到应用程序中。这样可以减少应用程序的体积。
- “方式”：显示模式。一共有 6 种显示模式，关于图形的显示模式，请参看 2.4 节。
- “颜色”：色彩设置。包括两个选项：
 - “前景”：前景色，通过单击前面的  按钮打开“颜色”对话框，从对话框中选取前景色。
 - “背景”：背景色，通过单击前面的  按钮打开“颜色”对话框，从对话框中选取背景色。
- “文件大小”：该图形文件的大小。
- “文件格式”：该图形文件的类型。
- “颜色深度”：该文件的色彩表达能力。

2. “版面布局”标签页

在如图 2-11 所示的图片属性对话框中单击“版面布局”标签页，打开如图 2-12 所示的“版面布局”标签页。



图 2-12 图片属性对话框的“版面布局”标签页

- “选项”下拉框：控制图片的显示类型，它包括三个选项：
 - “比例”：当选择该选项时，图片可以任意缩放。
 - “裁切”：当选择该选项时，图片的大小不变，当调整图片时，相当于对图片进行剪切。
 - “原始”：当选择该选项时，图片将保持原来大小。如果要改变图片的大小，系统将弹出一个如图 2-13 的提示框，询问用户是否真的改变图片大小。如果单击“确定”按钮，图片的显示类型自动改变为“比例”类型。

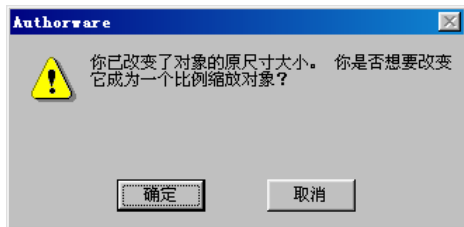


图 2-13 改变图片大小的提示框

- “位置”：在这两个文本框中分别显示出图片在 X、Y 方向上的位置。如果用户要改变图片的

位置，可以直接在展示窗口中直接拖动图片到满意位置，也可以在这两个文本框中输入图片的位置以改变图片位置。

- “大小”：在这两个文本框中分别显示出图片在 X、Y 方向上的大小。如果用户要改变图片的大小，可以直接在展示窗口中直接拖动图片改变图片大小，也可以在这两个文本框中输入图片的大小数值以改变图片大小。
- “非约束比例大小”：在这两个文本框中显示出缩放前图片的原来大小。
- “比例%”：在这两个文本框中分别显示出图片在 X、Y 方向上的缩放比例。

2.3 使用绘图工具箱绘制简单图形

Authorware 提供了一些基本的绘图工具，你可以用它们来完成基本的直线、矩形、圆、多边形等图形的绘制。

要使用绘图工具箱，可以先创建一个新文件，拖动【显示】图标到流程线上，把它命名为“图片”，如图 2-14 所示。

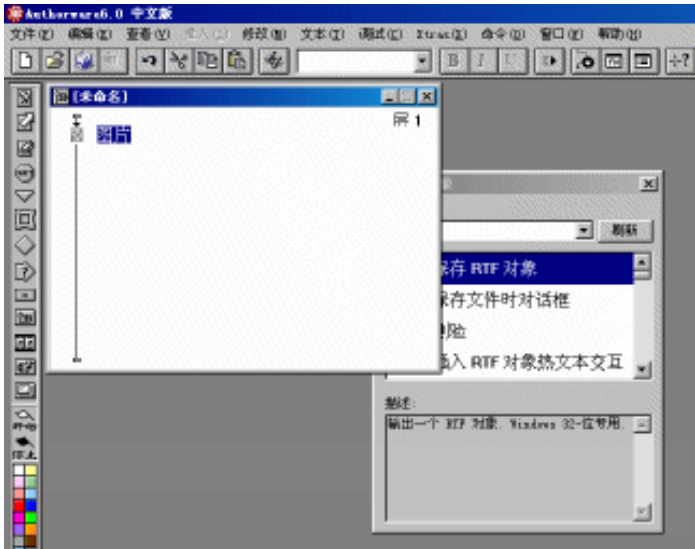


图 2-14 创建一个新文件

如果【显示】图标中还没有内容，则该图标灰显，如图 2-14 所示。其他图标如果没有内容也一样灰显。

双击显示图标，屏幕出现该图标的演示窗口，同时屏幕右上角出现一个含有几种绘图工具的绘图工具箱，其标题即是我们为显示图标命名的图标名“图片”。

绘图工具箱如图 2-15 所示。



图 2-15 绘图工具箱

2.3.1 指针工具

用鼠标单击指针工具，则该工具块高亮显示，如图 2-16 所示。



图 2-16 选择指针工具

指针工具在这里充当选择工具，它与设计按钮图标显示框不同，双击显示图标可以选中展示窗口中所有的对象，而指针工具只能选中一个对象。如图 2-17 所示。

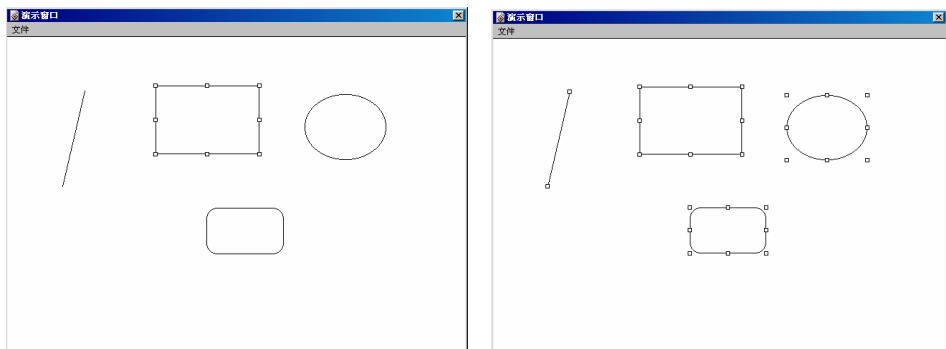


图 2-17 指针工具选中对象（左）与双击显示图标选中对象（右）的区别

当选择多个对象时，可以用指针工具先选择一个对象，然后按住 shift 键，再单击其他对象，这样就可选择多个对象。

指针工具另一个选择多个对象的方法是区域选择对象。单击鼠标并拖动，出现一个虚矩形框，当松开鼠标时，位于矩形框中的所有对象都被选中了。如图 2-18 所示。

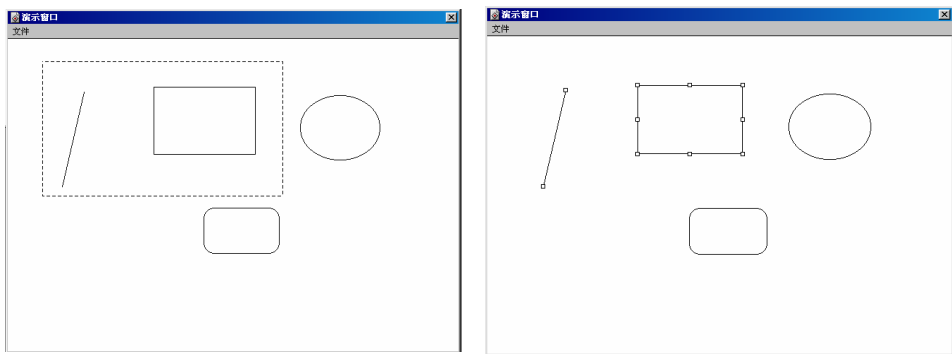


图 2-18 区域选择：圈定一个区域（左），选定的对象（右）

2.3.2 文字工具

用鼠标单击文字工具，该工具块高亮显示。将鼠标移到展示窗口中，鼠标指针形状变为“ I ”

形光标。在你想输入文字的地方单击鼠标，会出现图 2-19 所示的画面，其中两端带有黑色三角箭头和句柄的直线称为“缩排线”。在缩排线下方为文字输入区域，光标是文字输入的起始位置。

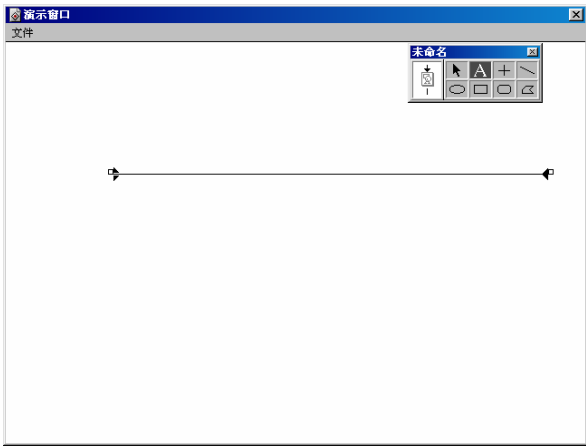


图 2-19 文字输入

在屏幕上输入“欢迎来到多媒体世界”。利用缩排线可以定义文字的起始位置和终止位置。如图 2-20 所示。

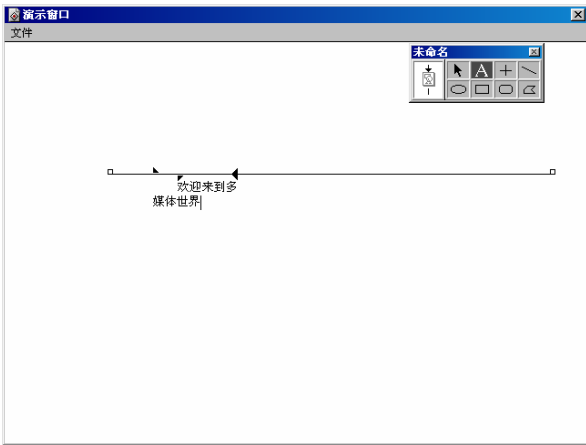


图 2-20 文字设置

其中，缩排线左边下面的三角形是控制首行缩进，上面的三角形是左缩进，右边的三角形是右缩进。这与 Word 中相似。

单击绘图工具箱中的指针工具即可取消文字输入状态。此时文字周围出现了六个句柄。同样，用指针工具可以调整文字的长度，但高度不可调整。要移动文字，单击句柄中间部分拖动到满意位置即可。

现在我们看一下如何设置字体样式。

1．设置字体类型。

选择菜单【文本】→【字体】选项，将弹出图 2-21 所示的二级菜单，在二级菜单中列出了用过的几种字体。当前所用字体的前面打了一个对号。如果你想选择更多的字体，选择【其他】选项，将会为你提供更多的字体。

选择【其他】选项，弹出“字体”对话框，如图 2-22 所示。“字体”下拉菜单中列出了系统中所

有的字体类型。从“字体”的下拉菜单中选择自己喜欢的字体，如“宋体”，下边的编辑框显示出选择字体后字体的样式。单击确定后，展示窗口中的字体变为宋体。



图 2-21 字体设置菜单



图 2-22 【字体】对话框

2. 设置字体大小

选择【文本】→【大小】选项,弹出其二级菜单,在二级菜单中列出了可选择的字体大小,其中打对号的是当前字体大小。如图 2-23 所示。同样,选择【其他】选项也可以自己定义字体大小。



图 2-23 字体大小设置菜单

选择【其他】选项,弹出“字体大小”对话框,如图 2-24 所示。在“字体大小”空白框中输入你要求的字体大小,如“20”。下边的编辑框显示出此时字体的大小。单击确定后,展示窗口的字体大小变为 20。

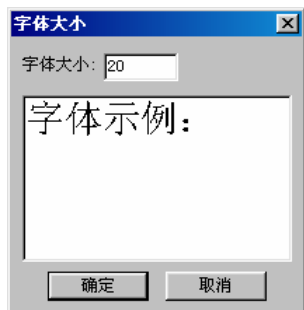


图 2-24 字体大小对话框

3. 设置字体风格

选择【文本】→【风格】选项,弹出其字体风格二级子菜单,如图 2-25 所示,在二级子菜单中列出了几种字体风格,如正常、粗体、斜体、下划线、上标、下标等等。根据需要选择字体风格。例如我们选择“粗体”。



图 2-25 字体风格设置菜单

快捷的方法是直接选取常用工具栏中的    等按钮来设置字体风格。

4. 设置字体对齐方式

选择菜单【文本】→【对齐】选项,弹出二级字体对齐方式子菜单,如图 2-26 所示。其子菜单中列出了几种对齐方式,如左齐、居中、右齐、正常等方式。其中,“左齐”、“右齐”、“居中”方式与 Word 中相似。“正常”是我们利用缩排线自己调整。由于开始时我们就是用缩排线自己调整的,因此【对齐】子菜单中“正常”前打了对号。现在我们选择“居中”。文字变成图 2-27 所示效果。

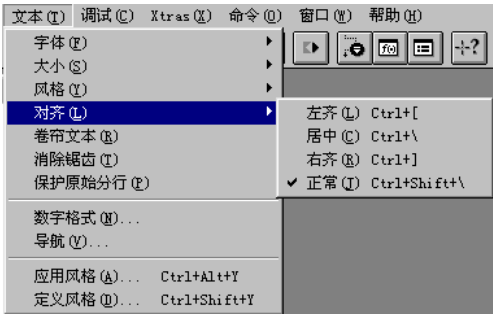


图 2-26 字体对齐方式

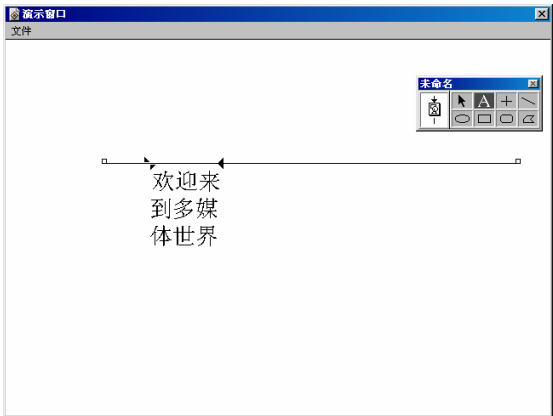


图 2-27 字体居中对齐

但是这并不是我们所希望的。我们想把“欢迎来到多媒体世界”几个字位于一行并把它们放在中间。此时我们需要改变缩排线位置。

在绘图工具箱中选中文字工具,单击文字部分,缩排线出现,此时可以对文字进行编辑。我们不需要对文字进行编辑,只改变缩排线的位置。改成图 2-28 所示。此时文字已经居中了。

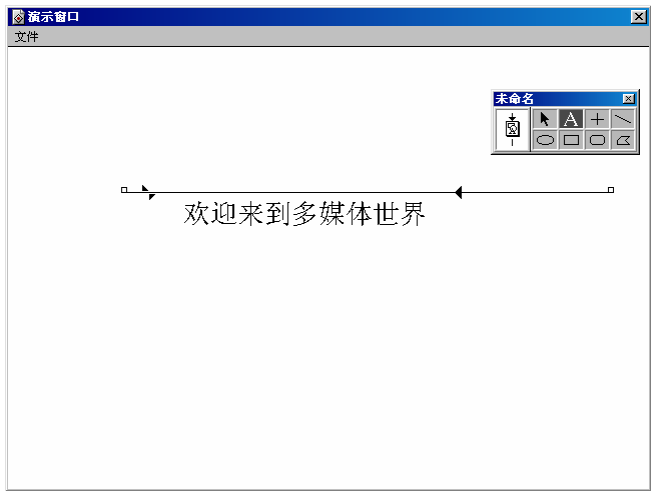


图 2-28 改变缩排线位置

5. 设置文字是否有滚动条

选择菜单【文本】→【卷帘文本】选项，可以为文字加上滚动条，如图 2-29 所示。

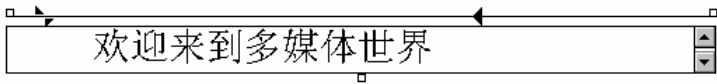


图 2-29 为文字加上滚动条

6. 设置文字具有抗锯齿功能

选择菜单【文本】→【消除锯齿】选项，可以为字体加上抗锯齿功能，如图 2-30 所示。将文字设置为抗锯齿文字，可以使文字变得更加平滑、更加美观。



图 2-30 文字没有加上抗锯齿功能（上）和加上抗锯齿功能（下）

2.3.3 直线工具

直线工具用于绘制水平线、垂直线以及任何 45°倍数角方向的直线。

1. 绘制直线

单击绘图工具箱中的直线工具，然后在展示窗口中单击左键并拖动，展示窗口中出现一条直线，

如图 2-31 所示。

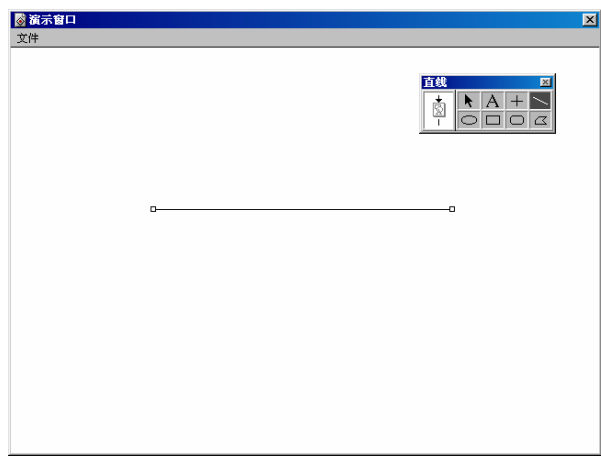


图 2-31 绘制直线

2. 设置线型

选择菜单【窗口】→【显示工具箱】→【线】选项，系统弹出线型线宽设置快捷工具板，如图 2-32 所示。



图 2-32 线型线宽工具板

在线型线宽工具板中分为两部分，上面部分为线宽设置，下面部分为线型设置，中间由一个横框隔开。可以直接双击绘图工具箱中的直线工具或斜线工具，打开线型线宽工具板。

选中绘制好的直线，然后在线型线宽工具板中选择合适的线型线宽，即可改变选中的直线。

2.3.4 斜线工具

斜线工具用于绘制任意方向的直线。如果同时按住<Shift>键，可以绘制水平线、垂直线以及任何 45°倍数角方向的直线，此时与直线工具相同。

单击绘图工具箱中的斜线工具，然后在展示窗口中单击左键并拖动，展示窗口中出现一条直线。同样，可以移动和改变直线，也可以通过线型线宽工具板来改变绘制直线的线型线宽。

2.3.5 椭圆工具

椭圆工具用于绘制椭圆和标准圆形。

1. 绘制椭圆和圆

选择绘图工具箱中的椭圆工具, 然后在展示窗口中, 此时鼠标指针变为“十”字形状, 按住鼠标左键, 拖动鼠标, 展示窗口中出现一个随着鼠标位置变化的椭圆, 等到合适位置释放鼠标, 绘制好的椭圆即可出现, 如图 2-33 所示。如果在绘制椭圆的同时按住<Shift>键, 可以绘制出标准的圆形。

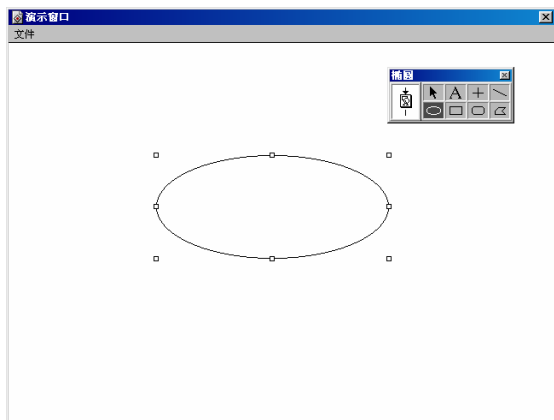


图 2-33 绘制椭圆

2. 移动和改变椭圆

如果要移动椭圆, 可以选中要改变的椭圆, 此时椭圆周围出现八个句柄, 点住句柄外椭圆的边, 然后拖动即可移动椭圆。

如果要改变椭圆的形状, 用鼠标点住句柄并拖动, 即可改变椭圆形状; 向外拖动可以使椭圆变大, 向内拖动可以使椭圆变小。

3. 设置椭圆的线宽

先选中要改变线宽的椭圆, 然后打开线型线宽工具板, 在工具板中选取合适的线宽即可。

4. 设置填充模式

选择菜单【窗口】→【显示工具箱】→【填充】选项 (CTRL+D), 打开填充模式工具板, 如图 2-34 所示。



图 2-34 填充模式工具板

要改变图形对象的填充模式, 可以先选择该对象, 然后在填充工具板中选择合适的填充模式, 即可改变该图形对象的填充模式。可以直接双击绘图工具箱中的矩形工具、圆角矩形工具或多边形工具, 打开线型线宽工具板



2.3.6 矩形工具

矩形工具用于绘制一般的矩形和标准的正方形。

1. 绘制矩形和正方形

选取绘图工具箱中的矩形工具，然后将鼠标移到展示窗口中，此时鼠标指针变成“十”字形，将鼠标移到矩形的起始位置，按下鼠标左键并拖动，此时屏幕上出现一个随着鼠标移动而变化的矩形，等到合适时松开鼠标，一个矩形就绘制成了。

2. 移动和改变矩形

选中指针工具，然后点取要移动的矩形，此时矩形周围出现八个句柄。点中句柄以外的部分并拖动，将矩形移到合适的位置。

要改变矩形的大小，可以点中矩形的句柄并拖动，向外拖动可以将矩形变大，向内拖动可以将矩形变小。

3. 设置矩形的线宽

先选中要改变线宽的矩形，然后打开线型线宽工具板，在工具板中选取合适的线宽即可。

4. 设置矩形的填充模式

同样，可以通过填充工具板来改变矩形的填充模式。

2.3.7 圆角矩形工具

圆角矩形工具用于绘制边角是圆角的矩形和正方形。

1. 绘制圆角矩形和圆角正方形

选取绘图工具箱中的圆角矩形工具，然后将鼠标移到展示窗口中，此时鼠标指针变成“十”字形，将鼠标移到圆角矩形的起始位置，按下鼠标左键并拖动，此时屏幕上出现一个随着鼠标移动而变化的圆角矩形，等到合适时松开鼠标，一个圆角矩形就绘制成了。

2. 移动和改变圆角矩形

选中指针工具，然后点取要移动的圆角矩形，此时圆角矩形周围出现八个句柄。点中句柄以外的部分并拖动，将圆角矩形移到合适的位置。

要改变圆角矩形的大小，可以点中圆角矩形的句柄并拖动，向外拖动可以将圆角矩形变大，向内拖动可以将圆角矩形变小。

3. 设置圆角矩形的线宽

先选中要改变线宽的圆角矩形，然后打开线型线宽工具板，在工具板中选取合适的线宽即可。

4. 设置圆角矩形的填充模式

同样，可以通过填充工具板来改变圆角矩形的填充模式。

5. 改变圆角矩形圆角的弧度

选取指针工具，选中圆角矩形，此时圆角矩形周围出现八个句柄，然后单击绘图工具箱中的圆角矩形工具，此时，圆角矩形周围的八个句柄被接近左上角的一个句柄代替，如图 2-35 所示。

用鼠标左键单击此句柄并拖动，可以发现圆角的弧度在改变。当向左上角移动时，圆角的弧度越小，直至圆角矩形成为一个矩形；当向内部移动时，圆角的弧度越大，直至圆角矩形变成一个椭圆。

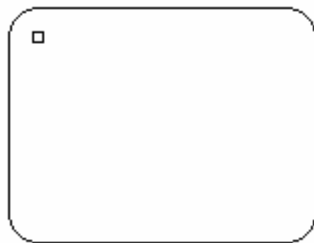


图 2-35 调整圆角矩形圆角的弧度

2.3.8 多边形工具

多边形工具用于绘制各种多边形。

1. 绘制多边形

选取绘图工具箱中的多边形工具，然后将鼠标移到展示窗口中，此时鼠标指针变成“十”字形状，将鼠标移到多边形的起始位置，按下鼠标左键，确定多边形的起始点，移动鼠标，此时屏幕上出现一个随着鼠标移动而变化的直线，等到合适时在按下鼠标左键，绘制成多边形的一条边；再移动鼠标，屏幕上又出现一条随着鼠标移动而变化的直线，等到合适时在按下鼠标左键，就又绘制成多边形的一条边；……要结束多边形的绘制，双击鼠标左键即可。

绘制的多边形可以是不封闭的。如果要将多边形形成封闭的多边形，在绘制多边形的过程中将鼠标移回起始点位置，然后单击左键确认即可。

2. 移动和改变多边形

选中指针工具，然后点取要移动的多边形，此时该多边形周围出现八个句柄。点中句柄以外的部分并拖动，将多边形移到合适的位置。

要改变多边形的大小，可以点中多边形的句柄并拖动，向外拖动可以将多边形变大，向内拖动可以将多边形变小。

3. 设置多边形的线宽

先选中要改变线宽的多边形，然后打开线型线宽工具板，在工具板中选取合适的线宽即可。

4. 设置多边形的填充模式

同样，可以通过填充工具板来改变多边形的填充模式。

2.4 【显示】图标其他功能

2.4.1 设置图形对象的显示模式

当多个图形、图片相互重叠时，可以通过显示模式工具板来改变图形、图片的显示模式来改变其重叠部分的显示效果。

选择菜单【窗口】→【显示工具箱】→【模式】选项，打开显示模式工具板，如图 2-36 所示。

各种显示模式介绍如下：

- 不透明：不透明显示模式。当选中该模式时，位于选中的图形图片会遮盖下面的图形图片。
- 遮隐：遮隐模式。当选中该模式时，选中对象边缘的白色会被遮蔽掉。
- 透明：透明模式。当选中该模式时，选中的图形图片中的白色部分变成透明的，位于该图形图片下面对象会通过白色部分显现出来。
- 反转：反转模式。当选中该模式时，选中的图形图片会与其下面的对象的重叠部分以互补色显示。
- 擦除：擦除模式。当选中该模式时，选中的图形图片和与下面对象重叠部分被擦除。
- 阿尔法：阿尔法通道模式。当选中该模式时，选中的图形图



图 2-36 显示模式工具板

片与下面的对象的重叠部分为阿尔法通道效果的重叠，如图 2-37 所示。

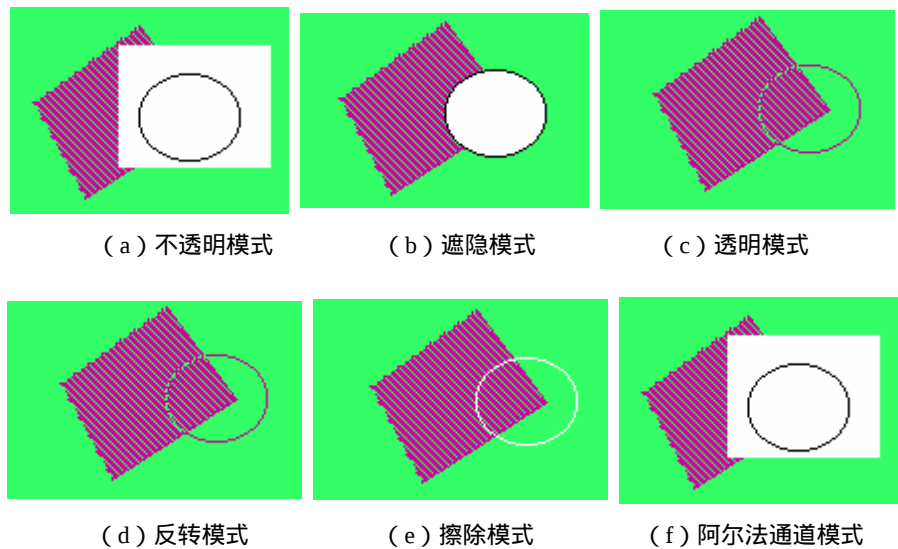


图 2-37 各种显示模式

2.4.2 设置图形对象的颜色

选择菜单【窗口】→【显示工具箱】→【颜色】选项，打开调色板，如图 2-38 所示。

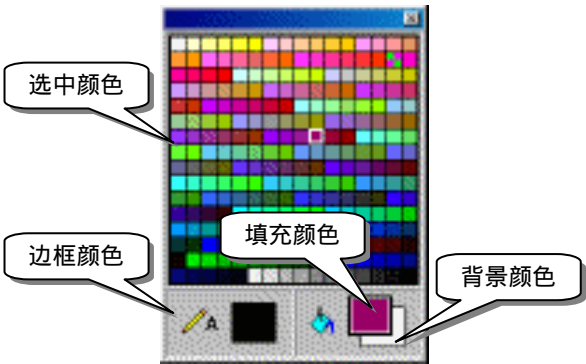


图 2-38 调色板

调色板分为三个部分，上面是颜色列表，用来设置颜色；下面左边为设置选中图形的边框颜色，同时也是文本所用的颜色；下面右边分别用来设置选中图形的填充颜色和背景颜色。

要设置选中图形的边框颜色，先选中该图形，然后打开调色板，点中边框颜色设置区的颜色块，在上面的颜色列表中选取合适的颜色即可。同样，如果要改变选中图形的填充颜色或背景颜色，只要点中填充颜色块或背景颜色块，然后在上方的颜色列表中选取合适颜色即可。

2.4.3 设置图形对象的对齐方式

选择菜单【修改】→【排列】选项，打开对齐方式控制板，如图 2-39 所示。

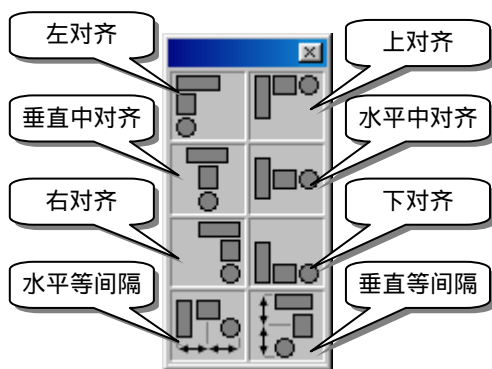


图 2-39 对齐方式控制板

假设展示窗口中有几个图形，如图 2-40 所示，选中其中一个图形，然后按住<Shift>键将其他图形全部选中或者直接双击绘图工具箱中的图标显示框选中所有图形，然后单击对齐方式控制板中的左对齐方式，所有选择的对象按照最上面对象的左边缘为准对齐排列，如图 2-41 所示。

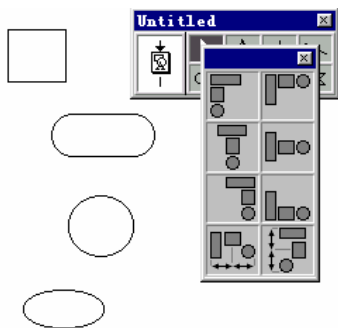


图 2-40 使用对齐方式前

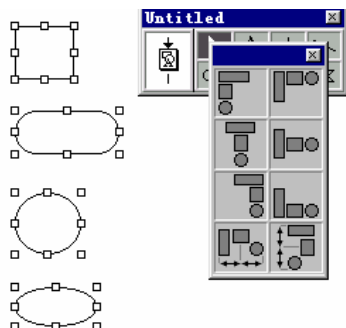


图 2-41 左对齐方式

2.4.4 组合/解除组合图形对象

当某些图形对象是一个整体时，往往由于不小心会将其中的某个部分给移动或改变了，需要重新进行组合，这是非常麻烦的事。虽然 Authorware 提供了“撤消”命令，但其只能撤消一次，这也是我觉得 Authorware 最大的缺点。现在我们可以通过“组合”命令，将几个简单图形组成为一个整体，这样就不怕由于不小心而破坏其完整性了。

首先，按住<Shift>键将要组合的图形全部选中，然后选择菜单【修改】→【群组】选项，将选中的图形对象进行组合，如图 2-42 所示。

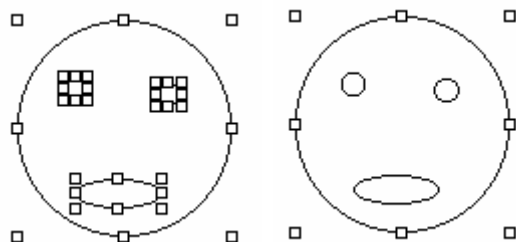


图 2-42 组合前的图形（左）和组合后的图形（右）

如果要对组合后的对象进行局部修改，必须先对其进行解除组合。首先选中要解除组合的图形，然后选择菜单【修改】→【取消群组】选项，将选中的图形对象解除组合；最后再对要修改的部分进行修改。

2.4.5 设置重叠图形的相对位置

在展示窗口中，如果几个图形有重叠的部分，除了可以用显示模式工具板来控制其显示模式外，还可以设置重叠图形的相对位置。

菜单【修改】→【置于上层】选项，可以将处于下面的图形图片放在最上边，图 2-43 是选中圆形后，执行菜单【修改】→【置于上层】命令将其放在最上边。

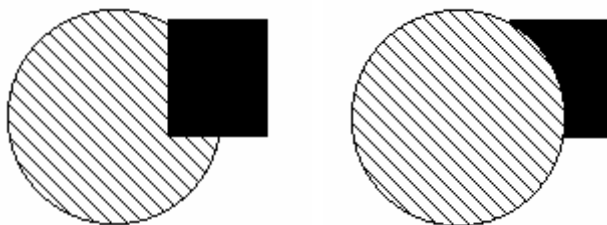


图 2-43 执行“置于上层”命令

菜单【修改】→【置于下层】选项，可以将处于上面的图形图片放在最下边，图 2-44 是选中矩形后，执行菜单【修改】→【置于下层】命令将其放在最下边。

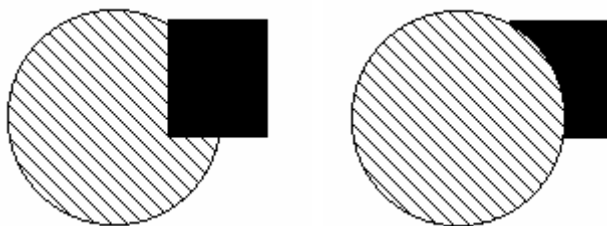


图 2-44 执行“置于下层”命令

2.4.6 设置显示过渡效果

在多媒体的作品中，通常需要使用大量的图片和动画，以吸引用户的兴趣。在 Authorware 中，【显

示】图标、【擦除】图标、【框架】图标和【交互】图标都可以设置一些特殊的动画效果。Authorware 作为一个功能强大的多媒体制作平台,为用户提供了大量的过渡效果功能。通过 Authorware 提供的 Xtras 可以实现窗帘、马赛克等常见的显示效果。Authorware 6.0 提供了大量的 Xtras 插件,而且第三方软件商也提供了很多附加的过渡效果, Authorware 的显示功能可以通过 Xtras 的方式来扩展。

1. 打开过渡效果对话框

打开过渡效果对话框有两种方法:

- 通过菜单命令直接打开过渡效果对话框

在设计窗口中的流程线上选择要实现过渡效果的设计图标,然后选择菜单【修改】→【图标】→【特效】命令,打开“特效方式”对话框,如图 2-45 所示。

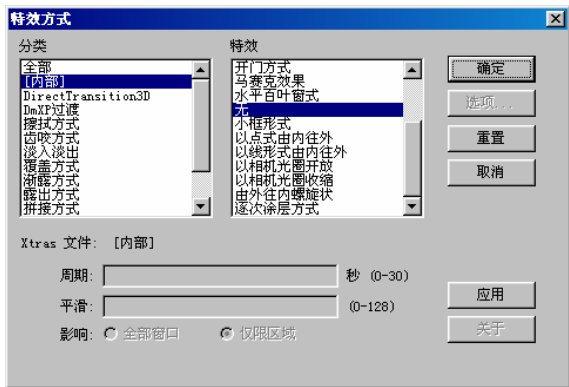


图 2-45 “特效方式”对话框

- 通过设计图标的属性对话框中的“特效”按钮打开过渡效果对话框

具有过渡效果的设计图标有【显示】图标、【擦除】图标、【框架】图标和【交互】图标。在设计窗口的流程线上选择要设置过渡效果的设计图标,然后选择菜单【修改】→【图标】→【属性】命令,打开该按钮的属性对话框,单击属性对话框中的“特效方式”按钮打开过渡效果对话框。

2. 特效设置

打开“特效方式”对话框,如图 2-46 所示。

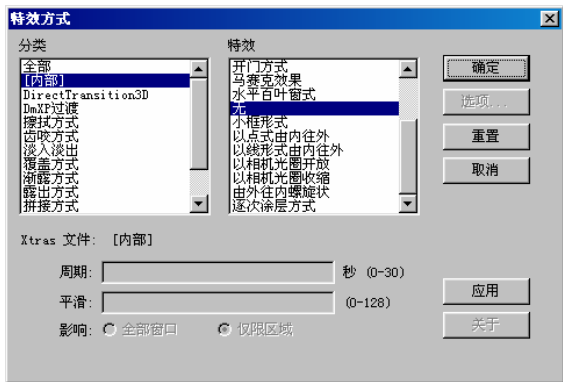


图 2-46 特效方式对话框

下面我们对该对话框中的各个选项分别予以介绍。

- “分类”列表框:过渡效果类别列表。
- “特效”列表框:过渡效果列表。

我们可以在过渡效果类别列表中选择一个过渡类别,然后再在该列表中可以选择合适的过渡效果,

并将该过渡效果应用到所选的设计图标上。

我们可以在“特效”列表框选择“无”选项，以使显示对象不使用任何过渡效果，而是将显示对象从展示窗口中直接显示或擦除。

- “Xtras 文件”选项：该选项显示了所选择的过渡效果的文件名及其所在目录位置，供开发人员在最终发行作品时参考。

如果是 Authorware 提供内部过渡效果，该信息区域显示的是“内部”。

对于“Xtras 文件”信息，用户可将所用到的 Xtras 文件的位置和文件名记录在自己的工作目录中，为最终作品的发行做准备。最终发行作品时，该 Xtras 必须同 Authorware 程序同时发行，并放置到合适的目录中。

- “周期”文本框：设置当前选中的过渡效果的持续时间。可以在文本框中直接输入数值、变量和数值型表达式来设定完成过渡效果所需的时间。用户可以使用 Authorware 中 Xtras 默认的持续时间。在该正文输入框中输入的数值、变量和数值型表达式的值最大不超过 20。
- “平滑”文本框：设置当前选中的过渡效果的平滑度。可以在文本框中直接输入平滑数值。平滑数值越小，过渡效果越平滑，持续时间越长。0 表示最平滑的过渡。
- “影响”选项：设置过渡效果作用的范围，它包括两个子选项：
“全部窗口”单选钮：选择该选项时，过渡效果作用于整个展示窗口。
“仅限区域”单选钮：选择该选项时，过渡效果只作用于显示对象所在的区域。
- “选项”按钮：单击该按钮，可以打开与当前过渡效果相关的附加选项设置，如选择“SharkByte Transitions”过渡类别时，该按钮被激活，单击该按钮，弹出附加选项设置对话框，如图 2-47 所示。



图 2-47 附加选项设置对话框

在附加选项设置对话框中，可以根据提示设置附加选项。

- “重置”按钮：单击该按钮，将选定的过渡效果的所有参数设置为 Authorware 的默认方式。当再次使用过渡效果对话框时，此时设定的默认方式会自动出现。
- “应用”按钮：单击该按钮，可以在展示窗口中预览设定好的过渡效果。如果感觉不合适，可以重新进行设置，然后再预览，直至满意为止。
- “关于”按钮：单击该按钮，可以打开一个附加信息对话框。如依然选择“SharkByte Transitions”过渡类别时，该按钮被激活，单击该按钮，弹出附加信息对话框，如图 2-48 所示。

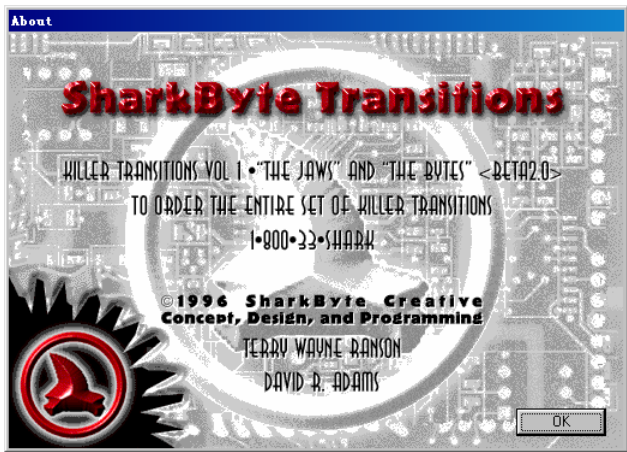


图 2-48 附加信息对话框

附加信息对话框中一般是创作者的一些信息，如地址、电话、制作时间、E-mail 地址等。

2.4.7 自定义文本风格

在多媒体作品中，一般都会有大量的文本需要编排处理。而不同文本风格的使用，可以为作品润色不少。而使用一些特殊的文本风格，如“注意”、“提示”、“技巧”等，以搭配其他的文本方式，体现出多媒体作品的风格统一。

在 Authorware 中，只提供了一种默认的文本风格“默认风格”。不过 Authorware 提供了定义和使用文本风格的功能，所以我们可以定义一个自己的文本风格，以满足程序的要求。

在定义文本风格时，需要设置文本的字体、字号、字色、字体格式等信息，并且我们可以为自定义的文本风格命名。

1. 定义一种文本风格

选择菜单【文本】→【定义风格】命令，打开“定义风格”对话框，如图 2-49 所示。

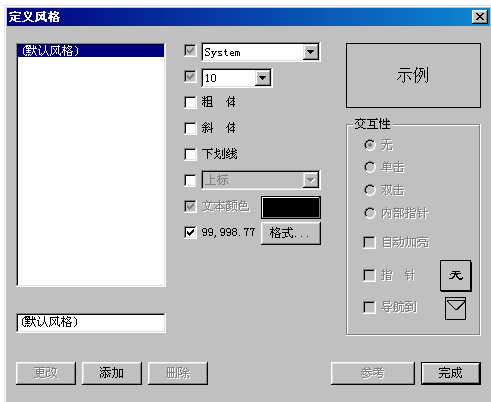


图 2-49 定义文本风格对话框

下面我们以一个例子来说明如何定义一种文本风格。

- (1) 单击对话框中的“添加”按钮，增加一种文本风格。
- (2) 此时，对话框的左侧的大文本风格列表框中增加了一个新的名称为“新样式”文本风格，同



时列表框下面的文本框中也显示出新增加的文本风格名称。

(3) 在下面的文本框中，将新增的文本风格名称改为“正文”，然后回车即可将上面的文本风格列表框中新增的文本风格名称改为“正文”。

当然，您也可以在下面的文本框中输入“正文”后，此时“更改”按钮被激活，单击该按钮，也可以将上面的文本风格列表框中新增的文本风格名称改为“正文”。

(4) 在文本风格列表框中选中“正文”格式，此时您可以发现，该对话框中的其他选项也被激活了。

(5) 选中“字体下拉框”复选框，然后在字体下拉框中选择“楷体_GB2212”字体。

(6) 选中字体大小复选框，然后在字体大小对话框中选择合适字体大小。如果下拉框中的所有选项都不合适，则可以选择下拉框中的“其他”选项，此时系统会弹出字体大小对话框，如图 2-26 所示。我们可以在字体大小对话框中输入合适的字体大小，然后单击“确定”按钮，即可将该字体大小添加到字体大小下拉框中，并且当前选中该字体大小。

(7) 如果希望字体为粗体，则可以选中“粗体”复选框。

(8) 如果希望字体为斜体，则可以选中“斜体”复选框。

(9) 如果希望字体带有下划线，则可以选中“下划线”复选框。

(10) 如果选中“上标”复选框，将激活“上标”下拉框，该下拉框中有两个选项：“上标”选项和“下标”选项。

(11) 选中“文本颜色”复选框，然后单击后面的  按钮，打开“文本颜色”调色板，如图 2-50 所示。我们可以在“文本颜色”调色板选择一种颜色，来作为新建文本风格的字色。

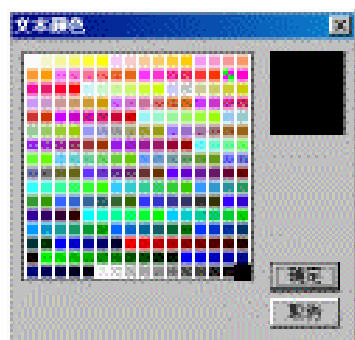


图 2-50 “文本颜色”调色板

(12) 选中“99,998.77”复选框，激活后面的“格式”按钮，单击该按钮，可以打开“数字格式”对话框。

(13) 当我们设置好以上选项后，该文本风格已经基本设置完成。此时，我们可以看一下“定义风格”对话框的右上角的预览区域，看此时设置好的文本风格是否满意。如果不满意，可以重新进行设置。

此时设置好的“定义风格”对话框如图 2-51 所示。



图 2-51 设置好的“正文”文本风格

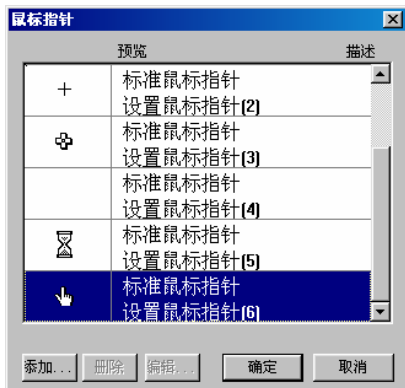


图 2-52 光标对话框

(14) 在“定义风格”对话框的右侧有一个“交互性”选项，该选项是用来设置该文本风格的交互信息，特别是设置该文本的超链接。

(15) 选择“无”单选按钮，表示该文本风格不是超链接文本。

(16) 选择“单击”单选按钮，表示该文本风格是超链接文本，单击该文本，即可跳转到链接的信息上。

(17) 当选择了“单击”单选按钮，下面的“自动加亮”复选框、“指针”复选框和“导航到”复选框被激活。

(18) 选择“双击”单选按钮，表示该文本风格是超链接文本，双击该文本，即可跳转到链接的信息上。

(19) 当选择了“双击”单选按钮，下面的“自动加亮”复选框、“指针”复选框和“导航到”复选框被激活。

(20) 选择“内部指针”单选按钮，表示该文本风格是超链接文本，当鼠标指针位于该文本之上时，即可跳转到链接的信息上。

(21) 当选择了“内部指针”单选按钮，下面的“导航到”复选框被激活。

(22) 选中“自动加亮”复选框，当单击或双击该文本而跳转到链接信息上时，该文本高亮显示。

(23) 选中“指针”复选框，可以单击其右侧的方框，可以打开鼠标光标对话框，如图 2-52 所示。

(24) 在“指针”对话框中选择一种鼠标光标，或者增加一种光标，然后单击“确定”按钮，即可指定鼠标移动到超链接文本上时鼠标光标的样式。一般情况下，选取“手型”鼠标光标作为提示超链接文本信息。

(25) 选中“导航到”复选框，单击右侧的方框，可以打开导航类型属性对话框，在该对话框中设置跳转目的地。关于导航类型属性对话框的具体设置，将在后面【导航】图标一章予以讲解。

2. 修改和删除自定义文本风格

如果要对已经定义好的文本风格进行编辑，可以选择菜单【文本】→【定义风格】命令，打开“定义风格”对话框。

在“定义风格”对话框的文本风格列表框中选择要编辑的文本风格名称，然后可以在下面的文本框中修改其名称，在右侧的各个选项中修改其文本风格，修改后，单击“定义风格”对话框中的“修改”按钮，即可完成修改。

如果要删除一种文本风格，可以在“定义风格”对话框的文本风格列表框中选择该文本风格名称，然后单击对话框中的“删除”按钮，即可将该文本风格删除。

3. 应用定义好的文本风格

当我们定义好所有的文本风格后，就可以在应用程序中使用它们。

选择菜单【文本】→【应用风格】命令，打开“应用风格”对话框，如图 2-53 所示。

在该对话框中列出了所有定义好的所有自定义文本风格。

如果要应用某一种文本风格，可以将该文本风格名称复选框选中即可使用。也可以首先选中展示窗口中的文本对象，然后在“应用风格”对话框中选中一种文本风格，即可改变选中的文本对象的文本风格。

可以再次选择选中的文本风格，可以取消文本块对该文本风格的应用。如果文本风格前的复选框中的“勾”是模糊的，则说明选中的文本不全是该指定的文本风格。

如果希望知道一种定义的文本风格由哪几个图标引用了，则可以打开定义文本风格对话框，在文本风格列表框中选中该文本风格，如“正文”文本风格，然后单击对话框中的“参考”按钮，打开“文本风格参考”对话框，如图 2-54 所示。

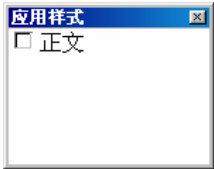


图 2-53 “应用风格”对话框图



2-54 文本风格引用对话框

在该对话框中，“从”下面的是引用该文本风格的图标类型和名称，“参考”下面的数值是对应的图标引用该文本风格的次数。

在列表框的下面的提示信息“5 涉及总数”，表示该程序共引用了该文本风格 5 次。

在列表框中选中一个引用文本风格的图标，然后单击“显示图标”按钮，则该按钮将会在程序中高亮显示。

单击“取消”按钮，可以关闭该对话框。

第3章 【移动】图标

- 【移动】图标简介
- 移动到固定点
- 移动到直线上的计算点
- 移动到固定区域中的计算点
- 沿着固定路径移动到路径终点
- 沿着固定路径移动到计算点

3.1 【移动】图标简介

在多媒体作品中，优美的二维三维动画可以增加观众对作品的兴趣，使观众加深对作品的印象，增进对作品的理解 and 提高学习效率。但是 Authorware 不能像其他动画制作工具，如三维的 3DS MAX，二维的 Macromedia Flash，Gif Animator 等，可以制作出复杂的动画效果。它为程序设计人员提供了简单的二维动画制作能力，这对于一般的动画要求已经足够了。

在 Authorware 设计的程序中，可以使用【移动】图标来实现显示对象的移动。通过【移动】图标不同的移动类型，以形成不同的动画效果。

在 Authorware 的程序中，对象主要有两种运动方式：

- (1) 对象沿着一定的路线运动。
- (2) 对象内部部分的运动。

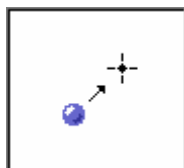
而 Authorware 中的【移动】图标可以实现显示对象沿着一定的路线运动，如一只小鸟从左边飞到右边，其特点是在运动过程中，对象的大小、形状不会改变。对于对象内部部分的运动，主要是运用外部制作工具制作出来，然后通过数字电影引入，这将会在后面的章节讲到。



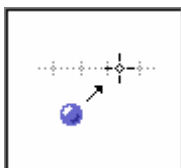
【移动】图标主要对【显示】图标或【交互】图标中的显示对象进行驱动，使其沿着一定的路线进行运动。要驱动某个对象，必须将【移动】图标放置在它所驱动对象所属图标的后面。

【移动】图标主要提供了五种移动类型：

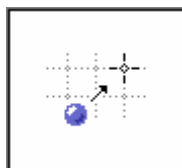
- 指向固定点：这种运动方式是将显示对象从展示窗口中的当前位置移动到另一点，如图 3-1 (a) 所示。
- 指向固定直线上的某点：这种运动方式是将显示对象从展示窗口的当前位置移动到指定直线的指定位置，如图 3-1 (b) 所示。
- 指向固定区域内的某点：这种运动方式是将显示对象从展示窗口中的当前位置移动到定义了固定区域中的一点上，如图 3-1 (c) 所示。
- 指向固定路径的终点：这种运动方式是显示对象从展示窗口中的当前位置沿着定义好的路径运动到路径终点，如图 3-1 (d) 所示。
- 指向固定路径上的任意点：这种运动方式是显示对象从展示窗口中的当前位置沿着定义好的路径运动到路径上的任意位置，如图 3-1 (e) 所示。



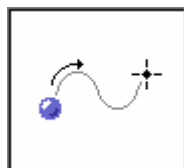
(a) 指向固定点



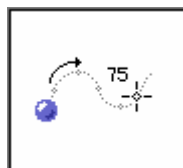
(b) 指向固定直线上的某点



(c) 指向固定区域内的某点



(d) 指向固定路径的终点



(e) 指向固定路径上的任意点

图 3-1 【移动】图标的五种运动类型

3.1.1 【移动】图标的使用

要在程序中产生动画效果，使用【移动】图标的一般步骤为：

(1) 在设计窗口的流程线上添加一个【显示】图标或【交互】图标，在该图标中创建要运动的图形或导入图片。

(2) 在【显示】图标或【交互】图标后面紧接着加入一个【移动】图标，并将【移动】图标与【显示】图标或【交互】图标中的运动对象建立链接。

(3) 打开【移动】图标属性对话框，在对话框中设置移动类型和其他设置。

3.1.2 【移动】图标属性对话框的设置

双击【移动】图标，打开【移动】图标属性对话框，如图 3-2 所示。

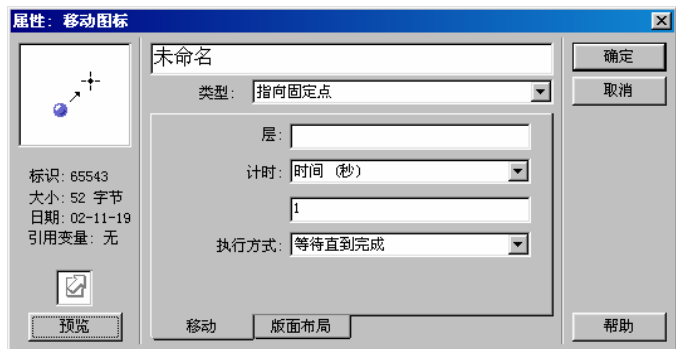


图 3-2 【移动】图标属性对话框

在【移动】图标的属性对话框的左上角为预览窗口，当没有选择移动对象时，预览窗口中显示的是移动类型图示，当选择移动对象后，预览窗口中显示的是移动对象。

单击左下角的“预览”按钮，可以立即在展示窗口中预览移动效果。

中间大的文本框是当前【移动】图标的图标名称。

中间的“类型”下拉框是用来选择移动类型的，如图 3-3 所示。

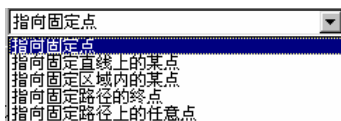


图 3-3 “移动类型”下拉框

在“类型”下拉框中的五个选项，分别对应着五种移动类型。用户可以从该下拉框中选择合适的移动类型，对于每一种移动类型，都分别对应着一个“移动”标签页。

下面分别介绍两个标签页：

1. “移动”标签页

【移动】图标属性对话框的“移动”标签页参看图 3-2。

- “层”文本框：该文本框是用来设置要移动的对象在展示窗口中所处的图层。在程序设计过程中，经常会遇到显示对象的重叠现象，我们就是通过图层的设置来决定显示对象的显示层次。在“层”文本框中可以输入正整数、零和负整数。层数大的显示对象显示在层数小的显示对象上面。

- “计时”下拉框和文本框：它主要用来控制动画效果的移动速度计算方式，它包括两种方式：
时间（秒）：以移动显示对象所需的时间（单位为秒）来计算，在下面的文本框中输入秒数。此为 Authorware 默认的控制方式。

速率（秒/英寸）：以移动显示对象的速度（单位为秒/英寸）来计算，在下面的文本框中输入任何数字类型的数值、变量或表达式。

- “执行方式”下拉框：该下拉框是用来控制执行【移动】图标时的同步方式。该下拉框包括三个选项：

等待直到完成：选择该选项时，Authorware 会执行到【移动】图标时，停止其他所有的动作，等到显示对象移动完成，【移动】图标执行完后再继续向下执行。

同时：选择该选项时，Authorware 在执行到【移动】图标时，其他所有的动作继续同步执行。

永久：选择该选项时，当被激励的对象得以显示并且给定的表达式为真的条件下，带有“永久”并发性设置的【移动】图标才被执行。

2. “版面布局”标签页

不同的移动类型有不同的“版面布局”标签页，下面是“指向固定点”移动类型的标签页，如图 3-4 所示。

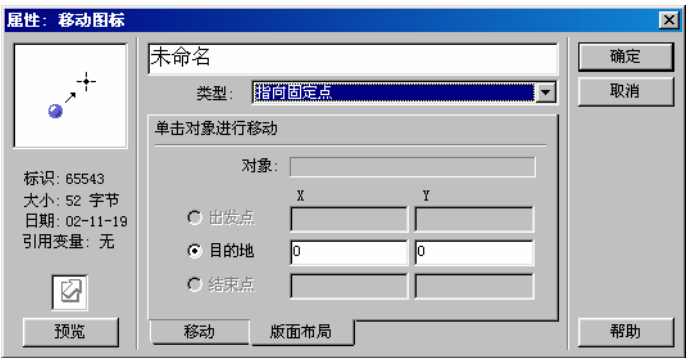


图 3-4 【移动】图标属性对话框的“版面布局”标签页

在“类型”下拉框的下面有一句提示语“单击对象进行移动”，说明在展示窗口中单击要移动的显示对象。当单击对象后，预览窗口中显示出选择对象，同时在“对象”中显示出单击对象所在的图标名，而且中间的提示变为“拖动对象到目的地”，如图 3-5 所示。

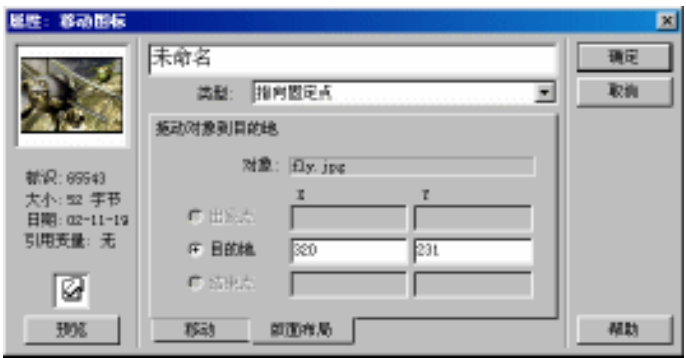


图 3-5 单击对象后的“版面布局”标签页

此时可以将对象拖动到终点，也可以在“目的地”单选按钮后面的 X、Y 文本框中输入终点坐标。如果拖动对象，X、Y 文本框中显示出当前目的地位置的坐标；如果没有单击任何对象，X、Y 文本框中显示的是 0，0；则如果单击对象后而没有拖动，X、Y 文本框中显示的是单击对象当前的坐标。

由于不同的移动类型，其【移动】图标属性对话框的“移动”标签页和“版面布局”标签页有所不同，具体区别将在讲解移动类型时讲到。

当设置完毕后，可以单击左下角的“预览”按钮，在展示窗口中预览移动效果。如果不满意可以继续修改，否则单击“确定”按钮确定。

3.2 指向固定点

指向固定点是最基本的一种移动方式，它是将显示对象从展示窗口中的当前位置移动到一个固定点。

3.2.1 选择“指向固定点”后的【移动】图标属性对话框

当将【移动】图标属性对话框的移动类型设置为“指向固定点”时，下面的“执行方式”下拉框只有两个选项，没有“永久”选项，其他选项没有变化，所以就不一一赘述。

3.2.2 指向固定点实例——蝶恋花

下面我们通过一个例子一只蝴蝶从左上角飞到花丛中来说明如何使用指向固定点移动类型。

- (1) 首先创建一个新文件。
- (2) 将【显示】图标拖动到设计窗口的流程线上，并将图标命名为“背景”。
- (3) 双击【显示】图标，打开展示窗口。
- (4) 单击工具栏中的“导入”按钮或按下快捷键<Ctrl>+<Shift>+R 打开导入文件对话框，如图 3-6 所示。



图 3-6 导入文件对话框

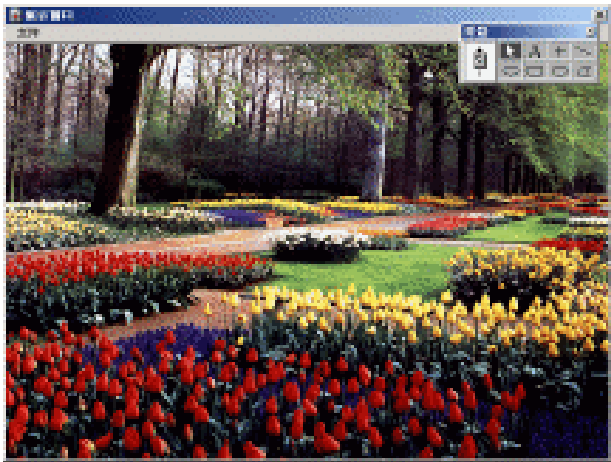


图 3-7 导入背景图案

- (5) 选择背景图案文件后，可以选择导入文件对话框中的“显示预览”复选框预览文件。如果只

生成一个指向该文件的链接，可以选择“链接到文件”复选框。单击“导入”按钮导入。此时在展示窗口中调节图片的大小以符合展示窗口的大小，如图 3-7 所示。

(6) 再拖动一个【显示】图标到流程线上，把它命名为“蝴蝶”，然后双击打开展示窗口。

(7) 单击“导入文件”按钮导入蝴蝶图片，如图 3-8 所示。

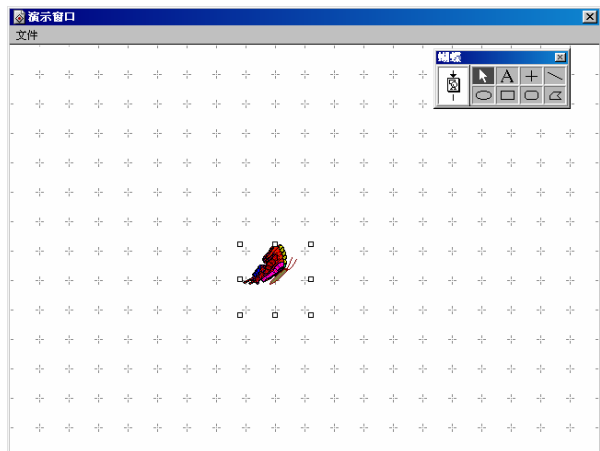


图 3-8 导入蝴蝶图片

(8) 双击“蝴蝶”图片，打开图片属性对话框，在“方式”下拉框中选择“透明”显示模式，如图 3-9 所示。关闭展示窗口。

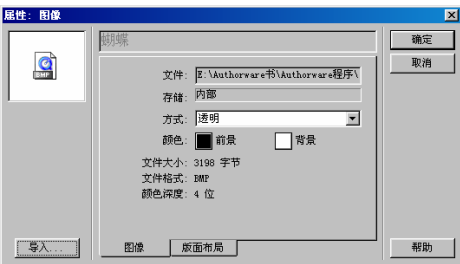


图 3-9 打开图片属性对话框

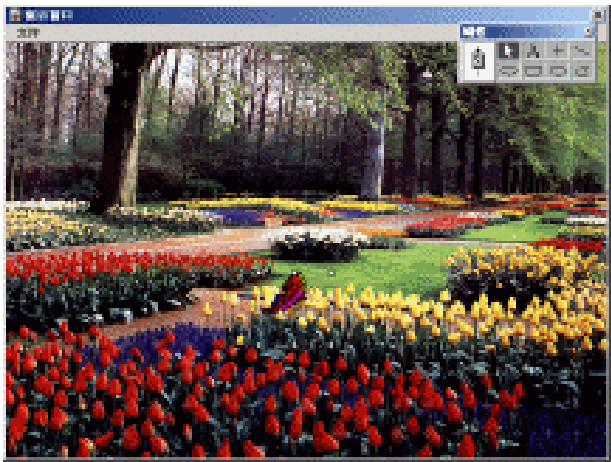


图 3-10 使“背景”图片与“蝴蝶”图片一同显示

(9) 双击打开“背景”图标的展示窗口再关闭。

(10) 按下<Shift>键双击“蝴蝶”图标打开展示窗口，使背景图片与蝴蝶图片一同显示，如图 3-10 所示。

(11) 将蝴蝶图片移到左上角后关闭展示窗口。

(12) 拖动【移动】图标到流程线上，将其命名为“移动蝴蝶”，此时设计窗口中的流程线如图 3-11 所示。



图 3-11 添加好图标后的流程线

(13) 双击【移动】图标，打开【移动】图标属性对话框。

(14) 在属性对话框中设置移动类型“类型”为“指向固定点”，时间控制方式“计时”选择“计时(秒)”，并在下面的文本框中输入“5”，设定为 5 秒。

(15) 单击“蝴蝶”图片并直接将其拖动到花丛中，如图 3-12 所示。



图 3-12 拖动“蝴蝶”图片到花丛中

(16) 单击【移动】图标属性对话框左下角的“预览”按钮预览移动效果。感觉满意后单击“确定”按钮确定。

(17) 单击工具栏中的“执行程序”命令按钮或直接按下快捷键<Ctrl>+R 运行程序观看蝴蝶移动效果。

(18) 如果感到满意后，保存文件为“蝴蝶.a6p”。

3.3 指向固定直线上的某点

这种运动方式是将移动对象从展示窗口的当前位置移动到指定直线的指定位置。

3.3.1 选择“指向固定直线上的某点”后的【移动】图标属性对话框

当将【移动】图标属性对话框中的移动类型设置为“指向固定直线上的某点”时，【移动】图标属性对话框的“移动”标签页发生了一些变化，如图 3-13 所示。

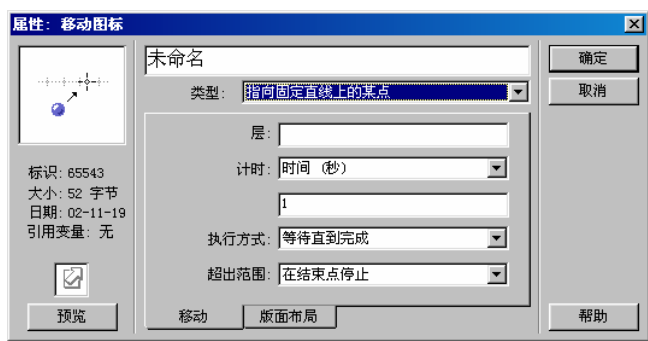


图 3-13 【移动】图标属性对话框

其中的一些选项已经介绍过了，就不再赘述。

增加了一个“超出范围”下拉框，它有三个选项，如图 3-14 所示。

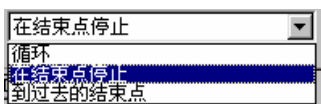


图 3-14 “超出范围”下拉框的选项

- 循环：该选项是将当前的控制值对终点与起点之间的距离求余，其余数就为移动的终点值。例如，如果起点位置值为 0，终点位置值为 50，控制移动的为 80，则移动对象应该移动到起点到终点的直线上的 30 ($80 \text{ Mod } (50-0)=30$) 位置。这样保证了显示对象只能在两个移动端点之间移动。
- 在结束点停止：选择该选项，移动对象会在终点处停止。这样保证了对象不会移到规定能够的线或区域之外。如果控制对象的数值、变量或表达式的值超过了线或区域终点的值，则对象只会移动到线或区域的终点位置并停止。
- 到过去的结束点：选择该选项，移动对象移动到目标值代表的位置上，不管目标值是否超出了两个端点。当目标值在直线的两个端点之外，甚至在展示窗口之外，移动对象也会沿着直线移动到目标值处。此时两个端点只是作为移动直线的参考点，并不限制移动对象的移动范围。

将【移动】图标属性对话框的移动类型“类型”设置为“指向固定直线上的某点”，“版面布局”标签页是用来设置移动对象，移动基准点和两个端点的对话框。“版面布局”标签页与“指向固定点”移动类型的“版面布局”标签页有所变化，如图 3-15 所示。

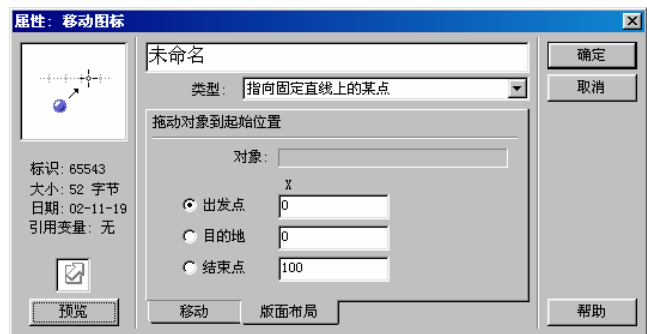


图 3-15 【移动】图标属性对话框的“版面布局”标签页

其中，“出发点”单选钮是用来定义直线的起始端点坐标的，可以先选择“出发点”单选钮，然后

拖动移动对象到一点上，该点即是直线的起始端点。

“结束点”单选钮是用来定义直线的终点端点坐标的，可以先选择“结束点”单选钮，然后拖动移动对象到一点上，该点即是直线的终点端点。

“目的地”单选钮是用来定义移动对象移动到直线上的端点坐标，它的值可以直接在后面的 X 文本框中输入，也可以先选择“目的地”单选钮，然后拖动移动对象沿着定义好的直线移动到直线上的一点上，该点即是移动对象的移动终点。

至于移动的起始点，是移动对象开始时所处的位置。

移动对象移动到直线上的一点上有两种方式：

- 移动对象在直线外

此时定义好直线的起始点和终点，以及目标点，移动对象会从展示窗口的当前位置直接移动到定义好的直线上的目标点上，如图 3-16 所示。

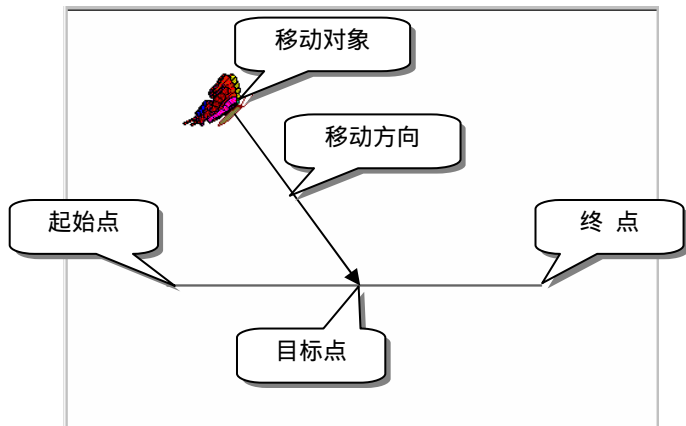


图 3-16 移动对象不在目标直线上

要改变移动对象的当前位置，可以通过在移动对象所在图标的展示窗口中改变。

- 移动对象在直线上

如果移动对象在直线上，移动对象会从当前位置沿着直线移动到目标点处，如图 3-17 所示。

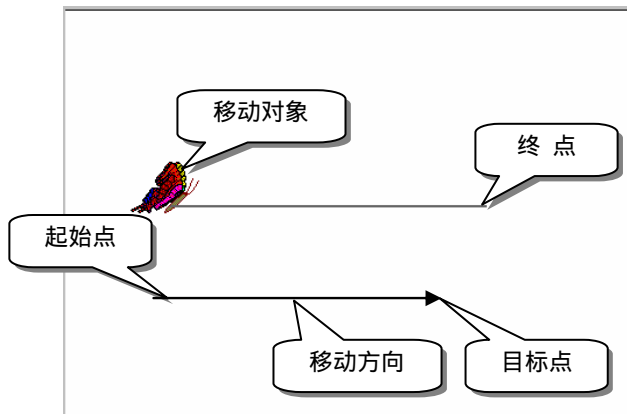


图 3-17 移动对象位于直线之上

3.3.2 指向固定直线上的某点实例——弹子球游戏

下面我们通过一个例子——弹子球游戏来说明如何使用指向固定直线上的某点移动类型。

- (1) 建立一个新文件，命名为“弹子球.a6p”。
- (2) 拖动一个【显示】图标到流程线上，命名为“背景”。双击该图标打开展示窗口，导入一幅图片作背景，如图 3-18 所示。

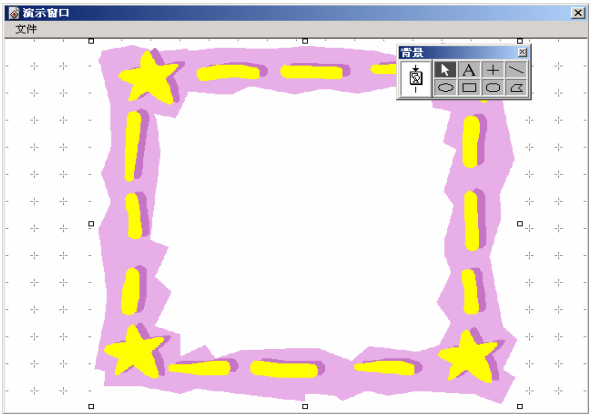


图 3-18 建立程序背景

- (3) 再拖入一个【显示】图标，命名为“杯子”。按下<Shift>键打开展示窗口。
- (4) 单击工具栏中的“导入文件”按钮，导入一个杯子形状图片，用鼠标对其进行调整。并且选择该图片并复制，然后粘贴出另外九个图片。
- (5) 将其中一幅图片拖动到背景图的左下角，以确定对齐方式的左边界和下边界。
- (6) 再将其中的另一幅图片拖动到背景图的右下角，以确定对齐方式的右边界。
- (7) 将它们都选中，选择菜单【修改】→【排列】选项，打开对齐方式工具板，从中选择下对齐方式，将所有图片先按底边对齐。
- (8) 点击对齐方式控制板中的水平等间隔方式，使所有对象等间隔排列。
- (9) 对图片进行细微的调整，如果不满意，可以继续使用对齐方式，最后将十个杯子形状的图片排列成图 3-19 形状。

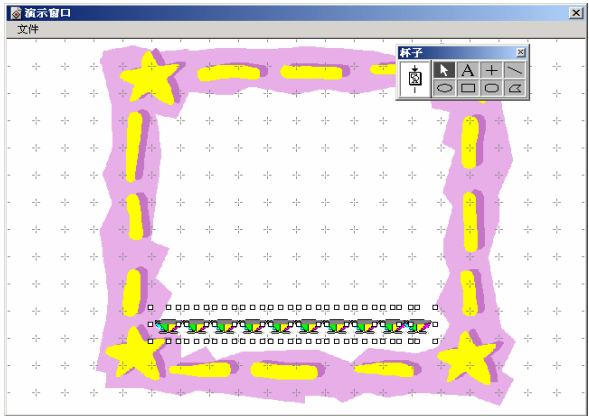


图 3-19 排列十个杯子

- (10) 拖动一个【显示】图标到流程线上，命名为“小球”，双击该图标打开展示窗口。
- (11) 双击绘图工具箱中的椭圆工具，在调色板中单击填充颜色块，然后在上面的颜色列表中选择紫色。
- (12) 选中椭圆工具，按下<Shift>键在背景图的上面中间画出一个圆形，如图 3-20 所示。

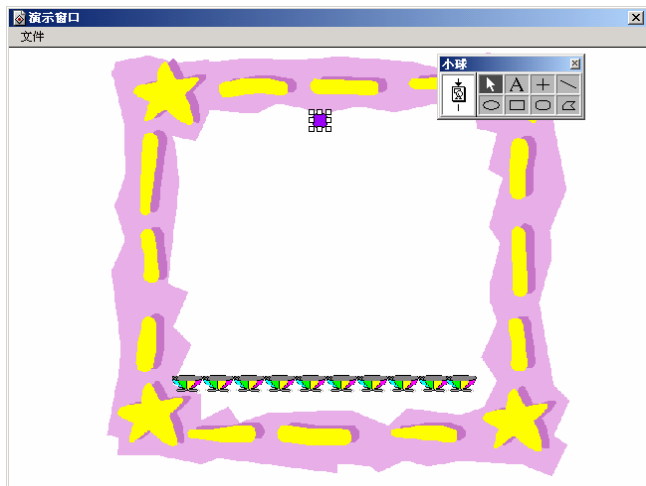


图 3-20 绘制小球

- (13) 拖入一个【交互】图标，命名为“控制”。
- (14) 拖动一个【计算】图标到【交互】图标右侧，出现“交互类型”对话框，如图 3-21 所示。由于系统默认选项为“按钮”类型，所以直接单击“确定”按钮即可。

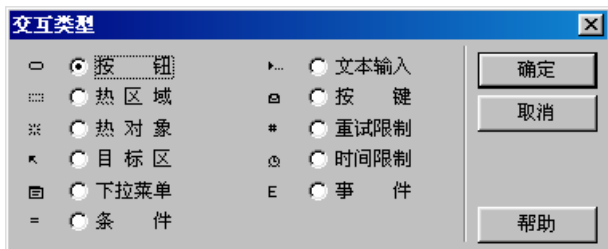


图 3-21 响应类型对话框

- (15) 将该【计算】图标命名为“复位”，双击打开它，如图 3-22 所示。

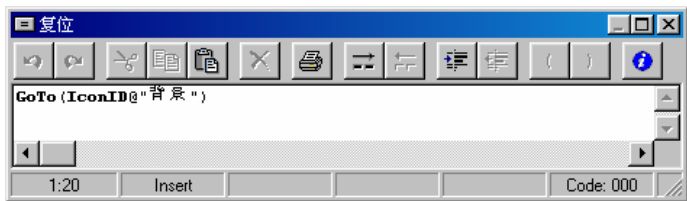


图 3-22 打开【计算】图标

- (16) 单击工具栏中的“函数”命令按钮，打开函数窗口，如图 3-23 所示。



图 3-23 函数窗口

(17) 从“分类”下拉框中选择“跳转”选项，然后在下面的函数列表中直接双击“GoTo”函数将它粘贴到【计算】图标中。

(18) 在【计算】图标中，将 GoTo(IconID@"IconTitle")中的“IconTitle”改为“背景”。这样当单击“复位”按钮后，程序会返回到“背景”图标继续执行。

(19) 关闭【计算】图标，出现一个提示信息对话框，询问用户是否保存对【计算】图标的修改，如图 3-24 所示。单击“是”按钮确定。

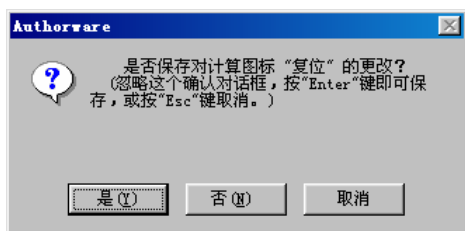


图 3-24 提示信息对话框

(20) 拖动一个【群组】图标到【计算】图标的右边，并将其命名为“开始”。

(21) 双击【群组】图标，打开二级流程，拖动一个【计算】图标到二级流程线上，将其命名为“产生随机数”。

(22) 双击打开该【计算】图标，在函数窗口的“分类”下拉框中选择“数学”选项，然后在下面的函数列表中直接双击“Random”函数将它粘贴到【计算】图标中。该函数是产生随机数的。

(23) 在【计算】图标中将“Random(min, max, units)”改为“p:=Random(1,100, 1)”，即在 1 到 100 范围内产生一个是 1 的倍数的随机数，并将附给变量 p，如图 3-25 所示。

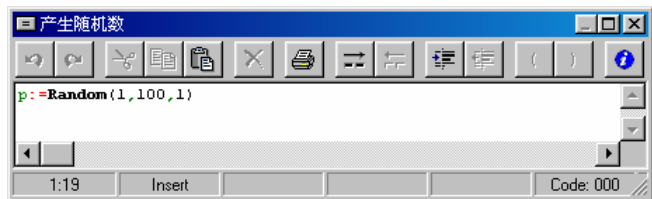


图 3-25 添加随机数函数

- (24) 关闭【计算】图标，出现提示信息对话框，直接单击“是”按钮确定。
- (25) 此时出现“新建变量”对话框，如图 3-26 所示，直接单击“确定”按钮确定。



图 3-26 新建变量对话框

- (26) 拖动一个【移动】图标，将其命名为“移动小球”。
- (27) 按住<Shift>键双击【移动】图标，打开【移动】图标属性对话框。在“类型（移动类型）”下拉框中选择“指向固定直线上的某点”选项，然后在“计时”下拉框中选择“时间（秒）”，并在下面的文本框中输入“3”，表示移动需要 3 秒时间。
- (28) 选择【移动】图标属性对话框的“版面布局”标签页，确定选中了“出发点”单选钮，拖动小球到十个杯子的最左边，如图 3-27 所示。

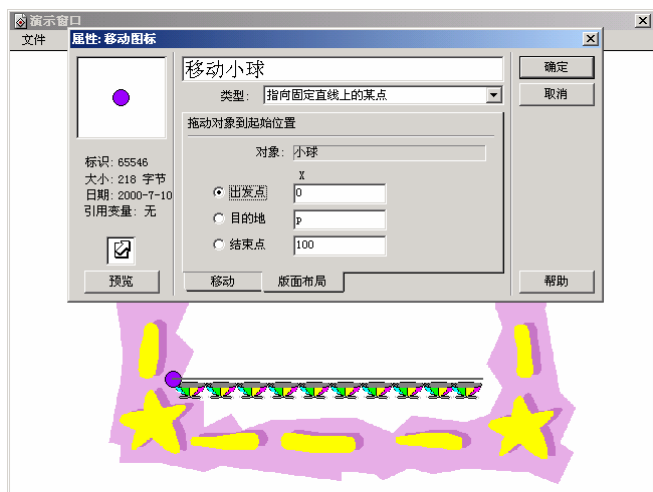


图 3-27 拖动小球到杯子的最左边

- (29) 选中“结束点”单选钮后，将小球拖动到十个杯子的最右边，如图 3-28 所示

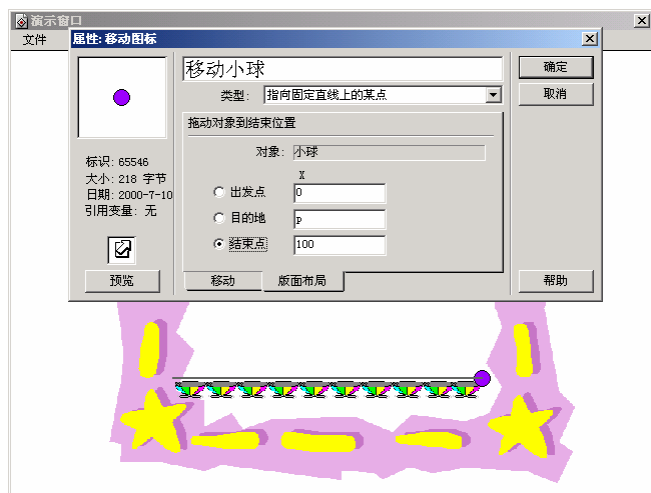


图 3-28 将小球拖动到最右边

(30) 单击“目的地”单选按钮，然后在后面的文本框中输入变量 p ，表示随机产生的数是目标点。单击“确定”按钮关闭【移动】图标属性对话框。

(31) 拖动一个【显示】图标到【移动】图标下面，将其命名为“得分”。

(32) 按住<Shift>键双击“得分”图标，打开展示窗口。双击绘图工具箱中的椭圆工具打开调色板。

(33) 单击调色板中的左边的“边框颜色板”，从上面的颜色列表列表中选取红色。然后选取工具栏中的“粗体”命令按钮，表示要输入的文本颜色是红色的，并且是粗体。

(34) 选取绘图工具箱中的文字工具，在展示窗口中单击一下并输入“您得了{p}分”，如图 3-29 所示。

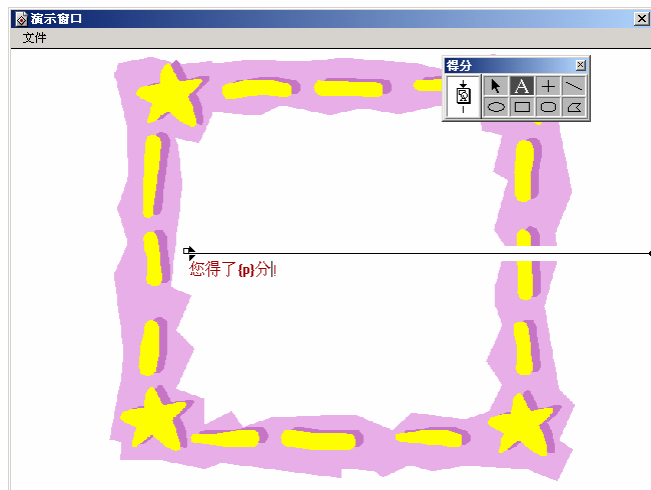


图 3-29 输入文本

(35) 关闭二级流程，在一级流程中，拖动一个【计算】图标到【交互】图标的【群组】图标右侧，将其命名为“退出”。

(36) 双击该【计算】图标，在【计算】图标中输入“Quit()”，然后关闭【计算】图标，并确定对【计算】图标的改变。

(37) 按下<Shift>键并双击【交互】图标，打开展示窗口，将三个按钮利用对齐方式工具板对齐排列在右上角，如图 3-30 所示。

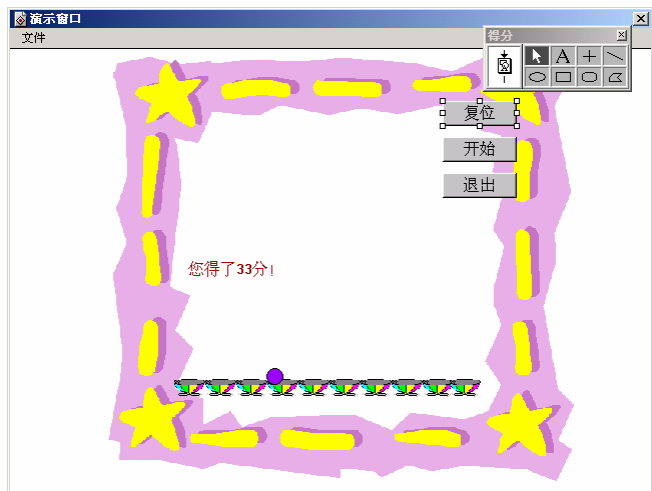


图 3-30 排列对齐三个按钮

(38) 此时整个程序设计完毕，设计好的流程图如图 3-31 所示。

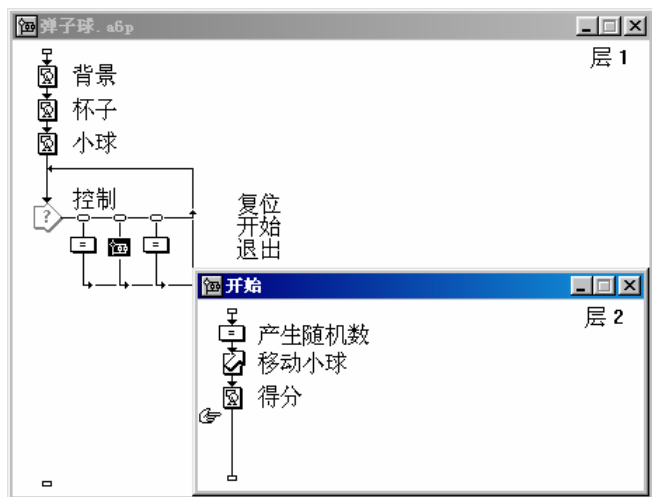


图 3-31 设计好的流程

(39) 按下工具栏中的“执行程序”命令按钮运行程序观看小球移动效果。

(40) 当单击“开始”按钮时，小球会随机地运动到一个杯子中，屏幕上显示出您的得分，如图 3-32 所示；如果单击“复位”按钮，程序会回到开始状态；单击“退出”按钮退出程序。



图 3-32 运行程序时的移动效果

3.4 指向固定区域内的某点

这种运动方式是将显示对象从展示窗口中的当前位置移动到定义了固定区域中的一点上。“指向固定区域内的某点”和“指向固定直线上的某点”两种移动类型十分相似，惟一区别是“指向固定直线上的某点”移动类型的目标点只能在一维的直线上变化，而“指向固定区域内的某点”移动类型可以在定义了区域的二维平面上移动。

3.4.1 选择“指向固定区域内的某点”后的【移动】图标属性对话框

当在【移动】图标属性对话框的“类型”下拉框中选择“指向固定区域内的某点”选项，“移动”标签页与“指向固定直线上的某点”移动类型的“移动”标签页相同，而“版面布局”标签页有所变化，如图 3-33 所示。

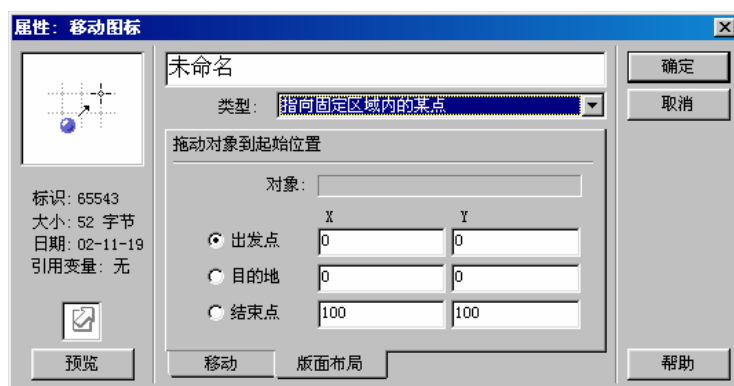


图 3-33 “指向固定直线上的某点”移动类型的“版面布局”标签页

该移动类型的功能是实现移动对象运动到平面上的某一点上，其含义是设置一个 X-Y 坐标平面，X 坐标和 Y 坐标的起始点和终止点的位置和数值都可以修改，而在定义好起始点和终止点之间的区域中的点由其相对于两端点所对应的 X-Y 坐标值所确定。

在【移动】图标属性对话框的“版面布局”标签页中，单击“出发点”单选钮，将移动对象拖动到移动区域的起始点，其起始点的 X-Y 坐标值在“出发点”单选钮后面的 X 文本框和 Y 文本框中，默认为 0、0；可以在“出发点”单选钮后面的 X 文本框和 Y 文本框中输入起始点的 X-Y 坐标值。

当将移动对象拖动到起始点后，中间的提示变为“选择终点至定义的结束位置处”，单击“结束点”单选钮定义终止点，此时提示又变为“拖动对象到结束位置”，将移动对象拖动到终止点，“结束点”单选钮后面的 X 文本框和 Y 文本框中是终止点的坐标，默认为 100、100；用户可以在 X 文本框和 Y 文本框中输入终止点的 X、Y 坐标值。

定义好移动对象的终止点后，单击“目的地”单选钮，提示信息变成“拖动对象到目的地”，将移动对象移动到目标点后，“目的地”单选钮后面的 X 文本框和 Y 文本框中显示出拖动后目标点的 X、Y 坐标值，当然您也可以在 X 文本框和 Y 文本框中直接输入目标点的坐标值。

3.4.2 指向固定区域内的某点实例——蝴蝶飞呀飞

该例子是蝴蝶在花丛中飞呀飞，单击屏幕上任一点都会退出程序，如图 3-34 所示。该例子运用了指向固定区域内的某点移动类型。

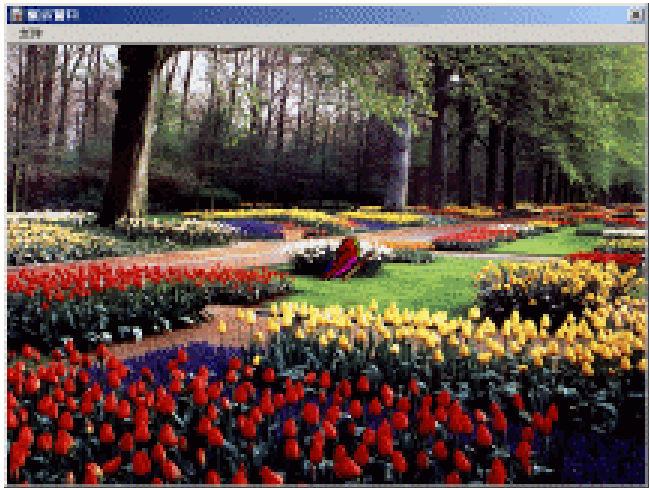


图 3-34 蝴蝶飞呀飞

- (1) 按照指向固定点实例——蝶恋花的步骤分别拖动两个【显示】图标，分别命名为“背景”和“蝴蝶”，并分别导入与例子“蝶恋花”相同的图片。
- (2) 拖动一个【交互】图标到流程线上，将其命名为“控制”。
- (3) 拖动一个【群组】图标到【交互】图标右侧，在出现的响应类型对话框中选择“条件”类型，然后单击“确定”按钮，将该图标命名为“True”。
- (4) 双击【群组】图标上面的响应类型图标，如图 3-35 所示，打开响应类型属性对话框，如图 3-36 所示。

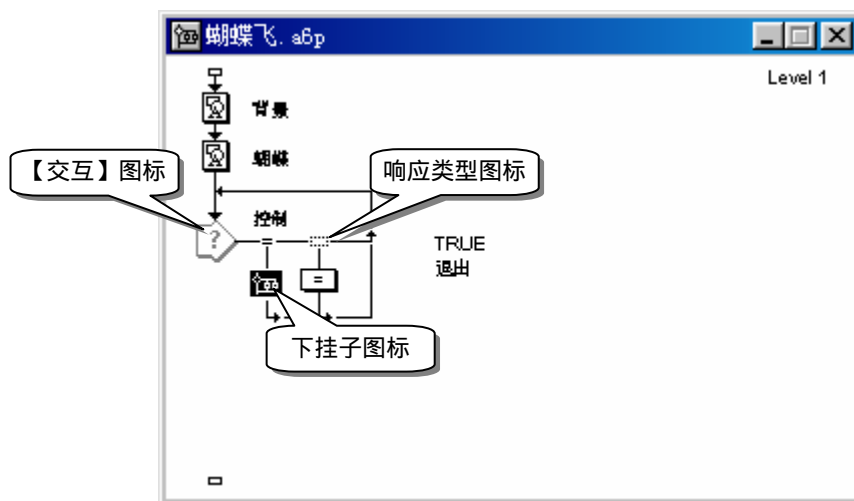


图 3-35 【交互】图标分支结构

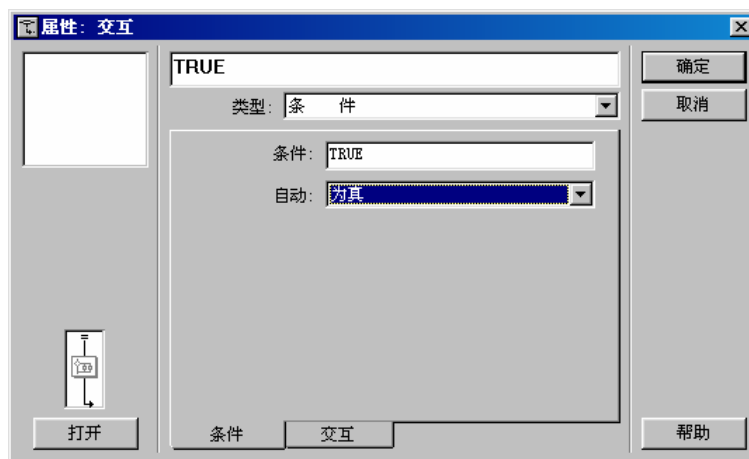


图 3-36 响应类型属性对话框

(5) 在响应类型对话框的“自动”下拉框中选择“为真”选项，然后单击“确定”按钮关闭对话框。

(6) 拖动一个【计算】图标到【群组】图标右侧，双击【计算】图标上面的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“热区域”响应类型。

(7) 在中间大的图标名称文本框中输入“退出”，将【计算】图标命名为“退出”。

(8) 单击响应类型属性对话框中“鼠标指针”后边的  按钮，打开“鼠标指针”对话框，如图 3-37 所示。

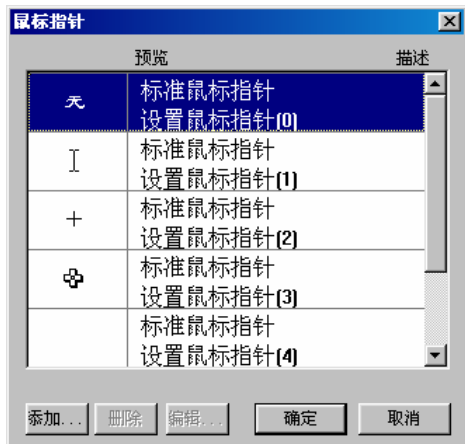


图 3-37 鼠标指针对话框

(9) 在光标对话框中选择手型光标，单击“确定”按钮退出。

(10) 选择响应类型属性对话框的“交互”标签页，在“交互”标签页中选择“永久”复选框，让该热区响应一直处于激活状态。

(11) 将响应类型属性对话框拖到一边，以便看到响应区域，将热区先拖到左上角，单击右下角的句柄并拖动到展示窗口的右下角，将热区增大到整个展示窗口，如图 3-38 所示。最后按下“确定”按钮关闭响应类型属性对话框。

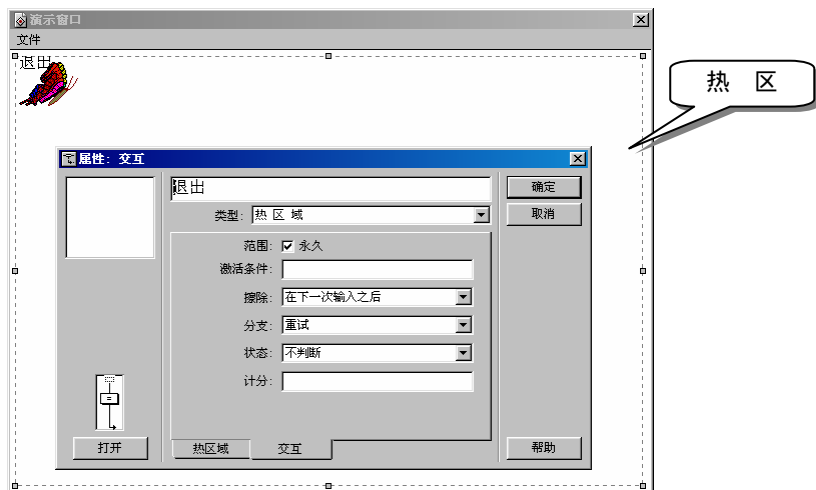


图 3-38 增大热区范围

(12) 双击【群组】图标，打开二级流程。

(13) 拖动一个【计算】图标到二级流程线上，将其命名为“产生坐标值”。

(14) 双击【计算】图标打开，在【计算】图标中输入下面语句：

`x:=Random(1,100,1)`

`y:=Random(1,100,1)`

(15) 关闭【计算】图标，确定对其的修改。此时会出现两个新建变量对话框，分别单击“确定”按钮确定。

- (16) 拖动一个【移动】图标到二级流程线【计算】图标下面，将其命名为“移动蝴蝶”。
- (17) 双击【移动】图标打开【移动】图标属性对话框，在“类型”下拉框中选择“指向固定区域内的某点”类型，选取“计时”下拉框中的“速率（秒/英寸）”选项，并在下面的文本框中输入“0.5”，表示移动对象的移动速度为 0.5 秒/英寸。
- (18) 选取【移动】图标属性对话框的“版面布局”标签页，拖动展示窗口中的蝴蝶图片到展示窗口的左上角；然后选择“结束点”单选钮，再将蝴蝶图片拖动到展示窗口的右下角；单击“目的地”单选钮，然后在“目的地”单选钮后面的 X-Y 文本框中分别输入“x”、“y”，如图 3-39 所示。单击“确定”按钮退出。

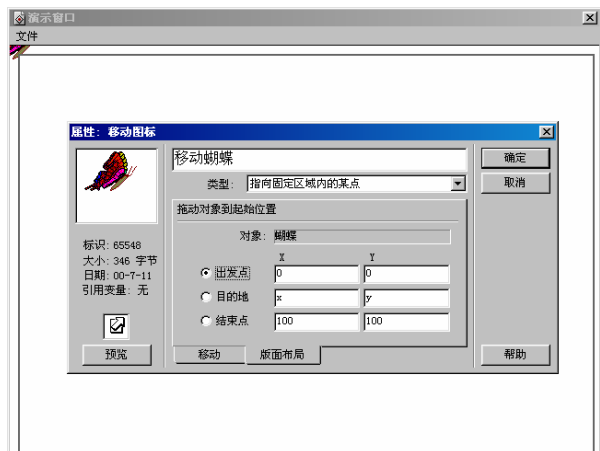


图 3-39 建立移动区域

- (19) 此时整个程序设计完毕，设计好的流程图如图 3-40 所示。

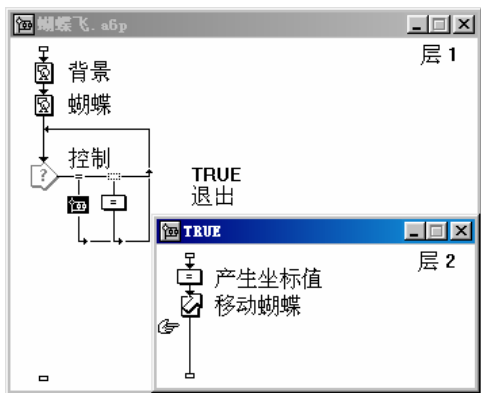


图 3-40 设计好的流程

- (20) 单击工具栏中的“执行程序”命令按钮运行程序观看蝴蝶移动效果。
- (21) 将文件保存为“蝴蝶飞.a6p”。

3.5 指向固定路径的终点

这种运动方式是显示对象从展示窗口中的当前位置沿着定义好的路径运动到路径终点。“指向固定

路径的终点”移动方式与前面的三种移动方式有所不同，前面三种移动方式都是直线运动；而“指向固定路径的终点”移动方式是基于自定义复杂路径的移动类型，移动路径可以是直线、折线以及曲线等，这样可以设计出复杂的运动效果。

3.5.1 选择“指向固定路径的终点”后的【移动】图标属性对话框

当在【移动】图标属性对话框的“类型”下拉框中选择“指向固定路径的终点”选项后，“移动”标签页与前面的有所不同，如图 3-41 所示。

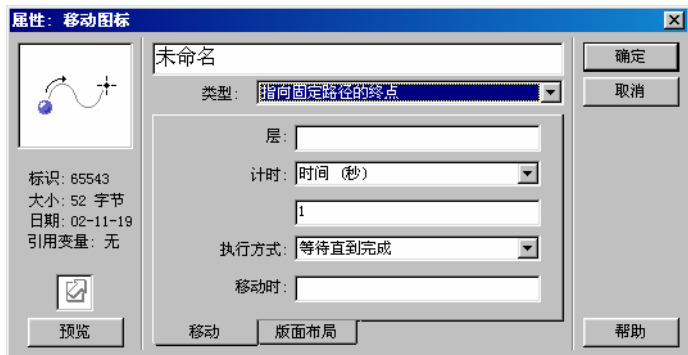


图 3-41 选择“指向固定路径的终点”后的“移动”标签页

其中的一些选项与前面的移动类型相同，就不再赘述。

最下面的“超出范围”下拉框变成了“移动时”文本框，该文本框是用来设置使移动生效的条件。用户可以在文本框中输入常量、变量以及表达式，当其为“真”时，移动对象开始移动。

当将【移动】图标属性对话框的移动类型“类型”设置为“指向固定路径的终点”时，“版面布局”标签页是用来设置移动对象，移动基准点以及编辑基准点的对话框。“版面布局”标签页与前面的移动类型的“版面布局”标签页有所变化，如图 3-42 所示。

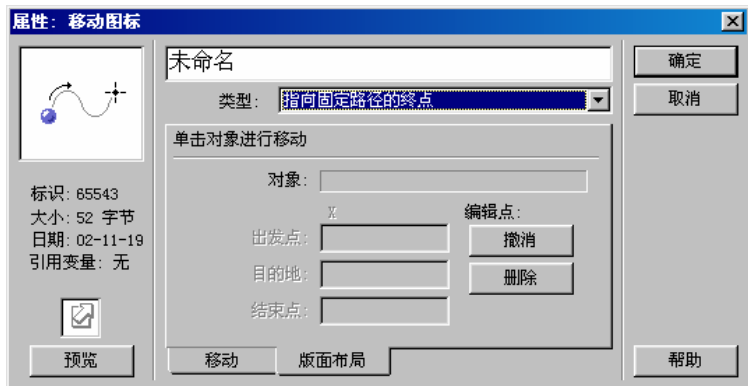


图 3-42 选择“指向固定路径的终点”后的“版面布局”标签页

当没有选择移动对象时，中间的提示为“单击对象进行移动”，要求用户选择移动对象，单击要移动的物体即可，当单击选择好移动对象时，移动对象中心会出现一个三角形，该三角形是要创建路径的起始点。如图 3-43 是单击展示窗口中的“蝴蝶”图片为移动对象时的情形。



图 3-43 实心三角形是移动路径的起始点

当选择移动对象后，提示变为“拖动对象以创建路径”，用户就可以拖动移动对象创建移动路径了。松开鼠标左键后，拖动后的新位置也出现一个三角形，同时两个三角形之间形成一条直线，即为移动路径，如图 3-44 所示。



图 3-44 产生移动路径

此时中间的提示又变为“拖动对象到扩展路径”，提示用户可以继续拖动移动对象以延长移动路径。单击移动对象继续拖动，产生新的路径……这样重复上面的操作，可以制作出复杂的移动路径，如图 3-45 所示。

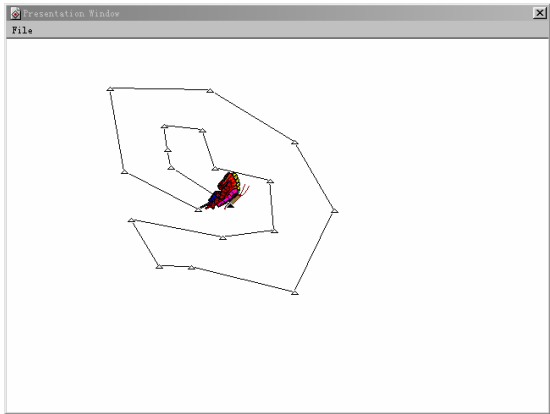


图 3-45 产生复杂的移动路径

在移动路径上，有一些三角形，这些三角形称为路径的关键点。从图 3-45 中可以看出，移动路径虽然十分复杂，但是移动路径仍然是折线，在每一段中都是直线。如果我们对关键点进行编辑，可以将移动路径变成曲线形状。

对关键点的编辑包括五种操作：

- 选择当前关键点：要对某个关键点进行操作，单击该关键点即可选择它。当前关键点是实心三角形形式，而其他关键点是空心三角形。

- 增加关键点：在移动路径上要添加关键点的地方单击即可在此处增加关键点。
- 删除关键点：先选择该关键点，然后单击【移动】图标属性对话框“版面布局”标签页中的“删除”按钮即可将该关键点删除。如果要撤消删除操作，可以单击“版面布局”标签页中的“撤消”按钮即可。
- 移动关键点：要移动某个关键点，可以单击该关键点并拖动到新位置即可。
- 某一段直线移动路径与曲线路径的互相转变：如果要将某一段直线移动路径转变曲线路径，可以双击该处的关键点，该关键点两端的直线会以曲线过渡，同时关键点由三角形改变为圆形，如图 3-46 所示。同样，如果要将某一曲线段转变为直线，可以双击该处的关键点即可，同时圆形关键点也变成了三角形关键点。

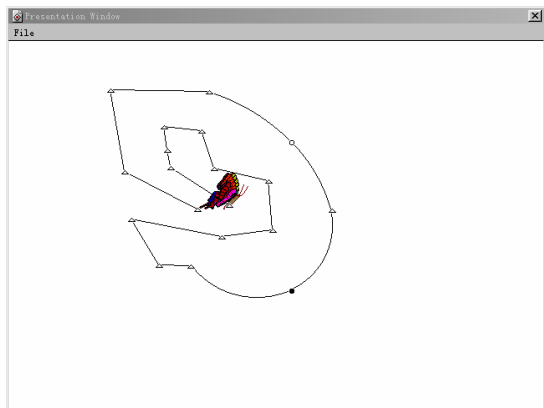


图 3-46 将直线段转变为曲线段

3.5.2 指向固定路径的终点实例——老鹰捉小鸟

这个例子是一只老鹰捉一只小鸟，老鹰和小鸟的飞行路径都是运用指向固定路径的终点移动类型，如图 3-47 所示。

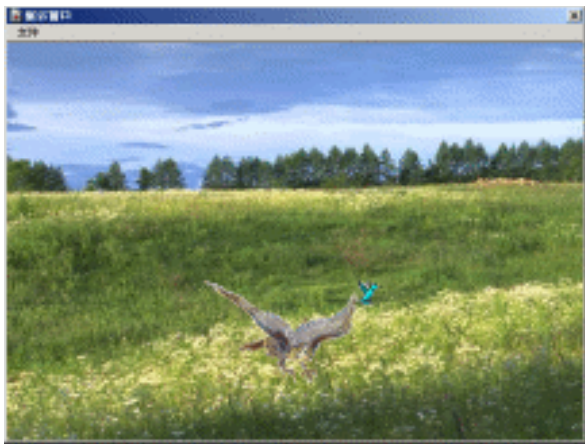


图 3-47 老鹰捉小鸟

- (1) 单击工具栏中的“新建文件”命令按钮，创建一个新文件。
- (2) 拖动一个【显示】图标到流程线上，将其命名为“背景”，双击打开其展示窗口，导入一幅

背景图片，如图所示。

(3) 再拖动一个【显示】图标到流程线上，命名为“小鸟”，按下<Shift>键双击打开其展示窗口，导入“小鸟”图片，如图 3-48 所示。



图 3-48 导入“小鸟”图片

(4) 双击绘图工具箱中的指针工具，打开显示模式工具板，选择“透明”模式，并将“小鸟”图片拖到展示窗口的左上部。

(5) 拖入一个【显示】图标，命名为“老鹰”，按下<Shift>键并双击该图标打开展示窗口，导入一幅“老鹰”图片，如图 3-49 所示。



图 3-49 导入“老鹰”图片

(6) 单击显示模式工具板中的“透明”模式，让“老鹰”图片透明显示，并将其拖动到左上角。

(7) 拖入一个【移动】图标，将其命名为“移动小鸟”。

(8) 按下<Shift>键并双击【移动】图标，打开其属性对话框。在“类型”下拉框中选择“指向固定路径的终点”选项，选择“计时”下拉框中的“时间(秒)”选项，并在下面的文本框中输入数值“5”。

(9) 为了能够让“小鸟”移动的同时也用另一个【移动】图标移动“老鹰”，所以在“执行方式”下拉框中选取“同时”选项，让“小鸟”和“老鹰”同时移动。

(10) 单击“小鸟”图片并拖动，产生移动路径，为了能够让“小鸟”飞的时候看起来比较平滑，所以分别双击其中的一些关键点，将直线段转变为曲线段，如图 3-50 所示。

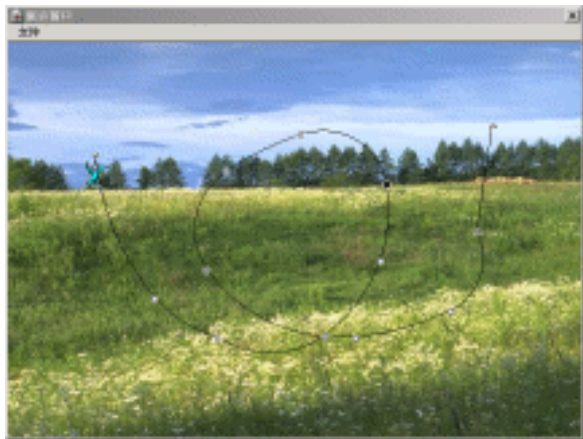


图 3-50 拖动“小鸟”产生移动路径

(11) 使用【移动】图标属性对话框的“版面布局”标签页中的“删除”按钮和“撤销”按钮以及其他操作对关键点进行编辑。

(12) 按下【移动】图标属性对话框左下角的“预览”按钮预览移动效果，感到满意后单击“确定”按钮确定。

(13) 拖动一个【移动】图标到流程线上，并将其命名为“移动老鹰”。

(14) 按下<Shift>键并双击【移动】图标，打开其属性对话框。在“类型”下拉框中选择“指向固定路径的终点”选项，为了能够让老鹰和小鸟同时到达移动路径终点，所以选择“计时”下拉框中的“时间(秒)”选项，并在下面的文本框中输入相同的数值“5”。

(15) 单击【移动】图标属性对话框中的“版面布局”标签页，单击并拖动“老鹰”图片产生移动路径，为了能够看起来在相同的时间里(5秒)“老鹰”比“小鸟”飞的快，应该将“老鹰”的移动路径拖动的比“小鸟”的移动路径长。并且“老鹰”在终点时必须看起来好象已经捉到了“小鸟”。

(16) 对移动路径的关键点进行编辑，如将一些直线段转变为曲线段、移动某个关键点等。编辑好的移动路径如图 3-51 所示。

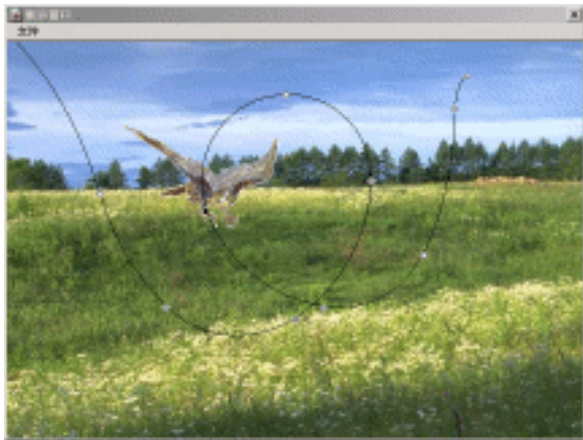


图 3-51 拖动“老鹰”产生移动路径

(17) 按下【移动】图标属性对话框左下角的“预览”按钮预览移动效果，感到满意后单击“确定”按钮确定。

(18) 此时整个程序设计完毕，设计好的流程图如图 3-52 所示。

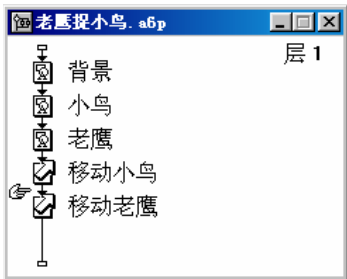


图 3-52 设计好的流程

(19) 单击工具栏中的“执行程序”命令按钮运行程序观看老鹰和小鸟的移动效果。

(20) 将文件保存为“老鹰捉小鸟.a6p”。

3.6 指向固定路径上的任意点

这种运动方式是显示对象从展示窗口中的当前位置沿着定义好的路径运动到路径上的任意位置。“指向固定路径上的任意点”移动类型和“指向固定路径的终点”移动类型十分相似，都是沿着自定义的移动路径移动对象，所不同的只是“指向固定路径上的任意点”移动类型并不一定要移动到路径终点，它移动的目标点是通过【移动】图标属性对话框的“版面布局”中的“目的地”项来确定。

3.6.1 选择“指向固定路径上的任意点”后的【移动】图标属性对话框

当在【移动】图标属性对话框的“类型”下拉框中选择“指向固定路径的终点”选项后，“移动”标签页与前面的相同，所以就不一一赘述。而“版面布局”标签页则发生了一些变化，如图 3-53 所示。

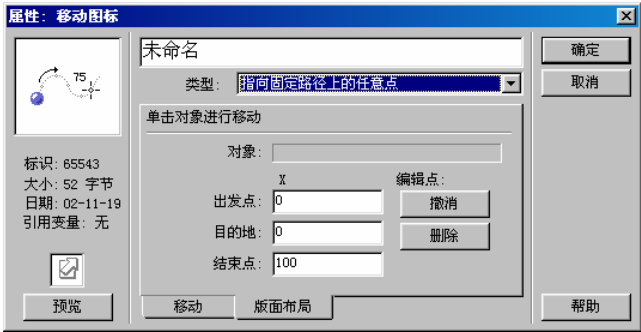


图 3-53 选择“指向固定路径上的任意点”后的“版面布局”标签页

“指向固定路径上的任意点”移动类型也需要先定义一段路径，定义方法与“指向固定路径的终点”类型相同，但“指向固定路径上的任意点”需要定义路径的起始点、终点和目标点，而定义方法又与前面“指向固定直线上的某点”类型相同。不过在【移动】图标属性对话框的“版面布局”标签页依然有对路径关键点编辑的“撤销”按钮和“删除”按钮。

3.6.2 指向固定路径上的任意点实例——蜗牛赛跑

下面我们通过一个例子——蜗牛赛跑来说明如何使用“指向固定路径上的任意点”移动类型，如图 3-54 所示。



图 3-54 蜗牛赛跑

- (1) 单击工具栏中的“新建文件”按钮，新建一个文件。
- (2) 拖动一个【显示】图标到流程线上，将其命名为“背景”，双击打开展示窗口，导入背景图片。
- (3) 拖入一个【显示】图标到流程线上，将其命名为“蜗牛 1”，按住<Shift>键并双击图标打开展示窗口，导入一幅蜗牛图片，将图片拖到展示窗口的左侧。
- (4) 双击绘图工具箱中的指针工具，打开显示模式工具板，选择“透明”模式，关闭展示窗口。
- (5) 再拖入一个【显示】图标到流程线上，命名为“蜗牛 2”，按住<Shift>键并双击图标打开展示窗口，导入蜗牛图片，将图片拖到展示窗口的左侧与“蜗牛 1”并排放置。
- (6) 选择显示模式工具板中的“透明”模式，然后关闭展示窗口。
- (7) 拖动一个【交互】图标到流程线上，将其命名为“控制”。
- (8) 拖入一个【计算】图标到【交互】图标右侧，在出现的“响应类型对话框”中直接单击“确定”按钮确定响应类型为“按钮”类型。
- (9) 将【计算】图标命名为“复位”，双击打开它，输入“GoTo(IconID@"背景")”，关闭【计算】图标并确定其改变。
- (10) 拖入一个【群组】图标到【计算】图标右侧，将其命名为“开始”。
- (11) 再拖入一个【计算】图标到【群组】图标右侧，将其命名为“退出”，双击打开它，输入“Quit()”，关闭该【计算】图标并确定其改变。
- (12) 按住<Shift>键双击【交互】图标，将三个按钮排列对齐于展示窗口的右侧，如图 3-54 所示。
- (13) 双击【群组】图标打开二级流程。
- (14) 拖入一个【计算】图标，命名为“产生目标点”，双击打开并输入下面的语句：

```
x:=Random(1,100,1)  
y:=Random(1,100,1)
```
- (15) 关闭【计算】图标并确定修改。
- (16) 拖入一个【移动】图标到二级流程线上，命名为“移动蜗牛 1”，按住“<Shift>键并双击打

开【移动】图标属性对话框。

(17) 在“类型”下拉框中选择“指向固定路径上的任意点”选项；在“计时”下拉框中选择“速率(秒/英寸)”，并在下面的文本框中输入“1”；在“执行方式”下拉框中选取“同时”选项，表示在移动蜗牛1的过程中继续执行下面的程序，即同时移动蜗牛2。

(18) 选取“版面布局”标签页，单击“蜗牛1”并拖动产生移动路径，然后对路径进行编辑，如将直线段转变为曲线段、移动关键点等，编辑好的移动路径如图3-55所示。



图 3-55 创建“蜗牛1”的移动路径

(19) 在“版面布局”标签页的“目的地”文本框中输入变量x，然后单击“确定”按钮关闭属性对话框。

(20) 同样拖动一个【移动】图标到二级流程线上，并命名为“移动蜗牛2”按住<Shift>键并双击打开【移动】图标属性对话框。

(21) 在“类型”下拉框中选择“指向固定路径上的任意点”选项；在“计时”下拉框中选择“速率(秒/英寸)”，并在下面的文本框中输入“1”。

(22) 选取“版面布局”标签页，单击“蜗牛1”并拖动产生移动路径，然后对路径进行编辑，如将直线段转变为曲线段、移动关键点等，编辑好的移动路径如图3-56所示。



图 3-56 拖动“蜗牛2”产生移动路径

(23) 在“版面布局”标签页的“目的地”文本框中输入变量 y ，然后单击“确定”按钮关闭属性对话框。

(24) 拖入一个【显示】图标到二级流程线上，将其命名为“显示结果”。

(25) 先选取该【显示】图标，然后选择菜单【修改】→【图标】→【计算】命令或者直接按下快捷键 $\text{Ctrl}+\text{+=}$ ，打开【显示】图标的计算编辑窗口，在窗口中输入下面语句：

```
if x>y then p:="蜗牛 1 获胜！"
else if x<y then p:="蜗牛 2 获胜！"
else p:="蜗牛 1 和蜗牛 2 平手！"
```

(26) 关闭计算编辑窗口，确定对计算编辑窗口的改变，出现新建变量对话框，单击“确定”按钮确定即可。

(27) 此时【显示】图标的左上角出现一个“=”号，如图 3-57 所示。

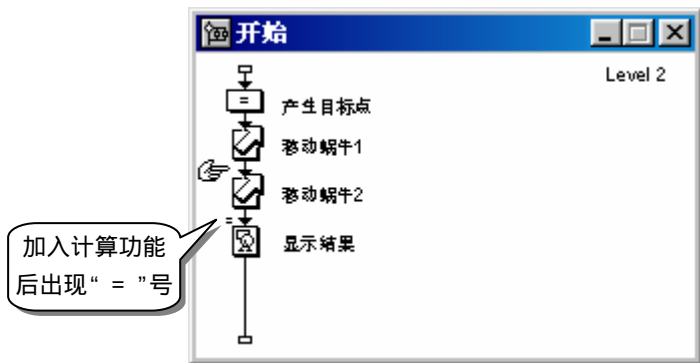


图 3-57 在【显示】图标中加入计算功能

(28) 按住 Shift 键双击【显示】图标打开展示窗口，先双击绘图工具箱中的椭圆工具，打开调色板，单击调色板中的左边的“边框颜色板”，从上面的颜色列表列表中选取紫红色。然后选取工具栏中的“粗体”命令按钮，表示要输入的文本颜色是紫红色的，并且是粗体。

(29) 选取绘图工具箱中的文字工具，在展示窗口中单击一下并输入“{p}”，如图 3-58 所示，关闭展示窗口。

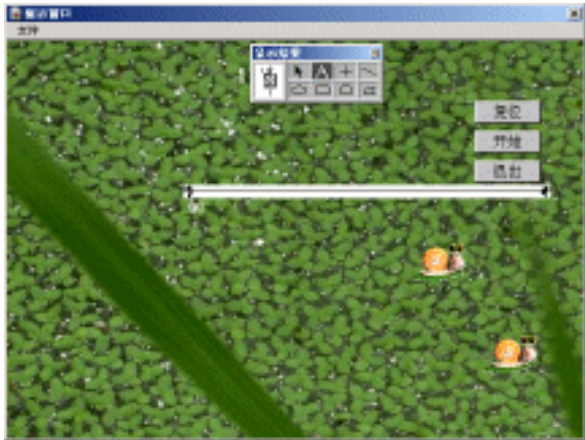


图 3-58 输入显示结果语句

(30) 此时整个程序设计完毕，设计好的流程图如图 3-59 所示。

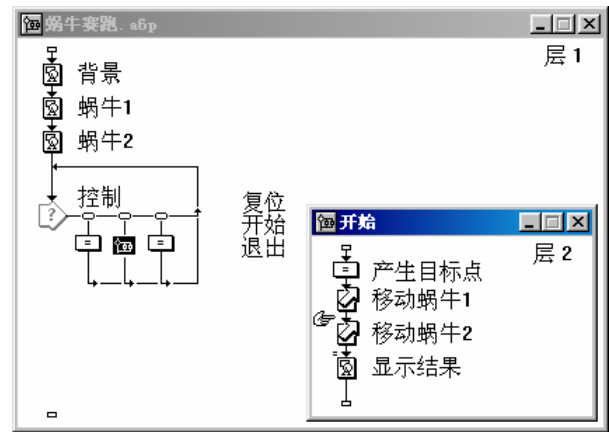


图 3-59 设计好的流程

(31) 单击工具栏中的“执行程序”命令按钮运行程序观看移动效果。

(32) 将文件保存为“蜗牛赛跑.a6p”。

第4章 【擦除】 【等待】 图标

■ 【擦除】图标简介

■ 【擦除】图标实例

■ 【等待】图标简介

■ 【等待】图标实例

4.1 【擦除】图标简介

在 Authorware 程序设计过程中，由于会有很多显示对象，它们之间经常会互相遮盖，而且有很多显示对象用过之后就不再使用，所以我们需要将它们擦除掉，此时就需要使用到【擦除】图标。尽管在【显示】图标的属性对话框中有“擦除以前的内容”选项，但是它是将该图标之前展示窗口中所有的显示对象都擦除。如果有些显示对象我们在后来还要使用，如果使用这种方法，就会将有用的内容也擦除了。此时就可以使用【擦除】图标。

【擦除】图标可以同时擦除几个【显示】图标或【交互】图标中的内容，但是如果擦除一个【显示】图标或【交互】图标中的内容，它就会将【显示】图标或【交互】图标中的所有内容都擦除掉，而不能只擦除【显示】图标或【交互】图标中的一部分对象。这同【移动】图标的性质相似。如果要单独擦除某个对象，可以将该对象单独放在一个【显示】图标中。

【擦除】图标擦除的对象可以用绘图工具箱中的工具绘制的图形，也可以是外部导入的图片、动画以及数字电影等。

4.1.1 【擦除】图标的使用

要在程序中擦除某些显示对象，使用【擦除】图标的一般步骤为：

- (1) 在设计窗口的流程线上添加一个【显示】图标或【交互】图标，在该图标中创建图形或导入图片。
- (2) 在【显示】图标或【交互】图标后面紧接着加入一个【擦除】图标，并将【擦除】图标与【显

示】图标或【交互】图标中的运动对象建立链接。
(3) 打开【擦除】图标属性对话框，在对话框中设置擦除过渡类型和其他设置。

4.1.2 【擦除】图标属性对话框的设置

双击流程线上的【擦除】图标，打开【擦除】图标属性对话框，如图 4-1 所示。

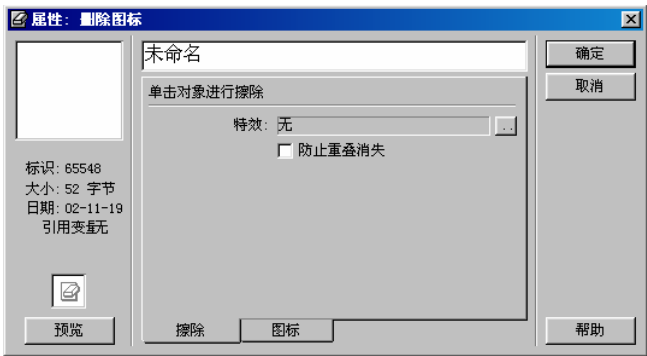


图 4-1 【擦除】图标属性对话框

在【擦除】图标属性对话框中，中间的文本框是该【擦除】图标的名称，下面是一句提示信息“单击对象进行删除”，提示用户单击展示窗口中要擦除的对象。

下面分别介绍两个标签页：

1. “擦除”标签页

在该标签页中只有两个项目，“特效”是用来定义擦除过渡效果的，单击后面的...按钮可以打开“擦除模式”对话框，如图 4-2 所示。

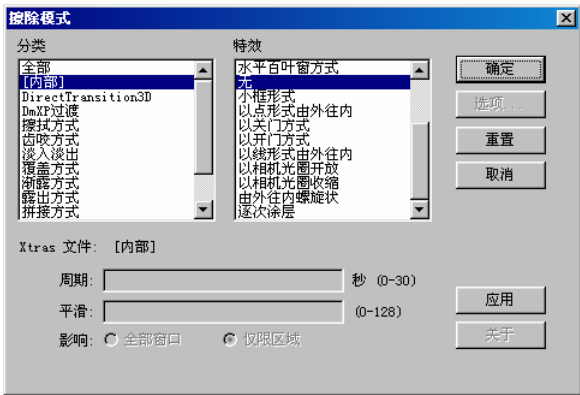


图 4-2 擦除过渡效果对话框

用户可以在擦除过渡效果对话框中设置擦除显示对象时的过渡效果，由于它与【显示】图标的过渡效果对话框设置相同，就不再一一赘述。

“防止重叠消失”复选框是用来控制擦除显示对象时程序应该如何进行。如果选择该选项，Authorware 会等到【擦除】图标擦除完设置的所有对象后才继续执行程序；如果不选择该选项，Authorware 会在执行【擦除】图标擦除显示对象的同时，开始继续运行下面的程序。

2. “图标”标签页

在【擦除】图标属性对话框中单击“图标”标签页，显示“图标”标签页，如图 4-3 所示。

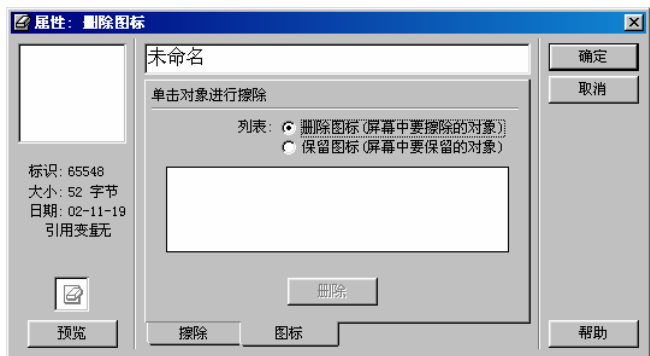


图 4-3 【擦除】图标的“图标”标签页

“列表”是标示下面大的文本框中显示内容的类型，它包括两个选项：

“删除图标”单选钮：当选择该选项时，下面的列表框中显示的是该【擦除】图标所要擦除的图标列表；当执行完该【擦除】图标，下面列表框中所有的图标都会被擦除。

“保留图标”单选钮：当选择该选项时，下面的列表框中显示的是该【擦除】图标所要擦除的图标以外的图标列表；当执行完该【擦除】图标，下面列表框中所有的图标都会被保留。

“删除”按钮是用来移除列表框中的某个或某些图标。要移除某个或某些图标，先在列表框中选择这些图标，然后单击该按钮即可将选中的图标从列表中移除。

4.2 【擦除】图标实例——风景图片展

下面的例子是运用【显示】图标和【擦除】图标制作的风景图片展，如图 4-4 所示。



图 4-4 风景图片展

- (1) 新建一个文件，命名为“风景图片展.a6p”。
- (2) 拖动一个【显示】图标到流程线上，将其命名为“主页”。

(3) 双击该图标，打开展示窗口，单击工具栏中的“导入文件”命令按钮导入一幅图片，调整图片大小以适应展示窗口的大小。

(4) 选择菜单【修改】→【图标】→【特效】命令，打开过渡效果对话框，如图 4-5 所示。从左边过渡效果类别列表中选择“填充方式”，从右边过渡效果列表中选择“快速细块填充展示”，其他设置不进行改变，单击“确定”按钮确定。

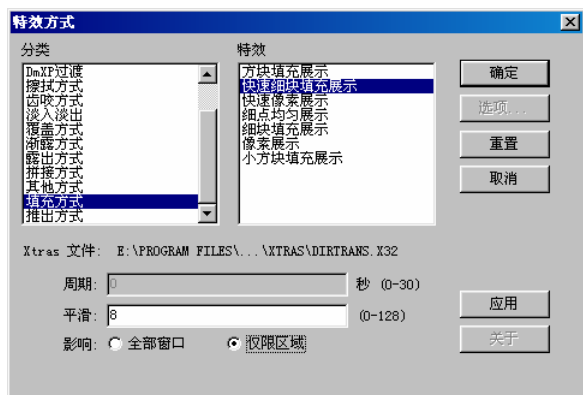


图 4-5 特效选择对话框

(5) 拖动一个【擦除】图标到流程线上，将其命名为“擦除主页”。

(6) 双击【擦除】图标打开【擦除】图标属性对话框，单击对话框后面展示窗口中“主页”图片，表示要擦除“主页”图片。

(7) 单击“特效”后面的  按钮打开“擦除模式”对话框，在擦除过渡效果对话框中选择“Dissolve/Dissolve, Pixels”，其他设置不进行改变，单击“确定”按钮确定。

(8) 拖动一个【显示】图标到流程线上，将其命名为“风景 1”。

(9) 双击该图标，打开展示窗口，单击工具栏中的“导入文件”命令按钮导入一幅图片，调整图片大小以适应展示窗口的大小。

(10) 选择菜单【修改】→【图标】→【特效】命令，打开过渡效果对话框，从对话框中选择“填充方式/像素展示”，其他设置不进行改变，单击“确定”按钮确定。

(11) 拖动一个【擦除】图标到流程线上，将其命名为“擦除风景 1”。

(12) 双击【擦除】图标打开【擦除】图标属性对话框，单击对话框后面展示窗口中“风景 1”图片，将“风景 1”图片擦除。

(13) 单击“特效”后面的  按钮打开“擦除模式”对话框，在擦除过渡效果对话框中选择“填充方式/像素展示”，其他设置不进行改变，单击“确定”按钮确定。

(14) 同理拖动两个【显示】图标和两个【擦除】图标到流程线上，分别命名为“风景 2”、“擦除风景 2”、“风景 3”和“擦除风景 3”。

(15) 将“风景 2”的过渡效果设置为“填充方式/像素展示”，将“擦除风景 2”的过渡效果设置为“填充方式/细点均匀展示”，将“风景 3”的过渡效果设置为“填充方式/细点均匀展示”，将“擦除风景 3”的过渡效果设置为“覆盖方式/由上向下覆盖展示”。

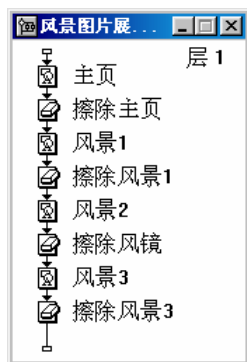


图 4-6 设计好的流程

(16) 程序的流程已经设计完成, 设计好流程如图 4-6 所示。

4.3 【等待】图标简介

在制作的多媒体应用软件, 有时运行时需要一定的停顿, 【等待】图标就是 Authorware 6.0 提供的实现程序暂停的方法。

使用【等待】图标属性对话框可以指定【等待】图标想要响应的事件的类型, 如鼠标单击或按键。也可以使用【等待】图标来指定是否等待一段时间后才继续执行程序, 以及是否显示一个小时钟来显示剩余的等待时间。当然, 您也可以使用该对话框设置屏幕上是否显示一个等待按钮。

Authorware 在流程线上第一次遇到一个新【等待】图标时, 并不会自动打开它。但是, 如果您在【等待】图标属性对话框中选中“显示倒计时”选项, 屏幕上出现一个小时钟, 此时可以直接双击这个小时钟, 来打开【等待】图标。

拖曳一个【等待】图标到流程线上, 双击该【等待】图标, 出现其属性对话框。如图 4-7 所示。

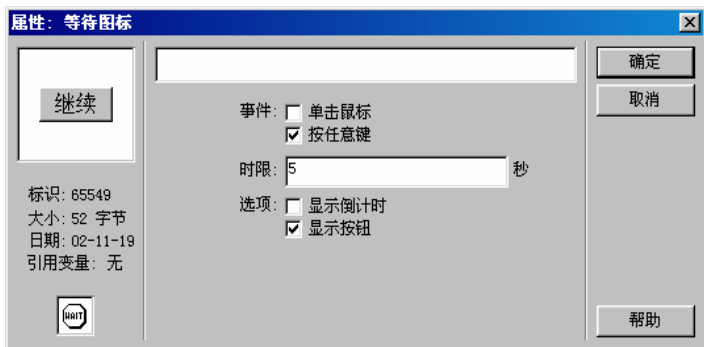


图 4-7 【等待】图标属性对话框

下面我们对该对话框予以简单介绍。

对话框的左上角是该【等待】图标的预览窗口。缺省时为一个“继续”按钮。

- “事件”选项: 设置事件响应结束等待, 其响应事件有两种类型:

“单击鼠标”复选框: 选中该选项, 当程序执行到【等待】图标时, 如果用户在展示窗口中单击鼠标左键, 则结束等待状态, 程序向下继续执行。

“按任意键”复选框: 选中该选项, 当程序执行到【等待】图标时, 如果用户按键盘上的任意一个键, 则结束等待状态, 程序向下继续执行。

- “时限”选项: 在右侧的文本框中输入等待时间, 此单位为秒。程序等待指定时间后继续向下执行。如果想结束等待, 则选择“事件”中的一种方式, 按鼠标或任意键即可。

为了使等待的时间由一个变量来控制, 可以在该文本框中输入一个变量或表达式来替换一个具体数值。例如, 可以在所有的【等待】图标属性对话框“时限”选项的文本框中输入一个自定义变量, 通过改变该变量的值来统一改变所有【等待】图标的等待时间, 使之全部等待相同的时间。这样, 就不必一个一个地单独改变【等待】图标的等待时间了。

- “选项”选项: 该选项包括两个子选项:

“显示倒计时”复选框: 当“时限”不为空时, 此项有效。选中该选项, 当程序执行到【等待】图标时, 演示窗口出现一个小时钟, 以显示剩下的等待时间。

您可以在作品运行时任意移动这个小时钟, 即直接拖动该小时钟到一个新位置即可。双击该小时钟则可以打开【等待】图标属性对话框。

“显示按钮”复选框：选中此项后，屏幕上会出现一个“继续”等待按钮，如果按下此键，则结束等待状态。

系统默认是显示该等待按钮，该按钮的标签为“继续”。该如果希望修改该等待按钮及其标签，可以选择菜单【修改】→【文件】→【属性】命令，打开文件属性对话框，然后选取对话框的“交互作用”标签页，如图 4-8 所示。



图 4-8 文件属性对话框的“Interaction”标签页

在“等待”选项中，可以单击其右侧的...按钮，打开按钮编辑器（也称为按钮对话框），然后使用按钮编辑器选择一个合适的等待按钮，或者自己编辑一个新的等待按钮。

在“标签”文本框中修改等待按钮的标签，可以在该文本框中输入中文，如“等待”。

下面是我们自己编辑好的等待按钮，如图 4-9 所示。

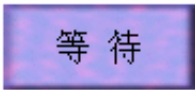


图 4-9 编辑的等待按钮

当将一个【等待】图标拖动到流程线上时，展示窗口中显示的等待按钮（如果您选择显示等待按钮的话）会自动显示在最后一次放置其他等待按钮的位置。

用户也可以在运行作品时定位等待按钮。在作品运行时，选择菜单【调试】→【暂停】命令，暂时暂停程序运行，然后拖动等待按钮到一个新的位置。在程序运行时定位等待按钮，可以统一整个作品中使用的等待按钮的位置。

4.4 【等待】图标实例——幻灯片的制作

该实例我们使用到了第 5 章的实例——风景图片展，在该流程上添加上【等待】图标，使之能够让用户看到风景一段时间后，自动播放下一幅风景图片，如同幻灯片一样效果。同时添加几个【交互】图标，使之能够控制播放时间间隔，如图 4-10 所示。

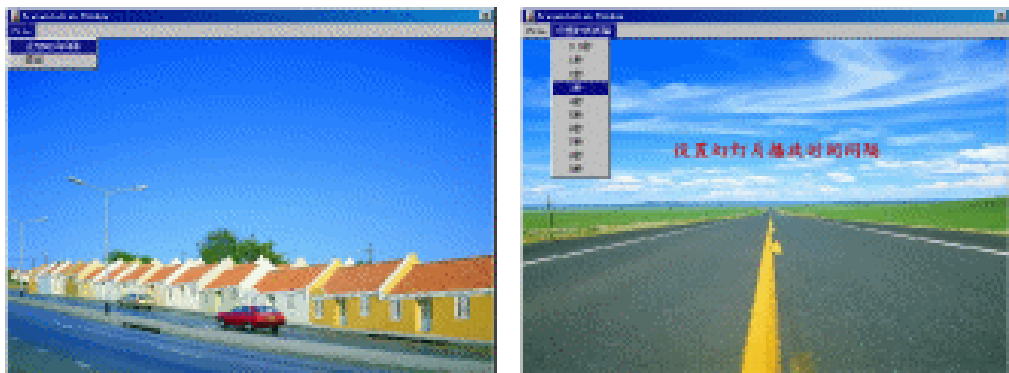


图 4-10 幻灯片

操作步骤为：

- (1) 新建一个文件，命名为“幻灯片.a6p”。
- (2) 拖动一个【交互】图标到设计窗口的流程线上，并将其命名为“File”。
- (3) 拖动一个【群组】图标到“File”【交互】图标的右侧，此时系统会弹出一个响应类型对话框，如图 4-11 所示。

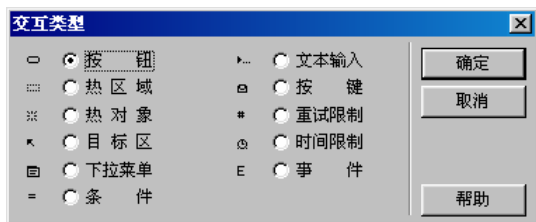


图 4-11 响应类型对话框

- (4) 在该响应类型对话框中选择“下拉菜单”单选钮，表示该响应类型为下拉菜单响应。单击“确定”按钮确定。
- (5) 将该【群组】图标命名为“设定时间间隔”。
- (6) 双击“设定时间间隔”【群组】图标上面的响应类型图标，打开响应类型属性对话框，选择“交互”标签页，将“永久”复选框选中，然后单击“确定”按钮确定。
- (7) 再拖动一个【群组】图标到“设定时间间隔”【群组】图标的右侧，然后双击其上面的响应类型图标，打开响应类型属性对话框。
- (8) 从“类型”下拉框中选取“条件”选项，选择“条件”标签页，在“条件”文本框中输入“TRUE”；然后从“自动”下拉框中选取“为真”选项。
- (9) 选择“交互”标签页，将“永久”复选框选中，然后单击“确定”按钮确定。
- (10) 拖动一个【计算】图标到【交互】图标的最右侧，然后双击其上面的响应类型图标，打开响应类型属性对话框。
- (11) 从“类型”下拉框中选取“下拉菜单”选项；选择“交互”标签页，将“永久”复选框选中，并选取“分支”下拉框中的“退出交互”选项，然后单击“确定”按钮确定。
- (12) 将该【计算】图标更名为“退出”。
- (13) 打开第 5 章的实例“风景图片展.a5p”，选择菜单【编辑】→【全选】命令，或者直接使用快捷键<Ctrl>+A，将流程线上的所有图标都选中。



(14) 单击工具栏中的“复制”按钮，将所有图标复制到剪贴板上。

(15) 重新打开“幻灯片.a6p”，然后双击“TRUE”【群组】图标，打开二级流程。

(16) 单击工具栏中的“粘贴”按钮，将复制的所有风景图片展文件中的所有图标粘贴到“TRUE”二级流程窗口中。

(17) 拖动一个【等待】图标到“主页”【显示】图标和“擦除主页”【擦除】图标之间，将其命名为“等待”。

(18) 双击该【等待】图标，打开【等待】图标属性对话框，在属性对话框中选中“单击鼠标”复选框和“按任意键”复选框，在“时限”文本框中输入一个自定义变量 Wait_Time，取消“显示按钮”复选框，然后单击“确定”确定。

(19) 此时系统会弹出一个新建变量对话框，在“初始值”文本框中输入“1”，设定该变量的初始值为1，如图4-12所示，然后单击“确定”按钮。



图 4-12 新建变量对话框

(20) 选中该【等待】图标，然后单击工具栏中的“复制”按钮，将其复制到剪贴板上。

(21) 将粘贴指针移到“风景1”【显示】图标下面，然后单击工具栏中的“粘贴”按钮，将赋值的【等待】图标粘贴到此处。

(22) 同样，按照上一步操作将复制的【等待】图标粘贴到“风景2”、“风景3”图标下面。

(23) 选中“主页”图标，然后单击工具栏中“复制”按钮将其复制。

(24) 双击“设定时间间隔”【群组】图标，打开其二级流程窗口。

(25) 单击工具栏中的“粘贴”按钮，将“主页”图标复制到二级流程窗口中，并将其更名为“背景”。

(26) 双击“背景”图标，打开其展示窗口。选取绘图工具箱中的文字工具，选取“风景图片展”几个字，然后将其改为“设置幻灯片播放时间间隔”几个字。

(27) 将这几个字选中，然后选择菜单[文本]→[字体]→[其他]命令，在“字体”对话框中选取“楷体_GB2312”，单击“确定”按钮确定。

(28) 选择菜单[文本]→[字体]→[大小...]命令，设置字体大小为18。

(29) 选择绘图工具箱中的指针工具，将“设置幻灯片播放时间间隔”几个字移动到展示窗口的中间，然后关闭展示窗口。

(30) 拖动一个【交互】图标到“背景”图标下面，将其命名为“设置时间间隔”。

(31) 拖动一个【计算】图标到【交互】图标右侧，在弹出的响应类型对话框中选择“下拉菜单”选项，将其更名为“0.5秒”。

(32) 双击其上面的响应类型图标，打开响应类型属性对话框，选择“交互”标签页，然后选取“分支”下拉框中的“退出交互”选项，单击“确定”按钮确定。

(33) 同样,在【交互】图标右侧拖动 9 个【计算】图标,分别命名为“1 秒”、“2 秒”、“3 秒”、“4 秒”、“5 秒”、“6 秒”、“7 秒”、“8 秒”和“9 秒”,并且在其响应类型属性对话框“交互”标签页的“分支”下拉框中都选取“推出交互”选项。

(34) 双击“0.5 秒”【计算】图标,打开计算编辑窗口,在该窗口中输入语句“Wait_Time := 0.5”,然后关闭该窗口,并确定其修改。

(35) 同样,在“1 秒”【计算】图标的计算编辑窗口中输入语句“Wait_Time := 1”;在“2 秒”【计算】图标计算编辑窗口中输入语句“Wait_Time := 2”;.....在“9 秒”【计算】图标计算编辑窗口中输入语句“Wait_Time := 9”。

(36) 选择第 1 级流程窗口,双击“退出”【计算】图标,在其计算编辑窗口中输入语句“Quit()”,然后关闭该窗口,并确定其修改。

(37) 此时程序已经设计完成,完成后的流程如图 4-13 所示。

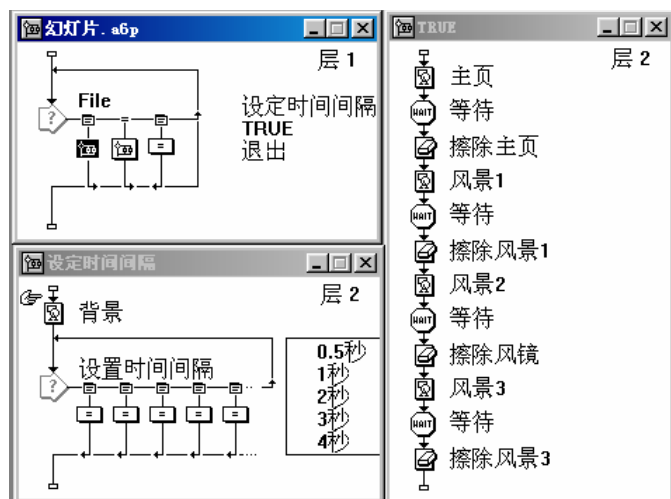


图 4-13 设计好的流程

第5章 【导航】图标

- 【导航】图标简介
- “最近”跳转方式
- “附近”跳转方式
- “任意位置”跳转方式
- “计算”跳转方式
- “查找”跳转方式
- 【导航】图标实例

5.1 【导航】图标简介

在具有交互作用的多媒体作品中，程序根据用户的不同输入或选择而跳转到不同的分支程序中执行，这就是所谓的“超媒体”或“超文本”。所谓“超媒体”或“超文本”，就是通过各种媒体文件如图形、文本等设置链接文件而跳转到另一个媒体文件中。

在 Authorware 中，可以使用【导航】图标来让用户选择不同的分支来控制应用程序的流向。我们可以使用【导航】图标和【框架】图标来实现“超媒体”，实现常见的课件、电子图书中的翻页功能。【导航】图标一般是和【框架】图标配合使用。



1. Authorware 中的跳转方式

【导航】图标允许用户选择不同的路径来控制应用程序的流向，通过【导航】图标的不同类型的导航方式，可以实现在多媒体作品中的不同跳转。

在 Authorware 中，一般有两种类型的跳转方式：

(1) 在 Authorware 文件内部，通过【导航】图标和【框架】图标来实现程序从一个地方跳转到另一个地方。

(2) 使用系统函数让程序从一个 Authorware 文件中跳转到另一个应用程序中，这个应用程序可以是一个 Authorware 程序文件，也可以是其他应用文件。

对于第一种跳转类型，Authorware 可以使用【导航】图标来控制跳转，它可以实现直接自动跳转，如图 5-1 所示。这样程序执行到该【导航】图标时，会自动跳转到“第 2 页”执行。

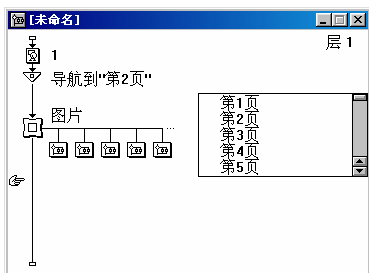


图 5-1 直接自动跳转

也可以通过一个【交互】图标，【交互】图标下面下挂多个【导航】图标，每一个【导航】图标指向【框架】图标中的一页，这样通过用户的选择来决定程序应该跳转到哪个分支，如图 5-2 所示。

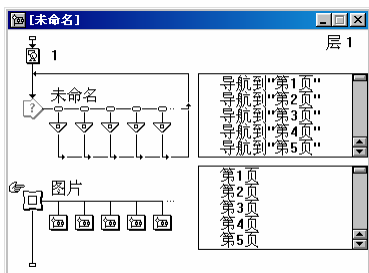


图 5-2 实现用户控制跳转

2. 【导航】图标的五种跳转方式及其对应的【导航】图标的标志

在 Authorware 中，【导航】图标对程序流向的影响有五种方式，对应于不同的程序流向。用户可以通过【导航】图标属性对话框来进行修改跳转方式。

【导航】图标的五种跳转方式：

- “最近”方式：该方式允许用户回溯，即可以返回到用户已经看到过的页面。
- “附近”方式：该方式允许用户在【框架】图标中的页面之间任意跳转和可以退出【框架】图标。
- “任意位置”方式：该方式允许用户跳转到任意【框架】图标中的任意页面中。
- “计算”方式：该方式允许用户设置一个可以返回图标名称的表达式，当 Authorware 程序执行到【导航】图标时，它会根据表达式计算出的图标名称而跳转到相应的页面中。

● “查找”方式：该方式允许用户查找指定的关键字，包含该关键字的页面即是要跳转的页面。
我们必须根据不同的实际情况来决定使用哪种跳转方式。

● 【导航】图标不同的跳转方式对应不同的【导航】图标标志

当用户刚开始拖动一个【导航】图标到流程线上，系统给一个默认名称“Unlinked”，此时【导航】图标是一个空心的三角形，表示该【导航】图标没有和其他图标链接；当用户为一个【导航】图标选择一个目标图标时，Authorware 会自动在【导航】图标上显示出一个标志，表明该【导航】图标的跳转方式与您所选取的图标相链接。在流程线上【导航】图标会反映出跳转方式，所以我们通过【导航】图标的标志就可以知道该【导航】图标的跳转方式。不同跳转方式不同链接对应的不同标志。

5.2 “最近”跳转方式

该方式允许用户回溯，即可以返回到用户已经看到过的页面。

拖动一个【导航】图标到流程线上，将其命名为“跳转”，双击打开其属性对话框，在“目的地”下拉框中选择“最近”选项，属性对话框变为如图 5-3 所示。



图 5-3 选取“最近”方式后的【导航】图标属性对话框

【导航】图标属性对话框中的预览窗口是空白的。

在【导航】图标属性对话框的左下角的白色方框是预览所选取的跳转方式和链接方式后【导航】图标的标志。

在【导航】图标属性对话框中间的大的文本框是该【导航】图标的名称。

“目的地”下拉框是用来控制跳转方式的，如图 5-4 所示。

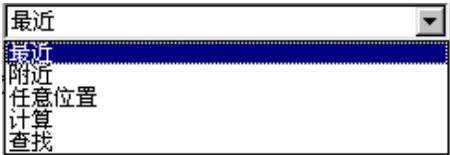


图 5-4 “目的地”下拉框中的选项

该下拉框中的选项即是【导航】图标的五种跳转方式。

“页”选项是用来控制链接页面的方式，它包括两种方式：

- “返回”单选钮：选择该选项，程序会返回到用户刚看过的页面。
- “最近页列表”单选钮：当选择该选项时，程序在执行到该【导航】图标时，会列出用户看过的页面列表，这样用户可以直接双击页面名称返回到该页面来查看该页的内容。

5.3 “附近”跳转方式

该方式允许用户在【框架】图标中的页面之间任意跳转和可以退出【框架】图标。

当在【导航】图标属性对话框的“目的地（跳转目标页方式）”下拉框中选择“附近”选项时，属性对话框变为如图 5-5 所示。



图 5-5 选取“附近”方式后的【导航】图标属性对话框

5.4 “任意位置”跳转方式

该方式允许用户跳转到任意【框架】图标中的任意页面中。

当在【导航】图标属性对话框的“目的地（跳转目标页方式）”下拉框中选择“任意位置”选项时，属性对话框变为如图 5-6 所示。

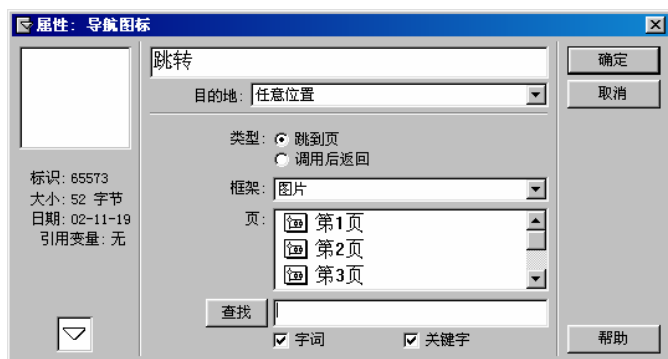


图 5-6 选取“任意位置”方式后的【导航】图标属性对话框

5.5 “计算”跳转方式

该方式允许用户设置一个可以返回图标名称的表达式，当 Authorware 程序执行到【导航】图标时，它会根据表达式计算出的图标名称而跳转到相应的页面中。

当在【导航】图标属性对话框的“目的地”下拉框中选择“计算”选项时，属性对话框变为如图 5-7 所示。

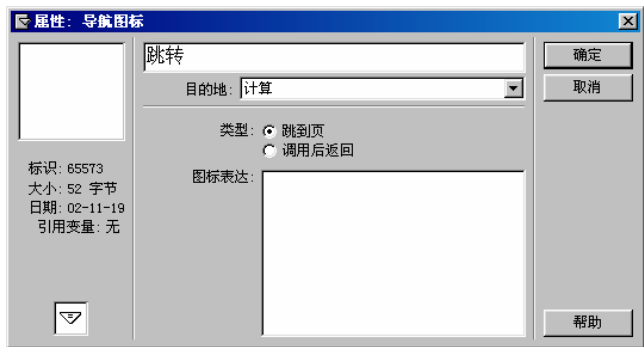


图 5-7 选取“计算”方式后的【导航】图标属性对话框

- “类型”选项中的两个单选按钮与“任意位置”方式中的相同，就不再赘述。
- “图标表达”文本框：在该文本框中输入表达式，程序执行到该处时，会根据表达式的值，返回目标图标的 ID 标识，程序然后跳转到目标图标处继续执行。

5.6 “查找”跳转方式

该方式允许用户查找指定的关键字，包含该关键字的页面即是要跳转的页面。

当在【导航】图标属性对话框的“目的地”下拉框中选择“查找”选项时，属性对话框变为如图 5-8 所示。

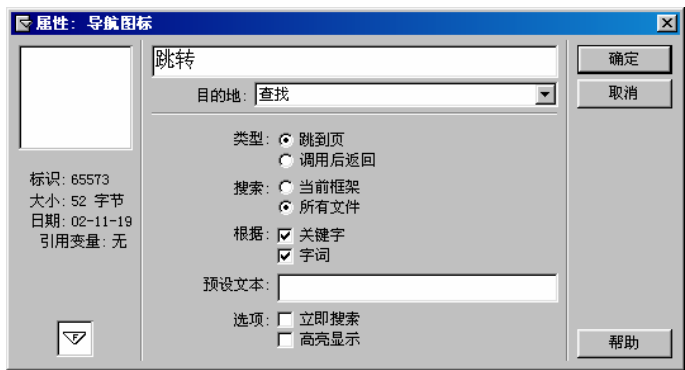


图 5-8 选取“计算”方式后的【导航】图标属性对话框

5.7 【导航】图标实例——制作网络教室

下面我们通过一个实例——网络教室，来说明【导航】图标的使用。

在该实例中，我们使用了【导航】图标实现章节跳转、【框架】图标提供课程、而【交互】图标提供用户选择要阅读的章节，如图 5-9 所示。



- (11) 单击绘图工具箱中的文字工具,在展示窗口中单击一下,然后单击工具栏中的(默认风格)下拉框,从下拉框中选择定义好的“标题”文本风格,然后输入文本“网络教室”。
- (12) 单击绘图工具箱中的指针工具,将“网络教室”文本移动到展示窗口的中上部居中,如图5-11 中所示。

- (13) 拖动一个【显示】图标到流程线上，并将其命名为“显示提示文字”。
- (14) 按住<Shift>键并双击该【显示】图标，打开展示窗口，选择绘图工具箱中的文字工具，并在工具栏的“文本风格”下拉框中选取“正文”文本风格，然后在展示窗口的右侧输入文本“请选择章节”，如图 5-11 所示。



图 5-11 输入提示文字

- (15) 拖动一个【交互】图标到流程线上，并将其命名为“选择章节”。
- (16) 依次拖动三个【导航】图标和两个【计算】图标到“选择章节”【交互】图标的右侧，并且选择“按钮”响应。
- (17) 将这六个下挂图标分别命名为“第 1 讲”、“第 2 讲”、“第 3 讲”、“第 4 讲”和“退出”。
- (18) 拖动一个【群组】图标到【交互】图标的最右侧，然后双击其上面的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“条件响应”选项，在“条件”文本框中输入一个自定义变量“EraseFlags”，然后在“自动”下拉框中选择“为真”选项，表示当变量 EraseFlags 的值为 True 时，执行该交互项。
- (19) 单击响应类型属性对话框中的“确定”按钮，系统弹出新建变量对话框，在该对话框的“初始值”文本框中输入 True，然后单击“确定”按钮确定。
- (20) 拖动一个【框架】图标到【交互】图标的下面流程线上，将其命名为“课程”。
- (21) 拖动四个【群组】图标到“课程”【框架】图标的右侧，依次将各个【群组】图标命名为“1 讲”、“2 讲”、“3 讲”和“4 讲”。
- (22) 双击【交互】图标中的“第 1 讲”【导航】图标，打开【导航】图标属性对话框，此时“目的地”下拉框中已经选择好了“任意位置”选项，选择“跳到页”单选钮，然后选择“页”下拉框中选择“课程”选项（即“课程”【框架】图标），此时下面的页面列表框中列出了“课程”【框架】图标中的四个页面“1 讲”、“2 讲”、“3 讲”和“4 讲”，选中“1 讲”页面，以使该【导航】图标链接到“1 讲”页面上，如图 5-12 所示，然后单击对话框中的“确定”按钮确定。

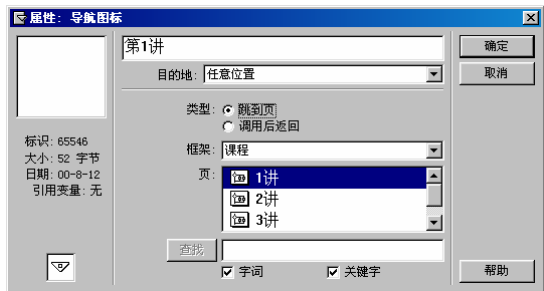


图 5-12 设置“第 1 讲”【导航】图标属性



(23) 同样,按照上面的步骤将“第2讲”【导航】图标和“第2讲”【导航】图标分别链接到“2讲”页面和“3讲”页面上。

(24) 双击“第4讲”【计算】图标,打开其计算编辑窗口,然后在窗口中输入下面语句:

```
GoTo(IconID@"4讲")
```

(25) 双击“退出”【计算】图标,打开其计算编辑窗口,然后在窗口中输入下面语句:

```
Quit()
```

(26) 双击【交互】图标下挂图标“EraseFlags”【群组】图标,打开其二级流程窗口。

(27) 拖动一个【擦除】图标到二级流程线上,将其命名为“擦除提示文字”。

(28) 双击该【擦除】图标,打开其属性对话框,单击对话框中“特效”选项右侧的按钮,打开“擦除方式”对话框,在该对话框的“分类”列表框中选择“推出方式”,在“特效”列表框中选择“从右往左推出展示”,在“周期”文本框中输入“5”,在“平滑”文本框中输入“3”,然后单击“确定”按钮确定。

(29) 回到【擦除】图标属性对话框后,单击其后面展示窗口中的“请选择章节”文本,表示要擦除提示文字。单击“确定”按钮确定。

(30) 拖动一个【计算】图标到【擦除】图标下面,将其命名为“取消擦除”。

(31) 双击该【计算】图标,打开计算编辑窗口,在窗口中输入语句:

```
EraseFlags:=FALSE
```

(32) 双击“课程”【框架】图标下面的“1讲”【群组】图标,打开其二级流程窗口。

(33) 拖动一个【显示】图标到二级流程线上,并将其命名为“显示1讲”。

(34) 按住<Shift>键并双击该【显示】图标,打开展示窗口,选取文字工具,在展示窗口上单击一下,然后在工具栏的“文本风格”下拉框中选取“正文”文本风格,并选择菜单[文本]→[卷帘文本]命令,然后在展示窗口的右侧输入文本:

(35) 单击指针工具,将输入的文本移动到展示窗口的中央,如图5-13所示。

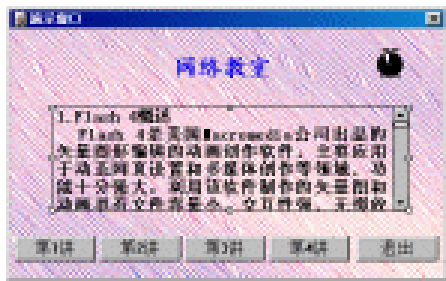


图 5-13 输入“1讲”内容

(36) 拖动一个【等待】图标到二级流程线上,将其命名为“等待”。

(37) 双击【等待】图标,打开其属性对话框,选中“单击鼠标”复选框,选中“按任意键”复选框,在“时限”文本框中输入“15”,并选中“显示倒计时”复选框,最后单击“确定”按钮确定。

(38) 拖动一个【计算】图标到二级流程线上,将其命名为“返回”。

(39) 双击【计算】图标,打开其计算编辑窗口,在窗口中输入语句:

```
GoTo(IconID@"选择章节")
```

(40) 将该二级流程窗口中的所有三个图标全部选中,然后单击工具栏中“复制”按钮,将它们复制到剪贴板上。

(41) 双击“课程”【框架】图标下面的“2讲”【群组】图标,打开其二级流程窗口。

(42) 单击工具栏中的“粘贴”按钮，将复制的三个图标粘贴到该二级流程线上。

(43) 将“显示 1 讲”【显示】图标更名为“显示 2 讲”。

双击该【显示】图标，打开展示窗口，选择文字工具，然后在展示窗口中选中“1 讲”的文本内容，并将其内容改变。

(44) 同样, 将复制的三个图标分别粘贴到“课程”【框架】图标下面的“3 讲”【群组】图标、“4 讲”【群组】图标的二级流程窗口中。

(45) 将“3 讲”【群组】图标二级流程窗口中的【显示】图标更名为“显示 3 讲”，打开该【显示】图标，在展示窗口中并将“1 讲”的内容改变。

(46) 同样, 将“4 讲”【群组】图标二级流程窗口中的【显示】图标更名为“显示 4 讲”, 打开该【显示】图标, 在展示窗口中并将“1 讲”的内容改变。

(47) 此时，整个程序设计完成，完成后的流程如图 5-14 所示。

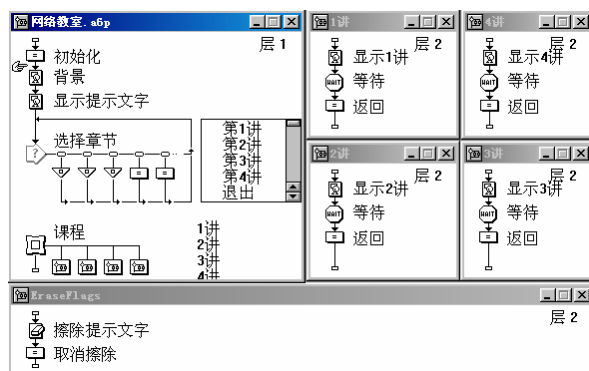


图 5-14 设计好流程

关于【导航】图标的其他几种跳转方式，将在【框架】图标中使用，所以不再列举实例。

第6章 【框架】 【群组】 图标

- 【框架】图标简介
- 【框架】图标实例
- 【群组】图标的特点
- 举例说明【群组】图标的使用

6.1 【框架】图标简介

在上一章中，我们提到了【框架】图标，Authorware 提供的【框架】图标可以实现页面之间随意浏览跳转功能的图标。许多多媒体课件、电子图书等中的结构都可以使用【框架】图标来实现。

6.1.1 【框架】图标的使用

通常，【框架】图标下面会下挂若干个图标，这些下挂图标称为页面，如图 6-1 所示，第一个下挂图标称为第一页，第二个下挂图标称为第二页，……最后一个下挂图标称为最后一页。整个下挂图标通常称之为框架结构。下挂图标可以是【显示】图标、【移动】图标、【擦除】图标、【等待】图标、【导航】图标、【计算】图标、【群组】图标、【数字电影】图标、【声音】图标、【视频】图标等，下挂图标中的内容可以是文本、图形、数字电影、声音或动画等。

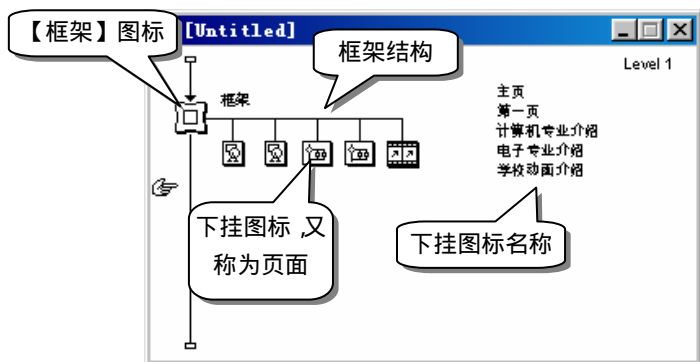


图 6-1 【框架】图标

6.1.2 【框架】图标属性对话框

选取流程线上的【框架】图标，选择菜单【修改】→【图标】→【属性】命令，打开【框架】图标属性对话框，如图 6-2 所示。

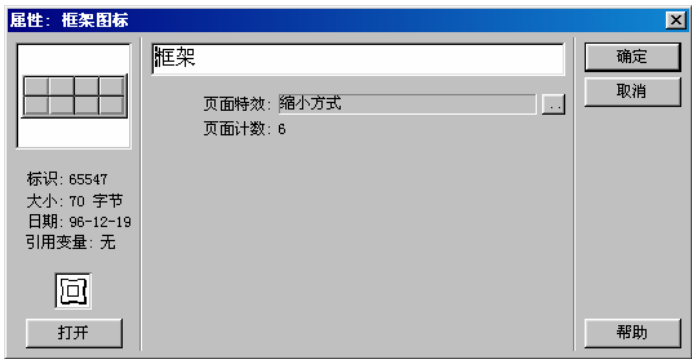


图 6-2 【框架】图标属性对话框

在【框架】图标属性对话框的左上角是导航面板预览窗口，中间的文本框中文字是该【框架】图标的名称。

“页面特效”中显示的是【框架】图标下挂页面之间相互跳转时的过渡效果，如果要改变默认的页面过渡效果，可以单击“页面特效”右侧的...按钮打开“页特效方式”对话框，如图 6-3 所示。

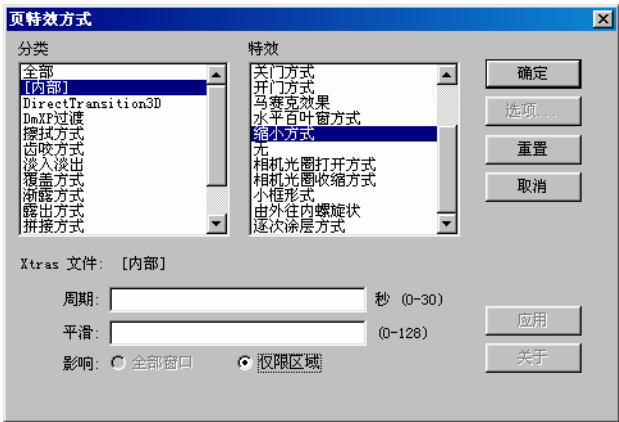


图 6-3 页面过渡效果对话框

页面过渡效果对话框与其他过渡效果对话框的使用相同，在此就不再赘述。
在“页面特效”下面有一句提示信息“页面计数：6”，表示该【框架】图标的下挂图标有 5 个。
单击【框架】图标属性对话框左下角的“打开”按钮，可以打开该【框架】图标的内部结构流程。

6.1.3 【框架】图标的内部结构

拖动一个【框架】图标到流程线，双击它打开该【框架】图标，如图 6-4 所示。

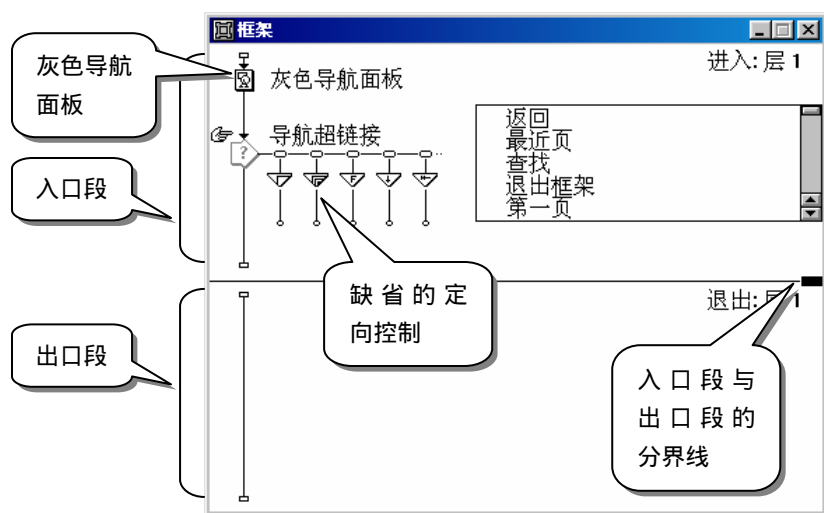


图 6-4 【框架】图标内部结构

【框架】图标在流程线上表现为一个图标，但是它实际上是由一组图标组成的。在【框架】图标内部也有一条流程线。当程序运行时遇到【框架】图标，就进入【框架】图标内部继续执行。

在【框架】图标的内部结构的入口段中，第一个【显示】图标是显示出跳转控制导航面板，双击打开展示窗口，显示出灰色导航面板，如图 6-5 所示。



图 6-5 灰色导航面板

下面是一个【交互】图标，它是用来提供用户要浏览的页面控制。【交互】图标下挂有八个【导航】图标，八个【导航】图标显示为八个控制按钮，用于导航作用，如图 6-6 所示。



图 6-6 八个导航按钮的功能

其实，这八个导航按钮执行的功能其实就是【导航】图标中不同的跳转方式。用户可以双击这些【导航】图标，打开其属性对话框，看一下其跳转方式，也可以在对话框中更改其跳转方式。

程序设计人员可以根据自己的需要来更改导航面板的样式和控制按钮的形状，也可以根据自己的需要来增加或减少控制按钮的个数和其在导航面板中的位置。要增加或减少控制按钮的个数，只需要改变【框架】图标内部结构中的【交互】图标下挂【导航】图标的个数即可。要改变控制按钮在导航面板中的位置，只需单击该控制按钮上方的响应类型图标，然后用鼠标直接拖动到合适位置即可。要更改控制按钮的形状，需要用到“按钮”对话框，关于按钮形状的设置和改变控制按钮在展示窗口中的位置，将

在第 10 章【交互】图标中详细讲解。用户可以在入口段中加入一些图标，让程序进入到【框架】图标时，首先执行这些图标。

入口段和出口段之间有一条分界线，用户可以拖动分界线右边的黑色矩形来改变入口段和出口段的大小。

出口段中的内容默认为空，当然用户也可以在出口段中加入一些图标，使得退出【框架】图标时执行这些内容。

需要注意的是，当使用【导航】图标或其他跳转函数跳转到【框架】图标中的某一页，程序是首先执行【框架】图标内部结构中的入口段部分内容，然后才跳转到目标页面去执行。同样，如果通过【框架】图标导航面板中的“退出当前【框架】图标”按钮或跳转函数退出当前【框架】图标，程序是先【框架】图标内部结构中的出口段部分内容，然后才转到目标图标继续执行。

6.2 【框架】图标实例——制作相册

下面我们通过一个实例——制作相册，来说明【框架】图标的使用。

在该实例中，我们使用了【框架】图标提供各个页面，每一个页面就是相册中的一幅照片，通过【框架】图标中的控制按钮来查看各个图片，如图 6-7 所示。



图 6-7 相册

操作步骤为：

- (1) 新建一个文件，命名为“相册.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“根据变量”选项，在“属性”标签页中取消“菜单栏”复选框，并将背景色设置为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“Add”按钮，增加一种新的文本风格，将该文本风格更名为“相册字体”，选择字体为楷体 GB_2312、字号为 18，粗体，字色为黑色，然后单击“修改”按钮确定修改。如图 6-8 所示。



图 6-8 定义新文本风格

(5) 同样, 单击定义文本风格对话框中的“添加”按钮, 增加一种新的文本风格, 将该文本风格更名为“正文字体”, 选择字体为楷体_GB2312、字号为 12, 粗体, 斜体、字色为蓝色, 然后单击“修改”按钮确定修改。

(6) 同样, 单击定义文本风格对话框中的“添加”按钮, 增加一种新的文本风格, 将该文本风格更名为“按钮字体”, 选择字体为宋体、字号为 12, 粗体, 字色为粉红色, 然后单击“修改”按钮确定修改。

(7) 设置好文本风格后, 单击定义文本风格对话框中的“完成”按钮确定。

(8) 拖动一个【计算】图标到设计窗口的流程线上, 并将其命名为“初始化”。

(9) 双击该【计算】图标, 打开其计算编辑窗口, 然后在窗口中输入语句:

```
ResizeWindow(450, 300)
```

(10) 拖入一个【显示】图标到流程线上, 将其命名为“背景”。

(11) 双击该【显示】图标, 打开展示窗口, 单击工具栏中的“导入图片”按钮, 从外部导入一幅图片, 并调整其大小。

(12) 选取绘图工具箱中的文字工具, 在图片上单击一下, 并单击工具栏中的[默认风格]“文本风格”下拉框, 从下拉框中选择定义好的“正文字体”文本风格, 然后输入文本。

(13) 在图片的下方单击一下, 选择“文本风格”下拉框中的“相册字体”, 然后输入文本“白雪影集”。

(14) 选取绘图工具箱中的指针工具, 将两个文本对象进行调整, 如图 6-9 所示。

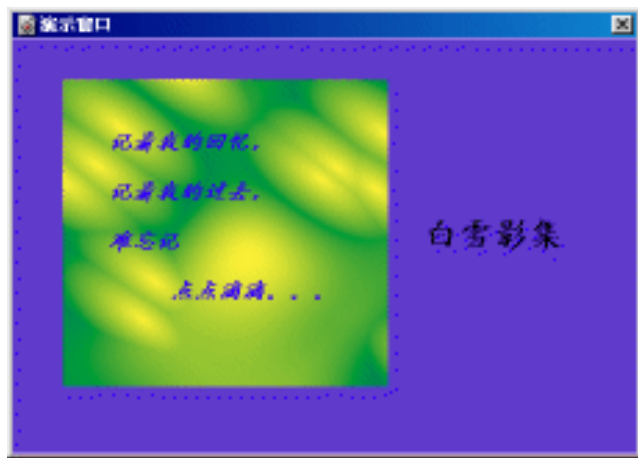


图 6-9 输入相册背景信息

- (15) 拖动一个【等待】图标到流程线上，将其命名为“等待”。
- (16) 双击该【等待】图标，打开其属性对话框，在“时限”文本框中输入“1”，然后单击“确定”按钮确定。
- (17) 拖动一个【擦除】图标到流程线上，并将其命名为“擦除背景”。
- (18) 双击【擦除】图标，打开其属性对话框，单击“特效”后面的...按钮打开“擦除方式”对话框，在擦除过渡效果对话框中选择“填充方式/像素展示”，在“周期”文本框中的数值改为“5”，其他设置不进行改变，单击“确定”按钮确定。
- (19) 单击【擦除】图标属性对话框后面展示窗口中的背景图片，表示要擦除背景信息。单击对话框中的“确定”按钮完成【擦除】图标属性的设置。
- (20) 拖动一个【框架】图标到流程线上，将其命名为“相册集”。
- (21) 选择菜单[修改]→[图标]→[特效...]命令，打开“页特效方式”对话框，在页面过渡效果对话框中选择“填充方式/像素展示”，其他设置不进行改变，单击“确定”按钮确定。
- (22) 拖动六个【显示】图标到【框架】图标的右侧，并将它们分别命名为“照片1”、“照片2”、“照片3”、“照片4”、“照片5”和“照片6”。
- (23) 双击“照片1”【显示】图标，打开展示窗口，导入外部照片图片，并将其调整在展示窗口的左侧。
- (24) 同样，在“照片2”、“照片3”、“照片4”、“照片5”和“照片6”中分别导入其他外部照片图片。
- (25) 双击【框架】图标，打开【框架】图标，展开其内部结构。
- (26) 将“灰色导航面板”【显示】图标更名为“按钮说明”。
- (27) 删除“导航超链接”【交互】图标中的“返回”、“最近页”、“查找”和“退出框架”下挂子【导航】图标。
- (28) 拖动一个【计算】图标到“导航超链接”【交互】图标的最右侧，将其命名为“Quit”。
- (29) 双击该【计算】图标，打开其计算编辑窗口，在窗口中输入语句“Quit()”。
- (30) 双击该【计算】图标上面的响应类型图标，打开响应类型属性对话框，单击对话框中的“按钮”按钮，打开“按钮(按钮)”对话框。
- (31) 在“按钮(按钮)”对话框中选择...按钮，然后单击“确定”按钮确定。
- (32) 单击“确定”按钮关闭响应类型属性对话框。

(33) 按住<Shift>键并双击“导航超链接”【交互】图标，打开展示窗口，将各个按钮进行排列对齐，如图 6-13 中所示。

(34) 按住<Shift>键并双击“按钮说明”【显示】图标，打开展示窗口，将导航面板删除。

(35) 选择文字工具，在展示窗口中单击一下，然后选择“文本风格”下拉框中的“按钮字体”，然后输入文本“首页”。

(36) 然后分别单击输入文本“上页”、“下页”、“尾页”和“退出”。

(37) 选中这五个文本对象，将其分别与各自对应的按钮对齐，如图 6-10 中所示。

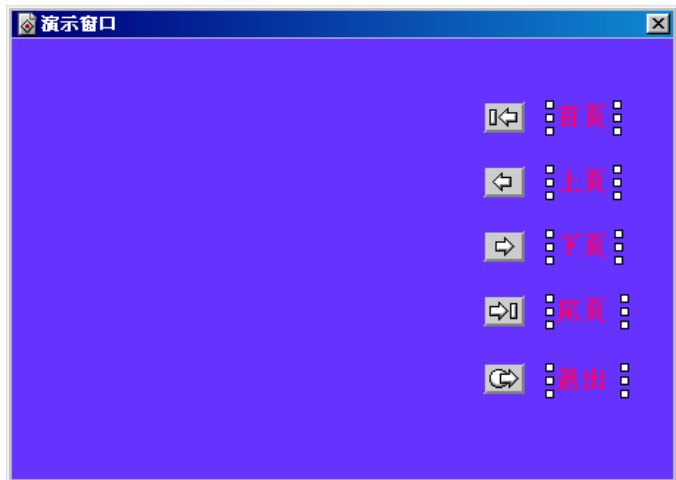


图 6-10 对齐按钮与按钮文本

(38) 程序的流程已经设计完成，设计好流程如图 6-11 所示。

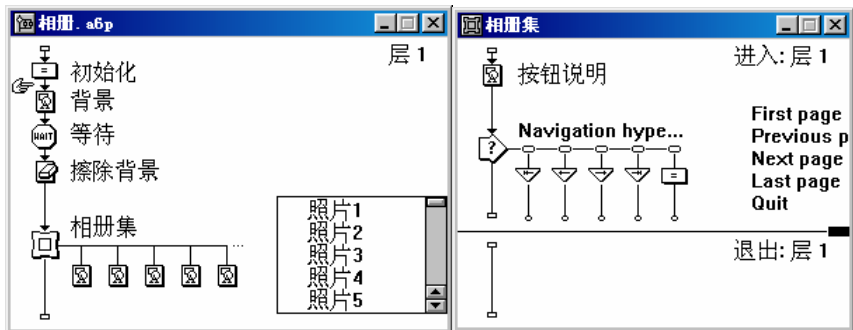


图 6-11 设计好的流程

6.3 【群组】图标简介

6.3.1 【群组】图标简介

当用户在设计一个多媒体作品时往往会遇到这样的问题：主流程线上的图标太多而程序窗口的大小是有限的，以至于处于流程线上下面的图标就会逐渐推到窗口之外而见不到了，我们在程序窗口中无法看到全部图标，这里我们将利用系统所提供的【群组】图标把流程线上的相关且联系紧密的部分图标组

合起来，从而达到缩短流程线，同时使结构更加紧骤的目的。

【群组】图标又称 Map Icon，可以将一系列图标集合在一起，你可以将程序流程片段组合进便利的模块，无论其中包含多少图标都可使作品看起来更有条理。可将【群组】图标放置在流程线上的任何位置，与【交互】图标、【决策】图标或者【框架】图标配合使用。同时你可以在【群组】图标中再放置二级【群组】图标。

● 【群组】图标的特点

每个【群组】图标都有它自己的流程线，双击图标可以打开其流程，在右上角显示层级（使用它用来创建流程等级——层级说明用来表明流程及该等级中【群组】图标的层次）。当系统运行到【群组】图标时，它将按【群组】图标中图标的顺序执行，当执行完最后一个图标时，系统将退出【群组】图标继续执行主流程线上下一个图标。

如果【群组】图标是隶属于一个【交互】图标，Authorware 在执行下一个图标前将执行【群组】图标所包含的所有内容。当它是属于一个【决策】图标的一条路径或是【框架】图标的一页时，Authorware 也将执行【群组】图标中所包含的所有内容。

有时我们要把【群组】图标中的所有图标解开到上一级程序窗口的流程线上，可以在流程线上选中该图标，选择菜单[修改]→[群组]命令，也可以用<Ctrl>+<Shift>+G 快捷键。

6.3.2 【群组】图标属性对话框

打开一个程序，然后按住<Ctrl>键并双击一个【群组】图标，或者选中一个【群组】图标，然后按下快捷键<Ctrl>+I，可以打开该【群组】图标属性对话框，如图 6-12 所示。

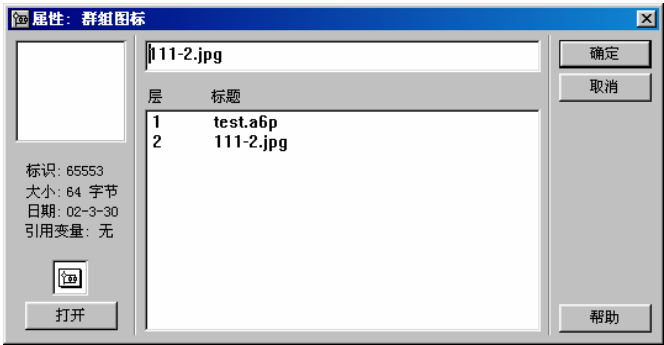


图 6-12 【群组】图标属性对话框

属性对话框中列出的是该【群组】图标所处的图层和标题名，以及该【群组】图标的图层名和标题名。

属性对话框的预览窗口始终为空。

6.3.3 【群组】图标的组合、取消组合以及层号

当某一层设计窗口中图标太多时，我们需要将其中一部分图标进行组合，即将它们组合成一个【群组】图标。

首先选择流程线上要组合的图标，有两种方法：

- 可以使用鼠标拖动法，按住鼠标左键并在设计窗口中拖动，拖动出一个虚线矩形框，如图 6-13 所示。



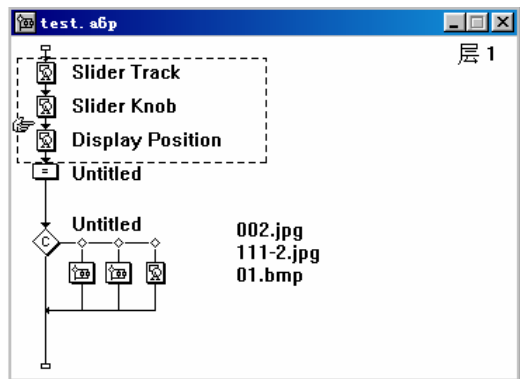


图 6-13 拖动一个虚线矩形框

松开鼠标后，位于矩形框中的图标都会被选中。

- 可以按住<Shift>键，然后一个一个单击要组合的图标，将图标一个一个选中。如果选中了不要组合的图标，可以按住<Shift>键再次单击一下该图标，该图标将取消选中状态。

当选中要组合的图标后，可以选择菜单【修改】→【群组】命令，或者直接按下快捷键<Ctrl>+G 将这些图标组合，组合后图标变成一个名称为“未命名”的【群组】图标，如图 11-14 所示。

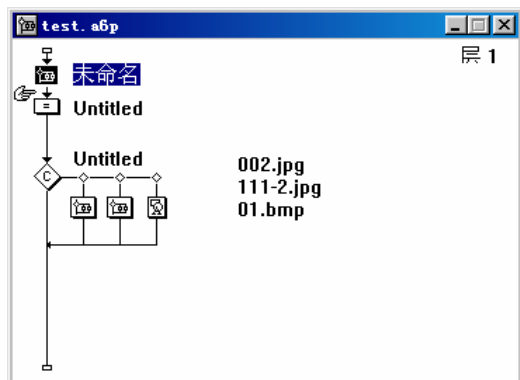


图 6-14 组合后的程序流程

当然，我们也可以先将选中的要组合的图标剪切，然后在该处拖入一个【群组】图标，双击该【群组】图标，打开下一层设计窗口，然后单击工具栏中的“粘贴”按钮，将剪切的图标粘贴到该【群组】图标中。显然，这种方法进行图标组合是比较烦琐的，而且，如果我们突然忘记了而先去复制或粘贴的其他内容，原来剪切的图标就无法恢复了。

如果我们希望将一个【群组】图标中的内容取消组合，可以首先选中该【群组】图标，然后选择菜单[修改]→[取消群组]命令，或者直接按下快捷键<Ctrl>+<Shift>+G 将该【群组】图标取消分组。

由于【群组】图标的使用，使得设计窗口可以分为很多个层，层是按照从主设计窗口为第 1 层，主设计窗口中的【群组】图标打开后是第 2 层，第 2 层中的【群组】图标打开后是第 3 层，……，第 n 层中的【群组】图标打开后是第 n+1 层。我们可以通过设计窗口左上角的提示，或者该【群组】图标属性对话框中的信息知道该窗口所处的层号。

第7章 【判断】图标

- 【判断】图标简介
- 【判断】图标实例

7.1 【判断】图标简介

7.1.1 【判断】图标的流程特点

【判断】图标和【交互】图标有一些相似，如图 7-1 所示。

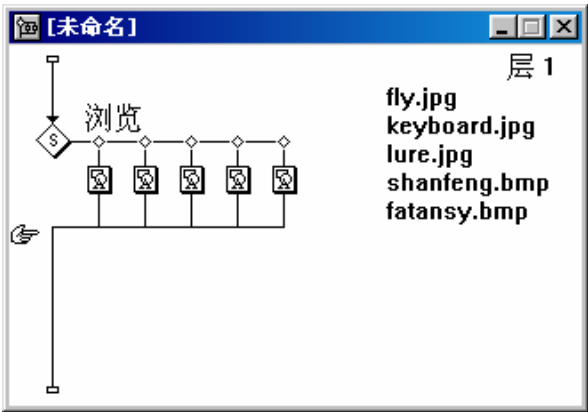


图 7-1 使用【判断】图标

在图 7-1 中，我们可以看到【判断】图标和【交互】图标都是通过其下挂子图标实现起功能的，通过对【判断】图标有关属性的相应设置也能够完成【交互】图标的某些功能。但是，我们知道，【交互】

图标的主要是通过按钮、热区、热体、条件响应等十一种类型和用户进行交互而进入各自的分支的，【判断】图标则主要是以顺序执行的方式依次进入每一个分支，然后在进入下一个流程。

7.1.2 【判断】图标属性对话框

开始一个新文件，拖一个【判断】图标到流程线上，双击【判断】图标，打开【判断】图标属性对话框，如图 7-2 所示。

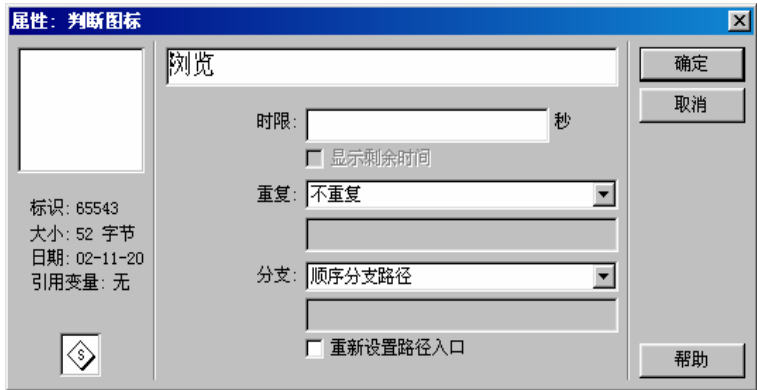


图 7-2 【判断】图标属性对话框

下面我们对【判断】图标属性对话框中的一些选项予以介绍。

- 在【判断】图标属性对话框中，最上面的文本框是该【判断】图标的名称，左边分别显示了该图标的 ID 号、大小、最近修改时间以及是否引用了变量等信息。
- “时限”文本框：在这里可设定一个时间值，决定执行此图标的时间，时间一到，不管此分支是否执行完，程序都将会中断当前的执行，跳到下一个流程去。
- 当用户设定了“时限”选项后，“显示剩余时间”复选框被激活。选中此项，在展示窗口屏幕的左下角出现一个显示当前剩余时间的小闹钟。
- “重复”下拉框：共有五种类型，如图 7-3 所示。

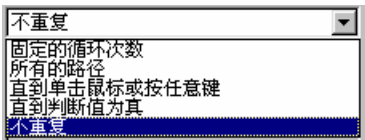


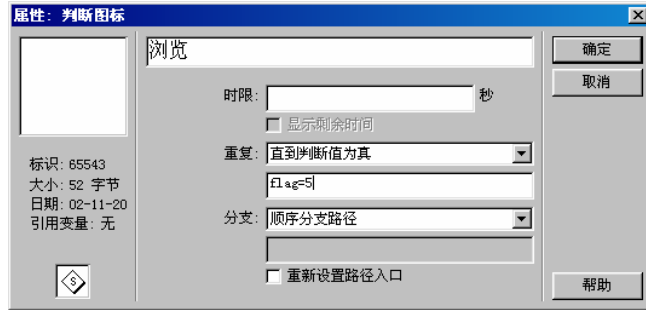
图 7-3 “重复”下拉框中选项

“固定的循环次数”选项：这一项可输入一个数值、变量、表达式以决定进入【判断】图标后，重复执行分支的次数。如果这个值小于 1，Authorware 将会不执行而越过此【判断】图标；如果这个值大于分支数目，程序将会顺序完执行分支，然后再从头执行分支直到分支的总数等于这个值为止。

“所有的路径”选项：在退出【判断】图标之前，所有的分支将会至少被执行一次。

“直到单击鼠标或按任意键”选项：反复执行分支直到用户按下一个键或者单击鼠标。由此用户可以用一组连续的差别不大的图片作为分支，以实现动画的效果。

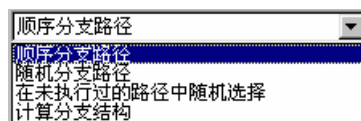
“直到判断值为真”选项：选中此项后，在“重复”下拉框下边的文本框被激活，如图 7-4 所示。



用户可以输入一个变量或者表达式,在每次执行分支前,先判断变量或者表达式的值是否为“真”,若不为“真”,就继续执行分支,否则退出此【判断】图标。例如在上图输入的是表达式 flag=5,执行分支前如果此表达式不为“真”,将继续执行,否则跳出【判断】图标进入下一个流程。

“不重复”选项：进入【判断】图标后，执行一次用户所选的分支，然后退出【判断】图标。用户可控制流程再进入【判断】图标，此时将根据分支项的选择执行相应的分支。

- “分支”下拉框：用来定义控制分支类型，共有四种类型，如图 7-5 所示。



“顺序分支路径”选项：从第一个分支开始顺序执行完每一个分支。

“随机分支路径”选项：进入【判断】图标后，将对下挂的判断分支随机的执行。由此其中的某一个分支可能会重复执行多次，而某一个分支却得不到执行。

“在未执行过的路径中随机选择”选项：随机的执行分支，而且这种随机执行限制在了还没有执行过的分支之间。

“ 计算分支结构 ” 选项：当用户选择了此项，“ 分支 ” 下拉选项下面的文本框变为可选的状态，可以在里面输入条件变量或者条件表达式，程序将根据这个值决定执行哪一条分支。选择此项，用户需要了解【判断】图标分支是怎样编号的。Authorware 中把【判断】图标的下挂分支从左到右顺序编号，最左边定义为 1，然后以此类推。如果值为 1，则执行第一个分支，为 2，则执行第二个分支。

选择不同的分支类型，【判断】图标呈现出不同的图标显示方式，见表 7-1 所示。

表7-1 图标显示方式和分支类型对应表

图标显示方式	分支类型
	“ 顺序分支路径 ” 选项
	“ 随机分支路径 ” 选项
	“ 在未执行过的路径中随机选择 ” 选项
	“ 计算分支结构 ” 选项

- “重新设置路径入口”复选框：如果用户选择了此项前面的复选框，当程序再次进入【判断】图标后，Authorware 将会不理睬前面记忆的内容，所有和判断分支有关的变量都将恢复初始值，就好像从没有执行过这些分支一样。例如，用户在“分支”框中选择了“计算分支结构”选项，系统将会擦除已经执行过的分支的信息记录。

7.2 【判断】图标实例

这个实例相对来说要稍微的复杂一点，程序随机的显示出十幅图画，要求拼写出对应的单词。这是一个必答类型的应用，也就是要求回答所有的十个问题。

当创建多媒体教学软件时，用【判断】图标管理试题库是一个很好的办法。用户通过学习这个实例，可以更好的理解【判断】图标提供的“重复”和“分支”下拉各选项的设置的含义，以及它们一些组合的应用。下面先简要的介绍一下这个实例所实现的功能。

首先出现的是一个“看图写单词”的背景画面，也注明了作者的名称。等 2 秒后或者点击鼠标、按下任意键后进入测试。总共有十道题，且每次测试时，这十道题的出现顺序是随机的。先是出现一个画面，在规定的时间内要求你输入一个单词（以左下角的时钟为准），然后回车确认，如果答对会出现“答对”的字样，同时会响起表示正确的音乐声；如图 7-6 所示。



图 7-6 程序界面 1

回答错误则响起一阵爆炸声，同时出现“错误”字样和正确的提示，如图 7-7 所示。

然后进行下一道题。测试完后将进行有关测试成绩的统计，不同的成绩会有不同的统计结果，并有相应的描述，等两秒后，弹出一个对话框，询问你是否要再试例如，如果一个也没有对，将会出现如图 7-8 所示的画面。

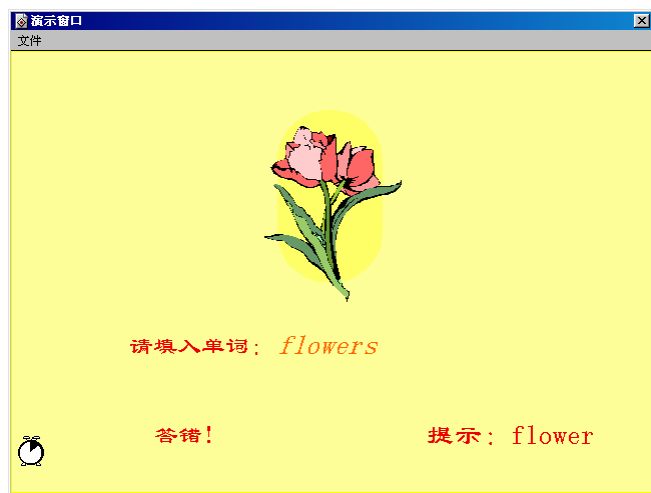


图 7-7 程序界面 2

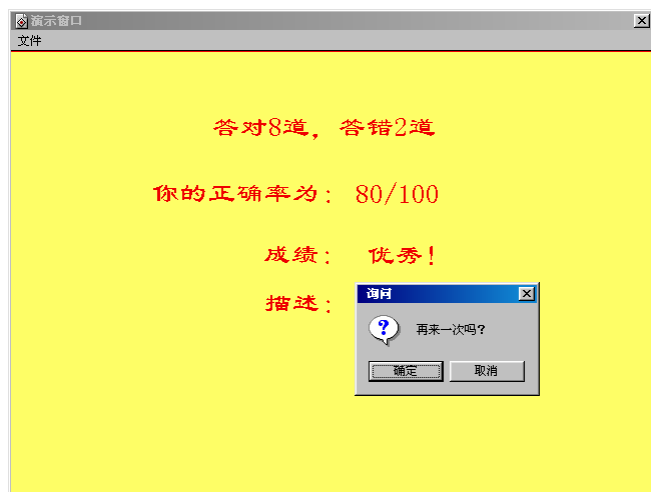


图 7-8 统计界面

由上图可知统计信息包括答对和答错的数目,也有正确率和成绩的计算,并有根据这些统计数据进行的描述。其中成绩将分为五个等级,一个也没答对将是“0”;答对的数目少于六个,等级为“极差”,描述是“很遗憾,不及格!”;多于六个而少于八个,是“及格”,描述是“你仍须努力!”;对了八个或者九个,等级为“优秀”,描述是“不错!”;第五个等级是“满分!”,描述改为“太完美了!”,此时还会响起一阵掌声表示鼓励。

下面我们进行程序的设计。

(1) 开始一个新文件,保存项目文件为“测试.a6p”。

(2) 拖动一个【显示】图标到流程线上放开,命名为“背景”,双击【显示】图标打开其展示窗口,导入图片并用绘图工具写入文字。

(3) 出现开始画面后需要一个时间停顿,加进一个【等待】图标,命名为“等 2 秒”,双击打开属性对话框,设置属性如图 7-9 所示。

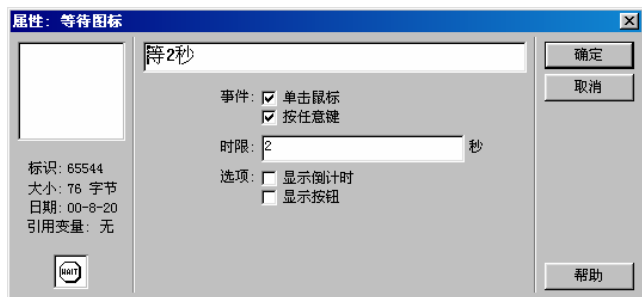


图 7-9 设置【等待】图标属性

(4) 在流程线下方加入一个【计算】图标，改名为“初始化”，双击后在出现的计算编辑窗口内加入代码如下：

```
right:=0
string1:=""
path:=0
score:=""
sting2:=""
string3:=""
```

(5) 再加入一个【群组】图标，名为“跳转点”，这个图标不作其他的用途，只是一个跳转的标志，上面介绍过程序在结束之前将会询问是否再进入测试，如果回答了“是”，程序就跳回到这里继续往下执行。到这里，我们已经完成了最初的工作，下面就将进入用【判断】图标控制出题的阶段了。目前完成的流程图如图 7-10 所示。



图 7-10 初步设计的程序流程

(6) 加入一个【判断】图标，命名为“出题”，双击打开【判断】图标属性对话框，设置属性如图 7-11 所示。

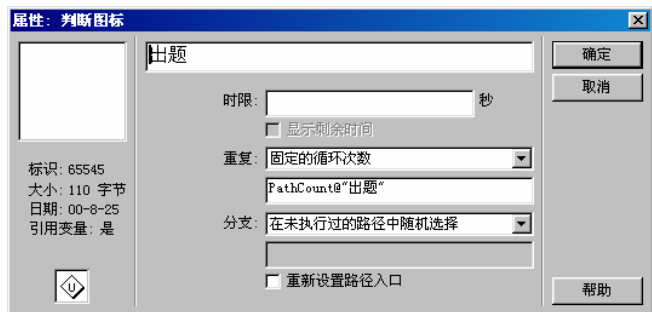


图 7-11 设置【判断】图标属性

其中“重复”下拉框中选择了“固定的循环次数”选项，下面的文本框里输入的是函数“PathCount@”



出题 ” ，这个函数返回的是【判断】图标 “ 出题 ” 的下挂分支数，表示此判断图标将随机不重复的执行所有的分支。这里最值得注意的是 “ 分支 ” 选项， “ 在未执行过的路径中随机选择 ” 使得【判断】图标在随机选择了一个分支后，接下来再随机选择一个没有进入过的分支继续执行，这是实现本例的关键。另外，在流程线上，用户还可以注意到【判断】图标中的字符显示，【判断】图标中的字符显示是和 “ 分支 ” 下拉框中的下拉选项密切相关的，例如在本例中，由于在 “ 分支 ” 的下拉选项选择了 “ 在未执行过的路径中随机选择 ” ，因此【判断】图标显示为 “ U ” ；如果选择了 “ 顺序分支路径 ” 选项，【判断】图标中将显示为 “ S ” 。以此类推，其他的选项亦会有相对应的字符显示。读者可以依次选择来看一下。

（7）设置了【判断】图标的属性过后，接下来要完成判断分支的流程线的设计。加入一个【群组】图标作为第一个分支，命名为 “ 第一题 ” ，双击【群组】图标进入二级流程线。

（8）拖动一个【计算】图标到二级流程线上，将其命名为 “ 初始化 string ” 。双击该【计算】图标，在打开的计算编辑窗口中输入如下语句：

```
string1:=""
string3:=""
--初始化这两个变量，避免其他题目的影响
```

（9）加入 “ 背景 1 ” 【显示】图标，在双击后的展示窗口中导入外部图形，并输入文本对象，如图 7-12 所示。

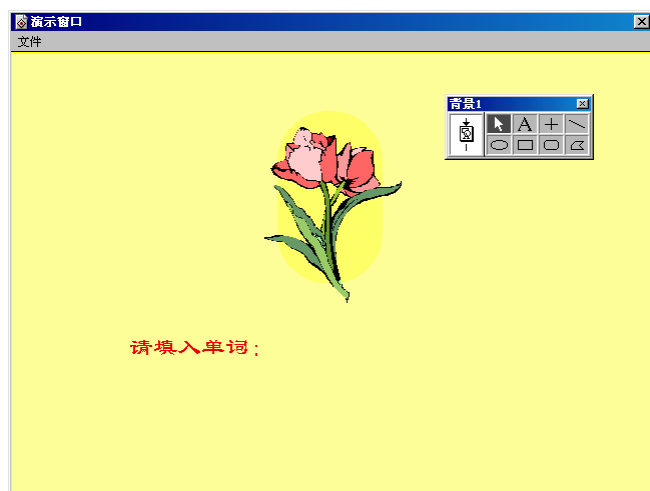


图 7-12 建立习题 1

（10）建立文本交互机制，接受用户输入的单词并进行判断。加入【交互】图标，名为 “ 填字 1 ” ，再拖动一个【群组】图标作为交互分支，这时将会出现 “ 交互类型 ” 对话框，选择 “ 文本输入 ” 后单击 “ 确定 ” 按钮确定。

（11）将该【群组】图标命名为 “ * ” 。为了能够接受任意的字符，此【群组】图标必须命名为通配符 “ * ”

（12）双击打开【群组】图标，在三级流程线上再加入一个【判断】图标，名为 “ 判断 ” ，也就是说此【判断】图标的功能是对输入的文本进行判断，以显示出相应的提示信息和音乐。

（13）在【判断】图标的右边加入两个【群组】图标分支，分别是 “ 对 1 ” 、 “ 错 1 ” ，由它们的意思，读者可以知道它们是对正确答案和错误答案的处理，为了选择性的进入这两个分支，相信读者也想到了，这需要设置【判断】图标的属性。双击【判断】图标打开【判断】图标属性对话框，并且设置如图 7-14 所示。



图 7-13 设置【判断】图标属性

“分支”下拉框中选择“计算分支结构”选项，表明了是依照所计算的值进入相应的分支，下面的文本框里将判断输入的文本是否是字符串“flower”，如果是，进入分支 1，否则进入分支 2。函数 Test 分别返回的是这两个分支的值。

(14) 双击打开【群组】图标“对 1”，加入一个【计算】图标，打开其计算编辑窗口，输入代码如下：

```
right:=right+1
--保存正确回答的题目个数
string1:="答对！"
--字符串“string1”记下了“答对”信息。
```

(15) 加入一个【声音】图标，名为“对的声音”，双击打开属性对话框，再加入声音文件。

(16) 同样双击【群组】图标“错 1”，加入【计算】图标和【声音】图标，在计算图标里加入代码如下：

```
string1:="答错!"
string3:="提示：flower"
--提示信息
```

现在可以运行程序来看一看效果，首先出现开始画面，按鼠标或者任意键，或者等 2 秒后，进入测试题。马上按<Ctrl>+P 快捷键使程序暂停，调整所出现的文本框到合适的位置，确保将要输入单词的地方紧跟在字样“请输入单词”的后面，如图 7-14 所示。

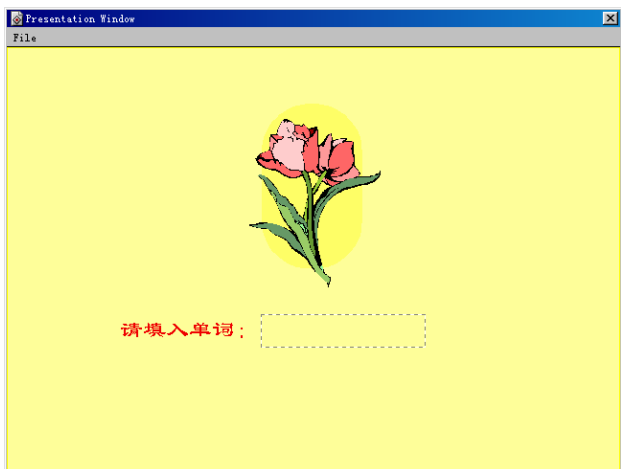


图 7-14 调整各个文本对象和图形

刚才介绍过变量 string1 和 string3，它们的作用如图 7-15 所示。

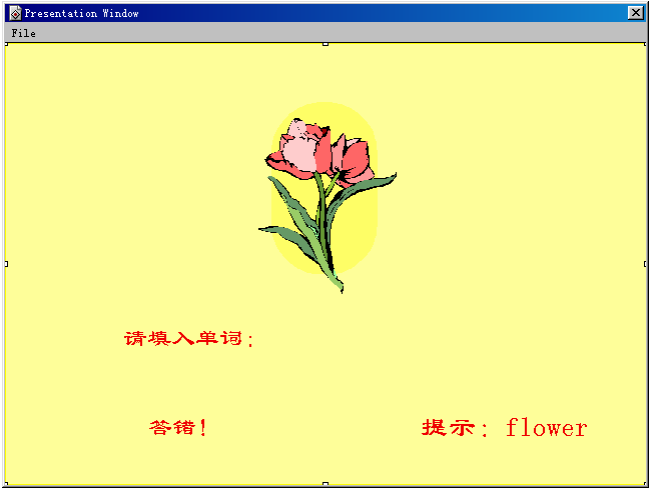


图 7-15 两个变量的作用

(17) 为了能显示 string1、 string2 所表示的值，双击“背景 1”【显示】图标，添加如图 7-16 所示。

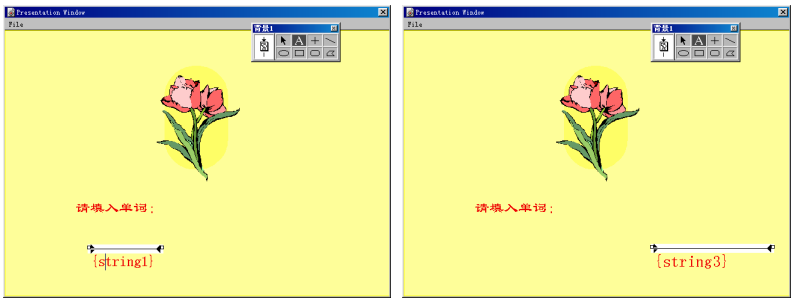


图 7-16 添加两个变量

(18) 选择菜单[修改]→[图标]→[属性...], 在弹出的【显示】图标属性对话框中，选择“显示”标签页，单击特效选项后的“...”按钮，打开“特效方式”对话框。

(19) “分类”下选择的是“内部”，在相应的“特效”栏中选择“水平百叶窗式”，这些是定义图片间转换的过渡类型。然后单击“确定”按钮返回到【显示】图标属性对话框。在对话框中，选择“显示”选项中的“更新变量显示”复选框。

(20) 添加一个【群组】图标到“填字 1”【交互】图标右侧，“*”【群组】图标的前面，在弹出的“交互类型”对话框中选择“时限”，命名为“时间限制 1”。

(21) 双击“时间限制 1”【群组】图标上边的响应类型图标，打开响应类型属性对话框，在“时限”文本框中输入对每道题的限制时间 5，单位是秒。在“中断”下拉选项中选择“保持计时”，在“选项”中选中“显示剩余时间”复选框，使得在时间计数时，屏幕的左下角出现一个时间计数器，提示使用者每道题还剩下的时间。如图 7-17 所示。

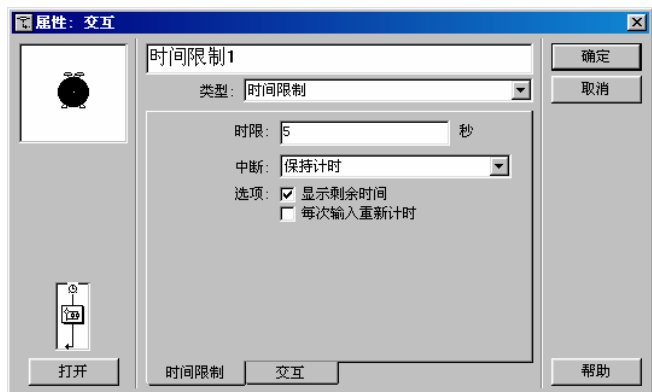


图 7-17 设置“限制时间”响应类型属性

(22) 假如用户对出现在屏幕上的时钟的位置不很满意的话,可以在程序的运行当中暂停程序后再调整位置。

(23) 双击“限制时间 1”【群组】图标,拖动一个【计算】图标和【声音】图标,双击计算图标后加入代码如下:

```
string3:="提示:flower"
```

```
--提示信息
```

(23) 在【声音】图标中载入声音文件。

(24) 此时“第一题”的编写工作完成了,程序流程如图 7-18 所示。

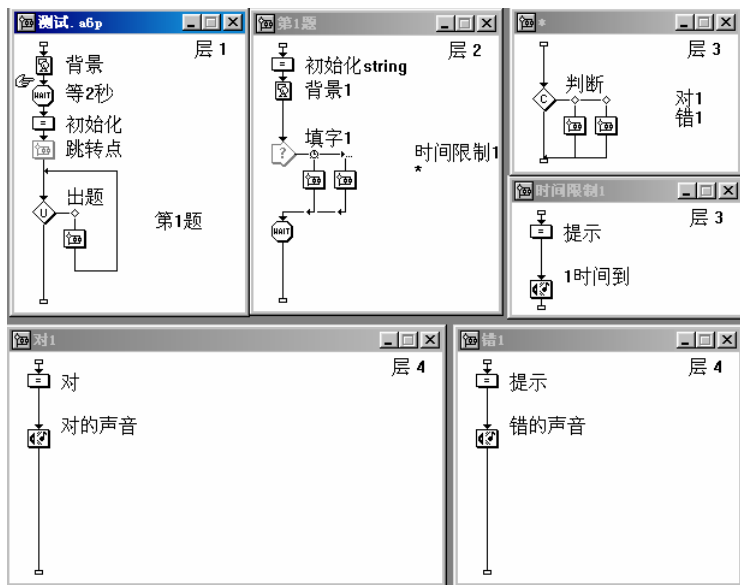


图 7-18 设置好的“第一题”程序流程

下面,将进入第二题的编制工作,由于第二题在工作流程和实现功能上和第一题是一样的,只是在背景图案和判别条件上有轻微的不同,所以我们完全可以把【群组】图标“第一题”复制过来,然后在此基础上进行轻微的改动即可,下面,我们进行复制工作。

(25) 用鼠标选中【群组】图标“第一题”，点击工具栏上的“复制”按钮，然后再在【群组】图标“第一题”的右侧单击一下，再按工具栏上的“粘贴”按钮，【群组】图标就粘贴过来了，并改变其名称为“第二题”，当用户打开其【群组】图标后会发现里面的内容也被粘贴过来了。把里面的“背景1”、“填字1”等有关“1”的地方统统该为“2”，以免混淆。

(26) 【计算】图标“初始化 string”里的内容不用改变，仍是对变量 string1、string3 初始化为“ ”空字符串，目的是不让前面的显示结果影响到本题，【显示】图标“背景 2”的内容和上题一样，所以需要改变，双击【显示】图标打开展示窗口，删除“狮子”的图像，引入一幅“猫”的图像，其余均可不改变，如图 7-19 所示。

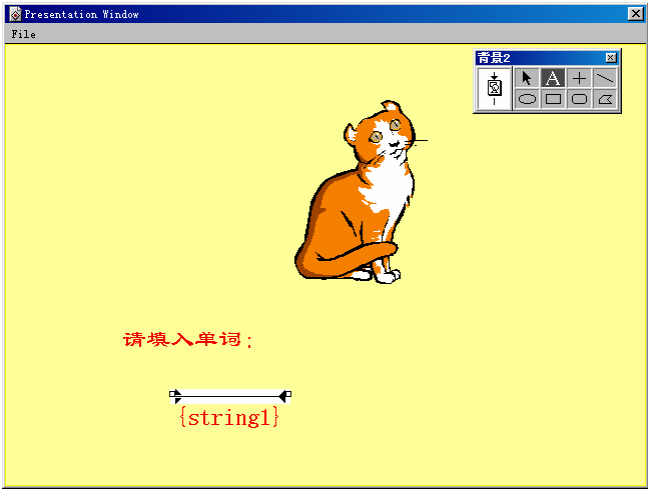


图 7-19 导入“猫”图像

由图可知，需要输入的单词改为了“cat”，这就需要改变判别条件，此外，我们已知道变量 string3 在答错的时候用来显示正确答案信息的，所以有关 string3 的所有计算图标里也需要该动一下。

(27) 首先改变判别条件。打开【交互】图标“填字 2”下挂的【群组】图标“*”，进入三级流程图线，双击【判断】图标，修改如图 7-20 所示。



图 7-20 修改【判断】图标属性

(28) “分支”下的文本框内的函数改为“Test(EntryText=“cat”，1，2)”；然后再改变【计算】图标里相应的值，如图 7-21 所示。

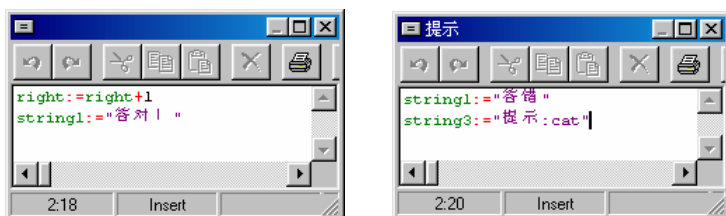


图 7-21 改变【计算】图标中信息

(29) 然后将其中的“flower”字符改为“cat”就行了，究竟在什么地方改，相信读者可以毫不费力的找到了。然后在依靠此方法加入第三道到第十道题。

(30) 到此为止，所完成的程序流程图如图 7-22 所示。

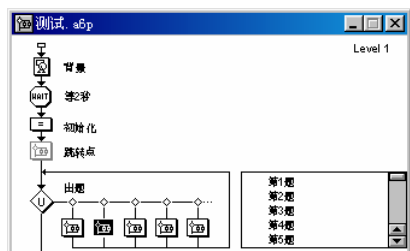


图 7-22 设计好的 10 道习题

(31) 再拖动一个计算图标到【判断】图标的下方，其作用是决定变量“path”的值，变量“path”又是用来存放进入下一个【判断】图标的分支的值的。加入代码如下：

--由变量决定判断分支

```
if right=0 then
    path:=1
else if right<6 then
    path:=2
else if right<8 then
    path:=3
else if right>=8 &right<>10 then
    path:=4
else if right=10 then
    path:=5
end if
```

由代码可知，变量“path”的值是由测试所答对的数目所决定的“right”变量就是用来存放所答对的数字的。请读者注意上述多重“if...else”语句的用法。

(32) 在该计算图标的下方加进【判断】图标“成绩打分”，再加进五个【群组】图标作为判断分支，分别命名为“=0”，“<6”，“<8”，“>8”，“=10”，每个分支针对不同的情况进行相应的处理，对变量“score”、“string2”进行相应的赋值，以备显示统计结果，如图 7-23 所示。

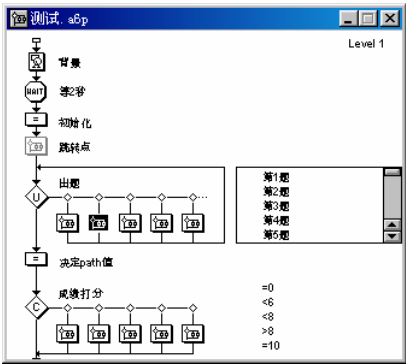


图 7-23 加入判断分支

(33) 双击【判断】图标“成绩打分”，设置【判断】图标属性如图 7-24 所示。

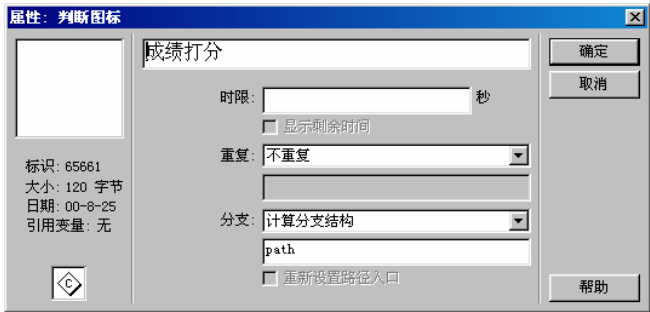


图 7-24 设置【判断】图标属性

当中所选项的意义相信读者已经能够理解，变量“path”就决定了进入判断图标的分支，“重复”项选择了“不重复”，表示程序进入判断图标后将在执行一条分支后退出判断图标。

(34) 为每一个作为【判断】图标分支的【群组】图标加入一个计算图标，在为每一个计算图标加入代码如图 7-25 所示。

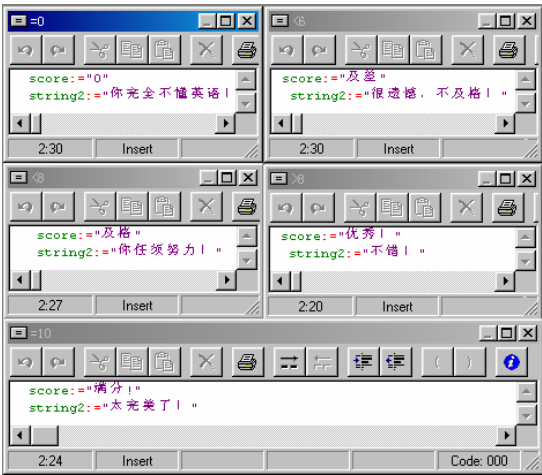


图 7-25 添加代码

请读者自己找到相应的计算图标并加入代码。

(35) 在流程线上再加入一个【群组】图标，名为“统计处理”，双击它打开二级流程图，加入【显示】图标“统计结果”和【交互】图标“判断”，再加入一个【声音】图标作为条件交互的交互分支，并载入相应的声音文件即可，如图 7-26 所示。

(36) 再设置交互条件为“right=10”，如图 7-27 所示。



图 7-26 添加【声音】图标

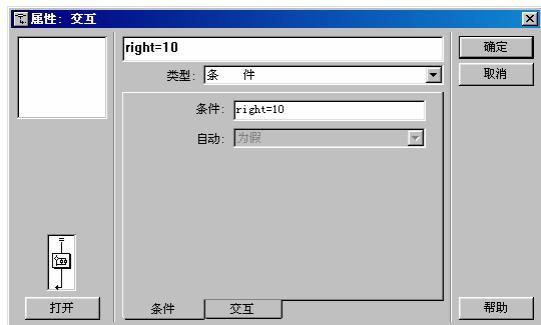


图 7-27 设置条件交互的条件

下面是对【显示】图标“统计结果”的设计，由图“5-21”可以知道，统计结果分别包括了答对和答错的数目，正确率是多少，成绩，以及对相应成绩的描述。在介绍【判断】图标“成绩打分”的时候，其判断分支队变量“score”、“string3”已经赋值，这些变量就准备在【显示】图标里显示统计结果的。如图 7-28 所示。

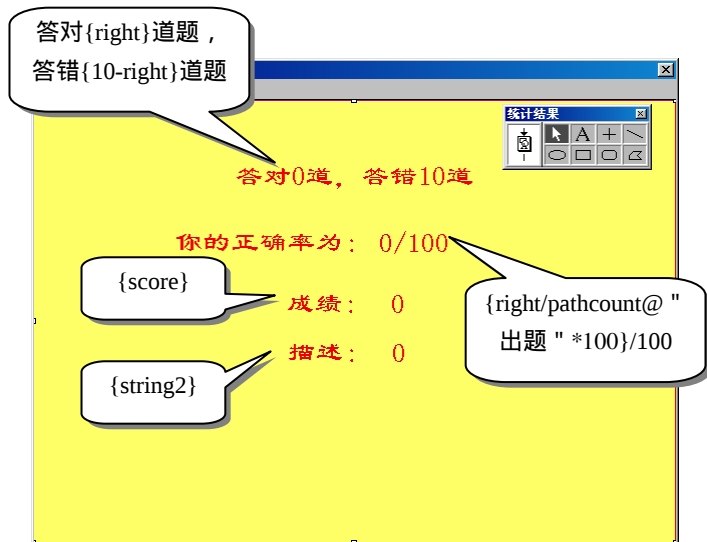


图 7-28 统计结果

相信读者对各个变量的意义一目了然了吧。用文本工具写上除了星云图所指向的另外部分，星云所指向的部分写成星云图里注释的形式，则【显示】图标“统计结果”的编制完成了。

(37) 在【群组】图标“统计处理”的下面加入一个【等待】图标，设置其属性让其等待 2 秒，选中“单击鼠标”复选框和“按任意键”复选框。

(38) 在知识对象中把“消息框”拖到流程线【等待】图标的下面，系统执行该知识对象的向导程序，单击“Next”按钮，出现“Modality”对话框，这里可以任意选择一种模式，选择“Task Modal”，单级“Next”按钮，出现“Buttons”对话框，要求选择出现在信息对话框上的按钮，选择“Ok，Cancel”，

再点击“Next”按钮。出现“Icon”对话框，要求选择出现在信息框中的小图标，由于本程序需要实现询问功能，这里选择“Question”。单级“Next”进入“Caption/Message”对话框，要求用户输入信息框的标题和内容，任意输入一些文字，如图 7-29 所示：



图 7-29 设置“Caption/Message”对话框

(39) 再继续按下“Next”进入“Return Value”对话框，要求定义返回值存放的变量，并选择返回值是按钮数值还是按钮名称，在这里我们选择了以按钮数值作为返回值。这个返回只是一个整数，返回值的具体意义读者可以参照“知识对象”一章中的有关对消息对话框的讲解。在变量栏输入“=result”，定义用变量“result”存储返回值。然后按下“Done”结束对话框的设置。

(40) 在程序流程图的末尾加上一个【计算】图标，命名为“是否决定退出”，并在该图标中加入如下语句：

```
if result=1 then
--如果继续测试，则跳转到“跳转点”图标
GoTo (IconID@"跳转点")
Else
--否则退出
Quit(0)
End if
```

变量“result”等于 1 表示用户在询问对话框按下的是“确定”按钮，然后程序就跳到【群组】图标“跳转点”后继续往下执行，否则将退出程序。

(41) 此时整个测试项目就编写完成了，主流程图如图 7-30 所示。

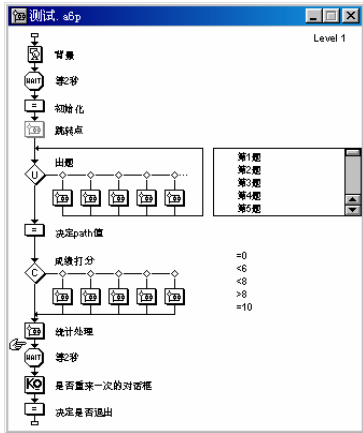


图 7-30 “测试”主程序流程

第8章 【交互】图标

■ 【交互】图标简介

■ 各种交互方式介绍

■ 【交互】图标实例

8.1 【交互】图标简介

随着多媒体技术的发展,教学方式也由原来单一方式转变为双向的交互方式。当前众多的多媒体课件大部分都在向双向交互方式转变。所谓“交互”,就是指用户可以与计算机进行信息的双向交流,以增强控制信息和使用信息的手段。Authorware 提供了强大的交互功能,这些交互功能都是通过【交互】图标来实现的。

【交互】图标是 Authorware 提供的功能图标中最复杂、最灵活、功能最强而又最难掌握的一个图标。它兼有【显示】图标和【决策】图标的功能,具有显示信息的功能,同时具有强大的分支功能,这样【交互】图标即可以提供丰富的提示信息,又可以对用户的响应作出正确的反应。用户必须在掌握不同交互响应类型知识的基础上通过一些具体实例来巩固所学的知识。

【交互】图标一共提供了 11 种交互响应类型:

- 按钮:通过用户单击屏幕上不同的按钮,Authorware 产生不同的响应,执行不同的响应分支。
- 热区域:当用户将鼠标移动到定义好的区域中、或单击、双击该区域,就会触发响应,Authorware 执行相应的响应分支。
- 热对象:当用户单击、双击或拖动定义好的热对象时,匹配相应的响应,Authorware 执行相应的响应分支。
- 目标区:当用户按照定义好的一定规则将一个对象移动到一个指定的区域中时,就会触发响应,Authorware 执行相应的响应分支。
- 下拉菜单:为菜单栏添加菜单和菜单项,当用户单击某个菜单项时,就会触发响应,Authorware



- 执行相应的响应分支。
- 条件：根据程序运行过程中所设置的条件是否得到满足来匹配响应。当条件为 True 时，就会触发响应，Authorware 执行相应的响应分支。
 - 文本输入：通过用户输入文本，当文本与定义好的匹配文本相匹配时，Authorware 就会执行响应的响应分支。
 - 按键：当用户在键盘上按下定义好的键时，就会触发响应，Authorware 执行相应的响应分支。
 - 重试限制：该响应一般与其他响应类型一起使用。当用户在指定的限制次数之内匹配了响应，就可以继续执行，否则 Authorware 就会执行相应的反馈信息分支。
 - 时间限制：该响应一般与其他响应类型一起使用。当用户在指定的限制时间之内匹配了响应，就可以继续执行，否则 Authorware 就会执行相应的反馈信息分支。
 - 事件：在应用程序中添加了 ActiveX 控件时，ActiveX 控件与 Authorware 应用程序之间的调用与响应，需要通过事件响应来控制的。
- 【交互】图标和【框架】图标、【决策】图标一样，都是通过其下挂的子图标来实现其功能，如图 8-1 所示。【交互】图标的下挂图标可以为【显示】图标、【移动】图标、【擦除】图标、【等待】图标、【导航】图标、【群组】图标、【数字电影】图标、【声音】图标和【视频】图标等。

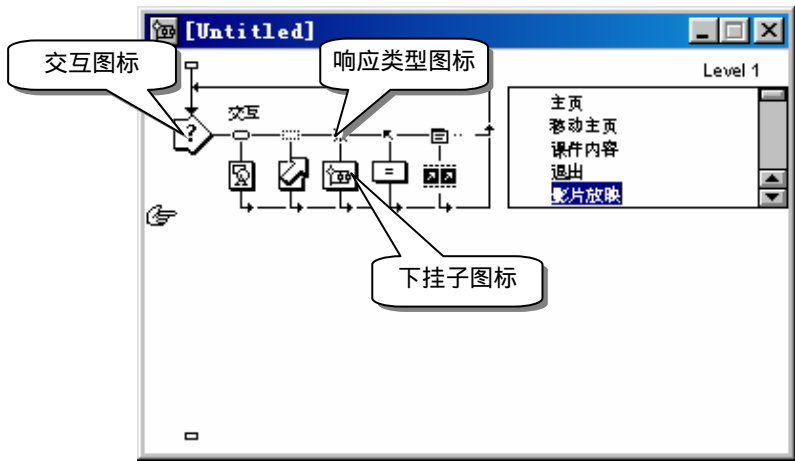


图 8-1 【交互】图标结构

当 Authorware 执行时遇到一个【交互】图标，它会显示出碰到的所有文本和图形元素，其中包括按钮、热区、文本输入框等，这些是由其响应类型图标所决定的。此时 Authorware 会停下来等待用户作出响应。当用户做出响应时，Authorware 会将这个响应同交互流程线上的响应类型进行比较，看是否与响应类型中的某个目标响应相匹配。如果与某一个响应目标相匹配，Authorware 会沿着该分支执行目标图标。

在“Response Type（响应类型）”对话框中，程序设计人员可以建立不同的响应类型。在这里，您可以通过定义每一种目标图标的响应类型来决定用户的响应和事件，当用户作出了一种响应或某个事件发生，Authorware 就会跳到相应的分支，执行目标图标，显示出它的内容。

【交互】图标提供了 11 种交互响应类型，不同的响应类型对应了不同的响应类型图标，表 8-1 列出了响应类型图标和响应类型的对应关系，这样，用户就可以在流程线上看出该响应类型是何种响应类型。

表8-1 响应类型图标和响应类型对照表

响应类型图标	响应类型	响应类型图标	响应类型
	按钮		文本输入响应（文本输入）
	热区域		按键响应（按键）
	热对象		限制次数响应（重试限制）
	目标区		限制时间响应（时间限制）
	下拉菜单		时间响应（事件）
	条件		

用户拖动一个【交互】图标到流程线上，然后拖动其他图标到【交互】图标右侧，出现“交互类型”对话框，如图 8-2 所示。

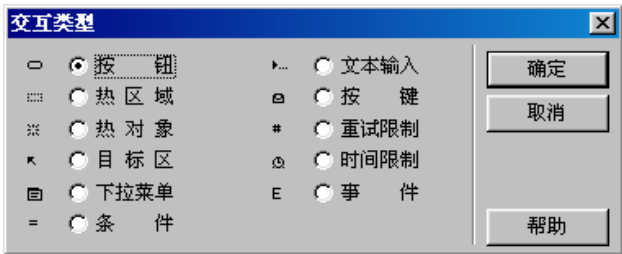


图 8-2 响应类型对话框

8.2 按钮响应（按钮）

按钮响应提供了一种最清晰，最简单的交互方式。当用户单击该按钮，就会触发该分支流程的响应，程序就沿着该分支流程运行。

一般来说，Authorware 提供的按钮类型有三种（如图 8-3 所示）：

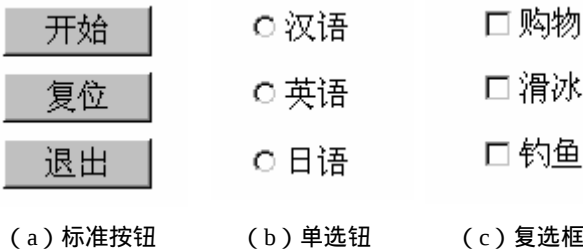


图 8-3 三种类型的按钮

8.2.1 选择“按钮”选项后的响应类型属性对话框

双击【交互】图标下挂图标的响应类型图标，打开“属性：交互”属性对话框，如图 8-4 所示。

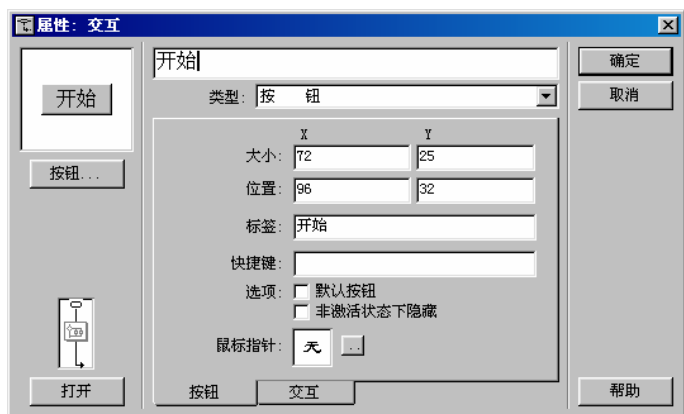


图 8-4 响应类型属性对话框

该对话框可以定义【交互】图标的 11 种响应类型及其他属性的设置。

“类型”下拉框中的选项就对应着【交互】图标的 11 响应类型，如图 8-5 所示。

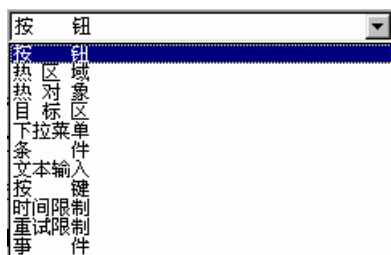


图 8-5 “类型”下拉框中的 11 个选项

按钮响应就是我们选择其中的“按钮”选项。下面介绍的就是我们选择“按钮”选项后的各个设置。

在中间的文本框中依然是用来输入图标/按钮的名称的。图标/按钮的名称中可以包含有空格。我们可以在这里修改按钮的名称，也可以在流程线上单击该下挂子图标，对子图标的名称进行修改也可以改变按钮的名称。

在【交互】图标响应类型属性对话框的左上角是按钮预览窗口，预览窗口下面有一个“按钮”按钮，单击该按钮可以打开“按钮”对话框，在该对话框中，用户可以选择自己喜欢的按钮形状，也可以通过按钮编辑器编辑自己的按钮，这些将在下面予以介绍。

响应类型属性对话框的左下角有一个“打开”按钮，单击该按钮，可以该响应类型图标所对应的下挂子图标。如果该子图标为【显示】图标，将打开该【显示】图标的展示窗口；如果该子图标为【计算】图标，将打开该【计算】图标的编辑窗口；如果该子图标为【群组】图标，将打开该【群组】图标，即打开二级流程窗口等。

选择“按钮”选项后，响应类型属性对话框包括两个标签页：

1. “按钮”标签页

“按钮”标签页见图 8-4 所示。

- “大小”选项：该选项是用来控制按钮的长宽大小的。

其中“X”文本框中的数值决定着按钮的横向长度，即按钮的长度；“Y”文本框中的数值决定着按钮的纵向长度，即按钮的宽度。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变按钮的大小，也可以通过在展示窗

口中直接拖动按钮的句柄来改变按钮大小。

- “位置”选项：该选项是用来控制按钮在展示窗口中的位置的。

其中“X”文本框中的数值决定着按钮的水平方向坐标，即按钮的X坐标；“Y”文本框中的数值决定着按钮的垂直方向坐标，即按钮的Y坐标。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变按钮的位置，也可以通过在展示窗口中直接拖动按钮的句柄以外的部分来移动按钮，改变按钮的位置。

- “标签”文本框：该文本框是用来输入按钮的标题的。在默认情况下，按钮的标题与该下挂子图标名称相同。

一个按钮的标题就是目标图标的标题与按钮标志相链接。按钮标题也会出现在“按钮”对话框的按钮名称域中。在“按钮编辑器”中用户可以显示或隐藏按钮标题。要改变按钮的标题，可以改变目标按钮的名称或响应类型属性对话框中“标签”文本框中的文本。如果您在“标签”文本框输入修改后的按钮标题，则该下挂子图标的名称也相应地变成了修改后的名称。

如果您不想为按钮的某个特定状态选择一个标题，Authorware 会用您已经从其默认图形中选择好的状态设置。

- “快捷键”文本框：该文本框是用来设置该按钮的快捷键的。也就是说，当设置好快捷键后，按下快捷键和单击该按钮会产生相同的效果。关于快捷键的设置方法，请参看“按键”响应一节。
- “选项”选项：该选项是用来设置按钮的一些特殊性质。它包括两个选项：

“默认按钮”复选框：如果选择该选项，系统会认为该按钮是默认按钮，此时 Authorware 会在该按钮四周显示出一个黑色边框，如图 8-6 所示。当用户直接按下<Enter>键和单击该按钮具有相同的效果。



(a) 没有选择“默认按钮”复选框的按钮 (b) 选择“默认按钮”复选框后的按钮

图 8-6 选择“默认按钮”复选框的效果

“非激活状态下隐藏”复选框：该选项来决定在不进行交互时是否将按钮隐藏。用户可以通过一个变量或表达式来控制按钮是否可用。当按钮不可用时，可以将该按钮灰显。可以关闭“非激活状态下隐藏”复选框，而定义一幅图形来表示按钮不可用的状态，当按钮不可用时，可以显示该图形。如果选择“非激活状态下隐藏”复选框，当按钮不进行交互时，该按钮将会在展示窗口中隐藏起来，直到该按钮起作用时，按钮自动显示出来。

- “鼠标指针”选项：该选项是用来定义鼠标指针形状的。当选择一种鼠标指针形状时，它会在后面的小预览窗口中显示出来。默认情况下，预览窗口中显示的是“无”，表示鼠标指针形状为系统默认的箭头形状。如果要改变鼠标指针形状，可以单击后面的按钮，打开“鼠标指针”对话框，如图 8-7 所示。

鼠标指针对话框中包含了系统标准的鼠标指针类型以及用户自定义的鼠标指针。在鼠标指针对话框中，左边一列是鼠标指针预览，右边一列描述栏指明了鼠标指针的类型、标

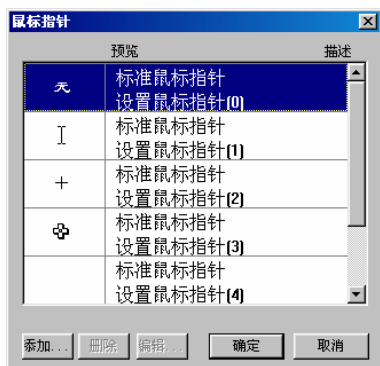


图 8-7 鼠标指针对话框

准的还是自定义的)和鼠标指针对应的类型号。用户可以通过系统函数 SetCursor(n)使用鼠标指针类型号 n 来设置鼠标指针形状。对于自定义鼠标指针,描述栏也会显示出有那些图标使用了这种鼠标指针。需要注意的是,自定义鼠标指针只能是黑白色的,不能是 256 色或真彩色,否则该鼠标指针会显示出一片黑色。

“添加”按钮:单击该按钮,系统会弹出一个标准对话框,如图 8-8 所示。

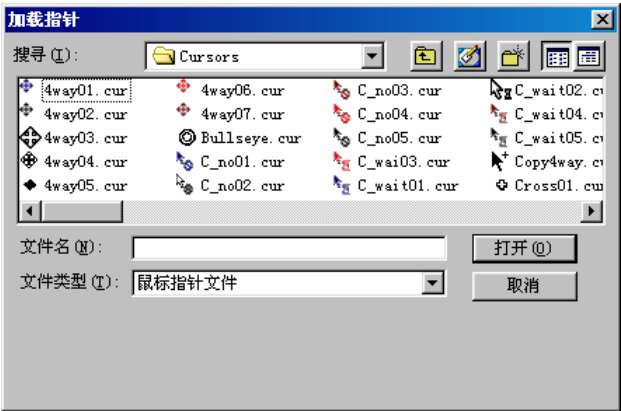


图 8-8 添加自定义鼠标指针

在对话框中,让用户选择应用程序,在该应用程序中有您想要复制的鼠标指针。选择好应用程序,找到该应用程序所使用的鼠标指针。选择一个鼠标指针并单击“确定”按钮,就可以将该鼠标指针添加到鼠标指针对话框中,此时您就可以使用该鼠标指针了。

当添加了新的鼠标指针后,系统会自动分配给该鼠标指针一个类型号。一般来说,用户自定义的鼠标指针类型号从 51 开始,然后类型号逐渐增加。

“编辑”按钮:单击该按钮,系统显示出一个标准对话框,在该对话框中选择一个鼠标指针文件然后确定,将会替换原来的鼠标指针。这个选项只能对自定义的鼠标指针进行操作。

“删除”按钮:单击该按钮,将会删除当前所选中的鼠标指针。用户不能删除系统鼠标指针,只能对自定义的鼠标指针进行操作。

2. “交互”标签页

选择响应类型属性对话框的“交互”标签页,打开该标签页,如图 8-9 所示。该标签页主要是控制 Authorware 何时擦除目标图标中的显示文本或图形,以及离开目标图标后的程序流程。



图 8-9 响应类型属性对话框的“交互”标签页

- “范围”选项：该选项只包括一个“永久”复选框，选择该复选框，会使响应类型标志在整个部分中都是可用的，或者在这一部分的一段中始终可用，这样就不需要在每一个交互中都创建一个相同的响应类型标志。永久响应不能被【交互】图标的擦除设置所擦除，如果要擦除一个永久响应，比如一个永久响应按钮或热对象，必须要使用一个【擦除】图标。
- “激活条件”文本框：该文本框是用来设置在何时该按钮才被激活。可以在该文本框中输入一个变量或表达式，当该变量或表达式为“TRUE（真）”时，该按钮才会被激活，此时按钮可用，可以单击该按钮或按下快捷键来触发分支流程的响应；当该变量或表达式为 FALSE（假）时，该按钮无效，此时无论单击该按钮还是按下快捷键都无法触发分支流程的响应。
- “擦除”下拉框：该选项主要是设置该交互作用分支结构的显示信息的擦除方式。它包括四个选项：

“在下次输入之后”选项：当选择该选项时，Authorware 会在进入下一个交互项后才擦除上一个交互项的显示内容。该选项是系统默认选项。

如果您选择的交互项分支方式为“重试”或“继续”，前面的显示信息会保留在屏幕上，直到用户又有响应。如果新的响应得到了匹配，Authorware 会擦除显示信息；而如果用户新的响应没有得到匹配，Authorware 不会自动擦除前面的显示信息。

如果您选择的交互项分支方式为“退出交互”，Authorware 会在离开当前交互时擦除目标图标的显示信息。然而，如果分支是让用户返回到另一个交互中，Authorware 不会擦除当前显示信息的，除非用户又发出新的响应。

“在下次输入之前”选项：当选择该选项时，Authorware 会在显示交互信息让用户进行下一个交互响应前擦除目标图标的显示内容。

如果您选择的交互项分支方式为“继续”，目标图标显示信息会一直保留在屏幕上直到 Authorware 匹配了下一个响应。如果没有其他响应被匹配，Authorware 会在用户进行下一次响应之前擦除显示信息。

如果您选择的交互项分支方式为“退出交互”，Authorware 将在程序离开当前交互时擦除目标图标的显示信息。

如果您为两个目标图标中的第一个选择了“在下次输入之前”擦除方式，并且用户同时触发了两个响应，Authorware 会在显示第二个显示信息出现之前擦除第一个图标的显示信息。

“在退出时”选项：选择该选项时，Authorware 会在离开该【交互】图标时才擦除其下挂图标的交互信息。目标图标的显示信息回保留在屏幕上，即使用户作出了另一个响应，除非 Authorware 离开当前交互才会被擦除。选择该选项可能会导致显示信息的重叠，但是当 Authorware 离开当前交互图标而执行主流程线上的下一个图标时，它会裁处所有的显示信息。

“不擦除”选项：选择该选项时，Authorware 在离开该【交互】图标时，不擦除其下挂图标的交互信息，让交互信息一直保留在展示窗口中，除非程序设计人员使用【擦除】图标将其擦除掉。

- “分支”下拉框：该选项是用来设置交互项的分支方式。使用下面的分支方式来决定当 Authorware 离开当前链接有响应类型图标的目标图标时程序的流向。一般来说，它包括三个选项；当选中“永久”复选框时，它包括四个选项：

“重试”选项：选中该选项，Authorware 会返回到【交互】图标的入口处，重新让用户作出新的响应。

“继续”选项：当选择该选项时，Authorware 会判断目标图标右侧的响应类型图标是否也与这次用户作出的响应相匹配，此时可能会出现同时有两个或多个分支发出响应。

“退出交互”选项：选择该选项时，Authorware 会退出当前【交互】图标而继续执行流程线上的下一个图标。

- “状态”下拉框：该下拉框是用来控制响应状态的，它包括三个选项：



∞

[
]

“不判断”选项：选择该选项时，Authorware 对这种响应不进行跟踪。

“正确响应”选项：选择该选项时，Authorware 会跟踪用户作出的正确响应，记录正确响应的次数，并将该值存储在表示“正确响应”的系统变量中。

选中该选项后，该下挂子图标名称前添加了一个“+”号。

“错误响应”选项：选择该选项时，Authorware 会跟踪用户作出的错误响应，记录错误响应的次数，并将该值存储在表示“错误响应”的系统变量中。

选中该选项后，该下挂子图标名称前添加了一个“-”号。

在跟踪用户信息过程中，判断响应是十分有用的。Authorware 提供了几个系统变量来存储响应类型的正确或错误信息。例如，系统变量 TotalCorrect 是累加在整个响应数目中的正确响应的数目，您可以保存这些信息，放在用户自己的一个文件中或将这些信息显示出来反馈给用户。关于其他系统变量的使用，可以参考光盘。

在流程线上，您可以看到每个目标图标的不同响应状态。含有一个“+”号的代表是正确响应，含有“-”号表示是错误响应，而没有符号显示的是对响应不作判断。

- “计分”文本框：在该文本框中，可以输入一个数值、变量或表达式来表示用户匹配该响应而获得的成绩。如果该响应为正确响应，可以输入一个正数来表示成绩。而如果该响应为错误响应，可以输入一个负数来表示错误响应的成绩。

当【交互】图标下挂多个子图标，子图标的响应类型也不一定相同，Authorware 执行程序时，遇到【交互】图标，等待用户的响应，当用户发出响应后，响应匹配顺序是按照从左到右逐个匹配。当前面的响应类型匹配上了，后面即使也有可以匹配上的响应类型，也要看前面匹配上的响应类型的分支方式。根据其响应方式来确定程序的流程。如果其响应类型为“继续”，则后面的响应类型可以继续匹配，这样就可以有多个同时匹配上用户的响应。

在【交互】图标属性对话框中的设置，大部分都可以表现在设计窗口的流程线上，如图 8-10 所示。

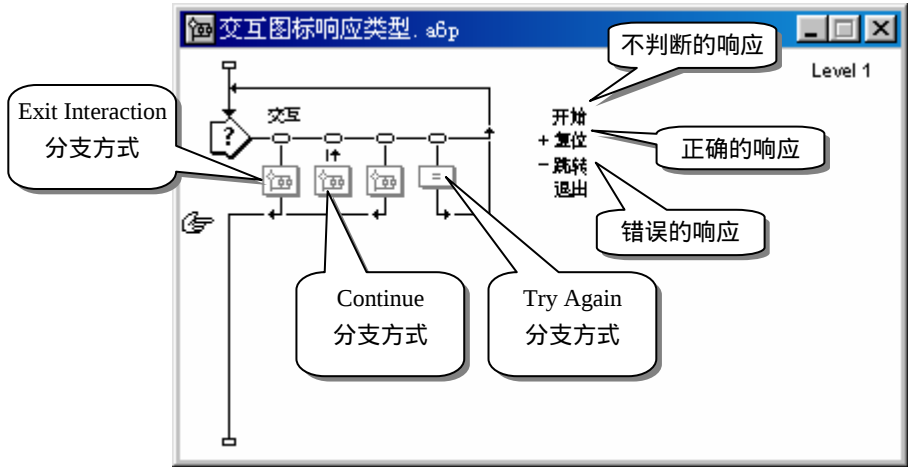


图 8-10 【交互】图标的不同响应类型

8.2.2 按钮对话框和按钮编辑器对话框——创建自己的按钮

1. 按钮对话框

在【交互】图标的响应类型属性对话框中，左上角有一个“按钮...”按钮，单击该按钮可以打开“按钮”对话框，如图 8-11 所示。

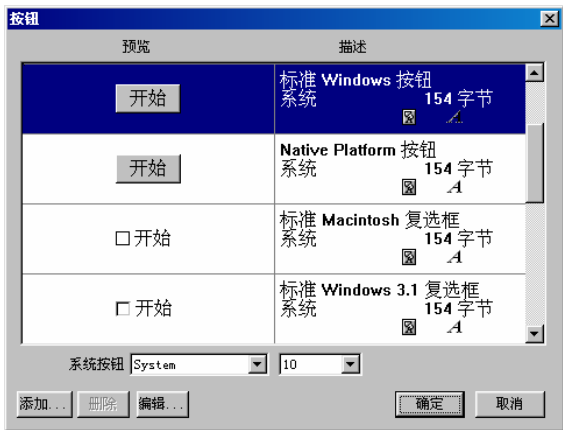


图 8-11 按钮对话框

2. 按钮编辑器对话框

单击按钮对话框中的“添加”按钮或“编辑”按钮，可以打开按钮编辑器对话框，如图 8-12 所示。

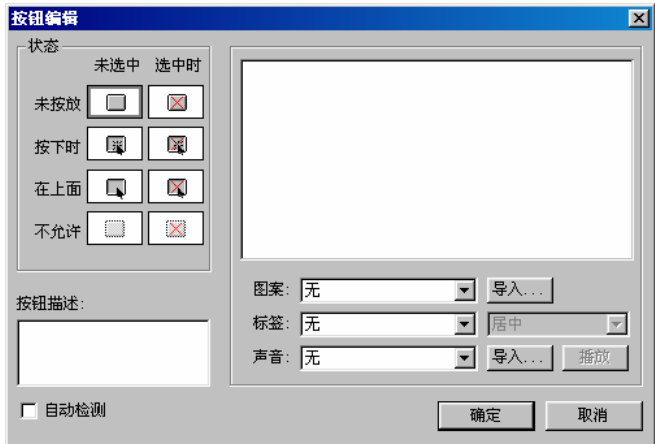


图 8-12 按钮编辑器对话框

● “状态”框架

一个按钮可以以正常状态或其他状态，如弹出、按下或不可用，显示。当要定义一个按钮时，用户必须首先为正常状态建立属性，然后再定义变化后的其他状态。单击按钮的一个状态，在预览窗口中预览按钮图形和标题对该状态的影响。如果要改变按钮的外形或按钮声音，可以使用预览窗口下面的选项。

标准按钮的四个状态分别为：

“未按放”：该状态定义了展览展示窗口中按钮的正常外形，也就是鼠标还没有单击该按钮并且鼠标指针也没有移动按钮的上方时的状态。

“按下时”：该状态定义了当用户按下该按钮时按钮的状态，包括按钮的外形和按钮声音。

“在上面”：该状态定义了当用户鼠标指针处于按钮上方时按钮的外形和按钮声音。

“不允许”：该状态定义了当按钮不可用时按钮的外形。不可用按钮一般是灰显的。

● “按钮描述”文本框

在“状态”框架下面是按钮描述信息文本框，在该文本框中，显示了该按钮的描述信息。用户可以



∞

【
】

在文本框中输入当前自定义按钮的描述信息。
在按钮描述信息文本框中，输入的描述信息不得超过 80 个字符。当用户编辑一个按钮，也可以编辑它的描述信息。描述信息中的第一行将会显示在按钮对话框的描述栏中。

● “自动检测”复选框

选中该复选框时，当前自定义按钮被定义为自动被选中，即当用户单击该按钮，按钮会在“选中”状态和“未选中”状态之间转换。用户可以使用系统变量 Checked 来查看按钮是否被选中了或按钮此时的状态是选中还是没有被选中状态。该选项对于复选框和单选钮特别有用。

● “预览窗口”

预览窗口是将用户编辑或添加的按钮外形显示给用户。如果用户创建一个新的按钮，预览窗口开始时是空白的。如果用户编辑一个按钮，预览窗口开始时显示出编辑前的按钮外形。创建按钮后或编辑好按钮后，预览窗口显示的是设计好的按钮外形。用户可以单击“状态”框架中的其他按钮状态，让预览窗口中显示出该状态的外形，然后对此时的状态进行修改编辑。当用户创建自定义的按钮，可以在预览窗口中拖动按钮不同的组件并将组件进行排列。但是用户不能在预览窗口中拖动系统按钮。

● “图案”下拉框

该下拉框是用来设置按钮图形的。系统默认为“无”，表示没有按钮图形。如果要为按钮添加按钮图形，可以单击该下拉框右侧的“导入...”按钮，打开导入文件对话框，如图 8-13 所示。



图 8-13 导入文件对话框

在该对话框中选择合适的按钮图形，然后单击“导入”按钮导入该文件。此时“图案”下拉框就有两个选项：“无”选项和“使用导入图”选项。选中“使用导入图”选项就可以将导入的按钮图形添加到此按钮的状态中。

● “标签”下拉框

该下拉框是用来控制按钮标题的。如果是用户添加的新按钮，默认为“无”；如果是编辑已有的按钮，默认为已有按钮的设置。该下拉框有两个选项：“显示卷标”选项和“无”选项。当选择“无”选项时，按钮不显示标题，右侧的下拉框灰显不可用；当选择“显示卷标”选项，按钮显示标题，此时右侧的下拉框被激活。右侧的下拉框包括三个选项：

- “左对齐”选项：选择该选项时，按钮标题左对齐。
- “居中”选项：选择该选项时，按钮标题中间对齐。系统默认为中间对齐。
- “右对齐”选项：选择该选项时，按钮标题右对齐。

● “声音”下拉框

该下拉框与“图案”下拉框相似，该下拉框用于设置按钮声音。系统默认为“无”，表示没有按钮

声音。如果要为按钮添加按钮声音，可以单击该下拉框右侧的导入按钮，打开导入文件对话框。

在该对话框中选择合适的按钮声音，然后单击“导入”按钮导入该文件。此时“声音”下拉框就有两个选项：“无”选项和“使用导入图”选项。导入声音后，系统自动选中“使用导入图”选项，此时该下拉框右侧的“播放”按钮被激活，单击该按钮可以播放导入的声音文件。如果导入的声音文件不合适，可以重新导入。如果此时选择“无”选项，Authorware 给出一个提示对话框，询问用户是否删除该导入的声音文件，如图 8-14 所示。

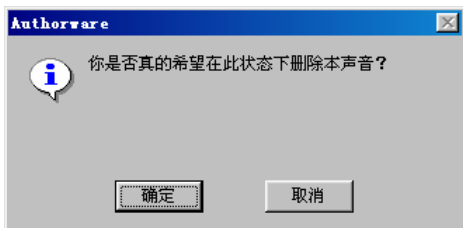


图 8-14 系统提示对话框

如果确实要删除导入的声音文件，单击提示对话框中的“确定”按钮，否则单击“取消”按钮。当用户选择“无”选项后，下拉框右侧的“播放”按钮灰显不可用。

8.2.3 按钮响应实例——制作“学习教程”

下面我们通过一个实例来说明如何使用按钮响应。

在该实例中，我们使用了按钮响应、按钮编辑器等，该实例的界面如图 8-15 所示。希望通过该实例，能够让读者掌握如何使用按钮响应。

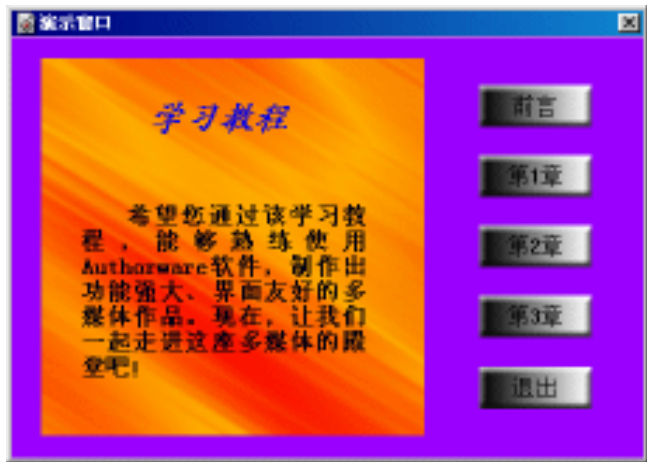


图 8-15 学习教程

操作步骤为：

- (1) 新建一个文件，命名为“学习教程.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“根据变量”选项，在“选项”中取消“显示菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (4) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：

ResizeWindow(450, 300)

(5) 选择菜单[文本]▶[定义风格...]命令, 打开定义文本风格对话框。

(6) 单击定义文本风格对话框中的“添加”按钮, 增加一种新的文本风格, 将该文本风格更名为“标题”, 选择字体为楷体_GB2312、字号为 18, 粗体, 斜体、字色为蓝色, 然后单击“修改”按钮确定修改。

(7) 同样, 单击定义文本风格对话框中的“添加”按钮, 增加一种新的文本风格, 将该文本风格更名为“正文”, 选择字体为宋体、字号为 12, 粗体, 字色为黑色, 然后单击“修改”按钮确定修改。

(8) 单击对话框中的“完成”按钮, 完成定义文本风格。

(9) 拖动一个【显示】图标到流程线上, 并将其命名为“背景”。

(10) 双击该【显示】图标, 打开展示窗口, 然后单击工具栏中的“导入图片”按钮, 从外部导入一幅背景图片, 然后调整图片的大小, 使之正好覆盖整个窗口。

(11) 按住<Ctrl>键并双击该【显示】图标, 打开其属性对话框, 选中“防止自动擦除”复选框, 单击“确定”按钮确定。

(12) 拖动一个【显示】图标到流程线上, 并将其命名为“提示信息”。

(13) 按住<Shift>键并双击该【显示】图标, 打开展示窗口, 选择文字工具, 在展示窗口上单击一下, 然后在工具栏中的“文本风格”下拉框中选取“标题”选项, 输入文本“学习教程”。

(14) 在展示窗口上再单击一下, 然后在“文本风格”下拉框中选取“正文”选项, 输入文本“希望您通过该学习教程, 能够熟练使用 Authorware 软件, 制作出功能强大、界面友好的多媒体作品。现在, 让我们一起走进这座多媒体的殿堂吧!”。

(15) 调整两个文本对象, 使之正好位于导入背景图片的中间, 如图 8-16 所示。

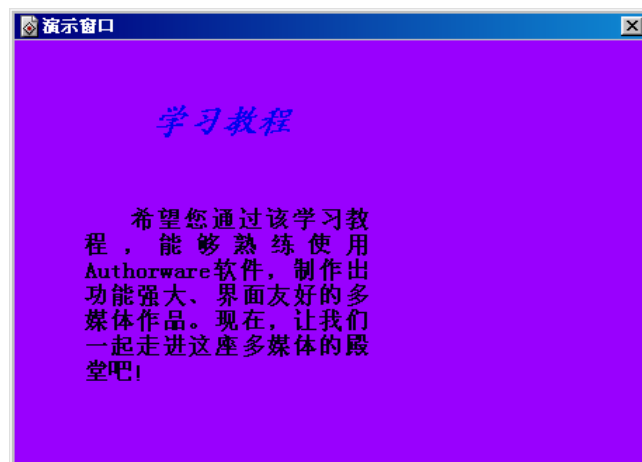


图 8-16 输入提示信息

(16) 拖动一个【交互】图标到流程线上, 将其命名为“课程”。

(17) 拖动一个【显示】图标到【交互】图标的右侧, 选取按钮响应类型, 并将其命名为“前言”。

(18) 双击“背景”【显示】图标, 显示出展示窗口, 然后将其关闭。

(19) 按住<Shift>键并双击“前言”【显示】图标, 打开展示窗口。

(20) 选择绘图工具箱中的文字工具, 在展示窗口中单击一下, 然后在“文本风格”下拉框中选取“标题”选项, 输入文本“前言”。

(21) 再在展示窗口中单击一下, 输入文本“Authorware 简介”。

(22) 在展示窗口中再单击一下, 在“文本风格”下拉框中选取“正文”选项, 输入文本“Authorware

是一种面向对象并基于图标和流程线的多媒体创作平台，具有丰富的函数及程序控制功能，融合了编辑系统和编程语言的特色。Authorware 提供多媒体基本元件的集成及多重分支功能，能够制作出功能强大的演示简报，而多样化对话模式适合于各种测试题目的编制，因此成为多媒体教育软件的最好的开发平台。”。

(23) 调整这三个文本对象，如图 8-17 所示。



图 8-17 输入“前言”文本信息

(24) 按住<Ctrl>键并双击该【显示】图标，打开其属性对话框，选中“擦除以前内容”复选框，单击“确定”按钮确定。

(25) 双击“前言”【显示】图标上面的响应类型图标，打开响应类型属性对话框，单击对话框中的“按钮”按钮，再打开“按钮”对话框。

(26) 单击“按钮”对话框中的“添加”按钮，打开按钮编辑器。

(27) 选择“未选中”的“未按放”选项，然后单击“图案”下拉框右侧的“导入”按钮，从外部导入一幅按钮图片，如图 8-18 所示。



图 8-18 导入按钮图片

(28) 在“标签”下拉框中选择“显示标签”选项。



∞

【
】

(29) 然后选择“未按下”的“按下时”选项，然后单击“图案”下拉框右侧的“导入”按钮，从外部导入另一幅按钮图片。

(30) 在“标签”下拉框中选择“显示标签”选项。

(31) 单击“声音”下拉框右侧的“导入”按钮，从外部导入一个声音文件作为按钮按下时的声音。

(32) 单击“确定”按钮确定，然后再将“按钮”对话框和响应类型属性对话框都按下“确定”按钮确定。

(33) 选择“前言”【显示】图标，单击工具栏中的“复制”按钮，将该图标复制到系统的剪贴板上。

(34) 将粘贴指针指到“前言”【显示】图标的右侧，然后单击工具栏中的“粘贴”按钮，将“前言”【显示】图标复制到此处，并将该【显示】图标更名为“第1章”。

(35) 同样，再粘贴两个“前言”【显示】图标到【交互】图标的右侧，将其分别更名为“第2章”和“第3章”。

(36) 双击“第1章”【显示】图标，打开展示窗口，将其中的“前言”文本改为“第1章”，将“Authorware 简介”文本改为“Authorware 功能”，修改的文本。

“Authorware 易用性强，开发效率高，网络开发和发布简捷、学习跟踪管理容易，其

(37) 同样，将“第2章”和“第3章”的标题分别改为“第2章”、“第3章”、“[显示]图标”、“[移动]图标”，同时修改文内容。

(38) 拖动一个【计算】图标到【交互】图标的最右侧，将其命名为“退出”。

(39) 双击该【计算】图标，打开计算编辑窗口，在窗口中输入语句“Quit()”，然后确定并关闭该计算编辑窗口。

(40) 按住<Shift>键并双击【交互】图标，打开展示窗口，将五个按钮进行排列对齐，如图 8-25 中所示。

(41) 此时程序已经设计完成，此时的程序流程如图 8-19 所示。

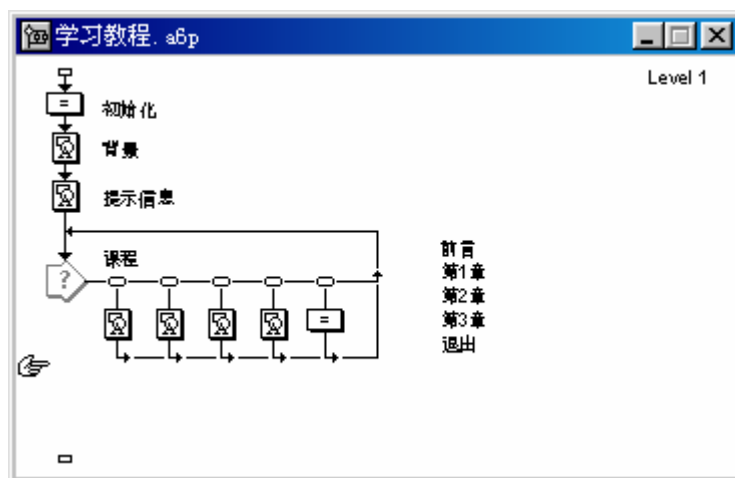


图 8-19 设计好的程序流程

8.3 热区域响应

热区是指用户可以在屏幕上定义一个矩形区域来作为响应用户的鼠标指针单击该区域的操作或将鼠标指针移动到该区域中。同按钮响应和热对象响应相同，这个区域不能是分离的图形。利用热区响应可以将背景图形的一部分作为热区，而不需要对背景图形进行任何编辑。热区响应也是建立全热区响应的理想方法。全热区响应通常是将热区面积拖动的如整个响应区域一样大，这样用户在屏幕上的任意响应区域位置单击都可以发出响应。

8.3.1 选择“热区域”选项后的响应类型属性对话框

双击【交互】图标下挂图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“热区域”选项，属性对话框变为如图 8-20 所示。



图 8-20 选择“热区域”后的响应类型对话框

在选择“热区域”选项后的响应类型对话框中，“热区域”标签页与按钮响应的“按钮”标签页十分相似。

- “大小”选项：该选项是用来控制热区的长宽大小的。

其中“X”文本框中的数值决定着热区的长度；“Y”文本框中的数值决定着热区的宽度。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变热区的大小，也可以通过在展示窗口中直接拖动热区的句柄来改变热区大小。

- “位置”选项：该选项是用来控制热区在展示窗口中的位置的。

其中“X”文本框中的数值决定着热区的 X 坐标；“Y”文本框中的数值决定着热区的 Y 坐标。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变热区的位置，也可以通过在展示窗口中直接拖动热区的句柄以外的部分来移动热区，改变热区的位置。

- “快捷键”文本框：该文本框是用来设置该热区的快捷键的。关于快捷键的设置方法，请参看“按键”响应一节。

- “匹配”选项：该选项是用来设置热区的匹配方式，它包含有三种方式：

“单击”选项：选择该选项时，用户用鼠标指针在热区中单击左键就会匹配该响应。此选项为默认选项。

“双击”选项：选择该选项时，用户用鼠标指针在热区中双击左键就会匹配该响应。

“指针处于指定区域内”选项：选择该选项时，当鼠标指针移动到热区中时就会匹配该响应。当选



择该选项后，“匹配时加亮”复选框灰显不可用。如果擦除方式设置为“在下一次输入之前”，直到用户将鼠标指针移出热区后，Authorware 才会擦除当前的显示信息。

下面还有两个复选框，其意义分别为：

“匹配时加亮”复选框：选择该选项后，当 Authorware 匹配上该热区响应后，热区会以反色高亮显示，表示该热区已经匹配过了。当用户松开鼠标后，热区才会原色显示。例如，假设热区颜色为白色，响应匹配时变为黑色；如果热区是彩色的，它会以反色显示。

“匹配标记”复选框：选择该选项后，在热区的左边界里边显示出一个小正方形标志。该正方形是空心的，如果用户的响应得到匹配，Authorware 会将正方形填充。Authorware 会在擦除交互显示信息时同时擦除小正方形。如果用户调整热区的大小，小正方形的大小也相应地调整。

- “鼠标指针”选项：依然是定义鼠标移动到热区上时鼠标指针的形状，与按钮响应的设置相同，就不再赘述。

由于选择“热区域”选项后响应类型对话框的“交互”标签页与按钮响应的完全相同，所以设置含义请参考按钮响应的“交互”标签页设置。

8.3.2 热区响应实例——制作多媒体界面

下面我们通过一个实例来说明如何使用热区响应。

该实例我们制作了一个多媒体界面，如图 8-21 所示，其中使用了热区响应的几种匹配方式。

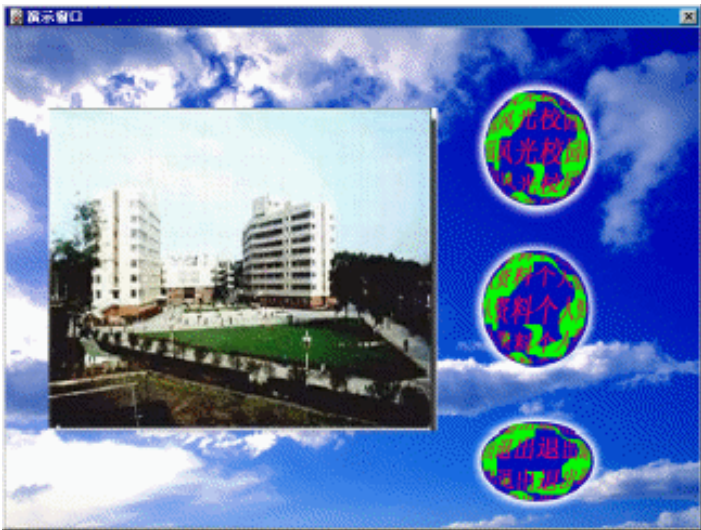


图 8-21 多媒体界面

操作步骤为：

- (1) 新建一个文件，命名为“多媒体界面.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，在“属性”标签页中取消“菜单栏”复选框，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“提示文字”，选择字体为楷体_GB2312、字号为 18，粗体，斜体、字色为紫色，然后单击“修改”按钮确定修改。
- (5) 单击对话框中的“完成”按钮，完成定义文本风格。

- (6) 拖动一个【显示】图标到流程线上，将其命名为“背景”。
- (7) 双击该【显示】图标，打开展示窗口，从外部导入一幅图形作为背景。
- (8) 拖动一个【交互】图标到流程线上，将其命名为“控制”。
- (9) 拖动一个【群组】图标到【交互】图标的右侧，选取热区响应，并将其命名为“校园风光”。
- (10) 再拖动两个【群组】图标到【交互】图标右侧，分别命名为“个人资料”和“退出”。
- (11) 按住<Shift>键并双击【交互】图标上面的响应类型图标，打开展示窗口。
- (12) 在展示窗口中调整三个热区的范围和位置，使其分别覆盖背景图片中的三个球型图形区域。
- (13) 选取文字工具，在背景图片的下方单击一下，然后在选取工具栏中“文本风格”下拉框中选取“正文”选项，输入文本“{文本}”，其中文本是我们要定义的变量，如图 8-22 所示。

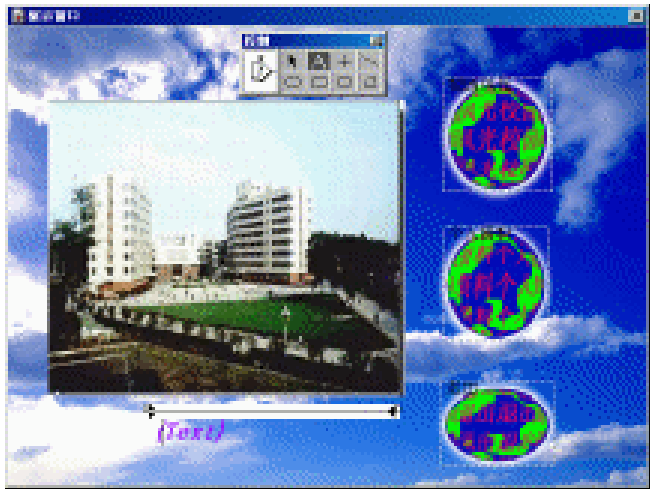


图 8-22 调整热区

- (14) 关闭展示窗口时，系统弹出新建变量对话框，在“初始值”文本框中输入“””，表示文本变量的初始值为空，即 $\text{Text}:=\text{""}$ 。单击“确定”确定。
- (15) 双击“校园风光”【群组】图标上面的响应类型图标，打开响应类型属性对话框，选取“匹配”下拉框中的“指针处于指定区域内”选项，并单击“鼠标指针”选项右侧的按钮，将鼠标指针形状选取为“小手”形状，然后单击“确定”确定。
- (16) 同样，将“个人资料”、“退出”的响应匹配类型也改为“指针处于指定区域内”选项，鼠标指针也为“小手”形状。
- (17) 双击“校园风光”【群组】图标，打开其二级流程窗口，拖动一个【计算】图标到二级流程线上，将其命名为“显示校园风光文字”。
- (18) 在该【计算】图标中输入语句“文本:=“校园风光””。
- (19) 拖动一个【声音】图标，命名为“滴”。
- (20) 双击该【声音】图标，打开其属性对话框，如图 8-23 所示。单击对话框中的“导入”按钮，导入外部声音文件作为热区响应的声音。
- (21) 选择“显示校园风光文字”【计算】图标和“滴”【声音】图标，单击工具栏中的“复制”按钮进行复制。
- (22) 在“个人资料”【群组】图标的二级窗口中，粘贴复制的两个图标，然后将【计算】图标更名为“显示个人资料文字”，并在【计算】图标中输入语句“ $\text{Text}:=\text{“个人资料”}$ ”。

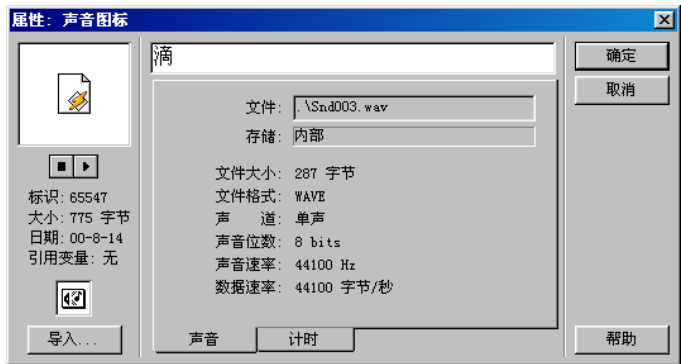


图 8-23 【声音】图标属性对话框

(23) 在“退出”【群组】图标二级窗口中，粘贴复制的两个图标，然后将【计算】图标更名为“显示退出文字”，并在【计算】图标中输入语句“文本:=“退出””。

(24) 拖动一个【交互】图标到“退出”【群组】图标二级窗口中，位于“滴”【声音】图标下面，将其命名为“控制退出程序”。

(25) 拖动一个【计算】图标到该【交互】图标右侧，将其命名为“退出程序”。

(26) 在该【计算】图标中输入语句“Quit()”。

(27) 双击该【计算】图标上面的响应类型图标，打开响应类型属性对话框，在“匹配”下拉框中选取“单击”选项，并单击“鼠标指针”选项右侧的按钮，将鼠标指针形状选取为“小手”形状。

(28) 选择“交互”标签页，选中“永久”复选框，并在“分支”下拉框选取“退出交互”选项。

(29) 调整该对话框后面热区在展示窗口中的位置，使其能够正好覆盖背景图片中的“退出”球形图形区域。单击“确定”按钮关闭响应类型属性对话框。

(30) 此时程序已经设计完成，流程如图 8-24 所示。

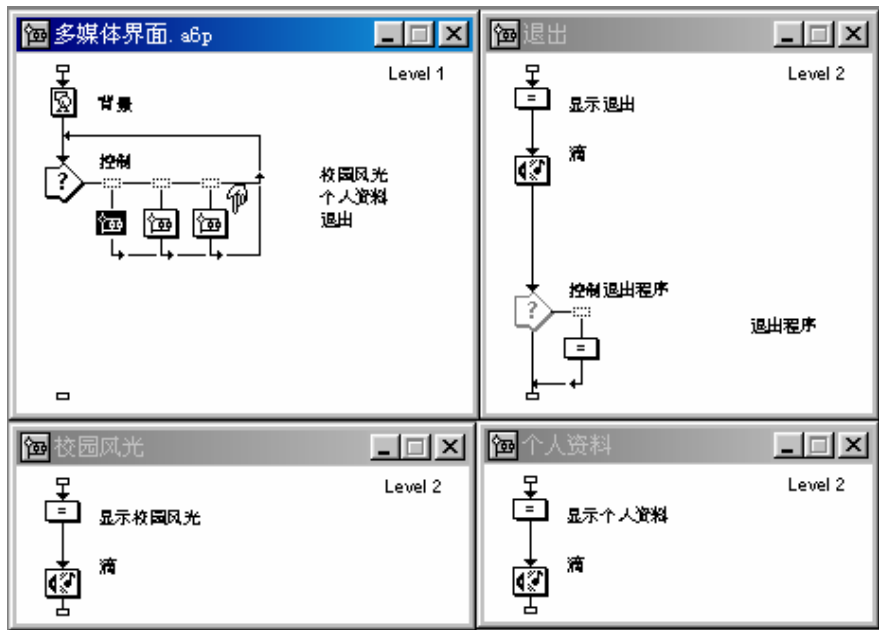


图 8-24 设计好的流程

8.4 热对象响应

热对象是指在展示窗口中定义一个二维的图形对象作为鼠标响应的区域。

尽管一个热对象是与背景相分离的，与热区响应一样，热对象不再必须是矩形形状了。使一个物体具有交互作用，使用热对象响应比拖动一个热区覆盖物体更有效。特别是，当物体不是矩形形状，而且又希望该物体的所有部分都具有交互响应动能，那么使用热对象响应是一个十分有用的方法。比如，如果您希望地图中每一个国家都当作一个按钮，使用热对象响应是最好的办法。而当您在屏幕中移动对象时，不论该对象位于何处，您都希望它仍然保留交互响应，使用热对象也是十分有用的。

需要记住的一个最重要的规则是：当您建立一个热对象，需要保证该对象必须独立位于一个【显示】图标中。实际上 Authorware 会将整个【显示】图标中包含的所有内容作为一个热对象，所以如果将几个对象放在一个【显示】图标中，它们都变成了热对象并且都可以响应交互信息。

8.4.1 选择“热对象”选项后的响应类型属性对话框

使用一个热对象响应可以定义一个热对象，当用户单击、双击或移动鼠标指针到热对象上面时作出响应。热对象响应与热区响应十分相似，特别是选择了一个矩形形状的物体。用户也可以将可以移动的物体定义为热对象。

为了使用这些热对象，可以将每个对象分别放在一个【显示】图标中，然后在响应类型对话框中设置热对象响应的属性。

双击【交互】图标下挂图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“热对象”选项，属性对话框变为如图 8-25 所示。

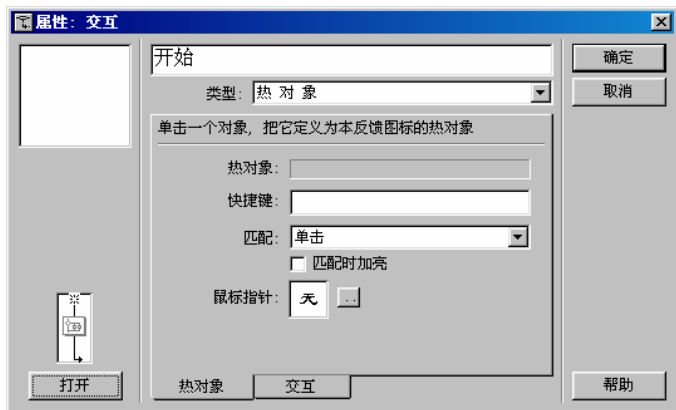


图 8-25 选择“热对象”选项后的响应类型属性对话框

- 在响应类型属性对话框中间有一句提示信息“单击一个对象，把它定义为本反馈图标的热对象”，因为在打开响应类型对话框的同时也打开了展示窗口，所以只需在展示窗口中单击一下想要定义为热对象的物体即可将该物体变为热对象。
- “热对象”文本域：该文本域中显示出了已经选择好的作为热对象的对象名称。一般来说该文本域中显示的是该热对象所处的图标的名称。
- “快捷键”文本框：该文本框依然是用来设置该热对象的快捷键的。

下面的选项以及“交互”标签页与选择“热区域”选项时的属性对话框相同，所以就不再一一赘述。

8.4.2 热对象响应实例——玫瑰情话

下面我们通过一个实例来说明如何使用热对象响应。

在该实例中，我们通过几幅玫瑰图片来作为热对象，当鼠标移动到一个玫瑰图形上时，就会显示出相应的文本信息，如图 8-26 所示。

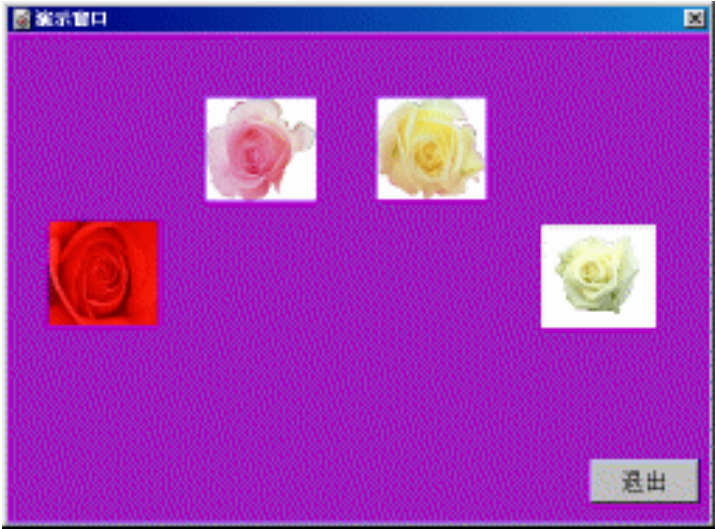


图 8-26 玫瑰情话

操作步骤为：

- (1) 新建一个文件，命名为“玫瑰情话.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“红玫瑰标题”，选择字体为楷体_GB2312，字号为 18，粗体，字色为红色，然后单击“修改”按钮确定修改。
- (5) 同样，再增加三个新的文本风格，分别更名为“粉红色玫瑰标题”、“黄色玫瑰标题”和“白色玫瑰标题”，字色分别为粉红色、黄色和白色，其他设置不变，然后单击“修改”按钮确定修改。
- (6) 再增加一种新的文本风格，将该文本风格更名为“正文”，选择字体为宋体，字号为 12，粗体，斜体，字色为蓝色，然后单击“修改”按钮确定修改。
- (7) 单击对话框中的“完成”按钮，完成定义文本风格。
- (8) 拖动一个【计算】图标到流程线上，在该图标中输入语句“ResizeWindow(500, 350)”。
- (9) 拖动四个【显示】图标到流程线上，分别命名为“红玫瑰”、“粉红色玫瑰”、“黄色玫瑰”和“白色玫瑰”，并且在四个图标中分别导入红玫瑰图片、粉红色玫瑰图片、黄色玫瑰图片和白色玫瑰图片，并调整它们的位置，如图 8-37 中所示。
- (10) 拖动一个【交互】图标到流程线上，将其命名为“选择玫瑰”。
- (11) 拖动一个【显示】图标到【交互】图标右侧，选择热对象响应类型，并将其命名为“显示红玫瑰信息”。

(12) 双击该【显示】图标, 打开展示窗口, 输入两个文本信息, 分别为“红玫瑰”和“象征美好、幸福和忠贞! ”。其中“红玫瑰”文本对象的文本风格为“红玫瑰标题”, 而“象征美好、幸福和忠贞!”文本对象中的文本风格为“正文”。

(13) 调整它们的位置, 如图 8-26 中所示。

(14) 双击该【显示】图标上的响应类型图标, 打开其属性对话框, 单击后面展示窗口中“红玫瑰”图形, 将其作为热对象。

(15) 选择对话框中“匹配”下拉框的“指针在对象上”选项, 并将鼠标指针形状设置成“小手”形状。单击“确定”按钮确定。

(16) 将“显示红玫瑰信息”【显示】图标复制, 然后在【交互】图标右侧粘贴出三个图标来, 将其名称分别更名为“显示粉红色玫瑰信息”、“显示黄色玫瑰信息”和“显示白色玫瑰信息”。

(17) 将【显示】图标中的“红玫瑰”文本对象分别改为“粉红色玫瑰”、“黄色玫瑰”和“白色玫瑰”, 并且文本风格相应改变成“粉红色玫瑰标题”、“黄色玫瑰标题”和“白色玫瑰标题”, 将“象征美好、幸福和忠贞!”文本对象改为“适合初恋时赠送, 而且要送花蕾, 象征美好的恋情含苞待放!”、“象征忌妒或不快, 适合恋情遭受挫折, 也用作爱的表白, 道歉和失望!”和“象征和谐美好, 纯洁的友谊, 适合一般友人赠送!”。

(18) 拖动一个【计算】图标到【交互】图标的最右侧, 将其命名为“退出”。

(19) 在该【计算】图标中输入语句“Quit()”。

(20) 单击【计算】图标上面的响应类型图标, 将响应类型改变成按钮响应, 并将按钮移动到展示窗口的右下角。

(21) 此时程序已经设计完成, 流程如图 8-27 所示。

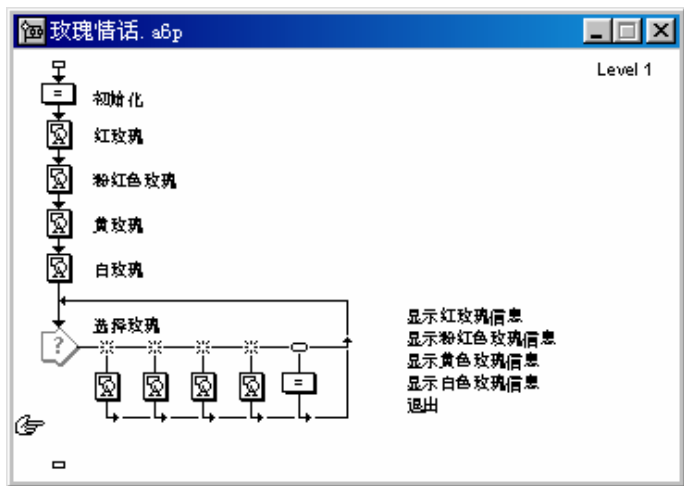


图 8-27 设计好的流程

8.5 目标区域响应 (目标区)

目标区域响应是指当用户拖动一个显示对象到一个指定的区域中时, 就会触发响应, 这个指定的区域就是目标区域。

让用户拖动一个对象到一个目标区域而产生响应, 这样可以创建一些十分有趣的交互。比如, 可以让用户拖动一个国家到创建的地图的合适位置; 可以让用户将一个机器的各个部件组装起来; 可以让用



户将各个阶段进行正确的排列而跟踪进程的流程；可以通过将各个事件按照它们实际发生的顺序排列而创建不同事件结果。当然，也可以创建一个交互让用户自由地排列花园中的风景或房间里的家具。这些都需要使用 Authorware 提供的目标区域响应类型。

对于每一个与【交互】图标相链接的目标区域响应类型标志，Authorware 会在屏幕上显示出一个目标区域。用户可以拖动与目标区域相链接的对象到屏幕上的合适位置，Authorware 会同时移动目标区域到用户指定的位置。

8.5.1 选择“目标区”选项后的响应类型属性对话框

当希望用户拖动一个对象到指定的区域而产生响应，可以使用目标区域响应。当用户拖动对象到目标区域中，Authorware 就会执行目标图标分支。

每一个链接有交互响应的目标区域，其外形是一个内有“X”号的矩形区域。其中目标区域的标题是目标图标的名称。由于在程序运行时，目标区域并不会显示出来，只有用户使用工具箱编辑时或暂停作品执行时才能看到目标区域。

当建立一个目标区域响应时，认为是建立了一个界线，防止用户将对象拖动到屏幕之外。如果用户拖动对象到了一个错误位置，可以让该对象自动回到原来位置。当然将对象拖动后也可希望该对象停止在拖动的新位置上。

在【交互】图标中建立一个包括整个屏幕的全目标区域响应是十分有用的。这样，不管用户将对象拖动到屏幕的什么位置，全目标区域都允许程序重新回到【交互】图标。与全热区响应相同，要将全目标区域响应的图标放在【交互】图标的下挂图标的最右边，这样用户输入的响应就会首先匹配小目标区域响应，等到所有小目标区域响应都没有得到匹配，才触发全目标区域响应。

双击目标图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“目标区”选项，显示出选择“目标区”选项后的响应类型属性对话框，如图 8-28 所示。



图 8-28 选择“目标区”选项后的响应类型属性对话框

- 在响应类型属性对话框中间有一句提示信息“选择目标对象”，因为在打开响应类型对话框的同时也打开了展示窗口，所以只需在展示窗口中单击一下想要定义为目标对象的物体即可将该物体变为目标对象。
- “大小”选项：该选项是用来目标区域的长宽大小的。

其中“X”文本框中的数值决定着目标区域的长度；“Y”文本框中的数值决定着目标区域的宽度。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变目标区域的大小，也可以通过在展示窗口中直接拖动目标区域的句柄来改变目标区域大小。

- “位置”选项：该选项是用来控制目标区域在展示窗口中的位置的。

其中“X”文本框中的数值决定着目标区域的X坐标；“Y”文本框中的数值决定着目标区域的Y坐标。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变目标区域的位置，也可以通过在展示窗口中直接拖动目标区域的句柄以外的部分来移动目标区域，改变目标区域的位置。

- “放下”下拉框：该下拉框是用来控制当用户停止拖动目标对象时，目标对象的响应方式。它包括三个选项：

“在目标点放下”选项：选中该选项后，当用户拖动目标对象时，目标对象会停留在用户释放时的位置。该选项为系统默认选项。

“返回”选项：选中该选项后，当用户拖动目标对象时，如果目标对象没有被拖动到目标区域，目标对象会自动返回到原来所处的位置。该选项对设置为“重试”分支方式的错误响应非常有效，因为当用户没有将目标对象移动到目标区域，它会让其重新回到原来位置，让用户重新作出选择。

“在中心定位”选项：选中该选项后，当用户拖动目标对象到目标区域中，目标对象会自动在目标区域中居中。该选项对于当用户将目标对象正确移动后而作出反应是十分有用的。

- “目标对象”文本框：该文本框中显示出了选择好的目标对象的名称，该名称一般是目标对象所处的图标名称。
- “允许任何对象”复选框：选中该选项时，当用户拖动任何对象到目标区域中，都会触发响应。如果选择了该选项，可以使用【显示】图标属性对话框中的“版面布局”标签页来创建可移动的对象。当选定该选项时，“目标对象”文本框灰显不可用，同时中间的提示信息消失。

当用户拖动一个对象到目标区域中，系统变量 ObjectMatched 中记录了该拖动对象所处的【显示】图标的标题；而系统变量 ObjectMoved 中记录了用户最近一次单击并拖动的对象的名称，不管该对象是否匹配了响应。

8.5.2 创建目标区域响应

1. 建立可拖动对象

- (1) 选择包括可拖动对象的【显示】图标。
- (2) 选择菜单[修改]→[图标]→[属性...]命令，打开【显示】图标属性对话框。
- (3) 选择“版面布局”标签页，并在“可移动性”下拉框中选择“任何地方”选项，然后单击属性对话框的“确定”按钮确定即可。

2. 创建目标区域标志

- (1) 拖动一个【交互】图标到设计窗口的流程线上，并将其命名。
- (2) 拖动一个图标到【交互】图标的右边。
- (3) 在弹出的响应类型对话框中选择“目标区”响应类型并单击“确定”。
- (4) 选择目标图标并将其命名。
- (5) 按照上面的步骤在将第一个目标区域与目标对象建立链接前创建其他目标区域；或者将第一个目标区域与目标对象建立链接后，然后按照上面步骤再建立其他目标区域。

3. 建立目标区域与目标对象的链接

- (1) 从程序的开头或从开始旗处运行程序。
- (2) 当程序运行到【交互】图标时，系统会弹出目标区域响应属性对话框，并且在展示窗口中有一个虚线矩形的目标区域出现。
- (3) 拖动对象——不是虚线矩形的目标区域，到希望标记为目标区域的地方。Authorware 会自动将虚线矩形的中心移动到拖动位置的中心而将其标记为目标区域。

(4) 单击目标区域响应属性对话框的“交互”标签页，选择合适的擦除方式和分支方式，然后单击“确定”确定。

8.5.3 目标区域响应实例——识物学单词

下面我们通过一个实例来说明如何使用目标区域响应。

在该实例中，通过将某个图片拖动到对应英语单词下面的方框中，就算做对了，如图 8-29 所示。



图 8-29 识物学单词

操作步骤为：

- (1) 新建一个文件，命名为“识物学单词.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“红玫瑰标题”，选择字体为楷体_GB2312，字号为 18，粗体，斜体，字色为紫色，然后单击“修改”按钮确定修改。
- (5) 同样，再增加一个新的文本风格，更名为“提示信息”，字体为宋体，字号为 12，粗体，字色分别为蓝色，其他设置不变，然后单击“修改”按钮确定修改。
- (6) 单击对话框中的“完成”按钮，完成定义文本风格。
- (7) 拖动一个【计算】图标到流程线上，在该图标中输入语句“ResizeWindow(400, 300)”。
- (8) 拖动四个【显示】图标到流程线上，分别命名为“背景”、“老鹰”、“小鸟”和“蝴蝶”，并且在四个图标中分别导入背景图片、老鹰图片、小鸟图片和蝴蝶图片，并调整它们的位置。
- (9) 在“背景”【显示】图标中输入四个文本对象，一个为“识物学单词”，文本风格为“标题”；其他三个分别为“Eagle”、“Bird”和“Butterfly”，文本风格为系统默认，然后在这三个文本对象下面分别绘制一个矩形，如图 8-30 所示。



图 8-30 添加文本对象

(10) 拖动一个【交互】图标到流程线上，将其命名为“选择”。

(11) 拖动一个【显示】图标到该【交互】图标右侧，选取目标区域响应，并将其命名为“移动老鹰”。

(12) 双击该【显示】图标上的响应类型图标，打开其属性对话框，单击后面展示窗口中的“老鹰”图形，并将其移动到文本对象“Eagle”下面的矩形中，然后调整目标区域，使之正好覆盖整个矩形。

(13) 按住<Shift>键并双击“移动老鹰”【显示】图标，打开展示窗口，输入文本“Right!”，文本风格为“提示信息”，并将该文本对象移动到三个矩形的正下方。

(14) 将“移动老鹰”【显示】图标复制，然后在【交互】图标右侧粘贴出五个该【显示】图标，分别将图标更名为“移动小鸟”、“移动蝴蝶”、“错误移动老鹰”、“错误移动小鸟”和“错误移动蝴蝶”。

(15) 双击“移动小鸟”上的响应类型图标，打开其属性对话框，然后单击展示窗口中的“小鸟”图形，并将其移动到文本“Bird”下面的矩形中。

(16) 双击“移动蝴蝶”上的响应类型图标，打开其属性对话框，然后单击展示窗口中的“蝴蝶”图形，并将其移动到文本“Butterfly”下面的矩形中。

(17) 双击“错误移动老鹰”上的响应类型图标，打开其属性对话框，然后单击展示窗口中的“老鹰”图形，并将目标区域调整到整个屏幕大小。

(18) 双击“错误移动老鹰”【显示】图标，将文本“Right!”改为“Wrong!”。

(19) 双击“错误移动小鸟”上的响应类型图标，打开其属性对话框，然后单击展示窗口中的“小鸟”图形，并将目标区域调整到整个屏幕大小。

(20) 双击“错误移动小鸟”【显示】图标，将文本“Right!”改为“Wrong!”。

(21) 双击“错误移动蝴蝶”上的响应类型图标，打开其属性对话框，然后单击展示窗口中的“蝴蝶”图形，并将目标区域调整到整个屏幕大小。

(22) 双击“错误移动蝴蝶”【显示】图标，将文本“Right!”改为“Wrong!”。

(23) 拖动一个【计算】图标到【交互】图标的最右侧，将其命名为“退出”。

(24) 将其响应类型改变为按钮响应，并在该【计算】图标中输入语句“Quit()”。

(25) 将“退出”按钮进行调整，并移动到展示窗口的右下角。

(26) 此时程序已经完全设计完成，完成后的程序流程如图 8-31 所示。

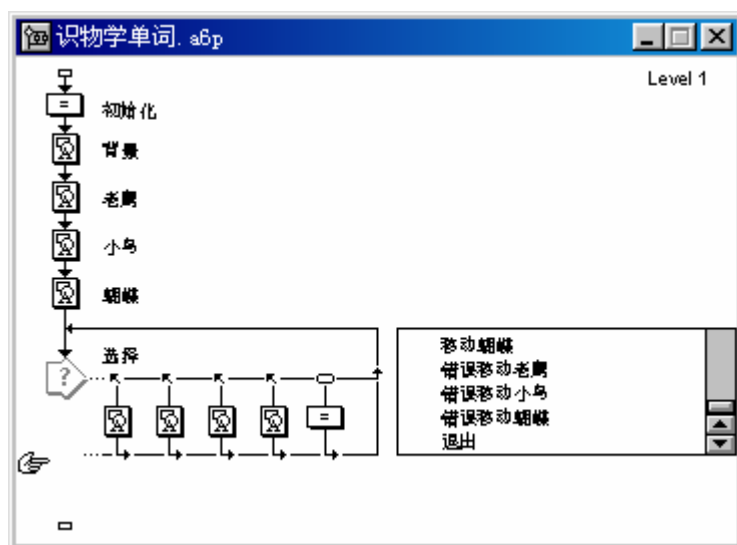


图 8-31 设计好的程序流程

8.6 下拉菜单响应（下拉菜单）

下拉菜单是提供在菜单栏中添加一些菜单命令，让用户直接在菜单中选择操作。它不占去展示窗口中的空间。在缺省状态下，Authorware 的展示窗口中的菜单栏只有一个“文件”菜单项，“文件”菜单中只包括一个“退出”命令。

用户在做一些操作的时间里，希望一些选项命令在这一段时间里都有效，此时下拉菜单响应应该是最好的方法。所以，下拉菜单响应通常是“永久”响应——不管用户在做什么，交互响应总是有效的。

如果为下拉菜单项设置了快捷命令，如同其他任何标准的 Windows 程序一样，Authorware 会自动在菜单中显示出快捷键。当创建下拉菜单响应时，Authorware 就会将该【交互】图标标题来作为下拉菜单的标题，而将该【交互】图标的下挂目标图标的名称作为该菜单各个菜单项的名称。

尽管在下拉菜单响应中的所有菜单项通常是下拉菜单响应，但并不都是。比如，在一个需要用户立即从菜单中作出选择的响应，可以在菜单响应中添加一个限制时间响应类型标志，如果用户不作出选择，就在过一段时间后自动执行响应。

8.6.1 选择“下拉菜单”选项后的响应类型属性对话框

使用下拉菜单响应来建立下拉菜单。用户可以从菜单中选择操作命令。对于每一个下拉菜单，都需要一个单独的【交互】图标，而该【交互】图标下挂的目标图标是该菜单的各个菜单项。

对于建立永久响应、该响应总是可用并且响应不十分显眼，这种情况下使用下拉菜单响应特别有效。对于需要用户立即作出响应的，而用户在给定时间内没有作出选择，可以使用一个限制时间响应。

在目标图标的响应类型图标上双击，打开响应类型属性对话框，选择“类型”下拉框中的“下拉菜单”选项，出现选择“下拉菜单”选项后的响应类型属性对话框，如图 8-32 所示。

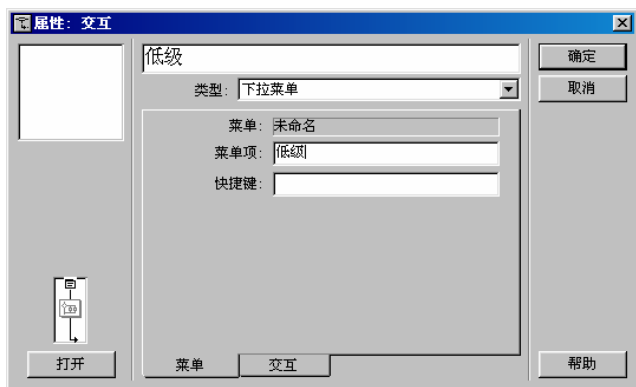


图 8-32 选择“下拉菜单”选项后的响应类型属性对话框

- “菜单”文本框：该文本框显示出了主菜单的名称，也就是该目标图标所处的【交互】图标的名称。
- “菜单项”文本框：在该文本框中输入下拉菜单中需要出现的菜单项名称。可以使用一些特殊符号来控制在下拉菜单中显示的标题情况。使用表 8-2 中的符号可以显示出相应的标题：

表8-2 特殊符号与相应菜单项对应表

特殊符号	对应菜单项
(菜单项名称)	该菜单项灰显
(或菜单项标题为空)	显示一个空白项
(-或-)	显示一个分隔线
&	将该符号放置在菜单项名称中的一个字符前，可以使该字符成为带有下划线的热键（或加速键）。如果要在菜单项中显示出一个&符号，必须输入两个&符号。

- “快捷键”文本框：使用该文本框来指定<Ctrl>键或<Alt>键组合的键来组成打开该菜单项的快捷键。

8.6.2 建立下拉菜单响应

一般来说，创建一个下拉菜单响应的步骤为：

- (1) 拖动一个【交互】图标到设计窗口的流程线上，并将其命名。该【交互】图标的名称即为菜单栏中主菜单的名称。
- (2) 拖动一个图标到【交互】图标右边，建立菜单中第一个菜单项。Authorware 是使用每一个目标图标的标题来作为菜单的菜单项名称。
- (3) 在弹出的响应类型对话框中选择“下拉菜单”选项，单击“确定”按钮确定。
- (4) 双击响应类型图标，打开响应类型属性对话框，在对话框的“菜单项”文本框中输入该菜单项的名称。
- (5) 单击响应类型属性对话框中的“交互”标签页，选择擦除方式和分支方式，然后单击“确定”按钮确定。
- (6) 如果需要其他菜单项，继续拖动图标到【交互】图标右边，并按照上面步骤进行。为每一个作为菜单项的目标图标建立信息。

8.6.3 为菜单添加分隔线、空白项和指定菜单项热键

1. 为菜单项添加分隔线

添加分隔线的步骤为：

- (1) 拖动一个【群组】图标到需要显示分隔线的位置。
- (2) 选择该【群组】图标，将其名称改为“(- ”或“- ”即可，如图所示。

2. 为菜单添加空白项

添加分隔线的步骤为：

- (1) 拖动一个【群组】图标到需要显示空白项的位置。
- (2) 选择该【群组】图标，将其名称改为“(”或名称为空即可，如图 8-33 所示。程序运行的效果如图 8-34 所示。

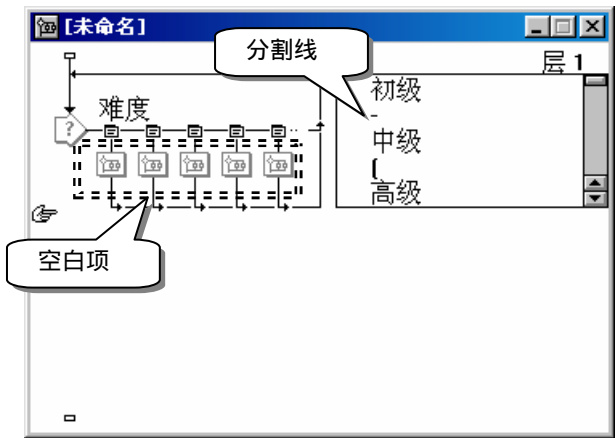


图 8-33 添加了分隔线和空白项的程序流程

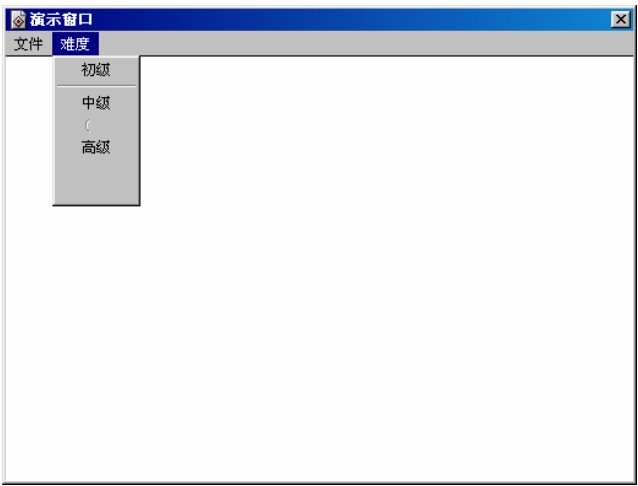


图 8-34 对应的菜单项

需要注意的是，使用左括号“(”添加的空白项，在程序运行时灰显示出一个左括号；而将图标名称改为空添加的空白项是真正的空白项，程序运行时不显示任何信息，就是一个空白项，如图 8-34，所

示。

3. 为菜单项指定热键/加速键

热键，就是一个单键，当单击该键时，会自动选择一个特殊的菜单项，就像用鼠标选择下拉菜单中的菜单项一样。

要为一个菜单项指定一个热键，就在该菜单项标题里使用一个&符号。将&符号放置在希望成为热键的字符前面即可。例如，一个菜单项名为“Test”，使用“&Test”可以将标题中的“T”指定为热键，在程序运行时，“T”带有下划线。如果使用“T&est”，将指定“e”作为热键，如图 8-35 所示。程序运行的效果如图 8-36 所示。

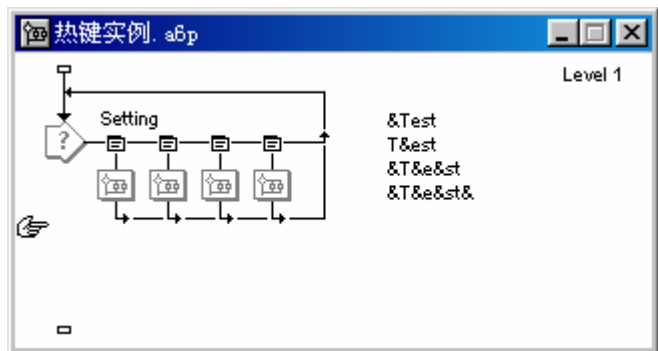


图 8-35 热键响应流程

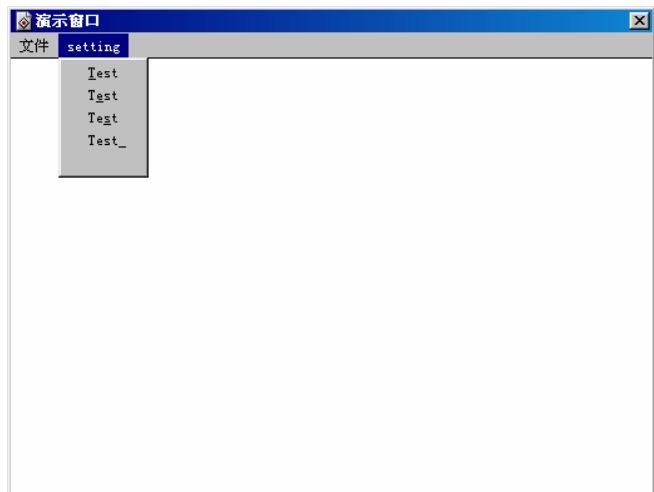


图 8-36 对应的菜单项

需要注意的是，如果在菜单项名称中使用了多个&符号，则系统将认为最后一个&符号后面的字符为热键，同时其他&符号并不显示。如果最后一个&符号后面已经没有字符了，则在菜单项标题中最后显示出一个下划线。例如，如果菜单项输入“&T&e&st”，则程序执行时，只有字符“s”带有下划线，即热键为“s”，其他&符号不在菜单项中显示；如果输入“&T&e&st&”，则菜单项会显示为“Test_”。

菜单项的热键对于大小写不敏感。例如，不要将 R 指定为一个菜单项的热键而将 r 作为另外一个菜单项的热键。

如果要在菜单项中显示&字符，即&字符是一个菜单项的一部分，则需要在显示的地方连续输入两

个&字符（&&）。

8.6.4 为菜单项指定快捷键和灰显/激活菜单项

1. 指定菜单项的快捷键

- 快捷键规则

要使用快捷键，只需要在响应类型属性对话框的“快捷键”文本框中输入要设置为快捷键的字符即可。如，在“快捷键”文本框中输入“R”，表示快捷键为<Ctrl>+R；如果输入“AltR”，表示快捷键为<Alt>+R；如图 8-37 所示为添加快捷键后的菜单。

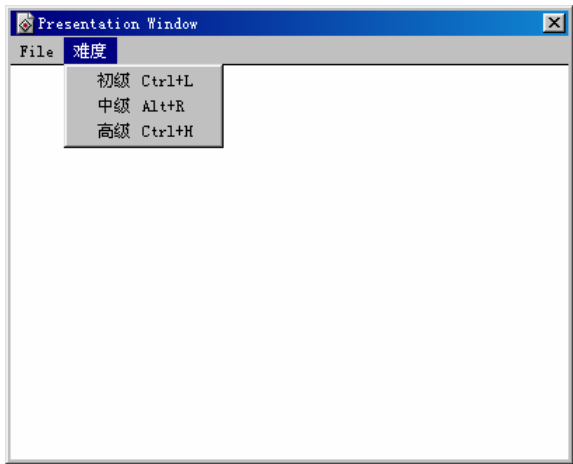


图 8-37 为菜单项添加快捷键

下拉菜单菜单项的快捷键对于大小写不敏感，即不要将 R 指定为一个菜单项的热键而将 r 作为另外一个菜单项的热键。但是 Authorware 会区别在同一个键上的不同字符。例如，可以将“+”号和“=”号分别指定给不同的菜单项作为快捷键，尽管它们在同一个键上。如果指定“+”号作为一个菜单项的快捷键，则该快捷键应该为<Ctrl>+<Shift>+“+/=”键；如果指定“=”号作为一个菜单项的快捷键，则该快捷键变为<Ctrl>+“+/=”键。

- 为菜单项指定快捷键的步骤：

- (1) 双击菜单项响应标志，打开响应类型属性对话框。
- (2) 选取“Menu”标签页，在“快捷键”文本框中按照快捷键规则输入希望作为快捷键的键名。

2. 灰显/激活菜单项

要灰显菜单项，可以在该菜单项名称前面添加一个左括号“（”，如图 8-38 所示。



图 8-38 灰显菜单项

该方法有一些缺点，就是虽然灰显了菜单项，但是左括号也在菜单项中显示出来了，而且该方法无法在程序中需要灰显菜单项的地方灰显。当需要激活该灰显菜单项时，又需要将该菜单项激活。

我们可以使用下面方法来控制灰显和激活菜单项：

(1) 双击目标图标的菜单响应类型图标，打开响应类型属性对话框。

(2) 选取对话框的“交互”标签页，在“激活条件”文本框中输入一个变量或表达式，用变量或表达式的值来控制该菜单项是灰显还是激活状态。如果变量或表达式的值为“TRUE”，则该菜单项处于激活状态；反之，如果该变量或表达式的值为“False”，则该菜单项处于灰显状态。

(3) 通过程序来控制变量或表达式的值来决定何时该菜单项灰显和激活。

8.7 条件响应（条件）

条件响应使用在当某个特定条件为“TRUE”时，程序就会执行一些操作。许多逻辑性变量、表达式、或可返回逻辑性值的函数都可以作为条件响应的目标条件。例如，可以检测用户在输入文本后是否按下了 Caps Lock 键，或者如果用户在完成一个课程的一节时有一个限制时间，此时可以在交互响应的开始告诉用户还有多少剩余时间；或者当某个条件为“TRUE”时，就跳过整个交互响应（如，用户已经正确地回答了所有问题）。

8.7.1 选择“条件”选项后的响应类型属性对话框

在目标图标的响应类型图标上双击，打开响应类型属性对话框，选择“类型”下拉框中的“条件”选项，出现选择“条件”选项后的响应类型属性对话框，如图 8-39 所示。

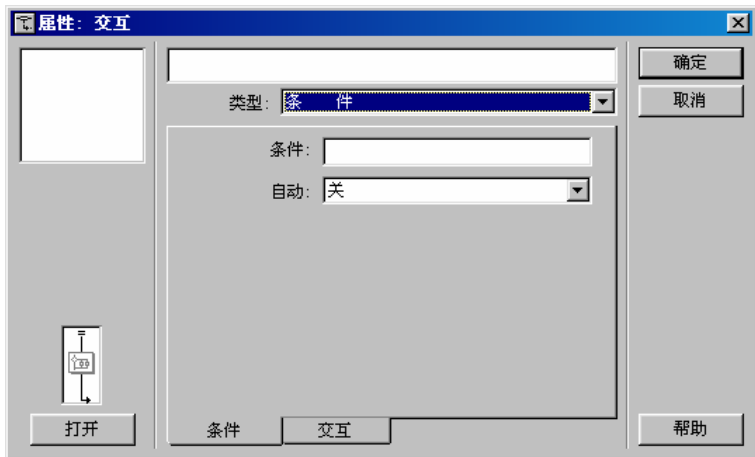


图 8-39 选择“下拉菜单”选项后的响应类型属性对话框

下面介绍“条件”标签页中各个选项的含义：

- “条件”文本框：该文本框中是设置该交互项的响应条件的。

在该文本框中输入条件语句，可以是一个数、变量、表达式、或逻辑性函数，当该文本框中的条件语句为“TRUE”时，Authorware 就会匹配该响应。

“条件”文本框中的内容与该目标图标的标题相同，当在“条件”文本框中修改了内容，该目标图标的标题也相应地发生改变；同样，如果改变了该目标图标的标题名称，“条件”文本框的内容也相应地改变。

可以在“条件”文本框中输入任何类型的变量或表达式，而不一定必须是逻辑性变量或条件性表达式，Authorware 会依照表 8-3 所示的规则来判断该变量或表达式是“TRUE”还是“FALSE”。

表8-3 条件规则

规 则	示 例
0 为“ FALSE ”，其他数值为“ TRUE ”	S<8——当自定义变量 S 的值小于 8 时为“ TRUE ”，否则为“ FALSE ”。
	SESSIONHOURS>=0.25——当系统变量 SESSIONHOURS 中的值大于等于 0.25 时为“ TRUE ”，否则为“ FALSE ”。
	WORDCOUNT<5 CHARCOUNT>25 —— 当 系 统 变 量 WORDCOUNT 的 值 小 于 5 个 单 词 或 者 系 统 变 量 CHARCOUNT 的值大于 25 个字母时，表达式为“ TRUE ”，否则为“ FALSE ”。
字符串 TRUE , T , YES , ON 为“ TRUE ”，其他字符串为“ FALSE ”	STR——如果自定义字符串变量 STR 的值为“ TRUE ”、或为“ T ”、或为“ YES ”、或为“ ON ”时为“ TRUE ”，否则为“ FALSE ”。

● “自动”下拉框：该下拉框是用来设置匹配该交互项的方式。它包括三个选项：

“关”选项：选择该选项时，只有当用户输入响应并且指定的匹配条件为“TRUE”时，Authorware 才会匹配该响应，执行该响应分支的内容。

“为真”选项：选择该选项时，Authorware 会在程序运行过程中不断监视匹配条件的值，只有匹配条件为“TRUE”，Authorware 就会执行该响应分支的内容。该选项对于依靠匹配条件的值来作出自动的反应很有用。当匹配条件为“TRUE”时，条件响应会不断匹配。如果希望 Authorware 去匹配其他响应或要退出交互，必须对匹配条件进行改变，使之其值为“FALSE”。

“为假”选项：选择该选项时，在 Authorware 执行【交互】图标过程中，只有匹配条件的值从“FALSE”变为“TRUE”时，Authorware 才会匹配该条件响应。

由于“交互”标签页与其他响应的“交互”标签页除了“永久”复选框是灰显的外，几乎完全相同，所以就不再一一赘述。

8.7.2 条件响应实例——完善识物学单词

在目标区域响应中，我们列举了一个实例——识物学单词，现在我们使用条件响应来完善它。操作步骤为：

- (1) 删除“错误移动老鹰”、“错误移动小鸟”和“错误移动蝴蝶”三个图标。
- (2) 按住<Ctrl>键单击“移动老鹰”图标下方，使其分支方式为“继续”。
- (3) 按住<Ctrl>键单击“移动老鹰”图标标题的右侧，使其变为正确响应，即标题前出现一个“+”号。
- (4) 同样，将“移动小鸟”、“移动蝴蝶”的分支方式改变为为“继续”，都改为正确响应。
- (5) 拖动一个【群组】图标到【交互】图标的最右侧。
- (6) 双击其上面的响应类型图标，打开其属性对话框，选择条件响应，然后在“条件”文本框中输入变量 AllCorrectMatched，然后单击“确定”确定。
- (7) 此时该【群组】图标名称变为 AllCorrectMatched，双击该【群组】图标，打开其二级窗口。
- (8) 拖动一个【显示】图标到二级流程线上，将其命名为“全部正确”。
- (9) 双击该【显示】图标，在展示窗口中输入文本“ All Correct! ”，文本风格为“标题”，将其移动到屏幕的最下方。
- (10) 拖动一个【计算】图标到二级流程线上，将其命名为“固定图形”。
- (11) 在该【计算】图标中输入如下语句：



可移动性@"老鹰":=FALSE
可移动性@"小鸟":=FALSE
可移动性@"蝴蝶":=FALSE
-固定“老鹰”、“小鸟”和“蝴蝶”图形

(12) 此时程序设计完成，程序流程如图 8-40 所示。

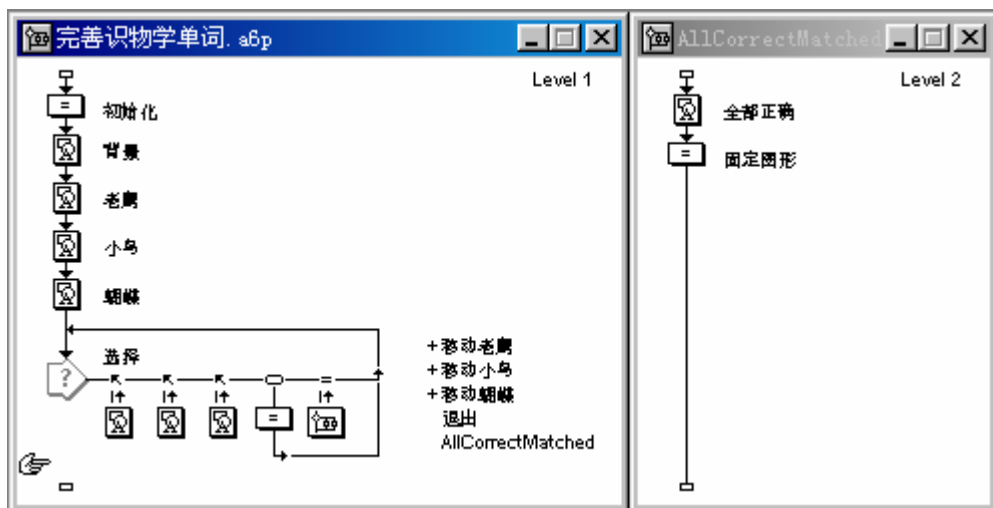


图 8-40 设计好的程序流程

8.8 文本输入响应（文本输入）

文本输入响应也就是创建一个文本域，用户可以在文本域中输入文本，如果输入文本与期待输入的文本相匹配，则程序就响应该输入，转而执行该交互项下挂目标图标的内容。

由于当程序希望用户输入文本，此时用户输入的信息范围太大。Authorware 在匹配文本时提供了很大的灵活性。例如，可以使用通配符来允许拼写上细微变化的文本；可以使用通配符来接受任何输入文本；也可以匹配代表相同意思的不同单词或词组；并且可以让 Authorware 忽略大小写、忽略多余的标点符号、空格、多余单词、或者忽略单词顺序。

文本输入响应与按钮响应有所不同。如果在【交互】图标下面添加四个按钮响应，那么在屏幕上就会出现四个按钮；而文本输入响应则不同，不管【交互】图标下面有多少个文本输入响应，屏幕上始终只有一个文本输入域。这是由于不管在程序中创建了多少个文本输入响应，用户在某一时刻只能输入一个文本。如果在程序中建立了几个不同的信息——比如用户名和密码，此时就需要创建两个独立的【交互】图标，其中一个匹配用户名，一个匹配密码。

当用户输入文本后，Authorware 将输入的文本信息存储在系统变量 Entry 文本中，让程序根据用户输入的信息而做出反应。当然，也可以将 Entry 文本中存储了文本信息赋给一个自定义变量，然后处理该信息——在屏幕上显示出来、将信息保存在一个文件中、或将信息写入到一个数据库中。

8.8.1 选择“文本输入”选项后的响应类型属性对话框

双击目标图标对应的响应类型图标，打开响应类型属性对话框，从“类型”下拉框中选择“文本输入”选项，属性对话框变为如图 8-41 所示。

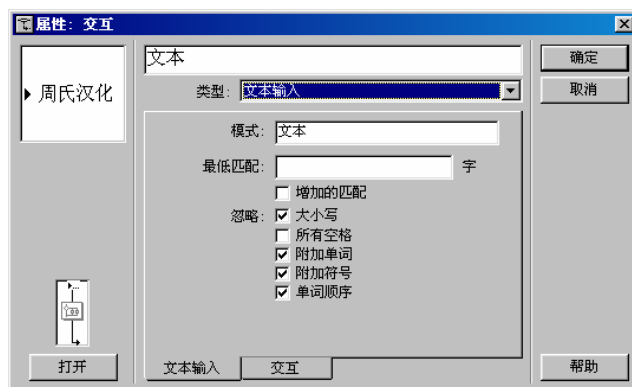


图 8-41 选择“文本输入”选项后的响应类型属性对话框

1. “文本输入”标签页

- “模式”文本框：该文本框中的信息是用来控制用户要输入的单词或短语。用户输入的文本不需要匹配所有的单词，匹配的多少取决于至少匹配设置。如果在该文本框中输入通配符“*”，将会响应用户输入的任何文本。如果在该文本框中输入多个独立的文本，各文本之间需要使用“|”分隔，表示用户输入的文本只要匹配其中一个文本即可响应。如：在“模式”文本框中输入“dong|xiaodong”，则程序运行时，用户输入“dong”或“xiaodong”都可以触发响应。该文本框中的信息将作为该目标图标标题。
- “最低匹配”文本框：使用该文本框来决定用户必须输入的最少单词个数（相对于“模式”文本框中单词总数）。可以在该文本框中输入数字、变量、表达式或数值型函数。
- “增加的匹配”复选框：如果在“模式”文本框中包含多个单词，则选择该复选框，让用户尝试输入多次才得到整个单词。例如，如果需要匹配的文本为“文本输入”，则用户需要一次尝试中输入“文本”，而在另一次尝试中输入“输入”即可触发响应。
- “忽略”选项：该选项是决定用户输入的文本可以忽略哪些因素，也就是说，在使用了下面各项后，系统在什么情况下才认为用户输入的文本与匹配文本相匹配了。

“大小写”复选框：选择该选项时，Authorware 将忽略大小写。如果用户输入的文本除了大小写外都与匹配文本相同，选择该复选框后，Authorware 认为此时已经匹配响应了。

“所有空格”复选框：选择该选项时，Authorware 将忽略用户输入的文本中的所有空格。在缺省情况下，Authorware 会将用户输入的文本与匹配文本一个字母一个字母地进行比较。一个空格表明另一个新单词的开始，多个空格将被忽略。

当选择该选项时，Authorware 会忽略用户输入的文本中的空格，而将整个输入文本作为一个单词或一个无间断的字符串。

“附加单词”复选框：选择该选项时，Authorware 会忽略用户输入文本中多余的单词。例如，选择该选项时，“Macromedia Authorware 6 is OK”可以匹配文本“Authorware OK”。

“附加符号”复选框：选择该选项时，Authorware 会忽略用户输入文本中的标点符号。例如，选择该选项时，如果匹配文本为“Authorware，OK”，而用户输入的是“Authorware OK!”，Authorware 由于忽略标点符号，所以仍然会匹配该响应。

“单词顺序”复选框：选择该选项时，Authorware 会忽略用户输入文本中单词的顺序。例如，选择该选项时，如果匹配文本为“Authorware OK”，而用户输入的文本为“OK Authorware”，Authorware 依然会匹配该响应。当然，如果不选择该选项，用户必须按照程序指定的单词顺序输入文本才可以触发响应。

2. “交互”标签页

单击响应类型属性对话框中的“交互”标签页，打开该标签页，如图 8-42 所示。



图 8-42 响应类型属性对话框的“交互”标签页

在该标签页中的选项，除了“范围”选项和“激活条件”文本框灰显不可用外，其余的选项与其他响应类型的“交互”标签页完全相同，所以就不再一一赘述。

8.8.2 建立一个文本输入响应

建立一个文本输入响应的步骤为：

- (1) 拖动一个【交互】图标到设计窗口的流程线上，并将其命名。
- (2) 拖动一个图标到【交互】图标的右边，从弹出的响应类型对话框中选择“文本输入”选项，然后单击“确定”按钮确定。
- (3) 如果响应类型对话框不自动出现，可以双击目标图标对应的响应类型图标，打开响应类型属性对话框，然后在“类型”下拉框中选择“文本输入”选项，然后单击“确定”按钮确定。
- (4) 在定义第一个目标文本之前按照上面的步骤建立其他文本输入响应，或者先定义第一个目标文本，然后再建立其他响应。
- (5) 双击文本输入响应图标，打开响应类型属性对话框，在对话框的“模式”文本框中输入匹配文本。
- (6) 在响应类型属性对话框中单击“交互”标签页，然后选择合适的擦除方式和分支方式，确定后单击“确定”按钮。
- (7) 建立目标图标的信息。

8.8.3 设置文本域——【交互】图标文本域属性对话框

按住<Ctrl>键并双击【交互】图标，打开【交互】图标属性对话框，然后单击属性对话框中的“文本区域”按钮，即可打开【交互】图标文本域属性对话框，如图 8-43 所示。

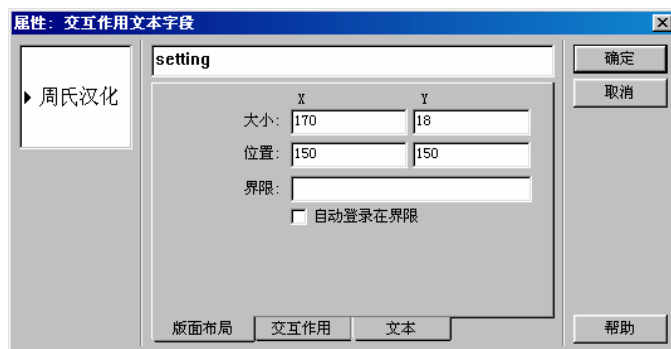


图 8-43 【交互】图标文本域属性对话框

该对话框包括了三个标签页：

1. “版面布局”标签页

该标签页主要是定义文本域的在展示窗口中的大小、位置等信息的。

- “大小”选项：该选项是用来控制文本域的长宽大小的。

其中“X”文本框中的数值决定着文本域的横向长度，即文本域的长度；“Y”文本框中的数值决定着文本域的纵向长度，即文本域的宽度。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变文本域的大小，也可以通过在展示窗口中直接拖动文本域的句柄来改变文本域大小。

- “位置”选项：该选项是用来控制文本域在展示窗口中的位置的。

其中“X”文本框中的数值决定着文本域的水平方向坐标，即文本域的 X 坐标；“Y”文本框中的数值决定着文本域的垂直方向坐标，即文本域的 Y 坐标。

用户可以在“X”文本框和“Y”文本框中直接输入数值来改变文本域的位置，也可以通过在展示窗口中直接拖动文本域的句柄以外的部分来移动文本域，改变文本域的位置。

使用“大小”选项和“位置”选项来定义文本域的大小和位置，可以直接在“X”文本框和“Y”文本框中输入数值或变量来控制。例如，可以使用系统变量 ClickX 和 ClickY 在用户最后一次单击的位置创建一个文本输入区域。

“位置”选项中的数值决定着文本输入区域左上角的位置，“大小”选项中的数值决定着文本输入区域的宽度和高度，所以必须在交互之前为这两个选项赋值。

您可以试着使用下面的系统变量：

CursorX,cursory,DisplayX,DisplayY,PositionX,PositionY,WindowHeight,WindowLeft,WindowTop,WindowWidth

- “界限”文本框：使用该文本框来决定用户可以在文本框中输入的最大数目的字符数。如果用户输入了比最大数目还多的字符数，Authorware 就会忽略多余的字符。

文本输入区域的大小也限制着用户可以输入的文本数目。如果“Character Limit”文本框中为空，则用户可以输入尽可能多的字符，只要文本输入区域可以容纳。

- “自动登录在界限”复选框：选择该选项时，用户输入文本，当文本的字符数达到了限制的数目时，Authorware 会自动结束这个文本输入响应的输入，而不需要用户按结束输入文本响应的功能键。该选项只能在通过“界限”选择指定了最大数目的字符数后才可以使

1. “交互作用”标签页

该标签页用来设置交互信息，如图 8-44 所示。

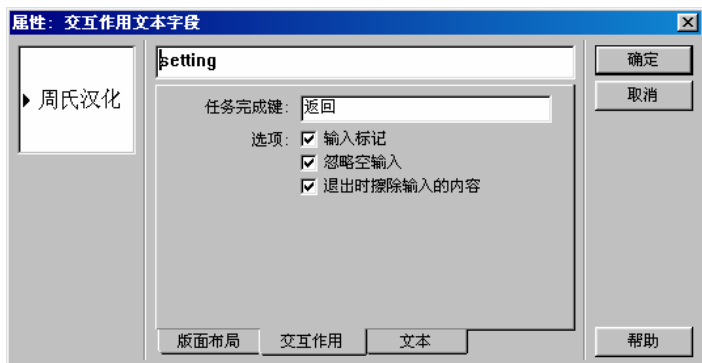


图 8-44 【交互】图标文本域属性对话框的“交互作用”标签页

- “任务完成快捷键”文本框：该文本框是用来指定哪一个键是用户结束文本输入的键。可以使用“|”来指定多个结束键，例如，在该文本框中输入“Enter|Tab”来指定 Enter 键和 Tab 键都可以作为文本输入的结束键。在 Authorware 中，默认的结束键为 Enter 键。

- “选项”选项：

“输入标记”复选框：该复选框是用来决定在指示用户输入文本的地方是否显示一个小三角形。当选择该复选框时，文本输入区域的左边会自动显示出一个标记——黑色小三角形（不管是否设置了该选项，在需要文本输入的地方总会显示一个闪烁的鼠标指针，来指示用户在该处输入文本）。系统默认是选择该选项，如图 8-45 所示。



(a) 选择“输入标记”复选框后显示出指示标记 (b) 没有选择该复选框时不显示指示标记

图 8-45 选择与不选择“输入标记”复选框的情形

“忽略空输入”复选框：选择该选项时，如果用户输入的文本没有匹配任何匹配文本，Authorware 会忽略输入的文本而要求用户重新输入文本。系统默认是选择该选项。

“退出时擦除输入的内容”复选框：选择该选项，当 Authorware 退出该【交互】图标时会擦除用户输入的文本。如果不选择该选项，用户输入的文本将会一直保留在屏幕上，除非使用【擦除】图标将其擦除。

2. “文本”标签页

使用“文本”标签页中的选项来指定用户输入的文本外形，包括文本的字体、字号、以及字型。也可以在该标签页中为文本域选择文本颜色和背景色。可以为文本对象选择一种显示模式，如图 8-46 所示。



图 8-46 【交互】图标文本域属性对话框的“文本”标签页

- “字体”下拉框：该下拉框中是系统提供的各种字体，可以从该下拉框中选择一个字体来指定用户输入的文本字体。系统默认字体为“System（系统字体）”。
在该下拉框中选择一些特殊的字体，如：Marlett 字体，可以将用户输入的文本显示为一些特殊的字符，来作为密码输入时用。
- “大小”下拉框和文本框选项：该下拉框是用来选择字号的。如果下拉框中没有所要的字号，则选择“其他”选项，此时后面的文本框被激活，在后面的文本框中输入合适的字号即可。系统默认字体的大小为“9”。
- “风格”选项：该选项是用来定义字体风格的。它包括三个选项：
“粗体”复选框：选择该选项，字体粗体显示，如图所示。
“斜体”复选框：选择该选项，字体斜体显示，如图所示。
“下划线”复选框：选择该选项，字体下划线显示，如图 8-47 所示。



图 8-47 定义字体风格

系统默认情况下，是不选择这三项，字体正常显示。我们可以组合使用这三项，以达到不同的效果。

- “颜色”选项：该选项包括两个选项：
“文本”按钮：该按钮是用来改变字体的颜色的。单击该按钮，可以打开“颜色”对话框，如图 8-48 所示，在该对话框中选择字色。
“背景”按钮：该按钮是用来选择文本域的背景色的。单击该按钮，也会打开颜色对话框，然后在该对话框中选择背景色即可。
- “模式”下拉框：该下拉框是用来定义输入文本的显示模式。它包括四个选项：不透明，透明，反转和擦除。

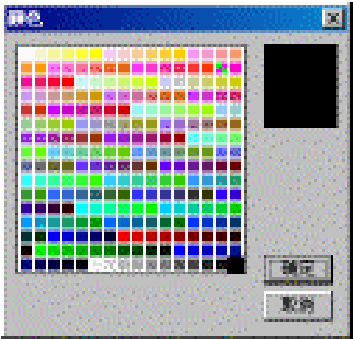


图 8-48 颜色对话框

则四个选项的含义我们已经在第3章【显示】图标中讲到，所以就不再赘述。

8.8.4 文本输入规则

在响应类型属性对话框的“模式”文本框中输入不同的文本来决定用户的输入。“模式”文本框中的文本称为匹配文本。

1. 对任何输入文本都作出反应

可以使用通配符“*”，输入“*”，告诉 Authorware 要接受用户输入的任何文本——一个字母、一个数字、一个单词、一个短语，甚至一个段落——都可以得到匹配。

如果只是收集一些资料，如一个用户名，可以使用通配符“*”，Authorware 会自动将用户输入的任何信息存储在系统变量 Entry 中。然后将存储在变量 Entry 中的文本传递给一个变量，这样就可以按照自己的方式处理用户信息。如果希望跟踪信息，由于 Authorware 提供的系统变量经常会改变，所以最好创建一个自定义变量。自定义变量会反映出用户赋给它的值。该值只有用户明显地改变它才会发生改变。例如，为了跟踪 Authorware 通过一个决策结构的次数，可以使用一个变量 Counter，然后根据表达式 Counter := Counter + 1 来计算 Authorware 每通过一次该决策结构，Counter 的值就加 1。自定义变量名可以跟在图标名和@标记后，表示该变量是该图标具有的，称为自定义图标变量。可以使用它们来查找一个文件中指定图标的自定义变量的值。

如果希望得到特定的输入信息，可以使用一个带有通配符“*”的响应——全响应——来接受用户输入的任何不正当响应。确定要将全响应放在文本输入响应的最右边，这样，Authorware 会首先检查是否匹配左边的文本输入响应，如果在左边得到响应，就执行该响应分支；否则就会匹配全响应，在全响应中可以输入一些提示信息，如告诉用户输入的文本错误等，然后重新返回到原来位置，让用户重新输入。

2. 只响应一个确定的匹配文本

可以在“模式”文本框中输入希望用户准确匹配的匹配文本。需要注意的是，如果匹配文本中包含有“*”号或“?”号，必须使用一个反斜杠“\”，输入格式为“*”和“\?”，此时就会只匹配“*”号或“?”号，Authorware 就不会将这两个符号当作通配符了。

3. 匹配等价的单词或短语

可以使用操作符“|或)”来定义等价的单词或短语。例如，如果希望接受一个问题的两个正确答案“Macromedia”和“Authorware”中的一个，用户输入其中一个正确答案就可以触发响应，可以输入“Macromedia|Authorware”即可。

8.8.5 特殊符号——通配符和其他符号的使用

在响应类型属性对话框的“模式”文本框中可以使用一些特定的字符或变量来建立不同的匹配文本。

- 使用“|或)”来分隔等价的响应。如果用户输入文本的文本框可以达到 400 个字符长，而匹配文本包括多个等价文本，可以使用“|”符号将等价文本分隔开。例如，匹配文本为“Macromedia|Authorware”，则用户输入“Macromedia”或“Authorware”都可以得到响应。
- 当用户输入了指定的匹配文本，系统不一定会作出响应，除非当系统变量 Tries 的值（变量 Tries 记录了用户输入文本的当前次数）与指定的数值相同时才会触发响应。如果希望用户尝试多少次才能匹配该响应，可以使用“#”符号，后跟一个数字作为匹配文本的开头。例如，在“模式”文本框中输入“#3Authorware”，则只有当用户第三次输入的文本为“Authorware”时，才会得到响应，而不管前面的输入。
- 为了让用户输入的文本可以匹配多个条件，混合使用“|”和“#”。例如，在“模式”文本框中输入“Mecro|media|#1 确定|#2Word|#3Authorware”，则无论何时输入“Mecro”或“media”



都可以得到匹配；或者当 Tries=1 时，即第一次输入的文本为“确定”，或 Tries=2 时输入的文本为“Word”，或 Tries=3 时输入的文本为“Authorware”都可以得到匹配。

- 可以使用两个连字符“-”(-)来为匹配文本添加注释。注释文字必须在匹配文本的后面。例如，在“模式”文本框中输入“Authorware—用户密码”，则 Authorware 会在检查匹配时忽略注释“用户密码”。
- 可以使用通配符“*”和“?”来考虑错误拼写和单词前缀或单词尾部的选择。通配符“*”可以替换一个单词中一个或多个字符，甚至可以替换整个单词。通配符“?”可以替换一个单词中的一个不可缺少的字符。通过使用一个反斜杠“\”，来输入“*”号和“?”号。

8.8.6 文本输入响应实例——用户登录程序

用户登录程序的设计是多媒体教学、培训类软件经常遇到的一个问题，我们这个实例只是比较简单是使用了文本输入响应，如图 8-49 所示，而真正的用户登录程序需要综合使用 Authorware 的知识，其中包括读写外部文件、用户密码判断、使用各种交互等等。

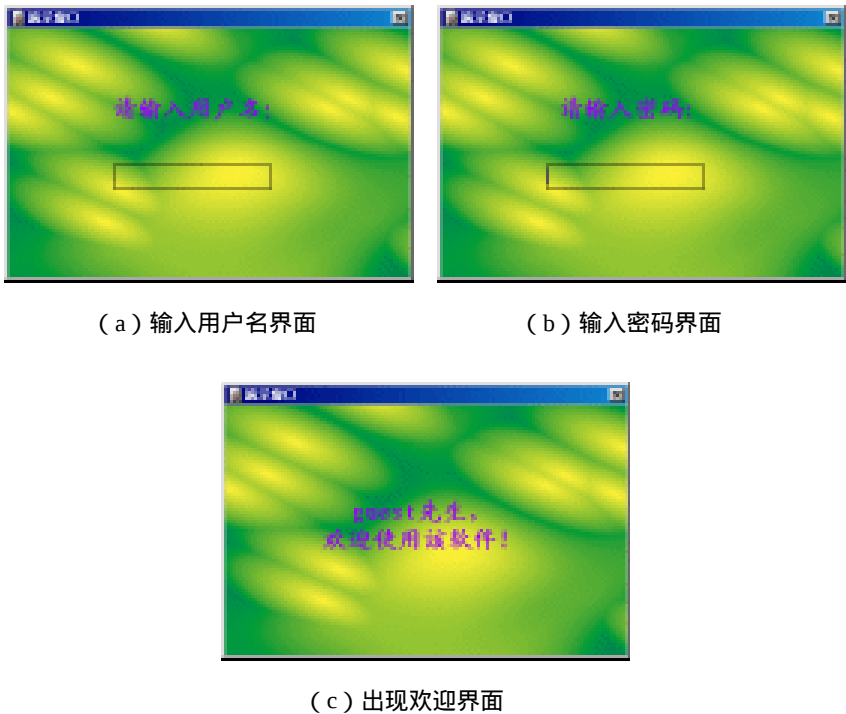


图 8-49 用户登录程序

操作步骤为：

- (1) 新建一个文件，命名为“用户登录程序.a6p”。
- (2) 选择菜单[修改][文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (4) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：
ResizeWindow(400, 250)

User_Name:= ""

Password:= ""

(5) 确定新建这两个变量 User_Name 和 Password, 并将其初始值设置为 “ ”。

(6) 选择菜单[文本]→[定义风格...]命令, 打开定义文本风格对话框。

(7) 单击定义文本风格对话框中的“添加”按钮, 增加一种新的文本风格, 将该文本风格更名为“提示文字”, 选择字体为楷体_GB2312、字号为 18, 粗体, 斜体、字色为紫色, 然后单击“修改”按钮确定修改。

(8) 单击对话框中的“完成”按钮, 完成定义文本风格。

(9) 拖动一个【显示】图标到流程线上, 将其命名为“背景”。

(10) 双击该【显示】图标, 打开展示窗口, 从外部导入一幅图形, 如图 8-65 中所示。

(11) 拖动一个【显示】图标到流程线上, 将其命名为“用户名提示信息”。

(12) 在该【显示】图标中输入文本“请输入用户名:”, 文本风格为“提示文字”, 显示模式为“透明”, 并在该文本下方绘制一个矩形。

(13) 拖动一个【交互】图标到流程线上, 将其命名为“登录用户名”。

(14) 拖动一个【计算】图标到该【交互】图标的右侧, 选择文本输入响应, 将其命名为“*”, 并将其响应的分支结构设置为“退出交互”。

(15) 在该【计算】图标中输入语句“User_Name:=EntryText”, 其中, EntryText 是一个系统变量, 用来记录用户输入的文本。

(16) 按住<Shift>键双击“登录用户名”【交互】图标, 打开展示窗口, 将文本域移动到绘制的矩形中。

(17) 双击文本域, 打开文本域属性对话框, 取消“交互作用”标签页中的“输入标记”复选框, 在“文本”标签页中设置字体为宋体, 字号为 12, 粗体, 字色为蓝色, 显示方式为“透明”方式, 然后单击“确定”按钮确定。

(18) 拖动一个【擦除】图标, 将其命名为“擦除用户名提示信息”, 设置其擦除“用户名提示信息”【显示】图标中的内容。

(19) 拖动一个【显示】图标, 命名为“密码提示信息”, 输入文本“请输入密码:”, 文本风格为“提示文字”, 并在该文本下方绘制一个矩形。

(20) 再拖动一个【交互】图标到流程线上, 将其命名为“输入密码”。

(21) 拖动一个【计算】图标到该【交互】图标的右侧, 选择文本输入响应, 将其命名为“*”, 并将其响应的分支结构设置为“退出交互”。

(22) 在该【计算】图标中输入语句“Password:=EntryText”。

(23) 按住<Shift>键双击“登录用户名”【交互】图标, 打开展示窗口, 将文本域移动到绘制的矩形中。

(24) 双击文本域, 打开文本域属性对话框, 取消“交互作用”标签页中的“输入标记”复选框, 在“文本”标签页中设置字体为宋体, 字号为 12, 粗体, 字色为蓝色, 显示方式为“透明”方式, 然后单击“确定”按钮确定。

(25) 拖动一个【擦除】图标, 将其命名为“擦除密码提示信息”, 设置其擦除“密码提示信息”【显示】图标中的内容。

(26) 拖动一个【显示】图标, 命名为“欢迎文字”, 输入文本“{User_Name}先生, 欢迎使用该软件!”, 文本风格为“提示文字”。

(27) 拖动一个【等待】图标到流程线上, 将其命名为“等待”, 在其属性对话框中选中“鼠标单击”复选框, 在“时间限制”文本框中输入“3”, 取消“显示按钮”复选框, 单击“确定”按钮确定。

- (28) 拖动一个【计算】图标到流程线上，将其命名为“退出”，在该图标中输入语句“Quit()”。
- (29) 此时程序已经设计完成，程序流程如图 8-50 所示。

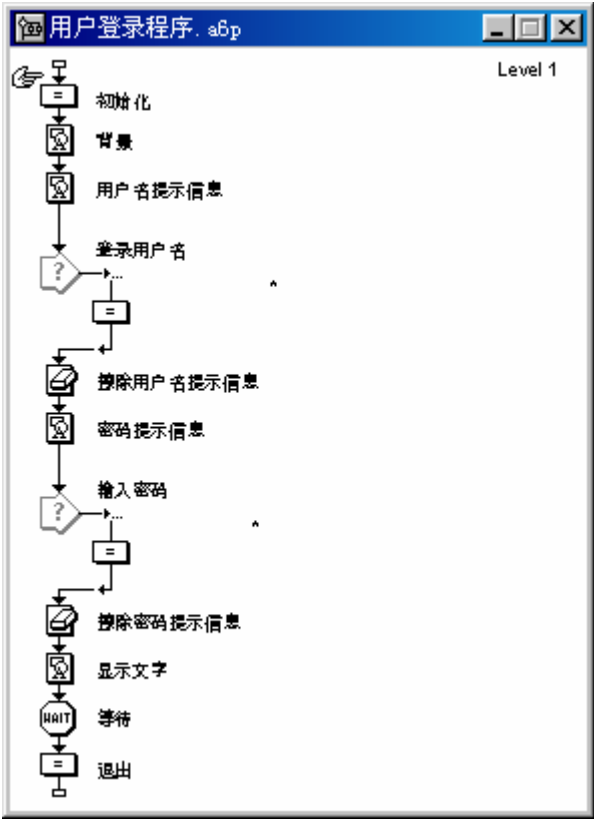


图 8-50 设计好的程序流程

8.9 按键响应（按键）

有时，让用户仅仅按下一个键就让程序做出响应是很简单的响应方式。例如，假如在屏幕上建立几个文本域，从一个文本域转换到另一个文本域的简单方法就是按一下 Tab 键，此时就需要使用到按键响应。

可以建立一个按键响应，让其对单个按键或组合键，如<Ctrl>+A，作出响应，也可以建立一个单个按键响应类型标志来对多个不同的键或不同的组合键作出响应（尽管一次只能对一个键或一个组合键作出响应），而且也可以创建一个全响应，使之能对用户按下的任何键都可以作出响应。

需要注意的是，不要使用按键响应来为其他响应类型创建响应。例如，不要使用按键响应来为一个按钮、热区、热对象、或下拉菜单项创建一个快捷键。因为上面的每一个响应在对话框中都有一个文本域是来指定快捷键的。也不要用户在需要输入文本的文本域中创建一个按键响应，因为这些可以使用文本输入响应来做。

8.9.1 选择“按键”选项后的响应类型属性对话框

双击目标图标对应的响应类型图标，打开响应类型属性对话框，从“类型”下拉框中选择“按键”

选项，属性对话框变为如图 8-51 所示。

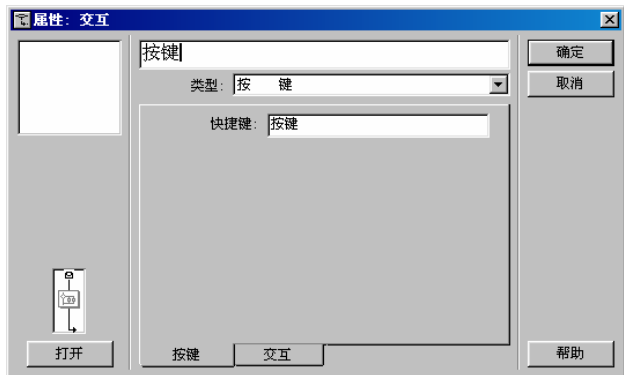


图 8-51 选择“按键”选项后的响应类型属性对话框

在“按键”标签页中只有一个选项：

- “快捷键”文本框：该文本框是用来定义匹配的键。

8.9.2 按键响应规则

在响应类型属性对话框的“快捷键”文本框中输入不同的键名来决定用户的按键。“快捷键”文本框中的键名称为匹配键。

1. 创建单个键或单个组合键

在“快捷键”文本框中直接输入一个字母、数字、或键盘上的一个按键符号。Authorware 会自动分辨大写字母和小写字母，例如，h（按下 H 键）和 H（同时按下<Shift>键和 H 键）是不同的。如果希望 Authorware 能够同时匹配按下<Shift>键和没有按下<Shift>键的字母，可以同时“快捷键”文本框中输入大写字母和小写字母，如“h|H”。

为了让 Authorware 能够匹配？键，必须在“快捷键”文本框中输入“\?”，而不是直接输入“？”，因为 Authorware 会将“？”符号当做通配符。

对于一些控制键，如 Tab 键、Return 键、或箭头键等，只需要在“快捷键”文本框中直接输入键名即可。Authorware 所能认识的完整的键名，请参考后面的表 8-5。

如果希望用 Alt 键、Ctrl 键、或 Shift 键和一个字母键作为一个组合键，则需要“快捷键”文本框中输入“Alt”、“Ctrl”、或“Shift”，后面直接跟上作为组合键的字母即可。如，如果希望使用<Ctrl>+a 来触发响应，可以在“快捷键”文本框中输入“Ctrl a”，Authorware 会认为是 Ctrl 键和 a 键组成的组合键。

2. 创建多个单个键或组合键

在“快捷键”文本框中直接输入所有的单个键或组合键，单个键或组合键之间使用“|”操作符隔开，Authorware 会自动识别。例如，如果希望同时创建按键 h 和按键 H，则在“快捷键”文本框中输入“h|H”；如果希望 Authorware 能同时识别 PageDown（向后翻页）键和 DownArrow（向下箭头）键，则在“快捷键”文本框中输入“PageDown|DownArrow”。

3. 创建能够识别任何键或任何组合键

由于？号是通配符，所以在“快捷键”文本框中直接输入“？”，Authorware 会响应任何一个键。

由于使用？号可以响应任何响应，所以可以使用它来建立一个全响应，让它处理不正当响应。同样，要将全响应放在其他按键响应的最右边，这样 Authorware 会首先去匹配其他按键响应，如果匹配到其中一个响应，就会执行该响应对应的分支内容；如果没有匹配到，则会执行全响应，在全响应分支中建立

一些提示信息，显示该提示信息后，重新回到原处让用户重新进行选择。

8.9.3 按键响应实例——完善用户登录程序

在讲解文本输入响应时，我们列举了一个实例——用户登录程序，在该程序中，我们使用了一种特殊的字体来作为密码显示。虽然这样做可以实现保密的作用，但是并不是我们习惯的方式，我们依然习惯输入一个字符，屏幕上以星号*显示，如图 8-52 所示。现在我们就使用按键响应来实现这一功能。

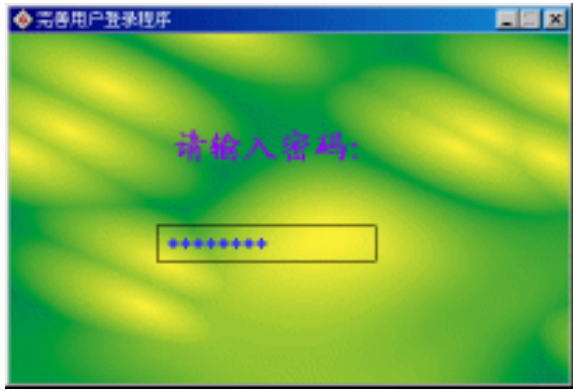


图 8-52 密码输入

操作步骤为：

- (1) 将“用户登录程序.a6p”另存为“完善用户登录程序.a6p”。
- (2) 将“输入密码”【交互】图标下的“*”【计算】图标的响应类型改为按键响应，将其更名为“？”，并且将其分支方式改为“重试”。
- (3) 将该【计算】图标中的语句改为如下语句：
 Password:=Password^Key
 -记录用户密码
 ScreenText:=ScreenText^"*"
 -在屏幕上再显示出一个*号
- (4) 在出现的新建变量对话框中，将变量 ScreenText 的初始值赋为空字符串，既在“初始值”文本框中输入“ ”。
- (5) 拖动一个【群组】图标到“输入密码”【交互】图标右侧，并选择按键响应，将其命名为“Enter”，并且将其分支方式改为“退出交互”。表示当用户按下回车键时退出该交互。
- (6) 再拖动一个【计算】图标到该【群组】图标的右侧，并将其命名为“BackspaceDelete”，将其分支方式改为“重试”。
- (7) 在该【计算】图标中输入如下语句：
 -当用户按下了 Backspace 键或 Delete 键时
 Password:=SubStr(Passwrd, 1, CharCount(Passwrd) - 1)
 -将用户密码减少一个字符
 ScreenText:=SubStr(ScreenText, 1, CharCount(ScreenText) - 1)
 -在屏幕上少显示出一个*号
- (8) 定义一种文本风格，命名“密码文字”，将其字体设置为宋体，字号为 12，粗体，字色为蓝色。

(9) 先双击“密码提示信息”【显示】图标，然后关闭。

(10) 按下<Shift>键并双击“输入密码”【交互】图标，打开展示窗口，用文字工具在窗口中输入“{ScreenText}”（显示星号*），文本风格为“密码文字”，并将该文本对象移动到“密码提示信息”【显示】图标的矩形中。

(11) 在“初始化”【计算】图标中添加一个语句“ScreenText:=“””。

(12) 此时程序已经设计完成，程序流程如图 8-53 所示。

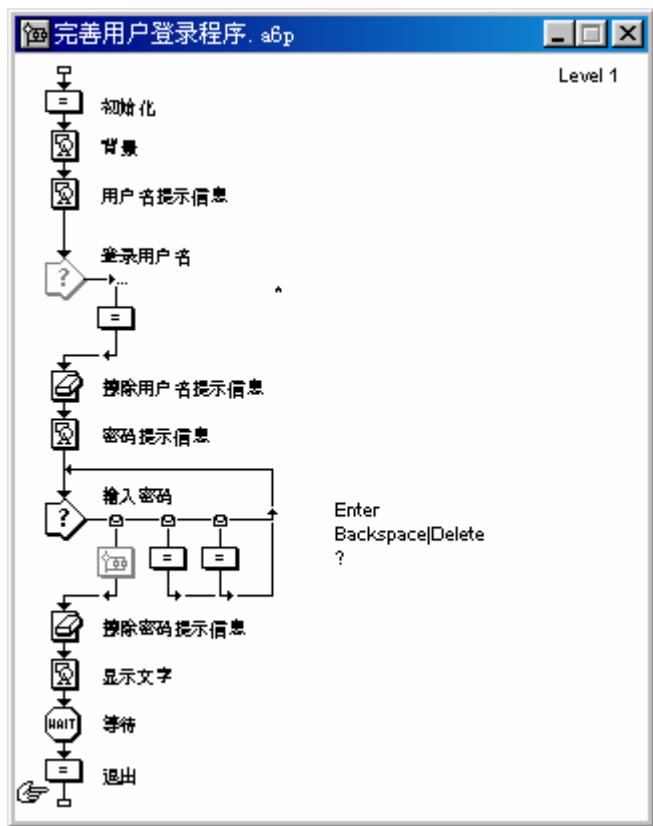


图 8-53 设计好的程序流程

其实，该程序还有许多需要改进的地方，如判断用户是否为存在用户，我们的程序只能算是新用户登录过程，但作为新用户登录也是不完整的，因为它需要输入两次密码，并且要验证两次密码是否相同。不过，相信通过这些实例的学习，您一定可以自己设计出一个完整的用户登录程序的。

8.10 限制次数响应（重试限制）

当建立一个交互，该交互只有一个正确答案，既不是让用户只有一次输入机会，也不是让用户一直输入，直到输入正确的答案，此时可以使用限制次数响应类型来限制用户的输入次数。

当每一次用户输入错误答案，也可以使用该响应类型给用户反馈信息，然后让他们继续尝试输入。例如，假如用户只有三次机会输入正确答案，此时就可以设置一个限制次数响应来显示信息，如当用户输入一次错误答案，就在屏幕上显示提示信息：“还有两次机会！”；而当再次输入错误信息后，可以显示信息：“这是您的最后一次机会！”。甚至更好的是，可以在提示信息中为用户提供一些反馈信息，

可以帮助用户想起正确答案。

限制次数响应一般情况下很少单独使用，它主要作为其他响应类型的辅助响应。当将限制次数响应类型标志放在交互流程中，让反馈流程与限制次数响应链接起来，同时也要将反馈流程与其他响应类型链接起来。例如，为了能够将链接了限制次数响应的反馈信息显示出来，必须将限制次数响应放在其他响应右边，如图 8-54 所示。

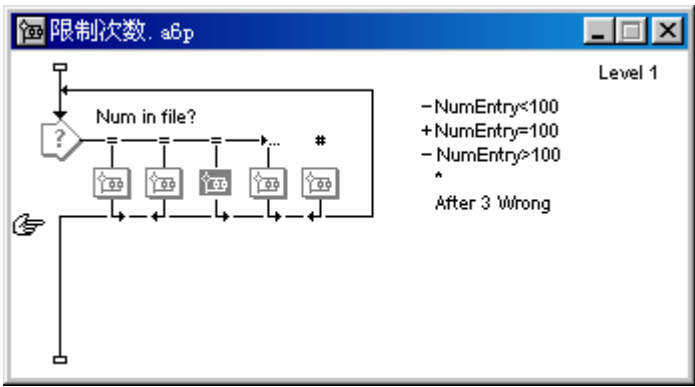


图 8-54 限制次数响应样例 1

如果希望将链接了限制次数响应的反馈信息用错误答案的反馈信息代替，必须将限制次数响应放在其他响应的左边，如图 8-55 所示。

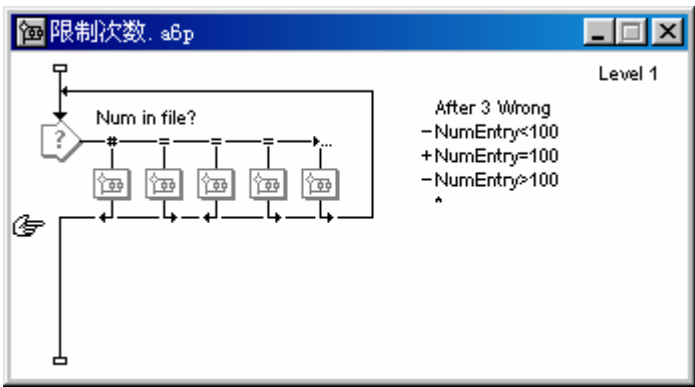


图 8-55 限制次数响应样例 2

Authorware 不但可以使用限制次数响应来跟踪用户输入的次数，它还可以使用系统变量 Tries 来记录用户输入的数值。

8.10.1 选择“重试限制”选项后的响应类型属性对话框

双击【交互】图标下挂图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“重试限制”选项，属性对话框变为如图 8-56 所示。

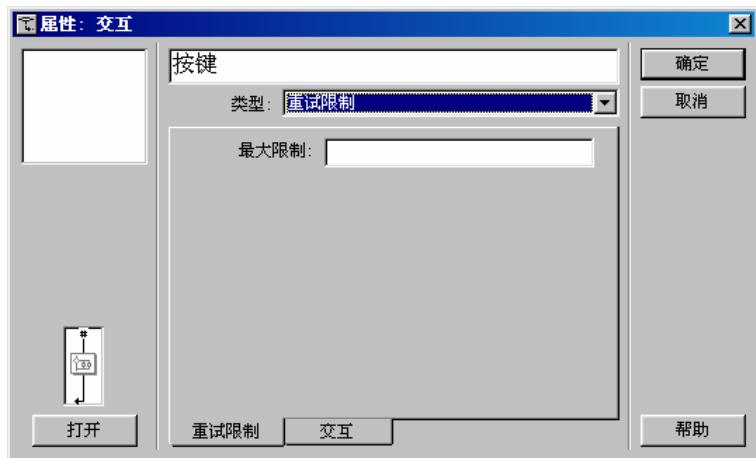


图 8-56 选择“重试限制”后的响应类型对话框

在“重试限制”标签页中只有一个选项。

- “最大限制”文本框：该文本框是用来输入用户最多输入数据的次数。

“交互”标签页与其他响应的“交互”标签页除了“永久”复选框和“激活条件”文本框是灰显的外，几乎完全相同，所以就不再一一赘述。

8.10.2 在交互中创建限制次数响应

1. 建立正确与错误答案

- (1) 拖动一个【交互】图标到设计窗口的流程线上，并将其命名。
- (2) 为每一个答案建立一个响应类型图标和目标图标。
- (3) 双击每一个响应类型图标，打开响应类型属性对话框，选择“交互”标签页，然后选择一种擦除方式。
- (4) 将每一个响应都设置成正确或错误响应。
- (5) 为每一个错误答案的响应设置成“重试”分支方式。

2. 为交互添加限制次数响应

- (1) 拖动一个图标到【交互】图标的右边。
- (2) 双击目标图标上面的响应类型图标，打开响应类型属性对话框。
- (3) 在对话框的“类型”下拉框中选择“重试限制”选项，并单击“确定”确定。
- (4) 为该目标图标命名，以描述其功能。
- (5) 在定义第一个按键响应之前按照上面的步骤建立其他按键响应，或者先定义第一个按键响应，然后再建立其他响应。

3. 放置限制次数响应在交互流程中的位置

限制次数响应在交互流程中的位置决定着限制次数响应和其他响应的反馈信息。

例如，为了显示错误答案的反馈信息以及限制次数响应的反馈信息，需要将限制次数响应放置在其他响应的右边；如果希望用限制次数响应的反馈信息来代替错误答案的反馈信息，则需要将限制次数响应放置在其他响应的左边。

4. 设置用户输入次数

- (1) 双击限制次数响应图标，打开响应类型属性对话框。
- (2) 在“重试限制”标签页的“最大限制”文本框中输入希望用户输入的次数。当用户输入的系

数匹配了允许最多输入的次数，链接到该限制次数响应的反馈信息就会出现。

(3) 单击“交互”标签页，从中选择合适的擦除方式和分支方式，然后单击“确定”按钮确定。

5. 在目标图标中建立反馈信息

双击打开目标图标，然后在目标图标中建立反馈信息。

8.10.3 限制次数响应实例——猜数游戏

该实例是一个猜数小游戏，程序自动产生一个从 1 到 8-0 的整数，用户可以有 6 次机会猜测，如果用户在 6 次机会中猜中了该数，程序给出信息，然后退出。在猜数的过程中，程序会给出提示信息，告诉用户输入的数字与目标数字之间的关系，如图 8-57 所示。

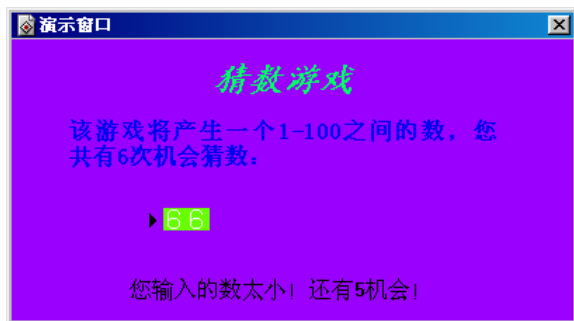


图 8-57 猜数游戏

操作步骤为：

- (1) 新建一个文件，命名为“猜数游戏.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“标题文字”，选择字体为楷体_GB2312，字号为 18，粗体，斜体，字色为绿色，然后单击“修改”按钮确定修改。
- (5) 再增加一种文本风格，命名为“提示文字”，字体为宋体，字号为 12，粗体，字色为蓝色。
- (6) 单击对话框中的“完成”按钮，完成定义文本风格。
- (7) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (8) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：
`ResizeWindow(400, 200)`
- (9) 拖动一个【显示】图标到流程线上，将其命名为“提示文字”。
- (10) 在【显示】图标中输入两个文本，一个为“猜数游戏”，文本风格为“标题文字”；另一个为“该游戏将产生一个 1-8-0 之间的数，您共有 6 次机会猜数：”，文本风格为“提示文字”。
- (11) 调整这两个文本对象。
- (12) 拖动一个【计算】图标到流程线上，命名为“初始化”，在该图标中输入如下语句：
`Num:=Random(1, 8-0, 1)`
 -产生随机数，记录在变量 Num 中
`Times:=0`

-记录用户匹配时已经尝试的次数

(13) 新建这两个变量，并将它们的初始值都设置为 0。

(14) 拖动一个【交互】图标到流程线上，并命名为“猜数”。

(15) 拖动一个【群组】图标到【交互】图标的右侧，选择条件响应类型，将其命名为“Num=NumEntry”，并将其分支方式改变为“退出交互”。

(16) 拖动一个【计算】图标到【交互】图标的右侧，选择文本输入响应，并将其命名为“*”，可以接受任何字符的输入，然后将其分支方式改变为“重试”。

(17) 在该【计算】图标中输入如下语句：

```
if NumEntry=0 then Screen 文本:=""  
if NumEntry<Num then  
    Screen 文本:="您输入的数太小！还有"^String(6 - TimesMatched)^"机会！"  
else if NumEntry>Num then  
    Screen 文本:="您输入的数太大！还有"^String(6 - TimesMatched)^"机会！"  
end if  
Times:=TimesMatched
```

(18) 新建变量 ScreenText，并将其初始值设置为空字符串。

(19) 双击【交互】图标，打开展示窗口，调整文本域，然后双击文本域，打开其属性对话框，在“文本”标签页中设置文本风格为字体为 System，字号为 8-，粗体，字色为黑色，显示方式为“透明”，然后单击“确定”按钮确定。

(20) 在展示窗口中输入文本“{ScreenText}”，文本风格为“默认”，对齐方式为中间对齐，然后将其移动到展示窗口的最下端，关闭展示窗口。

(21) 拖动一个【群组】图标到【交互】图标的最右侧，选择限制次数响应类型，将其命名为“6 次输入机会”，然后再将其分支方式改变为“退出交互”。

(22) 拖动一个【计算】图标到流程线上，将其命名为“显示结果”。

(23) 在该【计算】图标中输入如下语句：

```
if Times=0 then  
    ScreenText:="Wonderful!"  
-如果第 1 次就输入正确，显示 Wonderful！  
else if Times=1 then  
    ScreenText:="Great!"  
-如果第 2 次就输入正确，显示 Great！  
else if Times=2 then  
    ScreenText:=" Very Good!"  
- Very Good！  
else if Times=3 then  
    ScreenText:="确定!"  
-如果第 4 次就输入正确，显示确定！  
else if Times=4 then  
    ScreenText:="Oh! You're Luck!"  
-如果第 5 次就输入正确，显示 Oh! You're Luck！  
else if Times=5 then  
    ScreenText:="Amazing! You're very Luck!!!"
```

-如果第 5 次就输入正确，显示 Amazing! You're very Luck!!!
else
 ScreenText:="Sorry! You lost!"
-如果 6 次都没有输入正确，则显示 Sorry! You lost !
end if

- (24) 拖入一个【显示】图标，命名为“显示信息”，在该图标中输入文本“{ScreenText}”，文本风格为“默认”，对齐方式为中间对齐，然后将其移动到展示窗口的最下端。
- (25) 拖入一个【等待】图标，命名为“等待”，在其属性对话框中选中“鼠标单击”复选框和“按键”复选框，在“时间限制”文本框中输入“3”，并取消“显示按钮”复选框。
- (26) 拖入一个【计算】图标，命名为“退出”，在该图标中输入语句“Quit()”。
- (27) 此时程序已经设计完成，程序流程如图 8-58 所示。

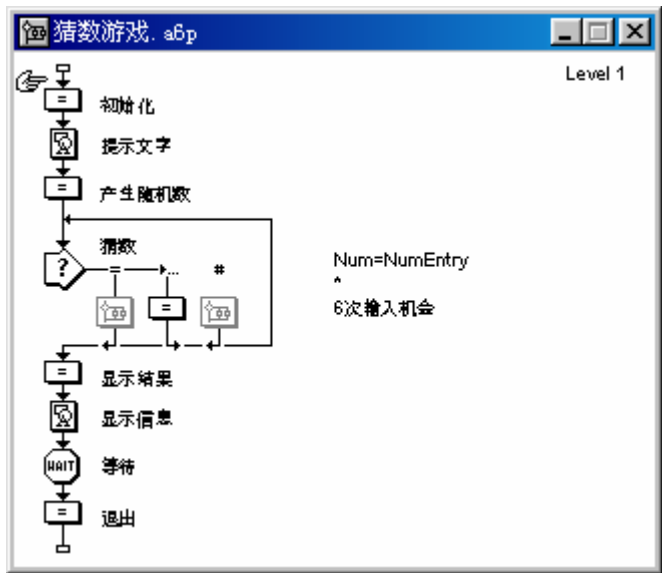


图 8-58 设计好的程序流程

8.11 限制时间响应（时间限制）

如果希望用户在一个限定时间里回答一个问题，可以使用限制时间响应。如果在一个特定长时间之后，用户没有作出任何响应，此时也可使用限制时间响应来警告用户，告诉他们时间不多了，请快速作出响应。

当在一个交互中使用了限制时间响应，Authorware 会使用系统变量 TimeRemaining 来记录剩余的时间，Authorware 也会将用户在一个交互中花去的时间记录在一个系统变量 TimeIn 交互作用中（即使在交互中没有使用限制时间响应）。

8.11.1 选择“时间限制”选项后的响应类型属性对话框

双击【交互】图标下挂图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择“时间限制”选项，属性对话框变为如图 8-59 所示。

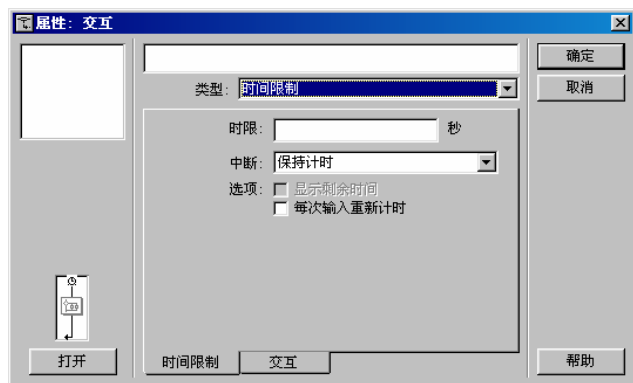


图 8-59 选择“时间限制”后的响应类型对话框

在“时间限制”标签页中只有几个选项。

- “时限”文本框：该文本框是用来输入该响应所限制的时间，单位是秒。可以在该文本框中输入数字、表达式、数值型变量、以及可以返回数值型函数等。

“交互”标签页与其他响应的“交互”标签页除了“永久”复选框和“激活条件”文本框是灰显的外，几乎完全相同，所以就不再一一赘述。

- “中断”下拉框：使用“中断”选项可以在交互过程中用户跳转到其他操作时关闭记时交互。如果在时间交互中，用户跳转到了一个永久交互中（如选择了全下拉菜单响应中一个命令），此时可以使用下面的选项来建立出现的情况。

“保持计时”选项：选择该选项时，Authorware 会在执行一个永久交互时继续记时。该选项为缺省选项。

“暂停，在返回时恢复计时”选项：选择该选项时，Authorware 会在跳转到永久交互时停止记时。当 Authorware 重新回到时间交互图标时，限制时间响应会继续从离开时的时间开始记时。

“暂停，在返回时重新开始计时”选项：选择该选项时，Authorware 会在跳转到永久交互时停止记时。当 Authorware 重新回到时间交互图标时，限制时间响应会重新开始记时，不管在跳转到永久交互响应前限制时间响应是否已经记时结束。

“暂停，如运行时重新开始计时”选项：该选项与“暂停，在返回时恢复计时”选项十分相似。选择该选项时，Authorware 在跳转到永久交互时停止记时。只有在跳转到永久交互响应前限制时间响应还没有记时结束，Authorware 才会重新回到时间交互图标时，限制时间响应重新开始记时。

- “选项”选项：该选项下面有两个复选框子选项：

“显示剩余时间”复选框：如果选择该选项，会在屏幕上显示出一个时钟，时钟上显示出剩余的时间，如图 8-60 所示。



(a) 属性对话框中显示出的小时钟



(b) 执行程序时显示的小小时钟

图 8-60 使用小时钟

可以为每一个限制时间响应建立一个独立的小时钟。双击【交互】图标，打开其展示窗口，在展示



窗口中将各个小时钟放置在合适的位置。双击一个小时钟，可以打开该小时钟对应的限制时间响应属性对话框。

需要注意的是，当一个【交互】图标中有多个小时钟，如果各个小时钟中（其实就是“时间限制”文本框中定义的限制时间）的时间不同，将会按照小时钟中最小的时间为准。

该选项开始时灰显，只有在“时间限制”文本框中输入限制时间后才会被激活。

“每次输入重新计时”选项：选择该选项时，当用户在限制时间里匹配了一个响应，重新进行计时，同时系统变量也重新赋值。

如果在一个交互中有多个限制时间响应，除了最后个限制时间响应外，其他响应的分支方式都要设置为“重试”方式，这样用户在得到限制时间响应的反馈信息后可以继续进行交互。如果用户在一个文本输入响应的文本域中输入了一部分文本，此时一个限制时间响应的限制时间到了，用户可以通过单击文本域尾部而继续进行交互响应。

8.11.2 限制时间响应实例——抢答题

下面我们通过一个实例——抢答题，来说明如何使用限制时间响应。

在该实例中，用户需要在限定的时间内对问题做出反应（按下正确答案的键），根据用户的不同反应来显示不同的效果，如图 8-61 所示。

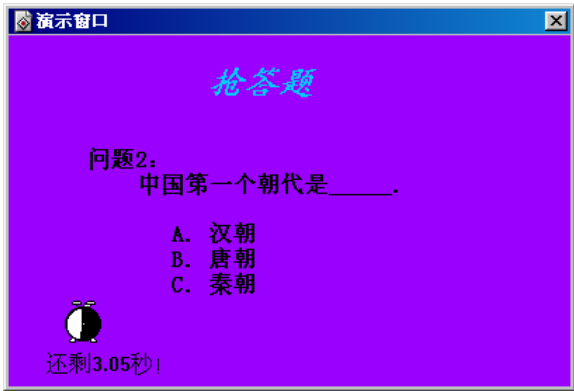


图 8-61 抢答题

操作步骤为：

- (1) 新建一个文件，命名为“抢答题.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“标题文字”，选择字体为楷体_GB2312，字号为 18，粗体，斜体、字色为浅兰色，然后单击“修改”按钮确定修改。
- (5) 再增加一种文本风格，命名为“正文文字”，字体为宋体，字号为 12，粗体，字色为黑色。
- (6) 单击对话框中的“完成”按钮，完成定义文本风格。
- (7) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (8) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：

ResizeWindow(400, 250)

CorrectNum:=0

-设置学生答对题目的个数

(9) 新建这个 CorrectNum 变量，并将其初始值赋为 0。

(10) 拖动一个【显示】图标到流程线上，将其命名为“标题”，并在该图标中输入文本“抢答题”，文本风格为“标题文字”。

(11) 拖动一个【显示】图标到流程线上，将其命名为“提示文字”。

(12) 在【显示】图标中输入两个文本，一个为“问题 1：请问中国最大的岛屿是_____。”，另一个为“A. 海南岛；B. 台湾岛；C. 崇明岛”，文本风格都为“提示文字”

(13) 拖入一个【交互】图标，将其命名为“抢答题 1”。

(14) 在【交互】图标右侧拖入四个【群组】图标，从左到右依次命名为“控制时间 1”、“A|a”、“B|b”和“C|c”。其中，“控制时间 1”【群组】图标设置为限制时间响应，其他三个为按钮响应。它们的分支方式都为“退出交互”。

(15) 由于 B 答案是正确的，所以双击“B|b”【群组】图标，打开其二级流程窗口，在该窗口中拖入一个【计算】图标，命名为“答对问题”，并在该图标中输入语句“CorrectNum:=CorrectNum+1”。

(16) 双击“控制时间 1”【群组】图标上面的响应类型图标，打开其属性对话框。在“时间限制”文本框中输入“5”，然后选中“显示剩余时间”复选框，然后单击“确定”按钮确定。

(17) 双击“抢答题 1”【交互】图标，打开展示窗口，将显示的小时钟放置在屏幕的左下角，并在小时钟下面输入一个文本“还剩{TimeRemaining@“控制时间 1”}秒！”，其中 TimeRemaining 是一个系统变量，用来记录剩余的时间，如图 8-62 所示。

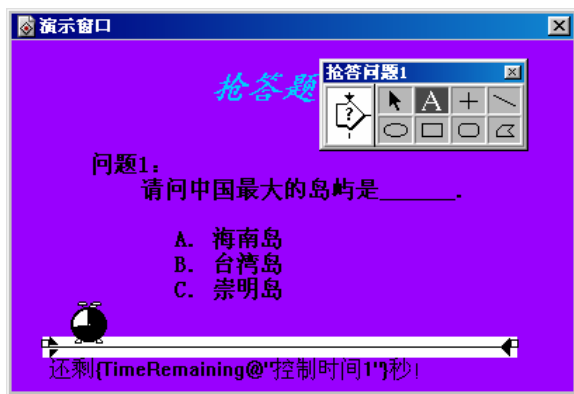


图 8-62 调整小时钟和输入文本

(18) 按住<Ctrl>键并双击“抢答题 1”【交互】图标，打开其属性对话框，在“显示”标签页中选中“更新变量”复选框，然后单击“确定”按钮确定。

(19) 拖入一个【擦除】图标，命名为“擦除问题 1”，选择擦除“问题 1”【显示】图标中的内容。

(20) 此时，第一个抢答题设计完成。

(21) 按照上面的步骤设计第二个抢答题，步骤略。

(22) 在擦除第二个问题后，拖入一个【显示】图标来显示答问题的结果，然后拖入一个等待图标，让结果显示一段时间。

(23) 在最后拖入一个【计算】图标，命名为“退出”，并在该图标中输入语句“Quit()”。

(24) 此时程序设计完成，程序流程如图 8-63 所示。

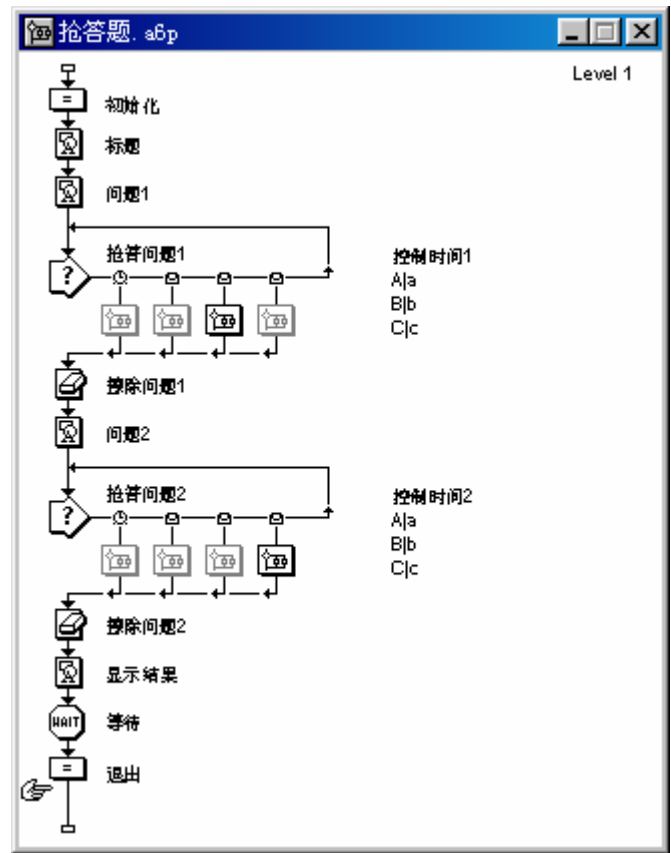


图 8-63 设计好的程序流程

8.12 事件响应（事件）

Sprite Xtras, Scripting Xtras 和 ActiveX 控件都可以发出事件。一个事件就是一个事情的发生。一些事件是用户所做操作的结果，例如，Microsoft 的日历 ActiveX 控件（Calendar ActiveX Control）可以跟踪用户在日历上的单击位置，并且为每一次的单击发出一个事件。其他事件是由 Sprite、Script 或者控件本身发出的，例如，Microsoft Web 浏览器控件（Microsoft Web Browser Control）会在一个信息下载完成时，而另一个下载信息正好开始的时候发出一个事件。

这些可以产生事件的 Sprite、Script 或者控件一般被称为发生器。您可以使用一个事件响应类型来跟踪用户对一个发生器的交互。当发生事件时，可以象建立的按钮、热区、或者文本域一样做出响应。您也可以建立一个交互来响应发生器自身产生的事件。

8.12.1 选择“事件”选项后的响应类型属性对话框

事件响应属性对话框列举了在 Authorware 应用程序中使用的所有发生器（Sender），以及这些发生器所产生的所有事件。

双击【交互】图标下挂图标的响应类型图标，打开响应类型属性对话框，在“类型”下拉框中选择

“事件”选项，属性对话框变为如图 8-64 所示。

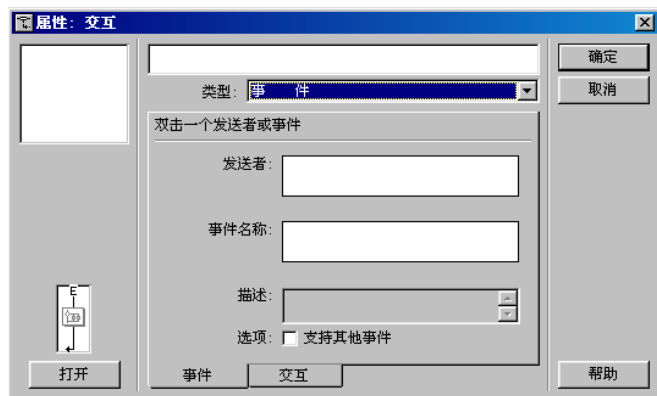


图 8-64 选择“事件”后的响应类型对话框

由于该对话框的“交互”标签页与前面响应类型的对应标签页完全相同，所以在此就不再一一赘述。对于“事件”标签页，其各个选项个含义如下：

- “双击一个发送者或事件”：提示信息，告诉用户在下面的发生器列表框和事件列表框中双击发生器或者一个事件来打开它。
- “发送者”列表框：发生器列表框。在该列表框中双击一个发生器打开它，此时该发生器前面会出现一个 X 号，表示为当前选中的事件发生器。
- “事件名称”列表框：事件列表框。当在“发送者”列表框中打开一个发生器后，该列表框中列出该发生器所有的事件。在该列表框中双击一个事件即可打开它，此时该事件前面会出现一个 X 号，表示为当前选中的发生器事件。
- “描述”提示框：当打开一个事件后，该提示框中显示出该事件的描述信息，一般为函数形式。
- “选项”选项：设置事件选项。它包括有一个子选项：

“支持其他事件”复选框：选中该复选框时，当 Authorware 接收到某一个指定事件时，将会挂起其他事件而不去响应它们。

当我们在“发送者”列表框中打开一个发生器，然后在“事件名称”列表框中打开其一个事件，此时响应类型属性对话框如图 8-65 所示。

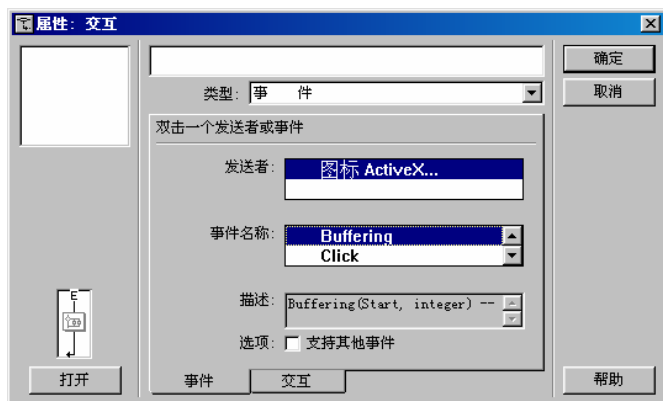


图 8-65 打开一个发生器及其事件



8.12.2 在交互中创建事件响应

1. 创建 Xtra 和交互

- (1) 创建一个希望发送事件的 Xtra。
- (2) 在流程线上拖入一个【交互】图标，并将其命名。
- (3) 在该【交互】图标右侧拖入一个图标。
- (4) 将该图标的响应类型设置成事件响应。

2. 创建事件

- (1) 双击事件响应类型图标，打开其属性对话框，并选择“事件”标签页。
- (2) 在“发送者”列表框中选择一个事件发生器。

在“发送者”列表框中显示出所有可以发出事件的对象。

单击一个事件发生器，使之高亮显示，并且显示出它的可以发生的事件。

双击一个事件发生器，选中它，它的前面会出现一个 X 号，表示它已经被选中了。

按下<Alt>键并在“发送者”列表框双击，可以选中所有显示的事件发生器。

双击一个选中的事件发生器，可以取消选中状态。取消选中一个事件发生器，自动取消了其发出的所有事件。

按下<Alt>键并在“发送者”列表框双击，可以取消所有选中的事件发生器。

- (3) 在“事件名称”列表框中选择一个事件。

在“事件名称”列表框中显示了 Sprite Xtra 对象和 Script Xtra 对象可以发出的所有事件，您可以选择任意个事件或全部选中。

单击一个事件，使之高亮显示，此时“描述”提示框显示出该事件的描述信息。

双击一个事件选中它，一个 X 号会出现在选中的事件前面。

按下<Alt>键并在“事件名称”列表框双击，可以选中所有显示的事件。

双击一个选中的事件，可以取消选中状态。

按下<Alt>键并在“事件名称”列表框双击，可以取消所有选中的事件。

选中一个事件，会自动选中了发生该事件的事件发生器。取消一个事件发生器，会自动取消所有该发生器发出的事件。

- (4) 在对话框的“交互”标签页中设置其其他响应选项。
- (5) 单击“确定”按钮确定即可。

8.12.3 事件响应实例——日历

现在我们使用一个实例来说明如何使用事件响应。

在该实例中，当用户单击屏幕上的某一个日期，下面就会显示出您单击的日期，如图 8-66 所示。

操作步骤为：

- (1) 新建一个文件，命名为“日历.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格...]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“显示信息”，选择字体为楷体_GB2312，字号为 12，粗体，斜体、字色为浅兰色，然后单击“修改”按钮确定修改。

∞

【

】



图 8-66 日历

(5) 单击对话框中的“完成”按钮，完成定义文本风格。

(6) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。

(7) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：

ResizeWindow(400, 300)

(8) 选择菜单[插入]→控件→[ActiveX...]命令，打开“选择 ActiveX 控件”对话框，如图 8-67 所示。

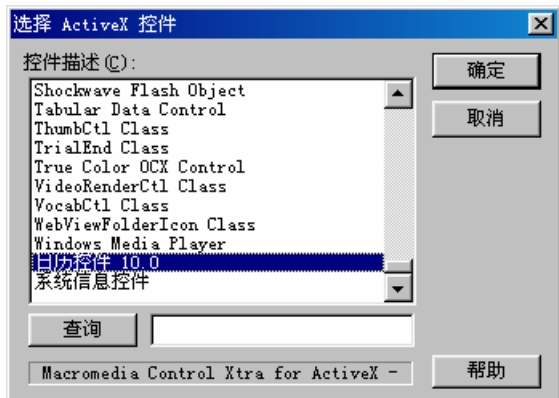


图 8-67 “选择 ActiveX 控件”对话框

(9) 在该对话框中选择“日历控件 10.0”，然后单击“确定”按钮导入。

(10) 此时出现“日历控件 10.0 属性”对话框，如图 8-68 所示。在该对话框中，显示出了该日历控件 10.0 的所有属性、方法和事件，我们可以在该对话框中预览这些信息，以备后来使用。单击“确定”按钮确定。

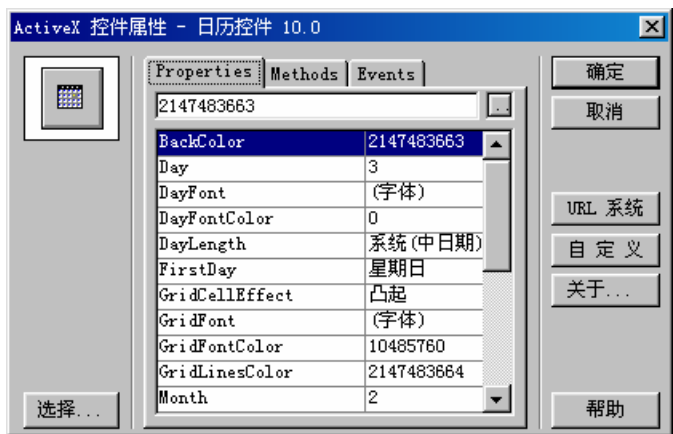


图 8-68 “Calendar Contro 8-0 属性”对话框

(11) 此时就将该 Calendar 控件插入到程序中，将该 ActiveX 控件图标更名为“日历”。

(12) 选中“日历”图标，然后按下快捷键<Ctrl>+1，打开展示窗口，调整日历的大小和位置，如图 8-69 中所示。



图 8-69

(13) 拖入一个【交互】图标到流程线上，将其命名为“控制事件”。

(14) 拖动一个【群组】图标到【交互】图标的右侧，选取事件响应，并将其命名为“单击日期”。

(15) 双击事件响应类型图标，打开其属性对话框，在“事件”标签页中，双击“发送者”列表框中的“图标 ActiveX”，选中该事件发生器，然后在“事件名称”列表框中双击“Click”，选中单击事件。最后单击“确定”按钮确定。

(16) 双击该【群组】图标，打开其二级窗口。

(17) 在二级窗口的流程线上拖动一个【计算】图标，命名为“记录日期”，并在该图标中输入如下语句：

Value:= ""

-初始化

Value:= GetSpriteProperty(@"日历", #Year)^ "年"

-在该字符串变量中加入单击的年

Value:=Value^ GetSpriteProperty(@"日历", #Month)^"月"

-在该字符串变量中加入单击的月

Value:=Value^ GetSpriteProperty(@"日历", #Day)^"日"

-在该字符串变量中加入单击的日

(18) 新建这个变量 Value，并将其初始值赋为空字符串。

(19) 再在该二级窗口的流程线上拖入一个【显示】图标，并输入文本“ 您单击的日期是{Value}！”，文本风格为“ 显示信息 ”，并将其移动到日历的正下方。

(20) 在“ 控制事件 ”【交互】图标的最右侧再拖入一个【计算】图标，将其命名为“ 退出 ”，响应类型为按钮响应，并在图标中输入语句“ Quit() ”。

(21) 双击按钮响应类型图标，将按钮移动到屏幕的右下角。

(22) 此时程序已经设计完成，程序流程如图 8-70 所示。

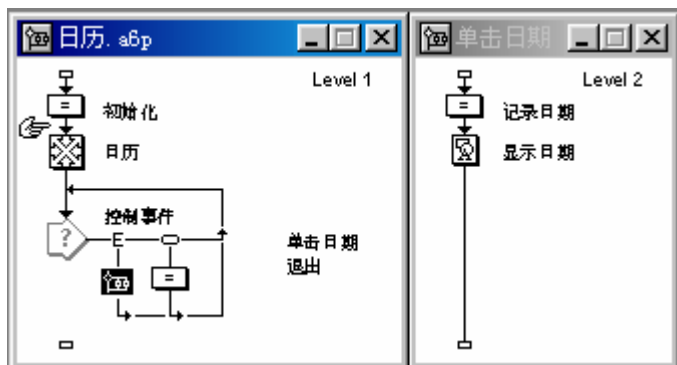


图 8-70 设计好的程序流程

第9章 【计算】图标

- 【计算】图标简介
- 【计算】图标的操作
- 计算编辑窗口介绍
- 【计算】图标实例

9.1 【计算】图标简介

使用【计算】图标可以执行表达式和语句,通过表达式和语句来改变变量的值,或者执行计算功能。可以在流程线上的任意位置插入【计算】图标,或者将计算功能附着在其他图标上。

可以为【计算】图标添加表达式。双击【计算】图标,打开计算编辑窗口,然后在窗口中输入表达式和 If...Then 语句以及 Repeat Loops 语句,也可以插入注释语句来描述表达式。

Authorware 通常会执行完【计算】图标中所有的语句后才退出,但可以在表达式中使用 Exit 语句来强制 Authorware 退出【计算】图标。如果【计算】图标是附着在其他图标上的,则该图标会正常地执行。

在大部分情况下,是将【计算】图标放置在另一个图标前面还是将它附着在该图标之上,只是一个优先级的问題。然而,如果将一个【计算】图标附着在一个【决策】图标或者【交互】图标上,如果该图标使用了“重试”分支形式,Authorware 就会不断重复地执行该计算功能。而如果将一个【计算】图标放置在一个【决策】图标或者【交互】图标之前,则 Authorware 只会执行它一次。

当 Authorware 执行一个【计算】图标时,不会有其他进程发生,并且 Authorware 不会对鼠标单击和按键做出响应。对于只包含一些相对简单的计算语句的【计算】图标来说,执行时的停顿时间是觉察不到的。然而,如果【计算】图标中包含有复杂的 Repeat Loops 语句,就需要执行较长一点的时间,也会有一个显著的停顿时间。

使用【计算】图标的功能有两种方法:

- 直接使用【计算】图标

将【计算】图标直接拖动到设计窗口流程线上的指定位置,然后打开计算编辑窗口进行编辑即可。



- 将【计算】图标附着在其他图标上

在流程线上，首先选中要附着计算功能的图标，如【显示】图标、【移动】图标、【交互】图标等，然后选择菜单[修改]→[图标]→[计算]命令或者按下快捷键<Ctrl>+=，即可打开计算该图标的计算编辑窗口，在该窗口中输入计算语句即可。与【计算】图标不同的是，该计算编辑窗口的标题栏显示的是该附着图标的标题。当在图标中添加了程序代码，该图标的左上角会出现一个“=”号，如图 9-1 所示，说明该图标附着有【计算】图标。

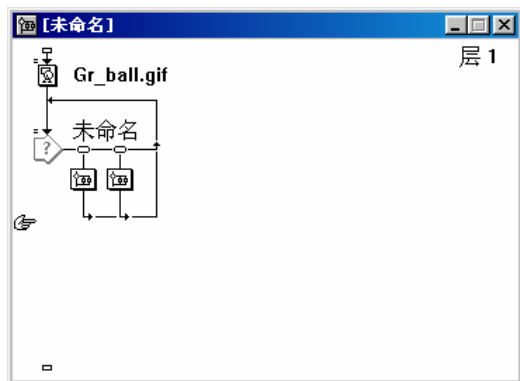


图 9-1 附着有计算功能的图标示例

如果希望编辑附着有计算功能的图标的计算语句，可以双击该图标左上角的“=”号，打开其计算编辑窗口，然后进行修改编辑即可。

9.2 【计算】图标的操作

9.2.1 编辑【计算】图标的计算语句

1. 打开计算编辑窗口

要打开计算编辑窗口，一般来说有两种方法：

- 直接双击【计算】图标

要编辑某个【计算】图标，可以双击该图标，打开计算编辑窗口，如图 9-2 所示。

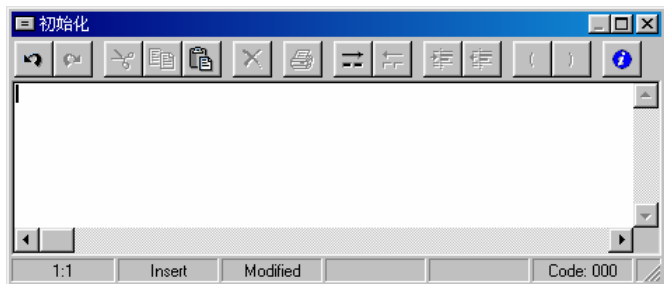


图 9-2 计算编辑窗口

如果是附着有计算功能的其他图标，可以双击该图标左上角的“=”号，打开其计算编辑窗口。

- 运行程序时打开计算编辑窗口

我们也可以首先运行程序，当程序遇到【计算】图标或附着有计算功能的其他图标，会自动打开其



计算编辑窗口。

2. 编辑计算语句

当打开计算编辑窗口后，就可以在该窗口中进行编辑计算语句。语句可以是赋值语句、表达式、函数等。

当输入一个语句后，最好换一行继续输入另一个语句，而不要将几个语句输入在一行中。

如果一个语句很长，当然可以在一行中全部输完，就算超出了窗口，计算编辑窗口会自动加上滚动条，但是这样不易读。所以我们可以将很长的语句分成几行输入，此时就需要在行尾使用一个继续符号 $\sim$ 。要创建一个继续符号，可以按下<Alt>+<Enter>键。例如：

```
Score:=JudgedResponses^Return^~  
JudgedInteractions^Return^PercentCorrect^Return~  
^FirstTryCorrect
```

在计算编辑窗口输入变量或函数时，变量名和函数名是不分大小写的，即，Score、score 和 SCORE 是相同的变量，Authorware 会自动将它们改变成定义时的格式。

当在计算编辑窗口中输入完语句后，Authorware 会弹出一个提示对话框，如图 9-3 所示。

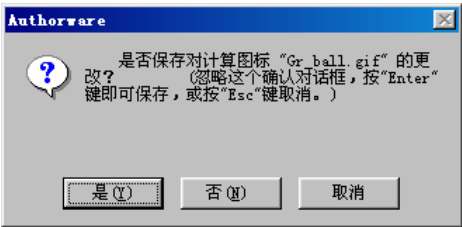


图 9-3 提示对话框

该对话框是询问用户是否保存对【计算】图标所做的修改，单击“是”按钮确定，单击“否”按钮则不保存修改，而单击“取消”按钮，则取消操作，重新回到计算编辑窗口中，继续进行修改。

9.2.2 改变计算编辑窗口中参数设置

您可以改变计算编辑窗口中的一些参数，一旦您改变了这些参数，它们就变成了默认的。例如，您改变参数使得计算文本大小写敏感（区分大小写），则函数或者变量会自动转换为正确拼写。

为了改变计算编辑窗口的参数，进行如下操作：

（1）打开计算编辑窗口参数设置对话框，如图 9-4 所示，可以使用如下方式：

- 选择菜单[文件]->[参数选择]->[计算]；
- 打开流程线中的任何一个【计算】图标，显示出计算编辑窗口，

然后单击按钮

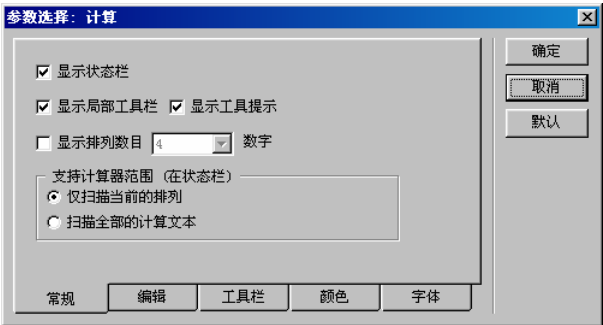


图 9-4 “参数选择: 计算”对话框

（2）在该对话框中设置好计算编辑窗口的各个参数。

（3）如果希望恢复默认设置，则单击“默认”按钮。

（4）否则单击“确定”按钮确认参数设置。此时就可以使用设置后的各个效果。

(5) 如果不想设置参数,则单击“取消”按钮退出。

9.3 【计算】图标实例——计算器

下面我们通过一个实例来说明如何使用【计算】图标。

该实例是制作出一个计算器,功能与一般的计算器相同,如图 9-5 所示。



图 9-5 计算器

操作步骤为：

- (1) 新建一个文件,命名为“计算器.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性]命令,打开文件属性对话框,选择“大小”下拉框中的“根
据变量”选项,在“属性”标签页中取消“菜单栏”复选框,选取背景色为黑色,然后单击“确定”按
钮确定。
- (3) 拖动一个【计算】图标到设计窗口的流程线上,并将其命名为“初始化”。
- (4) 双击该【计算】图标,打开其计算编辑窗口,然后在窗口中输入语句：

```

ResizeWindow(250, 300)
--初始化屏幕大小
Oper1:=0
Oper2:=0
--初始化两个操作数
Flags:=TRUE
--初始化标志数

```
- (5) 拖动一个【显示】图标到流程线上,将其命名为“背景”。
- (6) 绘制两个矩形,一个矩形作为显示屏,填充颜色为浅兰色,另一个矩形为控制面板,边框颜
色为紫色,无填充颜色,并将它们调整,如图 9-5 中所示。
- (7) 拖动一个【交互】图标到流程线上,将其命名为“控制按钮”。
- (8) 拖动 17 个【计算】图标到【交互】图标的右侧,依次命名为“1”、“2”、“3”、“4”、
“5”、“6”、“7”、“8”、“9”、“0”、“+”、“-”、“*”、“/”、“确定”、“复位”和
“退出”,响应类型都为按钮响应。
- (9) 双击“1”【计算】图标,打开计算编辑窗口,在该窗口中输入如下语句：

```

if Flags=FALSE then
--如果是已经输入了数字
    Oper1:=Oper1*10+1
--继续增加该数字
else
--如果是新一次输入数字
    Oper1:=1
end if
Flags:=FALSE
--操作数 1 已经输入了数字

```

(10) 同理, 将“2”、“3”、“4”、“5”、“6”、“7”、“8”、“9”和“0”【计算】图标中也输入上面的语句, 只是将“Oper1:=Oper1*10+1”和“Oper1:=1”中的“1”改为相应的数字。

(11) 双击“+”【计算】图标, 打开计算编辑窗口, 在该窗口中输入如下语句:

```

if Oper1<>0 & Oper2<>0 then
--如果两个操作数都设置好了
    if s="+" then Oper1:=Oper1+Oper2
    else if s="-" then Oper1:=Oper2-Oper1
    else if s="*" then Oper1:=Oper1*Oper2
    else Oper1:=Oper2/Oper1
    Oper2:=Oper1
--将运算结果赋给操作数 2
    s:="+"
--设置运算符号为加
else
--如果只设置了操作数 1
    Oper2:=Oper1
--将设置好的操作数 1 的值赋给操作数 2
    Oper1:=0
--初始化操作数 1
    s:="+"
--设置运算符号为加
end if
Flags:=TRUE
--设置标志数为 True, 重新输入另一个操作数

```

(12) 同理, 将“-”、“*”和“/”【计算】图标也输入上面的语句, 只是将上面的“s:=+”设置成相应的操作符即可。

(13) 双击“确定”【计算】图标, 打开计算编辑窗口, 在该窗口中输入如下语句:

```

if Oper1<>0 & Oper2<>0 then
--如果两个操作数都设置好了
    if s="+" then Oper1:=Oper1+Oper2
    else if s="-" then Oper1:=Oper2-Oper1
    else if s="*" then Oper1:=Oper1*Oper2

```

```

else Oper1:=Oper2/Oper1
else
--初始化
Oper1:=0
end if
Oper2:=0
Flags:=TRUE
--设置标志数为 True，重新输入操作数

```

(14) 双击“复位”【计算】图标,打开计算编辑窗口,在该窗口中输入如下语句:

```

Oper1:=0
Oper2:=0
Flags:=TRUE

```

(15) 双击“退出”【计算】图标,打开计算编辑窗口,在该窗口中输入如下语句:

```

Quit()

```

(16) 双击【交互】图标,打开展示窗口,将 17 个按钮调整排列对齐在绘制的控制面板中,并在显示屏的矩形中输入文本“{Oper1}”,如图 9-6 所示。

(17) 将文本调整在显示屏的中间,然后选择菜单[文本]→[对齐]→[右齐]命令,或者直接使用快捷键<Ctrl>+,将文本右对齐。

此时程序已经设计完成,程序流程如图 9-7 所示。

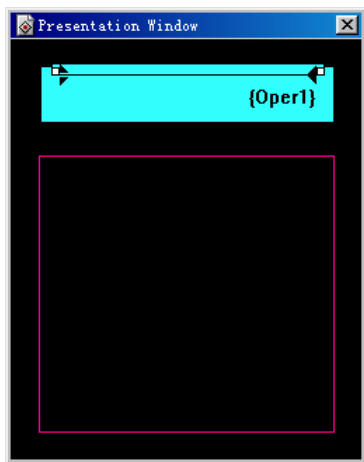


图 9-6 排列对其按钮且显示操作数

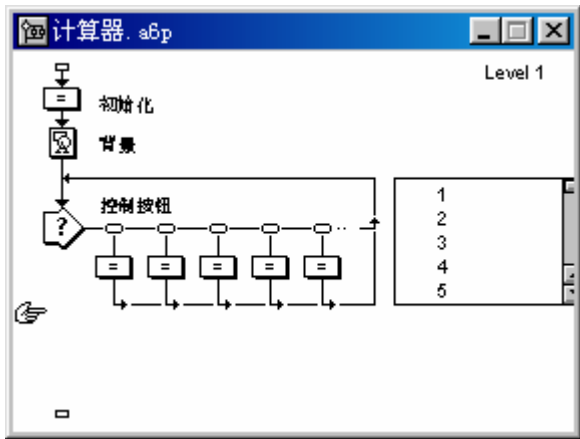


图 9-7 设计好的程序路程

第10章 【数字电影】图标、【声音】图标

- 【数字电影】图标、【声音】图标图标的属性设置
- 为程序添加数字电影、声音效果
- 根据作品的需要设置数字电影、声音属性
- 举例说明声音数字电影的协调实现和控制

10.1 图标简介

在 Authorware 中，除了最常见的文本和图形对象之外，系统还提供了对声音、数字电影以及视频的集成能力。在当前多媒体项目中，使用数字电影、声音和视频往往可以加强你的作品的表现力，将直接影响到你的作品的好坏，并且这三种媒体对象在多媒体项目中使用的非常广泛。我们常会把多媒体与漂亮的动画、优美动听的音乐联系在一起。在本章中，我们将讲述【数字电影】图标和【声音】图标，利用这些图标，我们可以自己动手制作好看好听的多媒体作品，可以轻松方便的加入音乐、导入动画，也可以自己尝试制作小电影播放器、CD 播放器，动手吧。

- 【数字电影】图标：利用它可以播放多种格式的数字电影和动画，如：AVI、FLI、FLC、MOV 等格式。
- 【声音】图标：此图标可以播放声音文件，并且可以对播放方式进行控制。利用它可以在多媒体作品中制作背景音乐，为文本配上解说词，也可以在用户的交互结果中给予提示。可以播放的声音文件格式有 WAV、AIF、VOX、SND 等。
- 【视频】图标：控制外接视频播放设备。



下面我们将分别介绍控制动画文件的【数字电影】图标、支持媒体对象的【声音】图标、以及控制视频设备的【视频】图标。

10.1.1 【数字电影】图标

数字电影图标又称【数字电影】图标，系统提供的【数字电影】图标用来导入除了使用 Authorware 提供的运动图标创建的简单的运动效果外，还可以是借助于其他应用软件创建的动画文件，如 3D Studio Max 创建的动画文件*. FLI、*. FLC，Director 创建的动画文件*. dir，以及其他的视频文件，如 AVI、MPEG 文件等。

在 Authorware 中导入数字电影文件。从图标栏上拖动一个【数字电影】图标到流程线上，双击该【数字电影】图标，打开“属性：电影图标”对话框，如图 10-1 所示。

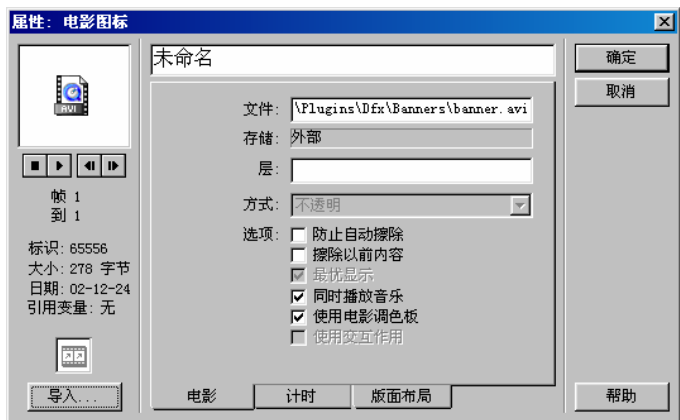


图 10-1 【数字电影】图标属性对话框

我们可以看到该属性对话框具有一个“导入”按钮和三个标签页：“电影”标签页、“计时”标签页和“版面布局”标签页。

在该对话框中单击“导入”命令按钮，系统将弹出“导入哪个文件？”对话框，单击该对话框中“文件类型”下拉列表右侧的向下箭头，我们可以看到 Authorware 所能支持也就是能够导入项目的动画文件类型。表 10-1 列出了 Authorware 所能支持的文件类型及其说明。

表10-1 Authorware 所能支持的文件类型及其说明

文件类型	说 明
BITMAP SEQUENCE	包含具有特定编号的有相似根文件名 (NAME001—NAME00N) 的 BMP 文件序列，可以通过选择一个开始文件帧，AUTHORWARE 可以用剩余的文件创建一个动画。
FLC/FLI	以内部文件方式存储的动画文件*.FLI、*.FLC，由 3D STUDIO MAX 创建。
DIRECTOR	以内部文件方式存储的动画文件*.DIR、*.DXR,如果 DIRECTOR 动画包含能够交互作用的动画序列，用户可以将其合并成 AUTHORWARE 文件。
MPEG FILES	以内部文件方式存储，它集成了压缩的数字动画和同步声频，需要安装 MICROSOFT ACTIVE MOVIE/SOFT PEG，或者是 MPEG 卡和驱动。
VIDEO/QUICKTIME FOR WINDOWS	(AVI/MOV) ,以外部文件方式存储 ,必须要确定已正确的安装了 VIDEO FOR WINDOWS/ QUICKTIME FOR WINDOWS
PICS FILES FOR MACINTOSH	当转换成 AUTHORWARE 文件时已内部方式存储，它不能在 AUTHORWARE 动画装载对话框中导入

当导入动画文件后，在属性对话框的左侧会显示出该动画文件的一些文件信息，如图 10-2 所示。



图 10-2 导入动画文件的文件信息

点击  按钮可以播放该动画文件。

帧 1：表示当前播放的帧数，即正在播放第几帧。

到 112：表示当前动画文件所有的帧数。

标识：表明该【声音】图标的惟一的标识 ID 号。

大小：表明该文件所占字节。

日期：表明该文件最后修改的时间。

引用变量：表明该文件是否引用参考变量。

下面分别介绍“电影”标签页、“计时”标签页和“版面布局”标签页。

(1) “电影”标签页

- “文件”文本框：显示了当前导入的动画文件的名称和路径。
- “存储”文本框：显示了动画文件的存储方式，有两种存储方式：“外部”方式和“内部”方式，它由动画文件本身决定，下面就此分别介绍。

“内部”方式：内嵌式动画直接以“Authorware”文件方式存储，执行速度快，增加了文件的大小，可以用内部存储动画使用擦除转换。

“外部”方式：不以“Authorware”文件方式存储，以外部文件单独存放，不会增加最终执行文件的大小，不能用 MPEG/AVI 动画使用擦除转换，要对其他类型的动画做擦除转换在擦除图标属性对话框中必须选择“擦除以前内容”选项。

“内部”方式支持 FLC、FLI、CEL、PIC 等动画文件格式，“外部”方式支持 DIR、AVI、MOV、MPG 等动画文件格式。同时，Authorware 还支持 QuickTime 动画和 Flash 动画，主要采用 FLC 和 AVI 两种格式的动画。

- “层”文本框：在该文本框中，用户可以为动画指定其所在层属性，可以重新设定一个值，也可以用变量来定义这个值。在默认情况下，动画文件将在其他显示对象前播放。其层属性决定在显示窗口中前后相关的其他对象的动画位置。
- “方式”下拉框：设置如同显示图标对象一样的动画对象显示模式，它包括四个选项：

“不透明”显示模式：可描述所有形式的动画，而不需要透明像素。这种动画执行更快，占据很少的空间，通常以外部文件方式存储的动画总是采用这种显示模式。

“透明”显示模式：它可以通过动画帧显示其他的对象。在默认情况下，AutoDesk 动画通过黑色像素显示对象。

“遮隐”显示模式：在动画框架后放置的一个非透明去光层，该各框架去光层应当与动画图像匹配。在默认情况下，AutoDesk 动画中白色像素从框架边缘切割出来，你可以将其从黑色像素改变为白色像素。

在你首次选中遮隐选项时，当系统为每个框架创建不规则形的无光泽层时将会出现“创建遮隐”框。

“反转”显示模式：在动画框架中将色素有白色转换成黑色或由黑色转换成白色，如同需要通过重放来显示下面的对象一样。如果一个有色动画放置在有色背景上时，这种显示模式可使动画显示出相反的颜色（这样可以产生意想不到的效果）。

- “选项”选项：用户可以根据该选项，设置数字电影文件的声音通道是否播放，以及动画文件的调色板。

“防止自动擦除”复选框：默认值为选中。如果选中此复选框，那么动画文件在播放结束之后不会被自动擦除，用户必须使用擦除图标来擦除该动画对象。

“擦除以前内容”复选框：默认值为选中。如果选中此复选框，在播放动画文件之前，将擦除演示窗口中已经存在的显示对象。

“最优显示”复选框：选中使动画对象可以在其他显示对象前播放，当你想在当前层前指定动画时可以关掉此选项。

“同时播放音乐”复选框：设置动画文件执行时声音通道继续播放，默认值选中。

“使用电影调色板”复选框：选择是否用动画调色板，可确定使用数字电影调色板还是 Authorware 调色板，该选项并不是对所有动画形式使用。

“使用交互作用”复选框：用来对 Director 动画进行交互作用，例如通过点击鼠标或键盘开始播放，通常不可选。

（2）“计时”标签页

在“计时”标签页中你可设置数字电影文件的播放同步以及播放速率等。打开的属性选项如图 10-3 所示。

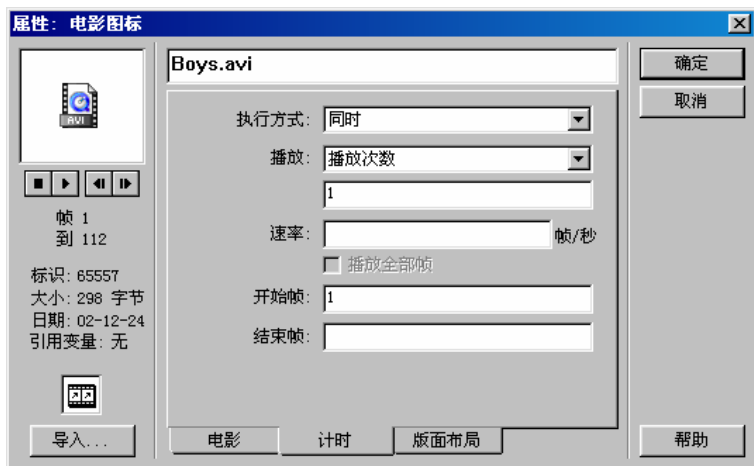


图 10-3 【数字电影图标】属性对话框的“计时”标签页

- “执行方式”下拉框：用来设置动画文件播放的同步问题。其中有三个选项：“同时”、“等待直到完成”和“永久”。

“同时”选项：如果选中，那么在动画文件播放的同时，程序流程不会在【数字电影】图标上等待，将继续执行流程线下面的图标。

“等待直到完成”选项：如果选中，当程序运行到【数字电影】图标时，程序将等待动画文件的播放结束。只有当动画文件播放完毕之后，程序才继续下面的流程。

“永久”选项：选中该选项时，用户可以在“Authorware”退出数字电影图标时激活动画，系统将



一直监控在【数字电影】图标属性对话框中所定义的变量，如果变量的值改变，系统将立即做出相应的调整。此时必须同“永久”选项下的“播放”下拉框中的“直到为真”结合使用。比如，你可以在“速率”文本框中输入一个变量来控制动画的重放速率。如果选中“永久”选项并且变量的值发生改变，动画播放速率将根据其值相应的做出改变。

- “播放”下拉框：用户可以设置动画文件播放的次数。它包括三个选项：

“重复”选项：重复播放动画，直到擦除或用“Media Pause”函数暂停。

“播放次数”选项：此为“播放”下拉框的默认选项，此时需要在“播放”下拉框下面的文本框中键入动画文件播放的次数。如果为0，系统将只显示第一帧动画。

“直到为真”选项：动画文件将一直播放，直到下面文本框中的条件变量或者条件表达式值为真。比如，如果你输入系统变量“Mouse Down”，动画文件将重复播放直到用户按下鼠标键。

- “速率”文本框：用户可以设置一个支持可调节速率形式的以外部文件存储方式的数字电影。通过输入一个数、变量名或者一个条件表达式来加快或减缓动画播放的速率。如果所设的速率太快，无法按所设速率显示所有动画帧，除非用户选中“播放全部帧”选项，系统将略过部分动画帧从而以接近所设速率播放动画。

“播放全部帧”复选框：系统将不掠过任何动画帧而尽可能快的播放动画，但并没有在“速率”文本框中所设定的速率快，该选项可以使动画在不同的系统中以不同的速率播放。它只对以内部文件存储方式保存的动画文件有效。

“开始帧”文本框和“结束帧”文本框：用来设定动画文件的播放起始帧和结束帧。

(3) “版面布局”标签页

可以在“版面布局”标签页中设定动画对象在演示窗口中是否可以被移动，以及其可以移动的区域。此选项卡与显示图标的“版面布局”选项卡设置相同，所以就不再一一赘述。

当我们拖动其他图标到【数字电影】图标右侧，使其成为该图标的子图标，如图 10-4 所示。

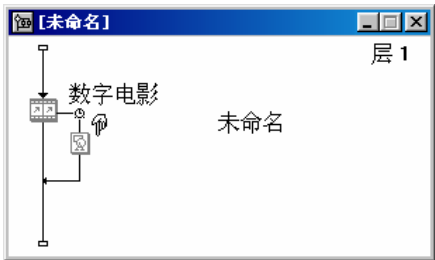


图 10-4 为【数字电影】图标添加子图标

我们可以为【数字电影】图标添加多个子图标，如图 10-5 所示。

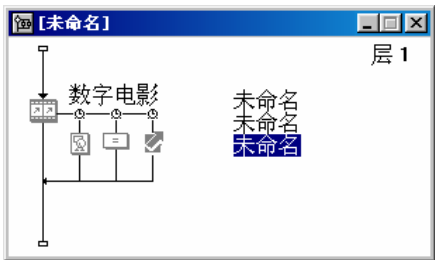


图 10-5 为【数字电影】图标添加多个子图标

双击这些子图标上侧的同步属性图标，系统弹出“媒体同步”对话框，如图 10-6 所示。

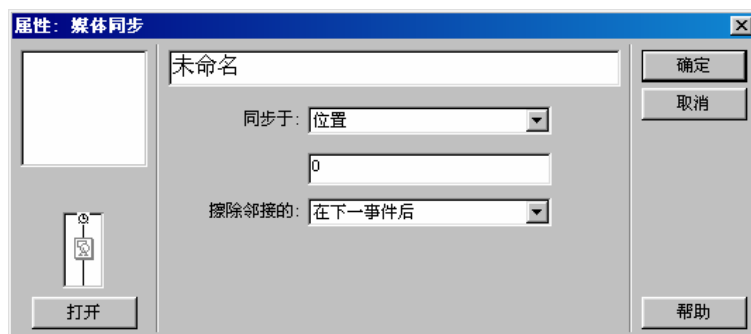


图 10-6 “媒体同步”对话框

如果您将一个图标放置在【数字电影】图标或者【声音】图标右侧，使其成为【数字电影】图标或者【声音】图标的子图标。当这些媒体开始播放时，您可以根据媒体位置或媒体播放的秒数来触发这些子图标。因为【数字电影】图标和【声音】图标有自己的分支图标，所以它们不再能够直接成为一个【交互】图标、【决策】图标或者【框架】图标的子图标。当您 Authorware 5 文件转化升级为 Authorware 6，任何直接是【交互】图标、【决策】图标或者【框架】图标的子图标的【数字电影】图标或者【声音】图标都将成为一个【群组】图标下的图标。

当您拖动一个图标到【数字电影】图标或者【声音】图标的右侧，一个同步属性图标就会显示出来，其形状像一个钟表。

当子图标添加后，【数字电影】图标或者【声音】图标定时就从永久变为并发。

下面就“媒体同步”对话框中的各个选项含义进行一一说明。

- “同步于”下拉框：该下拉框设置同步并发选项，其包括如下选项：
- “位置”选项：选择该选项，则根据【数字电影】图标或者【声音】图标的位置来并发，然后在下侧的文本框中输入一个位置值。如果媒体是一个数字电影，则位置值是帧数；如果媒体是音频，则位置值是毫秒数。
- “秒”选项：选择该选项，则根据媒体播放过的秒数值来并发，然后在下侧的文本框中输入一个秒数数值。
- “擦除邻接的”下拉框：设置擦除方式，其包括如下选项：
- “在下一事件后”选项：选择该选项，将在进入下次事件后擦除本图标的信息。
- “在下一事件前”选项：选择该选项，将在进入下次事件前擦除本图标的信息。
- “在退出前”选项：选择该选项，将在退出【数字电影】图标或者【声音】图标时擦除本图标的信息。
- “不擦除”选项：选择该选项，将不擦除本图标的信息。

当您单击该对话框中的“打开”按钮，可以打开展示窗口，来显示和编辑本图标的信息。

单击“确定”按钮，保存并发属性的改变。

10.1.2 【声音】图标

如同影视作品一样，多媒体作品中的声音也可分为配音和音效两种。配音主要实质单纯的语音介绍，如解说词等纯语音文件，音效是指一些用于烘托画面的背景音乐，当然有时我们将他们和在一起使用。Authorware 中，系统提供了【声音】图标来管理声音媒体对象，他为用户在作品中添加声音媒体提供了简单而快捷的方式。【声音】图标直接支持的声音文件格式主要有 AIFF、PCM、SWA、VOX、WAV 等，一般情况下是 WAV 文件，同时我们可以通过调用函数的方式来播放 MIDI 音乐。

当用户希望向项目文件中添加声音对象时，只要从图标栏上拖动一个【声音】图标到主流程线上，然后在流程线上双击该【声音】图标，打开其属性对话框，就可以根据用户的要求设定该对象的属性，如图 10-7 所示。



图 10-7 【声音】图标属性对话框

我们可以从打开的“属性：声音图标”属性框中看到左下角“导入”按钮，以及声音和计时选项卡，其中单击“导入”按钮可以导入你所需要的声音文件，系统将弹出“导入哪个文件？”对话框，这时我们可以通过对话框下面的“文件类型”下拉列表右侧的下箭头，会弹出 Authorware 所能够支持的声音文件的类型，其中包括：AIFF、PCM、SWA、VOX、WAVE。

从图 10-8 中我们可以看出在导入声音文件后有关其属性值，声音文件信息如下：

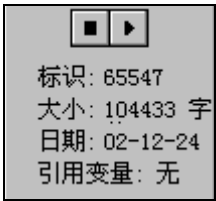


图 10-8 声音文件有关属性

标识：表明该【声音】图标的惟一的标识 ID 号。

大小：表明导入的声音文件所占字节。

日期：表明导入的声音文件最后修改的时间。

引用变量：表明该声音文件是否引用参考变量。

当你在修改已存在文件属性时如果需要确定没有删掉参考图标就可检查声音文件相关信息。选择 [修改]→[图标]→[链接]选项察看图标具体信息。

(1) “声音”标签页

在对话框最上侧的文本框中是该【声音】图标的名称，“文件”选项后是导入的声音文件所在的目录及文件名，可以包括空格；“存储”选项表示该声音文件是以内部方式存储的。还有声音文件的大小、格式以及播放速率。这些属性值已经确定，由所选的声音文件确定，除非更换声音文件，否则不能更改。我们可以单击 [] 按钮右箭头来确定是否已经导入声音文件，也可以用它来检查声音效果。

当我们完成声音文件的导入工作后，还需要在“属性：声音图标”对话框中设置声音的播放属性。我们将继续上面的导入声音文件为例来讲解如何控制声音的播放。

（2）“计时”标签页

完成声音文件的导入后，单击“属性：声音图标”对话框中的“计时”，打开“计时”标签页，对声音文件的播放控制主要是在“计时”标签页中完成的。我们可以看到“计时”标签页如图 10-9 所示：

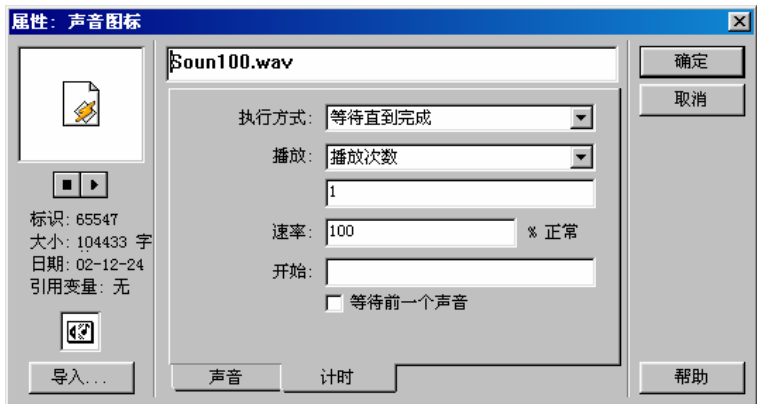


图 10-9 “计时”标签页

我们从图 10-9 可以看出，在“计时”标签页中用户可以设置声音文件播放的同步条件，声音文件的播放次数，以及声音文件的播放速度。

- “执行方式”下拉框：用来设置该声音文件播放的同步条件，它有三种选项：

“等待直到完成”选项：程序流程执行在执行到该【声音图标】后，将等待声音文件播放结束后才继续下面的流程。

“同时”选项：当程序执行到【声音】图标的时候，在声音文件播放的同时，程序流程并不停留，将继续向下运行。当用户在屏幕上运行其他同步动作而同时播放声音时这个选项非常有用。

“永久”选项：系统将一直监视控制声音文件播放的条件，该选项必须要和“播放”下拉选项中的“直到为真”配合使用。

- “播放”下拉框：是用户用来控制声音文件的播放次数及播放条件，它包括两个选项：“播放次数”和“直到为真”。

如果用户在“播放”下拉框中选择了“播放次数”（默认选项），那么用户在下面的文本框中间输入声音文件重复播放的次数，在默认情况下设为 1，代表声音知识播放一次。如果用户在“播放”下拉框中选择了“直到为真”，那么用户可以在“播放”下拉框下面的文本框中，键入控制声音文件播放停止的条件变量或者条件表达式，但条件变量或者条件表达式的值为 True 时，声音文件的播放就将停止。

- “速率”文本框：是用来控制声音文件播放的速率。通常是以声音文件正常播放速率的百分比来设置播放速率。例如当在“速率”文本框中输入 50，此时声音文件的播放速率就是声音文件原正常播放速率的一半。
- “开始”文本框：用户可以输入条件变量或者条件表达式，控制声音文件何时开始播放，当条件变量或条件表达式为 True 时，声音文件将播放。
- “等待前一个声音”复选框：用来指定是否要等待上一个声音文件播放文件结束之后才能播放。

10.2 声音动画实例——小电影

声音、动画是多媒体软件不可或缺的重要组成部分，我们所看到的绝大多数多媒体软件都是将声音动画和图像很好的结合在一起组成一部优美动听的多媒体作品。我们评价一个多媒体作品往往是凭视觉



和听觉来感受，对声音和动画使用的好坏直接影响到一个作品优劣，如何做到声音和动的有机结合就是我们着重强调的部分。

下面我们从例子来更具体直接深入的了解【数字电影】图标和【声音】图标的使用，用它们创作出吸引人的作品。

- (1) 新建一个 Authorware 文件，保存为“小电影.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，选择“大小”下拉框中的“根据变量”选项，在“属性”标签页中取消“菜单栏”复选框，选取背景色为紫色，然后单击“确定”按钮确定。
- (3) 选择菜单[文本]→[定义风格]命令，打开定义文本风格对话框。
- (4) 单击定义文本风格对话框中的“添加”按钮，增加一种新的文本风格，将该文本风格更名为“标题文字”，选择字体为华文彩云、字号为 36，字色为绿色，然后单击“修改”按钮确定修改。如图 10-10 所示。



图 10-10 定义文本风格

- (5) 单击对话框中的“完成”按钮，完成定义文本风格。
- (6) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (7) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：
ResizeWindow(400, 300)
--初始化屏幕大小
control:=1
voice:=0
--初始化变量
- (8) 从图标工具栏中拖入一个【显示】图标到主流程线上，命名为“背景”，然后使用文字工具输入文本“小电影”，文本风格为“标题文字”。然后使用指针工具将文本移动到屏幕的正上方。
- (9) 拖入一个【数字电影】图标到流程线上，将其命名为“小电影”。
- (10) 双击该图标，打开其属性对话框，然后单击“导入”按钮导入做好或者本来已经有的数字电影文件，在本例中导入的是 AVI 文件，同时是以外部方式存储。然后设置其属性，在“电影”标签页中，取消“擦除以前内容”复选框，以免在运行中删除背景，同时选中“同时播放音乐”复选框和“使用电影 调色板”复选框。

(11) 打开“计时”标签页，在“执行方式”下拉框中选取“同时”选项，在“播放”下拉框中选取“直到为真”选项，并在下面的文本框中输入条件语句“control=1”，在“开始帧”文本框中输入“1”，如图 10-11 所示。单击“确定”按钮确定。

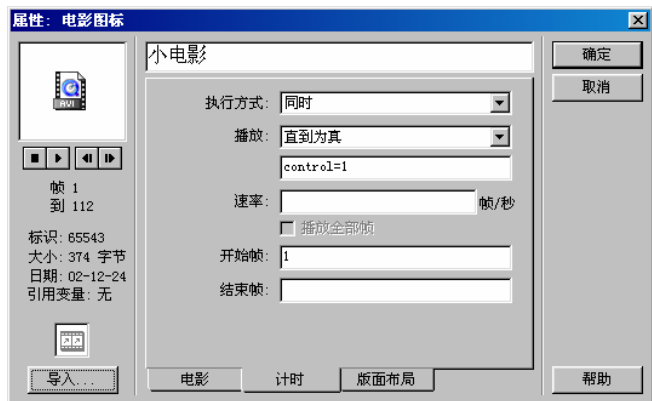


图 10-11 【数字电影】图标的“计时”标签页属性设置

(12) 在流程线上拖入【声音】图标，命名为“钟声”，导入声音文件，并选择“计时”标签页，在“执行方式”下拉框中选取“同时”选项，在“播放”下拉框中选取“直到为真”选项，并在下面的文本框中输入条件语句“voice=0”，在“速率”文本框中输入“100”，选中“等待前一个声音”复选框，如图 10-12 所示。单击“确定”按钮确定。



图 10-12 【声音】图标属性对话框的“计时”标签页

(13) 在主流流程线上拖入【交互】图标，命名为“控制”。

(14) 在【交互】图标的右侧拖入一个【计算】图标，选取按钮响应，并命名为“播放”，分支方式为“重试”，然后双击图标打开其计算编辑窗口，输入如下代码：

```
control:=0
voice:=1
--设置两个变量，使得能够播放小电影和声音
GoTo(IconID@"小电影")
--跳转到“小电影”图标处执行
```

(15) 在【交互】图标右侧继续拖入两个【计算】图标，分别命名为“暂停”和“退出”。其中的代码分别如下：

```
--“暂停”【计算】图标中语句
control:=1
```

```
voice:=0
----设置两个变量，使得停止播放小电影和声音
GoTo(IconID@"小电影")
--跳转到“小电影”图标处执行
--“退出”【计算】图标中语句
Quit(0)
```

(16) 存盘，运行程序过程中按 Ctrl+p 暂停程序运行，拖动按钮和动画显示窗口至合适位置。此时程序流程图如图 10-13 所示。

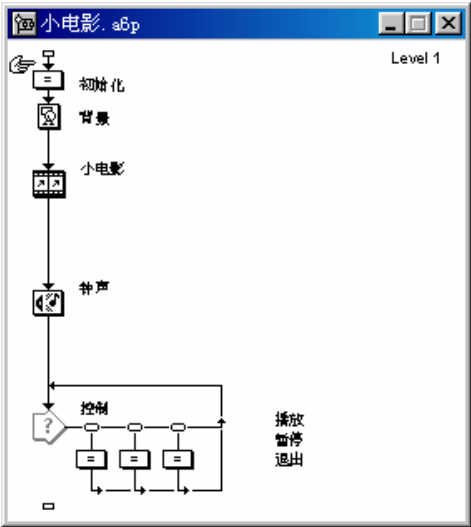


图 10-13 设计好的程序流程

现在我们可以执行该程序了，当我们按下“播放”按钮时，程序就会播放动画并伴随声音，当我们按下“暂停”按钮时，程序暂停。该程序只是简单的实现了动画和声音的配合，并且适当的加以播放控制，执行的结果如图 10-14 所示。



图 10-14 程序执行结果

第11章 开始旗、终止旗以及图标调色板

- 起始旗、终止旗图标简介
- 如何使用起始旗和终止旗调试程序
- 图标调色板的使用

11.1 “开始旗”和“终止旗”简介

11.1.1 “开始旗”和“终止旗”介绍

在图标板的下方是两个旗帜：“开始旗”和“终止旗”，它是用来供程序调试的工具，如图 11-1 所示。



图 11-1 “开始旗”和“终止旗”

在我们制作多媒体作品时，经常在基本完成时却出现这样那样的问题，怎样把作品中隐藏的毛病找出来是创作中一个非常重要的环节，这就需要程序调试。Authorware 为用户提供了多种程序调试方面的方法和手段，这里我们将详细讨论一下起始旗和终止旗的使用和技巧。

当我们制作的作品中流程线很长，要调试其中的某一部分时，不可能让我们从开头运行来调试程序，这样会很浪费时间，而且范围太大无法确定具体的位置，应从要调试的程序段起始位置开始调试，这样马上就可以确定问题的位置。在 Authorware 中为我们提供了开始旗和终止旗来实现这项功能。

- “开始旗”：在流程线上设置“开始旗”可以使程序在特定的位置开始运行。在调试程序时，设置开始旗对察看程序的特殊部分非常有效。



如果您在流程线上拖入“开始旗”，控制菜单中的“从标志旗处运行”命令将被激活，当您选择“从标志旗处运行”，程序将执行开始旗后的第一个图标，同时系统将所有变量的值置为初始值，并清除显示窗口，就好像您是从开始运行这个程序一样。

只能设置一个“起始旗”和一个“终止旗”，其标识将保留直到移开它为止。要移动标识可以拖动它到一个新的位置，要移走它可以在图标面板中“开始旗”位置处点击它（这种方法对当你忘记标识所在地方时要取消它很方便），“起始旗”和“终止旗”只是暂时的记号，并不包含在最终的打包程序中。当使用控制面板调试程序时，您可以使用开始旗和终止旗。

- “终止旗”：在流程线上拖入一个“终止旗”，可以在特定位置终止程序运行，当调试程序时，“终止旗”对于判断在某一段流程执行情况很有用，你可以在某一点停止流程检查运行情况。

当你的程序变得很长时，在流程中在制作的程序段的开始处放置“开始旗”及在结束的地方放置“终止旗”是很有帮助的。您可以使用控制面板中的“从标志旗处运行”命令来仅仅执行标记之间的图标。

11.1.2 “开始旗”和“终止旗”的使用

当我们制作比较复杂庞大的程序时，调试某一段程序如果都都需要从程序的开头执行，则会花费很多的时间，此时我们就可以使用“开始旗”和“终止旗”。

在要调试的程序段开头处拖入“开始旗”，在程序段的结尾处拖入“终止旗”，然后单击工具栏中的“执行程序”按钮，程序就会从“开始旗”处开始执行，执行到“终止旗”处停止。这样，我们可以将程序一段一段地进行调试，而不必总是从程序开头执行，如图 11-2 所示。

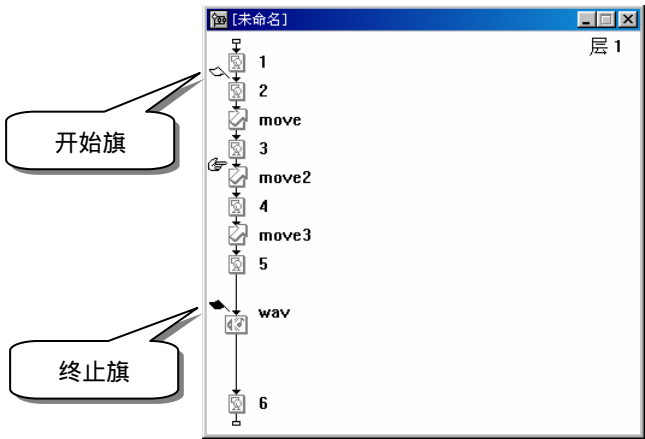


图 11-2 调试“调色板”程序段

如果程序的分层过多，找“开始旗”和“终止旗”是比较困难的，我们可以直接在图标板中“开始旗”或“终止旗”的位置用鼠标左键单击一下，即可让“开始旗”或“终止旗”自动返回到图标板中。当然，您也可以亲自将“开始旗”或“终止旗”拖动到图标板中。

当只使用一个“开始旗”时，Authorware 会从“开始旗”处开始执行程序，直到执行完整个程序；如果只使用一个“终止旗”，则 Authorware 会从程序的开头执行，直到执行到“终止旗”所在的位置。当然，如果全部使用了“开始旗”和“终止旗”，则 Authorware 会从“开始旗”处执行，直到执行到“终止旗”所在的位置。

11.2 图标调色板介绍及其使用

在图标板的最下方是一个图标调色板，它是用来标识图标的，如图 11-3 所示。



图 11-3 图标调色板

在图标调色板中，共有 16 种颜色。当我们将功能相近的图标染成相同的颜色，则在程序调试和查找图标时，都是非常有用的。

要将图标染色，必须首先选中设计窗口中该图标，然后在单击图标调色板中的一种颜色，则选中的图标就会变成指定颜色，如图 11-4 所示是我们将相同的图标都染成了相同的颜色。

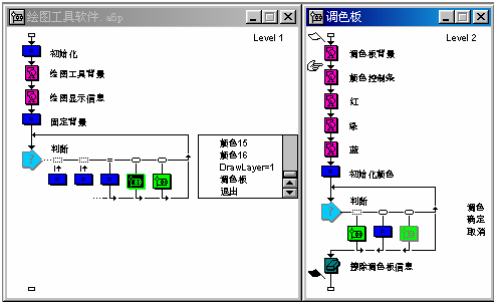


图 11-4 使用图标调色板将功能相近的图标染色

第12章 库的使用

■ 库的简介

■ 库的使用

-195-



12

12.1 库的简介

在制作多媒体作品中，往往会使用到一些相同的内容。在一般的高级程序设计语言中是使用函数或过程，而 Authorware 是使用库和模块来实现这些功能的。

库 (Library) 可以理解为是各种图标的集合。它是一个外部文件，独立于用户作品之外。库中可以存放【显示】图标、【交互】图标、【计算】图标、【声音】图标和【数字电影】图标等五种图标。由于库文件可以与 Authorware 应用程序相分离，所以可以将应用程序进行分工，让不同的设计人员进行合作开发，这样就使应用程序的内容更加容易管理和组织，同时也可以提高应用程序的开发周期。

12.1.1 库的特点

一般来说，库有以下几个特点：

● 节省空间

在应用程序中使用库，可以在库文件之间和库文件与应用程序之间互相拖动图标，而应用程序中使用库文件中的图标，只是建立了应用程序与库文件中该图标的链接，而不是将该图标的副本复制到应用程序中。特别是，当程序中多处使用库文件中的图标时，会极大地降低应用程序的存储空间，使得程序加载和运行时速度都较快。

● 节省时间

当应用程序中多处使用了相同的图标内容，则可以将该图标添加到库文件中，然后将库文件中该图标与应用程序中需要使用的地方建立链接即可，而不必重复建立重复的内容。

同时，如果要修改与库文件相链接的图标内容，只需要修改库文件中的图标内容即可，而不必将应用程序中所有与该图标相同的都进行修改。只要将库文件中直接对该图标进行编辑，应用程序中所有链接的图标会自动得到更新。这样会大大节省修改时间。

- 便于分工合作

由于库文件是与应用程序独立的，所以可以让不同的设计人员设计应用程序不同的部分，最后再将它们进行合成。

12.1.2 库窗口

要在应用程序中使用一个库文件中的内容，必须打开库文件，选择菜单[文件]→[打开]→[库...]命令，弹出“打开库”对话框，如图 12-1 所示。



图 12-1 打开库文件对话框

在该对话框中，选择要打开的库文件，然后单击“打开”按钮打开该库文件，如图 12-2 所示。



图 12-2 打开的库文件窗口

12.2 库的使用

库是一个外部文件，该文件中包含有图形、文本、图像、声音、数字电影等多媒体资源，可以存放【显示】图标、【交互】图标、【计算】图标、【声音】图标和【数字电影】图标等五种图标中的信息。

创建一个新库，然后对新库进行编辑，最后再将库进行保存。如果有某个应用程序要使用该库中的内容，只要打开该库文件，然后在库和应用程序之间进行操作就可以了。

12.2.1 创建一个新库

要创建一个新库，必须首先要创建一个新的库文件。

选择菜单[文件]→[新建]→[库]命令，或直接按下快捷键<Ctrl>+<Alt>+N，弹出新建库窗口，如图 12-3 所示。

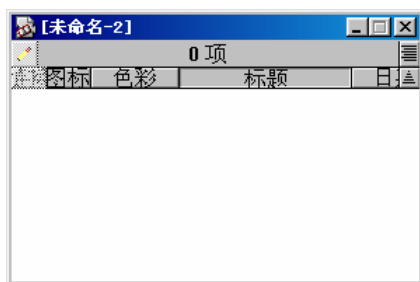


图 12-3 新建库窗口

此时库窗口中，库窗口中的读/写按钮、“连接”按钮都是灰显的，同时标题栏下方的图标个数为“0 项”，表示此时库窗口中还没有图标。

如果要为库窗口中添加图标，可以将应用程序设计窗口的流程线上的图标直接拖动到库中。当将图标拖动到库窗口中后，在库窗口中该图标的名称与拖动的图标名称相同，同时，左边有链接标记，表示库窗口的图标与应用程序设计窗口中的对应图标建立了链接，而且，应用程序设计窗口的流程线上的对应图标的标题变成了斜体，表示该图标是库窗口中图标的一个映像副本，也就是库窗口中的一个链接图标，而该图标的原型则被移动到了库窗口中，如图 12-4 所示。

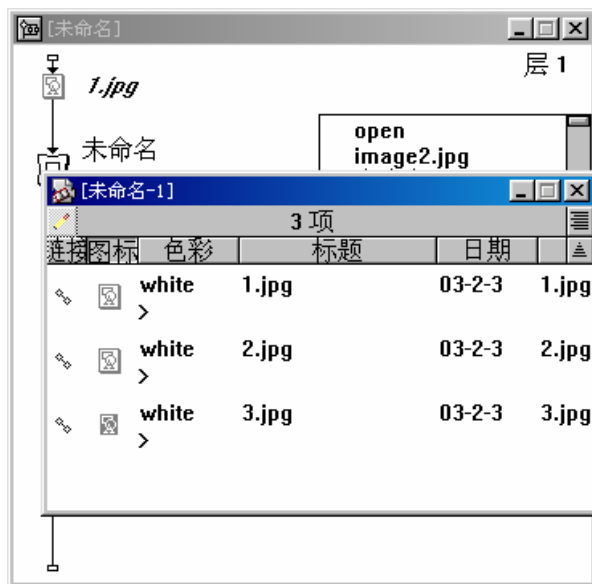


图 12-4 为库窗口中添加图标

当库创建好以后，可以菜单[文件]→[保存]命令，弹出保存库文件的对话框，如图 12-5 所示。



图 12-5 保存库文件对话框

在该对话框中选择要保存库文件的位置，然后在“文件名”文本框中输入库文件名，单击“保存”按钮即可将该库文件保存。库文件的默认扩展名为“.a6l”。

1. 删除库窗口中的图标

删除库窗口中图标的步骤一般为：

- (1) 打开库窗口。
- (2) 在库窗口中选择要删除的图标。
- (3) 单击工具栏中的“剪切”按钮或直接按下键盘中的<Delete>键，将选中的图标删除。
- (4) 如果选中要删除的图标中包含有与应用程序有链接关系的图标，则 Authorware 会弹出一个提示对话框，告诉用户该操作会断开链接关系，如图 12-6 所示。

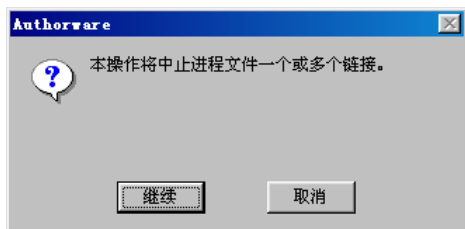


图 12-6 断开链接提示对话框

- (5) 如果单击“Continue”按钮，将会将选中的所有图标删除。
- (6) 如果单击“取消”按钮，将取消删除操作。

当我们剪切掉一个与应用程序有链接关系的图标，应用程序中对应的图标左边会出现断开链接标记；如果我们将剪切掉的与应用程序中有链接关系的图标重新粘贴到原库中，Authorware 发现是原来被剪切掉的，会弹出一个提示对话框，询问用户是否恢复图标之间的链接关系，如图 12-7 所示。

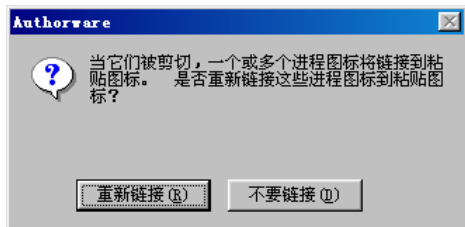


图 12-7 是否恢复链接关系对话框

当单击“重新连接”按钮时，该按钮将会重新粘贴到原库中，并且也恢复了原来的链接关系。此时

应用程序中与该图标链接的图标左边的断开链接标记消失了。

当单击“不要连接”按钮时，将该图标重新粘贴到原库中，但不恢复原来的链接关系。此时此时应用程序中与该图标链接的图标左边的断开链接标记依然存在。

12.2.2 建立链接和断开链接

1. 建立链接

将库窗口中的图标与应用程序设计窗口中的图标建立链接，一般来说有三种方法：

- 拖动应用程序设计窗口中的图标到库窗口中

按照上面为库窗口添加图标的方法，将应用程序设计窗口中的图标拖动到库窗口中，则 Authorware 会自动将库窗口中的该图标与原图标建立链接。

- 从库窗口中将图标拖动到应用程序的设计窗口中

这种方法使用的比较多。在库窗口中将一个图标进行编辑后，然后将它拖动到应用程序设计窗口流程线上的合适位置，Authorware 会自动将新拖入的图标作为链接图标。此时可以对链接图标的其他特性进行设置。

- 从一个库窗口中拖动一个具有链接关系的图标到另一个库窗口中。

按照上面为库窗口添加图标的方法，将两个库都打开，同时显示在屏幕上，然后从从一个库窗口中拖动一个具有链接关系的图标到另一个库窗口中，此时系统会弹出一个提示对话框，询问用户是否维持链接关系，如图 12-7 所示，单击“重新连接”按钮确定保持链接关系。

当然，还有一些其他方法，比如使用剪切、复制和粘贴命令等，这就不一一赘述了。

2. 断开链接

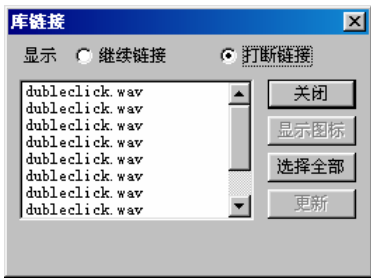
要断开库窗口中的图标与应用程序设计窗口中的链接图标之间的链接，一般只有通过将库窗口中的图标进行删除或剪切，然后在弹出的系统提示对话框中确定断开链接；或者通过将设计窗口中的链接图标进行删除或剪切即可。

3. 使用库链接对话框

选择菜单[Xtras]→[库连接]命令，打开库链接对话框，如图 12-8 所示。



(a) 未断开链接的图标列表



(b) 断开链接的图标列表

图 12-8 库链接对话框

- 在库链接对话框中，“显示”有两个选项：
 - “继续链接”单选钮：选择该单选钮，下面的图标列表中列出的图标是未断开链接的图标。
 - “打断链接”单选钮：选择该单选钮，下面的图标列表中列出的图标是断开链接的图标。
- “关闭”按钮：用于关闭该对话框。
- “显示图标”按钮：当在图标列表中只选择一个图标时，该按钮被激活，此时可以单击该按钮，

查看应用程序中选中标在流程线上的位置。

- “选择全部”按钮，单击该按钮，选中图标列表中所有的图标。
- “更新”按钮：该按钮是用于更新链接用的。在图标列表中按照上面提供的四种方法选择图标，然后单击该按钮，系统也会弹出图 12-9 所示的提示对话框，单击提示对话框中的“更新”按钮，将更新链接；如果单击“取消”按钮，将取消更新。需要注意的是，在断开链接的图标列表中，无法对其中的图标进行更新，此时“更新”按钮灰显。

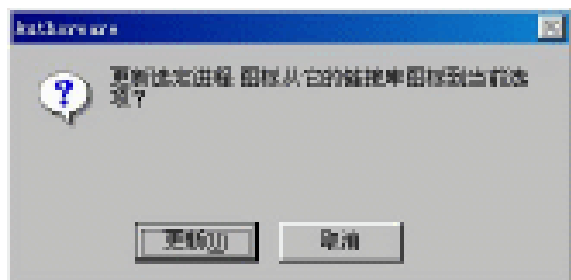


图 12-9

第13章 变量、表达式、函数的使用

- 变量简介
- 变量类型
- 系统变量和自定义变量
- 表达式的使用
- 函数简介
- 系统函数
- 应用程序中使用系统函数
- 常用系统函数介绍

13.1 变量简介

在高级程序语言中，经常会用到一些变量和函数。Authorware 也提供了许多系统变量和系统函数，同时，用户自己也可以定义自己的变量和函数，这就为设计一些程序设计提供了很大的灵活性。

Authorware 提供的系统变量，可以跟踪应用程序中运行的各种信息，如某个图标的状态、用户所做的动作、响应触发的条件等，它们的值在程序运行过程中自动由 Authorware 更新。

在 Authorware 中用户可以定义自己的变量，可以在程序运行时更改变量的数值，以跟踪用户希望得到的信息或来触发一定的响应条件。

Authorware 6 一共提供了 224 个系统变量。关于每个变量的含义，请参看本书后面的附录。

13.1.1 变量的类型

变量是其值可以改变的量。一个自定义变量名必须是惟一的，既不能与任何系统变量重名，也不能与任何已定义的自定义变量重名。

Authorware 规定，每一个变量名必须以一个字母开头，中间可以是字母、数字、下划线和空格的任意组合。变量名中不区分大小写，但有无空格则是不同的，例如，变量 Score 和 score 是相同的变量，而 Score1 和 Score 1 则是不同的变量。变量名最多可以包括 40 个字符长。

在一般的高级程序语言中，变量都包括很多中数据类型，以 C 语言为例，变量包括整型、浮点型、字符型等，每一种类型又包括几种，如整型可以分为：基本型、短整型、长整型和无符号型四种。又根据变量的使用范围，可以将变量分为全局变量和局部变量等。

Authorware 中的变量都是全局变量，而且没有区分变量是整型还是实型，统一定义为数值型变量。

在 Authorware 中，变量的类型一般可以分为七种：

1. 数值型

数值型变量是用来存储数值的。数值可以是一个整型（如 10）、实型（如 0.001）、或负数（如 -10、-10.05）。

在 Authorware 中，变量的数值范围可以从 -1.7×10^{308} ~ 1.7×10^{308} 。

当将两个变量进行加减时，Authorware 会将它们当做数值型变量，因为只有数值型变量才能使用算术运算符。数值型变量可以存储用户的分数、或一个算术表达式计算的结果。

2. 字符型

字符型变量用来存储字符串。字符串是一个或多个字符组成的字符序列，包括字母序列（如“Macromedia Authorware”）、数字序列（如“123456”）、特殊字符（如“*&/= ”）、以及它们的组合（如“Total is \$5.00”）。

一个字符型变量最多可以包含 30,000 个字符。

字符型变量可以存储一个用户名、一个字符序列、或一个网址的 URL 地址等。当使用连接运算符 ^ 将两个变量进行连接时，Authorware 会将这两个变量当作字符型变量，因为只有连接运算符 ^ 只使用于字符型变量。

3. 逻辑型

逻辑性变量用来存储 True（真）或 False（假），也就是说，逻辑性变量只有两种值：True 或 False。逻辑性变量相当于一个电灯开关，只能在两种状态中间转换：开和关。数字 0 认为是 False，而其他不等于 0 的数字（通常是 1）认为是 True。

当将一个变量放置在一个 Authorware 需要一个逻辑性变量的地方（如在响应类型属性对话框“交互”标签页的“激活条件”文本框中）时，Authorware 认为该变量是一个逻辑性变量。如果该变量中存储了一个数值，只要该数值不为 0，Authorware 认为该变量为 True。如果该变量存储的是一个字符串，只有该字符串为“True”、“T”、“Yes”、或“On”（大小写不重要）时，该变量才为 True，否则就为 False。

4. 列表型

列表型变量用来存储一组变量值或一组常量的值。

Authorware 支持下面的两种列表类型：

- 线型列表。在线型列表中，表中的每一个元素都是一个单值，例如，[1, 2, 3, "a", "b", "c"]。
- 属性列表。在属性列表中，表中的每一个元素都是一个属性和由冒号“：”分隔开的属性值组成，例如，[#Firstname:"Jack", #Lastname:"Smith", #Photo:5551212]。

在这两种列表中，元素可以是未排序的，也可以是按照字母顺序进行排序的。



在跟踪和更新数据数组时列表型变量是特别有用的。

5 . 符号型

符号型变量是这样一种由井号开头、后面可以是一个字符串数据类型或其他数值的数据类型。由于 Authorware 处理符号型变量比字符型变量要快，所以符号型变量还是比较有用的。

例如，在语句 MyVariable:=#StringLen 中符号型变量#StringLen 比语句 MyVariable:="StringLen"中的字符串 " StringLen " 运行要快。

赋予了符号的变量与其他变量起的作用不同。例如，运行 MyVariable:=#StringLen 语句后，由于 #StringLen 是变量 MyVariable 的值，所以可以将变量 MyVariable 嵌入到一个【显示】图标中以显示值 #StringLen。但是，如果运行 MyVariable + 1 语句将会返回一个数值，因为 Authorware 会将用运算符连接的变量当作是一个数值型变量，而将用户名当作是一个数值。

6 . 坐标变量

一个 Point 变量是由 Authorware 系统函数 Point 返回的数据类型，在设置点位置时是十分有用的。

用法：MyPoint:=Point(x , y)

含义：在屏幕坐标 (x , y) 处创建一个点。

7 . 矩形变量

一个 Rect 变量是通过 Authorware 系统函数 Rect 返回的数据类型。Rect 变量在定义矩形区域时是非常有用的。

用法：MyRect:=Rect(Left , Top , Right , Bottom)

或 MyRect:=Rect(Point1 , Point2)

含义：利用指定的值或点绘制矩形，其矩形和点限制为列表类型。

13.1.2 变量的赋值

变量的赋值方法一般包括以下两种：

- 在通过新建变量对话框来定义一个新的自定义变量时，在对话框中直接输入新建变量的初始值。

单击变量窗口中的“新建”按钮，可以打开新建变量对话框，如图 13-1 所示。

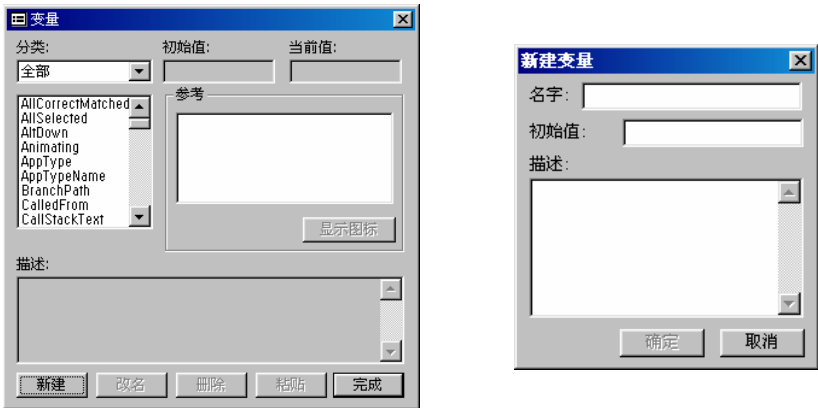


图 13-1 新建变量对话框

在“名字”文本框中输入新建变量名，然后在“初始值”文本框中输入该变量的初始值。

- 在计算编辑窗口中输入赋值语句

在 Authorware 中，赋值语句的格式为：

变量:=变量值，其中“:=”是赋值符号。

在计算编辑窗口中，用户也可以直接输入：变量=变量值，Authorware 自动检测输入语句，当发现语句中缺少了一个“:”号，它会自动将“:”添加上。

使用赋值语句，可以是直接将一个常量赋值给一个变量，可以将一个变量的值赋值给另一个变量，可以将一个表达式的值赋值给一个变量，也可以将一个有返回值的函数赋值给一个变量，如图 13-2 所示。

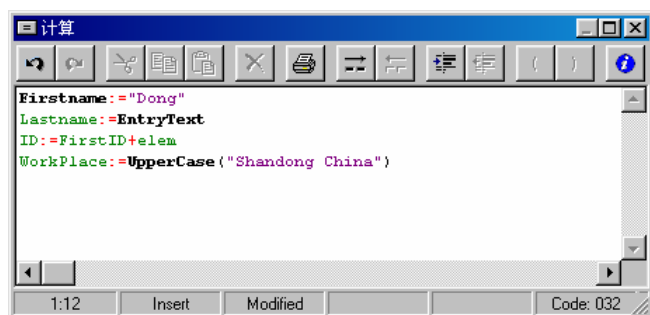


图 13-2 在计算编辑窗口中使用赋值语句

13.2 系统变量

Authorware 6 一共提供了 224 个系统变量，可分为 11 种类型：

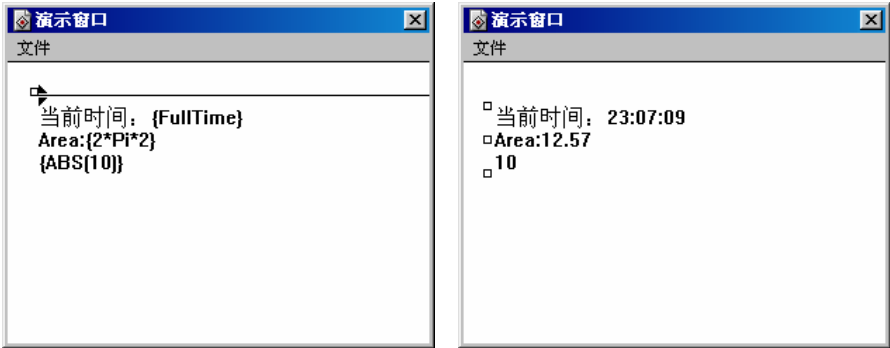
- CMI：计算机管理教学变量
- Decision：【决策】图标变量
- File：文件管理变量
- Framework：【框架】图标变量
- General：通用变量
- Graphics：图形控制变量
- Icon：图标控制变量
- Interaction：【交互】图标变量
- Network：网络控制变量
- Time：时间控制变量
- Video：视频控制变量

Authorware 提供的系统变量不但可以在【计算】图标中使用，也可以在赋予了计算功能的其他图标中使用。可以在展示窗口中将变量、表达式、或者函数的值显示出来，同时还可以将系统变量同【交互】图标、【决策】图标、【框架】图标等配合使用，实现用户信息的跟踪、学习成绩控制等功能。

13.2.1 在应用程序中显示变量的值

在应用程序中，我们可以对用户信息进行跟踪、分析、汇总，然后对数据进行评价和操作。

因为 Authorware 支持直接将变量、表达式、或函数直接写到文本中，可以直接在展示窗口中显示变量的当前值或表达式和函数的返回值。为了区分文本与变量、表达式和函数，Authorware 规定在要显示结果的变量、表达式和函数两边用大括号“{”、“}”括起，如：{User_Name}、{User_ID}等，如图 13-3 所示。



(a) 展示窗口中输入变量、表达式和函数 (b) 运行时的显示结果

图 13-3 显示变量、表达式和函数的值

上式中，Fulltime 是一个系统变量，显示当前系统时间；Pi 也是一个系统变量，其值是圆周率（3.1415926536...）；ABS(x)是一个系统函数，该函数返回参数 x 的绝对值。

如果要不断更新变量，可以打开【显示】图标属性对话框，在“显示”标签页中的“更新变量显示”复选框选中即可，如图 13-4 所示。

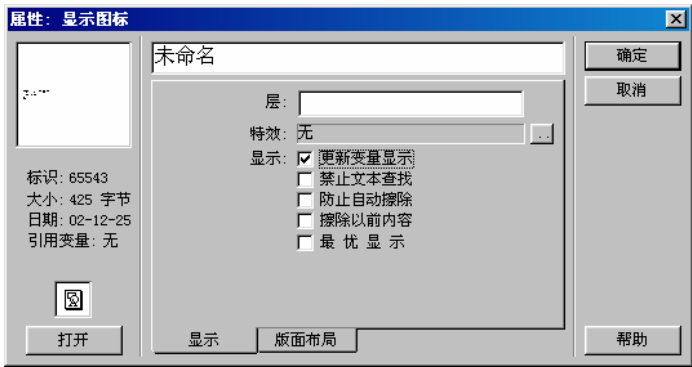


图 13-4 选中“更新变量显示”复选框

这样变量就可以根据程序的运行而不断更新，如变量 FullTime 的显示值会随着时间的变化而不断更改。

13.2.2 常用系统变量介绍

下面介绍一些常用的系统变量。

1. AllCorrectMatched

类型：逻辑型

说明：当前【交互】图标中所有标记为“正确”的响应全部被用户的响应所匹配了，则该变量的值为 True。

可以使用 AllCorrectMatched @“IconTitle”可以获得“IconTitle”指定的【交互】图标的正确响应匹配情况。

2. AllSelected

类型：逻辑型

说明：当当前【决策】图标中的所有分支都被选择过了，该变量为 True。

可以使用 AllSelected@“IconTitle”来返回“IconTitle”指定的【决策】图标的分支选择情况。

3. ClickX 和 ClickY

类型：数值型

说明：这两个变量返回用户最后一次按下鼠标时，鼠标距展示窗口左边界和上边界的像素点，也就是单击时鼠标在展示窗口中的位置坐标。

4. CursorX 和 CursorY

类型：数值型

说明：这两个变量返回当前用户鼠标距展示窗口左边界和上边界的像素点，也就是当前鼠标在展示窗口中的位置坐标。

5. CurrentPageNum

类型：数值型

说明：该变量存储了当前【框架】图标中最近一次显示的页面的页号。如果当前【框架】图标没有页面被显示，则该变量值为 0。

可以使用 CurrentPageNum@“IconTitle”来返回“IconTitle”指定的【框架】图标中最后一次显示的页面页号。

Authorware 对【框架】图标下的下挂页面从左到右分别编号为 1, 2, 3, ...。

6. DisplayX 和 DisplayY

类型：数值型

说明：这两个变量返回一个显示对象的中心在展示窗口中的位置坐标。

可以使用 DisplayX@“IconTitle”和 DisplayY@“IconTitle”来返回“IconTitle”指定的图标显示对象的坐标。

7. EntryText

类型：字符型

说明：该变量存储了用户最近一次在输入文本响应时所输入的文本。

可以使用 EntryText @“IconTitle”来返回“IconTitle”指定的【交互】图标中用户最近一次在输入文本响应时所输入的文本。

当属于指定【交互】图标的任意一个交互项被触发时，Authorware 就设置该变量的值。即使用户没有触发文本输入响应（如按下了 Enter 键），Authorware 仍然更改该变量的值。这样，您就可以在用户输入完文本之前对用户输入的文本进行判断和操作。

若用户触发的是一个设定为“Return”永久响应的分支类型，则该变量的值不变。若将分支类型设定为“Continue”的输入文本响应的 EntryText 变量值进行了修改，或赋予了一个新值，则更改了的变量值会被后面的响应使用。

8. Key

类型：字符型

说明：该变量存储了用户最后一次按下的键的键名，如 a, D, Alt 等

9. Movable

类型：逻辑型

说明：该变量常用作 Movable@“IconTitle”，来指定“IconTitle”指定的图标显示对象是否可以移动。也可以使用该变量来判断指定的图标显示对象是否可移动，如果可以移动，则返回 True，否则返回 False。

10. NumEntry

类型：数值型



说明：该变量存储了用户最近一次在文本输入框中输入的第一个数值。

可以使用 NumEntry@“IconTitle” 来返回 “IconTitle ” 指定的【交互】图标中文本域中输入的第一个数值。

该变量的值是从变量 EntryText 中获得的。Authorware 将获得的头三个数值分别赋给变量 NumEntry , NumEntry2 和 NumEntry3。

当 Authorware 遇到了非数字字符，则认为是一个数值的结束，将第一个数值赋给变量 NumEntry；若再遇到一个新的数字，Authorware 则认为是一个新的数值开始，并将其赋给变量 NumEntry2，依次类推。

如果在数字前面紧跟了一个减号“-”，Authorware 自动将该数值设成负数。

相关变量：NumEntry2，NumEntry3，EntryText

11 . PathSelected

类型：数值型

说明：该变量存储当前【决策】图标中用户最后一次选择的分支编号。

可以使用 PathSelected@“IconTitle” 来返回 “IconTitle ” 指定的【决策】图标中用户最后一次选择的分支编号。

Authorware 对【决策】图标下的下挂分支从左到右分别编号为 1，2，3，...。

12 . TimeRemaining

类型：数值型

说明：该变量指定了当前【交互】图标中时间限制响应中的剩余时间，或者【决策】图标、【等待】图标的剩余时间，单位是秒。

可以使用 TimeRemaining@“IconTitle” 来返回 “IconTitle ” 指定的图标中的剩余时间。

若指定的是【交互】图标，而此【交互】图标中有多个时间限制响应，则该变量知识时间限制最短的响应的剩余时间。

13 . TimesMatched

类型：数值型

说明：该变量记录了当前【交互】图标中用户已经做出的匹配次数。

可以使用 TimesMatched@“IconTitle” 来返回 “IconTitle ” 指定的【交互】图标中用户的匹配次数。

14 . Tries

类型：数值型

说明：该变量记录了当前交互中用户尝试匹配响应的次数。

可以使用 Tries @“IconTitle” 来返回 “IconTitle ” 指定的【交互】图标中用户的尝试次数。

13.3 自定义变量

13.3.1 定义自定义变量

在 Authorware 中，定义自定义变量一般来说有两种方法：

1 . 使用变量窗口

如果想要创建一个自定义变量，可以在变量窗口中进行定义。这种定义方法是先定义变量后使用。

打开变量窗口，如图 13-1 所示，要创建新建变量，可以单击窗口中的“新建”按钮，打开新建变量对话框，如图 13-5 所示。

现在对新建变量对话框中的各个选项进行简单介绍：

- “名字”文本框：该文本框中是用来输入新建变量名的。在该文本框中如果直接输入中文，单击“确定”后，系统会弹出一个提示对话框，如图 13-6 所示，告诉用户变量名必须一字母开头，而且只能包含字母、数字、空格和下划线等。



图 13-5 新建变量对话框

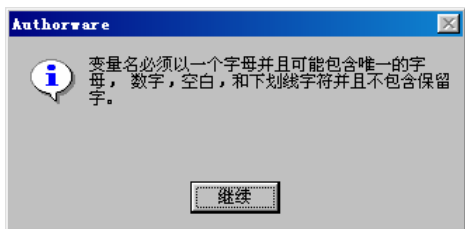


图 13-6 提示对话框

也就是说，Authorware 不认识中文式的变量。如果输入一个字母开头，后面跟有中文，如“Str 学生名”，Authorware 会忽略中间的中文，将变量认为是 Str。

- “初始值”文本框：在该文本框中输入定义好的新变量的初始值。如果在该文本框中输入一个数字，则 Authorware 认为新变量是一个数值型变量；如果输入一个字符串，Authorware 认为新变量是一个字符型变量；如果输入 True 或 False，Authorware 认为新变量是一个逻辑型变量等。
- “描述”文本框：该文本框是用来输入新变量的描述信息的。当在该文本框中输入描述信息后，在变量窗口中选择该变量，则变量窗口下面的变量描述框中就会显示出此时输入的描述信息。

2. 直接在需要新变量的地方输入

该方法是先使用变量，然后再定义。

首先在需要变量的地方（如计算编辑窗口中、一些属性对话框中等）直接使用新变量名，当关闭窗口或对话框时，Authorware 会自动检测用户的输入，当它发现不能识别的名称，它会认为该名称是用户自定义的新变量，此时会弹出新建变量对话框，在对话框的“名字”文本框自动添加了变量名，如图 13-7 所示。



图 13-7 Authorware 自动识别新变量并将其自动添加到新建变量对话框中

13.4 表达式的使用

在 Authorware 中，不但要使用变量，有时候还需要使用表达式，来达到一定的效果。

13.4.1 运算符

Authorware 一共提供了五种运算符：

1. 赋值运算符 “:=”

赋值运算符是将运算符右边的值赋给左边的变量。这与其他高级程序语言中的赋值语句相同。

如：StudentCount:=100

StudentName:= “张三”

StudentNum:=StudentNum+1

在程序设计时，如果将赋值语句中的赋值运算符写作了 “=”，Authorware 会自动将其变为 “:=”。

2. 算术运算符

使用算术运算符可以进行算术运算，结果为数值型数值。

算术运算符包括下面几个：

+: 将运算符两边的数值进行相加。如：R:=5+2 R 的值变为 7。如果是一目运算符，则表示为正号，如：R:=+5 R 的值为 5。

-: 将运算符左边的数值减去右边的数值。如：R:=5-2 R 的值变为 3。如果是一目运算符，则表示为负号，如：R:=-5 R 的值为-5。

*: 将运算符两边的数值进行相乘。如：R:=5*2 R 的值变为 10。

/: 将运算符左边的数值除以右边的数值。如：R:=5/2 R 的值变为 2.5。

: 将运算符右边的数值当做左边数值的指数。如：R:=52 R 的值变为 25。

可以将算术运算符混合使用。

3. 关系运算符

使用关系运算符可以进行关系运算，结果为逻辑值：True 或 False。

关系运算符包括下面几个：

=: 判断运算符两端的数值是否相等，如果相等则返回 True，否则返回 False。如：Con:=(StudentNum=100) （假设 StudentNum 的值为 101）返回 False。

<>: 判断运算符两端的数值是否不相等，如果不相等则返回 True，否则返回 False。如：Con:=(StudentNum<>100) 返回 True。

>: 判断运算符右边的数值是否大于左边的数值，如果大于则返回 True，否则返回 False。如：Con:=(StudentNum>100) 返回 True。

>=: 判断运算符右边的数值是否大于或等于左边的数值，如果大于或等于则返回 True，否则返回 False。如：Con:=(StudentNum>=100) 返回 True。

<: 判断运算符右边的数值是否小于左边的数值，如果小于则返回 True，否则返回 False。如：Con:=(StudentNum<100) 返回 False。

<=: 判断运算符右边的数值是否小于或等于左边的数值，如果小于或等于则返回 True，否则返回 False。如：Con:=(StudentNum<=100) 返回 False。

4. 逻辑运算符

使用逻辑运算符可以进行逻辑运算，结果为逻辑值：True 或 False。

逻辑运算符包括下面几个：

~: 逻辑非运算。该运算是一个一目运算符，其结果是运算符右边的值取反，即将 True 取反后变为



False，而将 False 取反后变为 True。逻辑非的运算规则见表 13-1 所示。

表13-1 逻辑非运算规则

FLAGS	~FLAGS
TRUE	FALSE
FALSE	TRUE

&: 逻辑与运算。它是将运算符两边的逻辑值进行与运算。逻辑与的运算规则见表 13-2 所示。

表13-2 逻辑与运算规则

FLAGS1	FLAGS2	FLAGS1 & FLAGS2
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

|: 逻辑或运算。它是将运算符两边的逻辑值进行或运算。逻辑或的运算规则见表 13-3 所示。

表13-3 逻辑或运算规则

FLAGS1	FLAGS2	FLAGS1 FLAGS2
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

5．连接运算符 “ ^ ”

连接运算符 “ ^ ” 是将两个字符串连接成一个新的字符串。

如：设 String1:= “ Macromedia ”，String2:= “ Authorware ”

则：String3:=String1^ “ ” ^String2，String3 的值为 “ Macromedia Authorware ”。

6．复合条件

当输入由关系运算符和逻辑运算符组成的复合条件时，我们可以使用一些速记格式。

例如：当 Flags 的值在 1 到 5 之间，下面的条件语句为 True。

长格式：Flags > 1 & Flags < 5

短格式：Flags >1 & <5

又如：当变量 Name 的值为 “ Jim ”、 “ james ” 或 “ James ” 中的一个，下面的条件语句为 True。

长格式：Name= “ Jim ” |Name= “ james ” |Name= “ James ”

短格式：Name= “ Jim ” | = “ james ” | = “ James ”

在一个表达式中经常会用到很多运算符，如果都使用括号将希望首先运算的括起，当然没有什么差错。但是，如果表达式很复杂，就会使用很多的括号。

Authorware 规定了运算符的优先级和结合性。在表达式求值时，先按运算符的优先级高低次序执行。如果在一个运算对象两侧的运算符的优先级相同，则将按规定的 “ 结合方向 ” 处理。Authorware 规定了各种运算符的结合方向（结合性）。算术运算符的结合方向为 “ 自左向右 ”，即先左后右（左结合性）。有些运算符的结合方向为 “ 自右向左 ”，即右结合性（如：**，赋值运算符等）。

运算符的优先级可见表 13-4 所示。

表13-4 运算符的优先级

优先级	运算符					
由高到低	()					
	~	+(正号)	-(负号)			
	**					
	*	/				
	+	-				
	^					
	=	<>	>	>=	<	<=
	&					
	:=					

13.5 函数简介

变量和函数的使用，是 Authorware 实现复杂多媒体程序的必备条件。函数可以完成一些特定的操作。这操作可以简单得只是将文本的每一行中所有单词的头一个字母转换成大写字母，也可以是创建一个数组这么复杂。

函数有两种类型：系统函数和自定义函数。系统函数是由 Authorware 提供给用户使用的；而自定义函数则是用户根据需要自己定义的，它可以包括 Xtras 和 DLLs。Macromedia 为 Authorware 提供了一小部分的自定义函数，大部分函数可以从其他厂商中直接使用。

系统函数和自定义函数的惟一真正的不同也就是它们的由来不同，但是它们却有着相同的目的：增加 Authorware 的灵活性和扩展作品使用的范围。

13.5.1 函数窗口

单击 Authorware 的工具栏中的 “函数窗口”命令按钮，或者选择菜单[窗口]▶[Functions]命令，可以打开变量窗口，如图 13-8 所示。

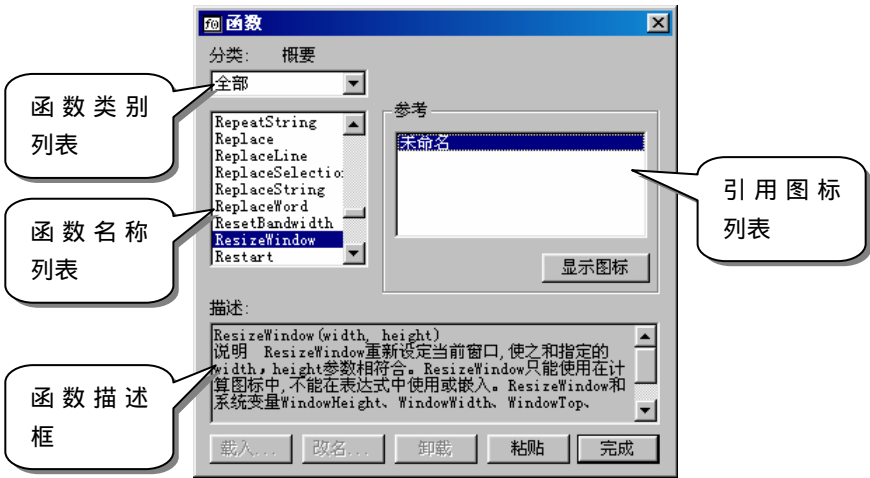


图 13-8 函数窗口

在函数窗口中，显示出了 Authorware 提供的所有系统函数和用户的自定义函数。

13.6 系统函数

Authorware 6 一共提供了 311 个系统函数，可分为 22 种类型：

- Character：字符管理函数
- CMI：计算机管理教学函数
- File：文件管理函数
- Framework：【框架】图标函数
- General：通用函数
- Graphics：绘制图形函数
- Icons：图标操作函数
- Jump：文件跳转函数
- Language：编程语言函数
- List：列表函数
- Math：数学运算函数
- Network：网络操作函数
- OLE：OLE 控制函数
- Platform：开发平台函数
- Target：目标类型函数
- Time：时间控制变量
- Video：视频控制变量
- Xtra ActiveX：ActiveX 外挂函数
- Xtra fileio：Xtra 文件输入输出函数
- Xtra Mui：Mui 外挂函数
- Xtra PWInt：PWInt 外挂函数
- Xtra QuicktimeSupport：支持 QuickTime 外挂函数
- Xtra SecurityInstallation：安全设置外挂函数

Authorware 提供的系统函数不但可以在【计算】图标中使用，也可以在赋予了计算功能的其他图标中使用。可以在展示窗口中将变量、表达式、或者函数的值显示出来，同时还可以将系统函数同【交互】图标、【决策】图标、【框架】图标等配合使用，实现用户信息的跟踪、学习成绩控制等功能。

各个系统函数的使用方法，请参考附录。

13.6.1 应用程序中使用系统函数

如同使用系统变量一样，要在应用程序中使用系统函数，可以使用下面的方法：

1. 直接在计算编辑窗口或属性对话框中输入

如果某个【计算】图标或赋予了计算功能的其他图标中需要输入语句，可以打开计算编辑窗口，然后直接在计算编辑窗口中输入语句即可。

例如，我们希望将退出应用程序，可以使用函数 Quit()，则在计算编辑窗口中输入“Quit()”即可，如图 13-9 所示。

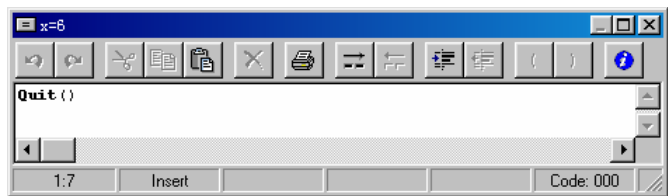


图 13-9 在计算编辑窗口中直接输入函数

当然我们也可以在一些图标属性对话框中输入函数。

2. 使用函数窗口

我们可以使用函数窗口，在窗口中选择要添加的函数，然后将它粘贴到合适位置即可。

使用函数窗口将函数粘贴到需要位置的操作步骤一般为：

(1) 将光标定位于需要输入函数的位置，如计算编辑窗口、属性对话框的相应文本框中等。

(2) 单击 Authorware 的工具栏中的  “函数窗口”命令按钮，或者选择菜单[窗口]→[函数]命令，可以打开变量窗口。

(3) 在函数窗口中首先在“分类”下拉框中选择目标函数所属的类别。

(4) 在下面的函数列表中选择目标函数。如果不知道该目标函数所属的类别，则直接在“全部”选项中查找。

(5) 可以直接双击该目标函数，将函数粘贴到光标定位的位置。

(6) 也可以单击函数窗口中的“粘贴”按钮将函数粘贴到光标定位的位置。

(7) 如果需要粘贴多个函数，则可以再将光标定位到其他需要函数的位置，然后按照上面的步骤操作。

(8) 当完成所有操作后，单击函数窗口中的“完成”按钮关闭函数窗口。

13.6.2 常用系统函数介绍

下面介绍一些常用的系统函数。

1. AppendExtFile

语法：Number := AppendExtFile (“Filename”, “String”)

说明：该函数是将字符串 “String” 添加到指定文本文件 Filename 的尾部。如果指定的文本文件不存在，则该函数会创建该文件，然后再将字符串 “String” 添加到该文件的尾部。如果没有为指定文件指定路径，Authorware 会使用变量 FileLocation 作为路径。

当使用 AppendExtFile 函数时，Authorware 会更新两个系统变量：IOStatus 和 IOMessage，并且返回 IOStatus 的值。如果无错误，则 IOStatus 为 0，IOMessage 为空。如果有错误发生，操作系统将错误号赋给 IOStatus。只要 IOStatus 不为 0，表明出错，IOMessage 中包含的将是出错信息。

需要注意的是，AppendExtFile 函数是将文本添加到一个文件中，使用 WriteExtFile 函数是将文本覆盖文件的内容，而使用 ReadExtFile 函数则可以读取文件的内容。

当 Authorware 的 Web Player 运行于非信任模式下，该函数无效。

示例：

AppendExtFile (RecordsLocation ^ “DATA.TXT”, NewUser)

将变量 NewUser 中的内容添加到文件 DATA.TXT 尾部。

相关变量：IOMessage，FileLocation，RecordsLocation

相关函数：ReadExtFile，WriteExtFile

2 . DeleteLine

语法：Result := DeleteLine (“String”, n [, m, delim])

说明：该函数是将字符串 “String” 中的第 n 行删除后，将剩余部分赋值给变量 Result。如果也定义了 m 值，则该函数是将字符串 “String” 中的第 n 行到第 m 行全部删除，然后将剩余部分赋值给变量 Result。若同时也定义了 delim，则将各行的分隔符设定为 delim 指定的值，如 Tab 符。系统缺省的行分隔符为回车符（Return）。

示例：

设 Lines := “Macromedia Authorware” ^ Return ^ “is a” ^ Return ^ “media software.” ^ Return。

Result := DeleteLine (Lines, 2)。

Result 的值变为 Result = “Macromedia Authorware\rmedia software\r”。

设 Lines := “Macromedia Authorware” ^ Tab ^ “is a” ^ Tab ^ “media” ^ Tab ^ “software.” ^ Tab。

Result := DeleteLine (Lines, 2, 3, Tab)。

Result 的值变为 Result = “Macromedia Authorware\t software\t”。

3 . DisplayIcon

语法：DisplayIcon (IconID@“IconTitle”)

说明：该函数是用于显示图标标题为 “IconTitle” 的图标的内容。如果该图标已经被显示了，则会更新所有已显示的变量。

该函数只能用到【计算】图标中。

示例：

DisplayIcon (IconID@“背景”)

显示标题为 “背景” 图标的内容。

4 . EraseIcon

语法：EraseIcon (IconID@“IconTitle”)

说明：该函数是用于擦除图标标题为 “IconTitle” 的图标的内容。

该函数只能用到【计算】图标中。

示例：

EraseIcon (IconID@“背景”)

擦除标题为 “背景” 图标的内容。

5 . Exit Repeat

语法：Exit Repeat

说明：该函数用于退出循环体而执行下面图标的内容。可以将该函数放在循环体中的任意位置。

6 . Find

语法：Number := Find (“Pattern”, “String”)

说明：该函数用于从指定的字符串 “String” 中查找子字符串 “Pattern” 的位置。如果查找到，则返回第一个匹配的字符位置；如果没有找到则返回 0。

该函数是区分大小写的。如果不想区分大小写，可以首先使用函数 LowerCase 或 UpperCase 将字符串全部变成小写或大写。

在使用 Find 函数时，可以使用一些特殊符号来扩大查找范围：

*：代表 0 个或多个字符

?：代表任何一个字符

\：查找特殊符号。如：使用 “\?” 表示查找字符 “?”。

注意：Authorware 将每一个空格和标点符号都当作是一个字符。

示例：

Location := Find (“media”, “Macromedia Authorware”)

返回 Location=6。

7 . GetLine

语法：Result := GetLine (“String”, n [, m, delim])

说明：该函数是将字符串 “String” 中的第 n 行赋值给变量 Result。如果也定义了 m 值，则该函数是将字符串 “String” 中的第 n 行到第 m 行全部赋值给变量 Result。若同时也定义了 delim，则将各行的分隔符设定为 delim 指定的值，如 Tab 符。系统缺省的行分隔符为回车符（Return）。

示例：

设 Lines := “Macromedia Authorware” ^ Return ^ “is a” ^ Return ^ “media software.” ^ Return。

Result := GetLine (Lines, 2)。

Result 的值变为 Result = “is a”。

设 Lines := “Macromedia Authorware” ^ Tab ^ “is a” ^ Tab ^ “media” ^ Tab ^ “ software.” ^ Tab。

Result := GetLine (Lines, 2, 3, Tab)。

Result 的值变为 Result = “is a\tmedia”。

8 . GoTo

语法：GoTo (IconID@“IconTitle”)

说明：该函数是使执行流程跳转到 “IconTitle” 指定的图标处运行。

示例：

GoTo (IconID@“背景”)

程序跳转到 “背景” 图标处运行。

9 . IconID

语法：Number := IconID (“IconTitle”)

说明：该函数返回由 “IconTitle” 指定的图标的 ID 号。

10 . If-Then

语法：If 条件 then 语句

或 If 条件 then
 语句

end if

或 If 条件 then 语句 1 else 语句 2

或 If 条件 then
 语句 1

else

 语句 2

end if

或 If 条件 1 then
 语句 1

else If 条件 2 then
 语句 2

else

 语句 3

end if

示例：

```
If Flags = True then
    GoTo (IconID@"风景 1")
else
    GoTo (IconID@"风景 2")
end if
```

11 . InsertLine

语法：Result := InsertLine (“String”, n, “NewString”[, delim])

说明：该函数是将字符串 “NewString” 插入到字符串 “String” 中的第 n 行位置，然后将结果返回给变量 Result。如果 n=0，则将 “NewString” 插入到字符串 “String” 的第 1 行前面；如果 n 的值大于了字符串 “String” 的总行数，则将 “NewString” 插入到字符串 “String” 的末尾和第 n 行之间。若同时也定义了 delim，则将各行的分隔符设定为 delim 指定的值，如 Tab 符。系统缺省的行分隔符为回车符（Return）。

示例：

```
设 String1 := “Monday” ^ Return ^ “Wednesday” ^ Return
String2 := “Tuesday”
String3 := InsertLine (String1, 2, String2)
返回 String3 = “Monday\rTuesday\rWednesday”
```

12 . JumpFile 和 JumpFileReturn

语法：JumpFile (“FileName”, [“Variable1, Variable2, ...”, “Folder”])

JumpFileReturn (“FileName”, [“Variable1, Variable2, ...”, “Folder”])

说明：函数 JumpFile 是使执行流程跳转到 “FileName” 指定的文件继续运行。函数 JumpFileReturn 则是使执行流程跳转到 “FileName” 指定的文件继续运行，运行该文件后，返回到原处继续执行初始文件。

使用这两个函数，Authorware 会自动搜索后缀名为 a5p 的文件，故指定文件时无需加入扩展名。同样，runa5w（打包文件）和 runa5m 文件自动搜索以 exe 或 a5r 为后缀的文件。

可以通过 Variable 参数在文件间传送参数，但必须保证两文件中有相同的变量。若传送多个参数，则参数间用分号分隔，且参数项需用引号引起来。还可使用*号代表具有相同子串的变量或所有变量。

若定义了 Folder 参数，则系统变量 RecordLocation 就会从缺省值变为定义好的 Folder 值，且这是惟一改变系统变量 RecordLocation 值的方法。

注意：如果在 Authorware 的 Web Player 运行状态下，还可以使用网络地址 URL 作为参数。

示例：

```
JumpFile (“history”, “subjects”)
JumpFileReturn (“history”, “”, FileLocation ^ “User Records”)
```

13 . JumpOut 或 JumpOutReturn

语法：JumpOut (“Program” [, “Document”])

JumpOutReturn (“Program” [, “Document”])

说明：函数 JumpOut 是用参数 “Program” 指定的应用程序打开文件 “Document”，同时退出 Authorware；函数 JumpOutReturn 是用参数 “Program” 指定的应用程序打开文件 “Document”，只是并不退出 Authorware。

这两个函数只能用在【计算】图标中。

示例：

JumpOutReturn ("Natepad.exe", "data.txt")

使用写字件打开文本文件 data.txt。

JumpOutReturn (" ", "data.txt")

使用系统默认的文本编辑程序打开文本文件 data.txt。

14 . MoveWindow

语法：MoveWindow (Top, Left)

说明：该函数是将当前窗口的左上角移动到 Top, Left 指定的位置。该函数只能使用到【计算】图标中。

注意：要使当前窗口定位在某一位置，必须在保存和打包文件前进行定位。

示例：

MoveWindow (0, 0)

将当前窗口移到展示窗口的左上角。

相关变量：WindowHeight, WindowWidth, WindowTop, WindowLeft

15 . Next Repeat

语法：Next Repeat

说明：该函数是跳过循环体中剩下未执行的命令，直接跳到下一次循环。可以将该函数放在循环体中的任意位置。

16 . Quit

语法：Quit ([Option])

说明：该函数是退出当前文件。参数 Option 有以下取值：

0：退出当前 Authorware 文件。如果该文件是从其他程序调用的，则返回到调用它的程序中，继续执行原来程序。当用户直接输入 Quit ()，Authorware 默认参数值为 0。

1：退出当前 Authorware 文件。

2：退出当前 Authorware 文件并重新启动计算机。如果操作系统为 Win31，则退回到 Dos 状态下。

3：退出当前 Authorware 文件，如果操作系统为 Win95、Win98、或 WinNT 时，则关闭计算机；如果操作系统为 Win31，则退出 Authorware 程序并显示出程序管理器。

17 . Random

语法：Random (Min, Max, Units)

说明：该函数返回一个介于 Min 和 Max 之间，并且是 Units 的倍数的数。

示例：

Num := Random (1, 10, 2)

返回 1 和 10 之间的数，并且该数是 2 的倍数（取值可能为 2, 4, 6, 8, 10），并将该随机数赋给变量 Num。

Value := Random (1, 5, .5)

返回 1 和 5 之间的数，并且该数是 0.5 的倍数（取值可能为 1, 1.5, 2, 2.5, 3, ...），并将该随机数赋给变量 Value。

18 . ReadExtFile

语法：String := ReadExtFile ("Filename")

说明：该函数是将文件 Filename 的内容赋给变量 String。若指定网络地址 URL，则必须为绝对地址，如：http://或 file:///。

该函数只能用到【计算】图标中。

使用 ReadExtFile 函数时，Authorware 会将相关信息保存到系统变量 IOStatus 和 IOMessage 中，并

且返回 IOStatus 的值。如果无错误，则 IOStatus 为 0，IOMessage 为空。如果有错误发生，操作系统将错误号赋给 IOStatus。只要 IOStatus 不为 0，表明出错，IOMessage 中包含的将是出错信息。

当 Authorware 的 Web Player 运行于非信任模式下，若文件为本地文件则该函数无效。

示例：

```
FileString := ReadExtFile ("D:\Data.TXT")
```

将 D 盘下的文本文件 Data.txt 中的内容赋给变量 FileString。

相关变量：IOStatus，IOMessage

相关函数：AppendExtFile，WriteExtFile

19. Repeat While

语法：Repeat While 条件

语句

End Repeat

说明：该函数建立一个循环。当判断条件为 True 时，就执行下面的语句；每开始一次循环，都要判断一下判断条件，直到判断条件为 False 时才退出循环。

示例：

```
i:=0
```

```
Repeat While i < 10
```

```
    i := i + 1
```

```
End Repeat
```

20. Repeat With

语法：Repeat With Counter := Start [down] to Finish

语句

End Repeat

说明：该函数建立一个循环。其循环次数 Counter 定义为一个范围，从小到大时使用函数 to，范围为从 Start 到 Finish（此时 Start<Finish），每执行一次 Counter 加 1；从大到小时函数使用 down to，范围为从 Start 到 Finish（此时 Start>Finish），（此时 Start<Finish）每执行一次 Counter 减 1。

示例：

```
Sum := 0
```

```
Repeat With i := 1 to 5
```

```
    Sum := Sum + i
```

```
End Repeat
```

```
Repeat With j := 5 down to 1
```

```
    Sum := Sum * j
```

```
End Repeat
```

21. ReplaceLine

语法：Result := ReplaceLine ("String", n, "NewString"[, delim])

说明：该函数是用字符串“NewString”替换字符串“String”中的第 n 行子串，然后将结果赋给变量 Result。同样，可以使用参数 delim 来设定行分隔符。

示例：

```
设 String1 := "Monday" ^ Return ^ "Tuesday" ^ Return ^ "Wednesday" ^ Return
```

```
ResultString := ReplaceLine (String1, 2, "Tuesday")
```

返回 ResultString = "Monday\tTuesday\tWednesday\t"

22 . ReplaceString

语法：Result := ReplaceString (“String”, Start, Length, “Replacement”)

说明：将字符串 “ String ” 中的从 Start 开始，长度为 Length 部分用字符串 “ Replacement ” 替换。

示例：

设 String1 := “Macromedia Freehand”

Result := ReplaceString (String1, 11, 8, “Authorware”)

返回 Result = “Macromedia Authorware”

23 . ResizeWindow

语法：ResizeWindow (Width, Height)

说明：该函数是重新设置程序窗口的大小，设置后窗口的宽度和高度分别为 Width 和 Height。

该函数只能使用在【计算】图标中。

示例：

ResizeWindow (800, 600)

将展示窗口的大小设置为 800×600。

24 . SetCursor

语法：SetCursor (Type)

说明：该函数是将鼠标光标设置成 Type 指定的光标类型。

Type 的取值一般为：

0：箭头

1：I (文本输入)

2：十字形

3：加号

4：空白

5：沙漏

6：手形

当更改鼠标光标后，Authorware 会一直使用该光标类型，除非用户再次改变光标。

如果添加了自定义的鼠标光标类型，Authorware 自动将类型定位从 51 开始。如果使用用户自定义的鼠标光标类型，可执行 SetCursor (n)，其中 n>=51。

示例：

SetCursor (6)

将鼠标光标类型设置成手形。

25 . ShowCursor

语法：ShowCursor (Display)

说明：该函数是控制显示或隐藏当前鼠标。

当 Display 值为 ON 时，显示鼠标；当 Display 值为 OFF 时，隐藏鼠标。

示例：

ShowCursor (OFF)

隐藏当前鼠标。

26 . String

语法：Result := String (Value)

说明：将 Value 的数据类型转化为字符串类型，并将其赋给变量 Result。

示例：



Result := String (12)

返回 Result = “12”。

27 . WriteExtFile

语法：WriteExtFile (“FileName”, “String”)

说明：该函数是将字符串 “ String ” 覆盖文件 Filename 的内容。

该函数只能用到【计算】图标中。

使用 WriteExtFile 函数时，Authorware 会将相关信息保存到系统变量 IOSTatus 和 IOMessage 中，并且返回 IOSTatus 的值。如果无错误，则 IOSTatus 为 0，IOMessage 为空。如果有错误发生，操作系统将错误号赋给 IOSTatus。只要 IOSTatus 不为 0，表明出错，IOMessage 中包含的将是出错信息。

当 Authorware 的 Web Player 运行于非信任模式下，该函数无效。

示例：

WriteExtFile (“D:\Data.TXT”, “Macromedia Authorware” ^ Return)

将字符串 “ Macromedia Authorware\t ” 覆盖 D 盘下的文本文件 Data.txt 中的内容。

相关变量：IOStatus，IOMessage

相关函数：AppendExtFile，ReadExtFile

第14章 Authorware 的外部函数

—— UCD

- UCD（用户代码文档）简介
- UCD 函数的使用
- 常用 UCD 函数详解
- 使用 UCD 函数实例

14.1 UCD（用户代码文档）简介

Windows DLL 并不是为非程序员设计的，标准的 DLL 格式和调用规则通常需要用户参考程序员的文档和 Windows 的 API（Application Program Interface，应用程序接口）调用。为了实现这些功能，Authorware 支持对于普通 DLL 的一种透明扩展，称之为 UCD（User Code Documents，用户代码文档）。在编写 Authorware 时使用到 DLL 的 C 程序员可以使用 UCD 来预装信息以使 Authorware 能够识别和链接 DLL，包括函数名、参数、返回类型、语法模块、以及在线描述等，而附加信息一般可以存储在 DLL 中或分开存储，并且这些信息对于其他程序来说也是透明的。

当装载一个 UCD 时，Authorware 会在函数窗口中列出函数列表。同使用系统函数一样，此时用户可以从函数列表中选择函数并将它们粘贴到一个计算编辑窗口中。

经常使用的 UCD 的文件一般有两种不同的类型，其后缀分别是 .ucd 和 .u32。其中后缀为 .ucd 的文件是使用在 Windows 3.X 这样的 16 位操作系统环境下的，而后缀为 .u32 是使用在 Windows 95/98/NT 这



样 32 位操作系统环境下。

14.1.1 装载 UCD 函数

同装载一般自定义函数一样，在 Authorware 使用一个 UCD 函数之前，必须将该函数装载。装载一个 UCD 函数如同打开一个 Authorware 文件一样，Authorware 从文件存储的位置读去函数。当函数被装载后，Authorware 会扩展函数信息，并控制何时执行函数。装载一个函数就地将 Authorware 和函数链接起来，并且允许它们之间进行通讯。

要装载一个 UCD 函数，首先打开函数窗口，在函数窗口的“分类”下拉框中选择当前应用程序文件名，此时窗口中的“载入”按钮被激活。单击“载入”按钮打开“加载函数”对话框，如图 14-1 所示。

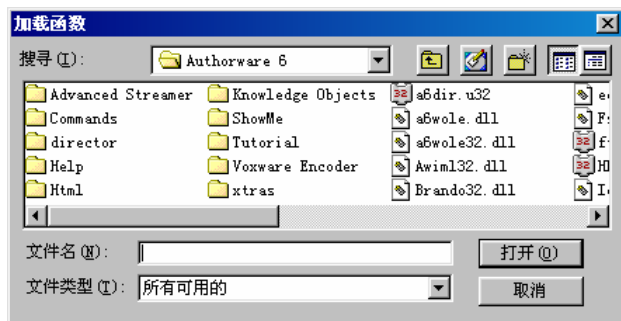


图 14-1 装载函数对话框

在 Authorware 的安装光盘上都有一个“\UCD”目录，该目录下面包含有十几个子目录，如图 14-1 所示，每一个目录中都包括一些特定功能的 UCD。

在目录中选择要装入文件的子目录，打开该子目录，然后在该子目录下选择 UCD 文件，如图 14-2 所示。

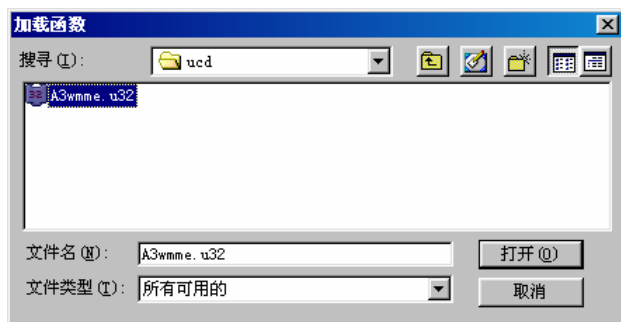


图 14-2 选择 UCD 文件

在子目录中选择好 UCD 文件后，单击“打开”按钮将该 UCD 文件打开，系统弹出该 UCD 文件中包含的函数列表对话框，如图 14-3 所示。

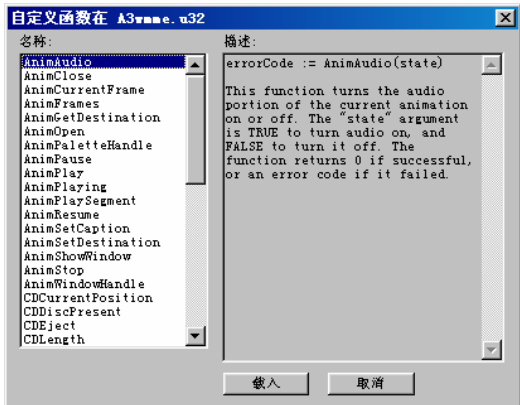


图 14-3 函数列表对话框

在该对话框中，左侧是该文件中所有的自定义函数，右侧是选中函数的描述信息。
在函数列表框中选择要装载的函数。可以使用<Ctrl>键或<Shift>键一次选择多个函数，单击函数列表对话框中的“载入”按钮可将选中的函数全部装入到用户应用程序中，如图 14-4 所示。



图 14-4 装载到函数列表框中的 UCD 函数

这样，用户就可以像使用系统函数一样使用装载的 UCD 函数。

14.1.2 常用 UCD 函数详解

1 . A5wmme.ucd/u32

该 UCD 中包括了对各种媒体播放的控制函数，其中包括对 Wave，CD，MIDI，Video 等的操作函数，同时还提供了一些 MCI 控制函数。

- CDCurrentPosition

语法：Result := CDCurrentPosition ()

说明：该函数返回当前 CD 音轨播放的时间，返回一个字符串，格式为“ TT:MM:SS:FF ”（音轨数：分钟数：秒数：帧数）。

- CDDiscPresent

语法：Result := CDDiscPresent ()

说明：该函数是判断是否将 CD 唱片放入了 CD-ROM 中。

如果 CD-ROM 中有 CD 唱片，返回值为 True，否则返回值为 False。

- CDEject

语法：ErrorCode := CDEject ()

说明：使用该函数可以弹出当前 CD-ROM。

如果该函数调用成功，返回值为 0，否则返回错误代码。

- CDLength

语法：Result := CDLength ()

说明：该函数返回当前 CD 唱片的总的时间长度，单位为秒。

如果有错误发生，返回值为 0。

- CDPause

语法：ErrorCode := CDPause ()

说明：该函数暂停当前 CD 音轨的播放。

如果函数调用成功，返回值为 0，否则返回一个错误代码。

- CDPlay

语法：Number := CDPlay (Track, Wait)

说明：该函数是播放 CD 唱片的。其中参数 Track 用来指定播放 CD 的音轨，参数 Wait 为 True 时，表示播放完 CD 后，Authorware 应用程序才继续执行，如果 Wait 为 False，表示播放 CD 唱片的同时应用程序继续执行。

该函数调用成功，返回值为 0，否则返回错误代码。

- CDPlaying

语法：Result := CDPlaying ()

说明：如果 CD 正在播放，该函数返回 True，否则返回 False。

- CDPlaySegment

语法：ErrorCode := CDPlaySegment (Track, FromSeconds, ForSeconds, Wait)

说明：该函数指定播放 CD 音轨的长度。参数 Track 用来指定播放的音轨，取值从 1 到 CD 的总音轨数；参数 FromSeconds 和 ForSeconds 分别用来指定播放的起始位置和播放长度；如果 ForSeconds 为 0，则表示要播放到该音轨的结尾。如果 ForSeconds 为 -1，则表示要播放到整个 CD 的结尾。参数 Wait 为 True 时，表示播放完 CD 后才让 Authorware 应用程序继续执行；若 Wait 为 False，则表示在播放 CD 的同时，应用程序继续执行。

如果函数调用成功，返回 0，否则返回一个错误代码。

- CDResume

语法：ErrorCode := CDResume ()

说明：如果使用了函数 CDPause 暂停了 CD 音乐的播放，可以使用 CDResume 函数让音乐继续播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- CDRetract

语法：ErrorCode := CDRetract ()

说明：该函数是收回已经弹出的当前 CD-ROM。

如果函数调用成功，返回 0，否则返回一个错误代码。

- CDStop

语法：ErrorCode := CDStop ()



说明：该函数是暂停当前 CD 音轨的播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- CDTrackLength

语法：Result := CDTrackLength (Track)

说明：该函数返回指定 CD 音轨的长度。

如果函数调用失败，返回 0。

- CDTracks

语法：Result := CDTracks ()

说明：该函数返回当前 CD 唱片的音轨数，即当前 CD 唱片的歌曲数。

如果函数调用失败，返回 0。

- MCIExecute

语法：ResultBool := MCIExecute (CmdString)

说明：该函数是向 MCI 设备发送一个 MCI 命令。

如果函数调用成功，返回 True，否则返回 False。

- MCISendString

语法：ErrorCode := MCISendString (CmdString)

说明：该函数是向 MCI 设备发送一个 MCI 命令字符串。

如果函数调用成功，返回 0，否则返回一个错误代码。可以从函数 MCIGetReturnString 得到 MCI 设备返回的字符串。

- MIDIPause

语法：ErrorCode := MIDIPause ()

说明：该函数是暂停当前 MIDI 音乐的播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- MIDIPlay

语法：ErrorCode := MIDIPlay (FileName, Tempo, Wait)

说明：该函数是从头到尾播放参数 FileName 指定的 MIDI 文件 (.MID)。参数 Tempo 使用一个百分数来指定播放的节拍，如 100 表示正常播放；参数 Wait 为 True 时，表示播放完 MIDI 文件后才让 Authorware 应用程序继续执行；若 Wait 为 False，则表示在播放 MIDI 文件的同时，应用程序继续执行。

如果函数调用成功，返回 0，否则返回一个错误代码。

- MIDIPlaying

语法：Result := MIDIPlaying ()

说明：如果正在播放一个 MIDI 音乐，该函数返回 True，否则返回 False。

- MIDIPlaySegment

语法：ErrorCode := MIDIPlaySegment (FileName, Tempo, FromSeconds, ForSeconds, Wait)

说明：该函数是播放参数 FileName 指定的 MIDI 文件 (.MID)。参数 Tempo 使用一个百分数来指定播放的节拍，如 100 表示正常播放；参数 FromSeconds 和 ForSeconds 分别用来指定播放的起始位置和播放长度；参数 Wait 为 True 时，表示播放完 MIDI 文件后才让 Authorware 应用程序继续执行；若 Wait 为 False，则表示在播放 MIDI 文件的同时，应用程序继续执行。

如果函数调用成功，返回 0，否则返回一个错误代码。

- MIDIResume

语法：ErrorCode := MIDIResume ()

说明：如果使用了函数 MIDIPause 暂停了 MIDI 音乐的播放，可以使用 MIDIResume 函数让音乐继

续播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- MIDIStop

语法：ErrorCode := MIDIStop ()

说明：该函数停止当前 MIDI 音乐的播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- WavePause

语法：ErrorCode := WavePause ()

说明：该函数暂停当前 Wave 音乐的播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- WavePlay

语法：ErrorCode := WavePlay (FileName, Wait)

说明：该函数是从头到尾播放参数 FileName 指定的 Wave 声音文件 (.wav)。参数 Wait 为 True 时，表示播放完 Wave 文件后才让 Authorware 应用程序继续执行；若 Wait 为 False，则表示在播放 Wave 文件的同时，应用程序继续执行。

如果函数调用成功，返回 0，否则返回一个错误代码。

- WavePlaying

语法：Result := WavePlaying ()

说明：如果正在播放一个 Wave 音乐，该函数返回 True，否则返回 False。

- WavePlaySegment

语法：ErrorCode := WavePlaySegment (FileName, FromSeconds, ForSeconds, Wait)

说明：该函数是播放参数 FileName 指定的 Wave 声音文件 (.wav)。参数 FromSeconds 和 ForSeconds 分别用来指定播放的起始位置和播放长度，参数 Wait 为 True 时，表示播放完 Wave 文件后才让 Authorware 应用程序继续执行；若 Wait 为 False，则表示在播放 Wave 文件的同时，应用程序继续执行。

如果函数调用成功，返回 0，否则返回一个错误代码。

- WaveResume

语法：ErrorCode := WaveResume ()

说明：如果使用了函数 WavePause 暂停了 Wave 音乐的播放，可以使用 WaveResume 函数让音乐继续播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

- WaveStop

语法：ErrorCode := WaveStop ()

说明：该函数停止当前 Wave 音乐的播放。

如果函数调用成功，返回 0，否则返回一个错误代码。

2. Copyfile.ucd/u32

该 UCD 是用来设置文件复制。

- CopyFile

语法：Result := CopyFile (SourceFile, DestinationFile)

说明：该函数是将参数 SourceFile 指定的源文件复制到参数 DestinationFile 指定的目标文件。如果目标文件存在，将会覆盖该文件。使用该函数最好使用绝对路径。

如果函数调用成功，返回文件复制的字节数，否则返回一个负数。



3. Filedlgs.ucd/u32

该 UCD 是用来设置标准的 Windows 文件对话框。

- OpenFileDialog

语法：Path := OpenFileDialog (Dlgtitle, Defname, Defpath, Defext, Filters)

说明：该函数是显示出一个标准的 Windows 打开文件对话框，让用户选择所要文件。如果选取了所要文件，返回该文件的完整路径；如果单击了对话框中的“取消”按钮或有其他错误发生，则返回一个空字符串。

参数 Dlgtitle 是该对话框的标题。如果该参数是一个空字符串，则标题默认为“打开文件”。

参数 Defname 是在打开文件名栏上的默认文件名。如果该参数包含了一个路径，则它会覆盖参数 Defpath 指定的路径，并且对话框以这个路径打开。该选项可为空字符串。

参数 Defpath 指定对话框打开的初始路径。如果该参数为空字符串，则对话框使用当前路径。

参数 Defext 用来指定默认的后缀。如果文件的扩展名没有明显给出，该参数会自动将默认的后缀加到文件名后。

参数 Filters 是指定文件类型列表中的文件类型。该参数的格式必须为：

“Description|*.ext| Description|*.ext; *.extl...”

即一个描述信息必须与其对应的文件类型相连，中间用|隔开。该字符串必须以|结束。如果一个描述信息有多个文件类型，则文件类型之间必须使用分号隔开。

如：“Bitmaps|*.bmp;*.dib|Text files|*.txt|All files|*.*|”

- SaveAsFileDialog

语法：Path := SaveAsFileDialog (Dlgtitle, Defname, Defpath, Defext, Filters)

说明：该函数是显示出一个标准的 Windows 保存文件对话框，让用户为文件命名，并将文件存储到一定的路径下。如果存储了文件，返回该文件的完整路径，存放到变量 Path 中；如果单击了对话框中的“取消”按钮或有其他错误发生，则返回一个空字符串。

参数 Dlgtitle 是该对话框的标题。如果该参数是一个空字符串，则标题默认为“另存为...”。

参数 Defname 是在保存文件名栏上的默认文件名。如果该参数包含了一个路径，则它会覆盖参数 Defpath 指定的路径，并且对话框以这个路径打开。该选项可为空字符串。

参数 Defpath 指定对话框保存的初始路径。如果该参数为空字符串，则对话框使用当前路径。

参数 Defext 用来指定默认的后缀。如果文件的扩展名没有明显给出，该参数会自动将默认的后缀加到文件名后。

参数 Filters 是指定文件类型列表中的文件类型。该参数的格式必须为：

“Description|*.extl Description|*.ext; *.extl...”

即一个描述信息必须与其对应的文件类型相连，中间用|隔开。该字符串必须以|结束。如果一个描述信息有多个文件类型，则文件类型之间必须使用分号隔开。

如：“Bitmaps|*.bmp;*.dib|Text files|*.txt|All files|*.*|”

14.1.3 其他一些常用 UCD 功能简介

- A5dir.ucd/u32：设置 Director 电影控制的函数。
- Anicur.ucd/u32：设置动画光标。
- Apwkeys.ucd/u32：识别 Windows 的一些特殊键，如 Alt，Ctrl 键等。
- Apwmouse.ucd/u32：设置鼠标键的功能。
- Budwav.ucd/u32：控制 Wave 文件的音量。
- Cover.ucd/u32：设置是否覆盖 Windows 的桌面。

- DDE.ucd/u32：提供支持应用程序之间动态传递数据的函数。
- Dib.ucd/u32：从 Dib 文件中获得调色板设置。
- Dib2pal.ucd/u32：从 Dib 文件中获得调色板设置。
- Disptext.ucd/u32：设置展示窗口的指定位置显示指定了字体、字型、字号、字色等特性的文本。
- Edit.ucd/u32：处理文本。
- Ftp.ucd/u32：提供对网络 FTP 下载服务的控制函数。
- Listbox.ucd/u32：设置带滚动条的列表框。
- Memtools.ucd/u32：提供各种管理和分配内存的函数。
- ODBC.ucd/u32：设置数据库操作。
- Palette.ucd/u32：设置调色板。
- Prnt.ucd/u32：设置打印机操作函数。
- Scrledit.ucd/u32：设置文本编辑框。
- Winapi.ucd/u32：提供 Windows 的 API 函数。

14.2 使用 UCD 函数实例——MIDI 播放器

该程序可以播放 MIDI 音乐，界面如图 14-5 所示。



图 14-5 MIDI 播放器

操作步骤为：

- (1) 新建一个文件，将其命名为“MIDI 播放器.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性...]命令，打开文件属性对话框，设置应用程序的背景色为灰色，不显示菜单项。单击“确定”按钮确定。
- (3) 拖动一个【计算】图标到设计窗口中流程线上，将其命名为“初始化”。
- (4) 双击【计算】图标，打开计算编辑窗口，在计算编辑窗口中输入下面语句：

```

ResizeWindow(304, 118)
File_Name:=""
Pause:=1

```
- (5) 关闭计算编辑窗口，系统弹出新建变量对话框，分别单击“确定”按钮即可。
- (6) 单击按钮，打开函数对话框，从“分类”下拉框中选中“MIDI 播放器.a6p”，然后单击函数对话框中的“载入”按钮，打开“加载函数”对话框，选择 Filedlgs.u32 文件，如图 14-6 所示。

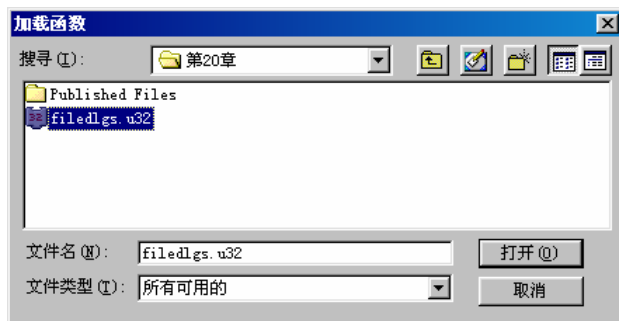


图 14-6 装载 Filedlgs.u32 函数

(7) 单击“打开”按钮后，打开“自定义函数在 Filedlgs.u32”对话框，从左侧函数列表选中“OpenFileDialog”函数，如图 14-7 所示。

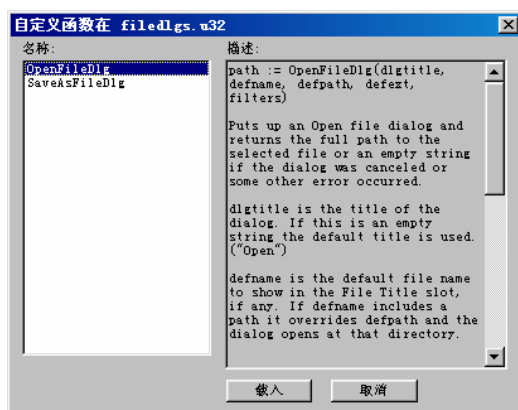


图 14-7 选中“OpenFileDialog”函数

(8) 单击该对话框中的“载入”按钮，将选中的函数装载到应用程序中。

(9) 同样在函数对话框中单击“载入...”按钮，打开“加载函数”对话框，选择 A5wmme.u32 文件，然后打开显示“自定义函数在 A5wmme.u32”对话框，选中“MIDIPause”、“MIDIPlay”、“MIDIPlaying”、“MIDIPlaySegment”、“MIDIResume”和“MIDIStop”函数，如图 14-8 所示。



图 14-8 选中 MIDI 控制函数

(10) 单击“载入”按钮将选中的函数装载到应用程序中。此时函数对话框“MIDI 播放器.a6p”程序中已经装载了我们需要的函数，如图 14-9 所示。



图 14-9 程序装载的自定义函数

(11) 拖动一个【交互】图标到流程线上，将其命名为“控制”。

(12) 依次拖动五个【群组】图标到【交互】图标的右侧，分别将其命名为“播放”、“暂停”、“关闭”、“打开文件”和“退出”，响应类型为按钮响应。

(13) 单击“播放”按钮响应类型图标，打开响应类型属性对话框，在“交互”标签页的“激活条件”文本框中输入“PlayFlags”。

(14) 关闭对话框时，系统弹出新建变量对话框，输入其初始值为 False，单击“确定”按钮。

(15) 同样，在“暂停”和“关闭”响应类型属性对话框“交互”标签页的“激活条件”文本框中分别输入“PauseFlags”和“StopFlags”。创建这两个新变量，并将其初始值都设置为 False。

(16) 双击“控制”【交互】图标，打开展示窗口。在展示窗口中排列五个按钮如图 14-5 所示。

(17) 在按钮的上方绘制一个黑色矩形，然后选取“文字工具”在黑色矩形的中间输入“{File_Name}”，并选择文字颜色为白色，粗体，并且其显示模式为“透明”模式，如图 14-10 所示。



图 14-10 显示变量

(18) 打开“打开文件”【群组】图标，进入二级流程线。在二级流程线上拖动一个【计算】图标，命名为“选择 MIDI 文件”。

(19) 打开计算编辑窗口，在窗口中输入如下语句：

```
File_Name := OpenFileDlg("", "Midi.mid", "", "*.mid", "MIDI 文件|.mid|所有文件|*.*)")
--显示“打开文件”对话框
```



```

if File_Name <> "" then
--选择了播放文件
    PlayFlags:=TRUE
    PauseFlags:=TRUE
    StopFlags:=TRUE
    Pause:=1
--激活“播放”按钮、“暂停”按钮和“关闭”按钮
else
--如果打开文件错误
    File_Name:="打开文件错误！"
end if

```

(20) 打开“播放”【群组】图标，进入二级流程线。在二级流程线上拖动一个【计算】图标，命名为“播放 MIDI 音乐”。

(21) 打开计算编辑窗口，在窗口中输入如下语句：

```

FileN:=LowerCase(File_Name)
--判断文件类型是否为 MIDI 文件类型
if Find(".mid",FileN) then
--如果是 MIDI 文件类型
    if Pause=0 then
--如果是暂停音乐播放
        MIDIResume()
--继续播放音乐
    else
--如果是第 1 次播放该 MIDI 文件
        MIDIPlay(File_Name, 100, FALSE)
--播放该文件
    end if
else
--如果不是 MIDI 文件类型
    File_Name:="无法打开该文件！文件类型错误！"
--显示出错信息
    PlayFlags:=FALSE
    PauseFlags:=FALSE
    StopFlags:=FALSE
--灰显“播放”按钮、“暂停”按钮和“关闭”按钮
end if

```

(22) 打开“暂停”【群组】图标，进入二级流程线。在二级流程线上拖动一个【计算】图标，命名为“暂停 MIDI 音乐”。

(23) 打开计算编辑窗口，在窗口中输入如下语句：

```
Pause:=MIDIPause()
```

(24) 打开“关闭”【群组】图标，进入二级流程线。在二级流程线上拖动一个【计算】图标，命名为“关闭 MIDI 音乐”。

(25) 打开计算编辑窗口，在窗口中输入如下语句：

```
MIDIStop()
--停止 MIDI 音乐播放
File_Name:=""
--显示空文件名
PlayFlags:=FALSE
PauseFlags:=FALSE
StopFlags:=FALSE
--灰显“播放”按钮、“暂停”按钮和“关闭”按钮
```

(26) 打开“退出”【群组】图标，进入二级流程线。在二级流程线上拖动一个【计算】图标，命名为“退出程序”。

(27) 打开计算编辑窗口，在窗口中输入如下语句：

```
MIDIStop()
--停止 MIDI 音乐的播放
Quit()
--退出程序
```

(28) 此时程序已经设计完成，流程如图 14-11 所示。

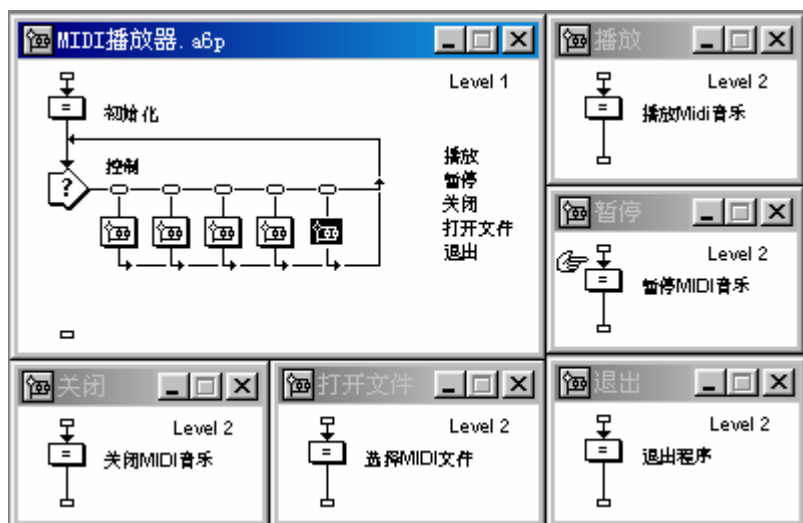


图 14-11 设计好的流程

第15章 知识对象的使用

- 知识对象简介
- 使用知识对象
- 系统知识对象简介
- 知识对象应用实例

15.1 知识对象简介

知识对象（知识对象）是一个逻辑包——包装在一个模块中，该逻辑包可以插入到 Authorware 应用程序中。一个知识对象与一般的模块有所不同，因为它是与一个向导相链接的。向导其实是一个 Authorware 程序，它为应用程序的建立、改变、甚至添加新的内容和逻辑提供接口。

知识对象是一个强大的工具。使用它，非程序员可以快速、高效地制作出一个完整的培训软件，可以实现许多系统函数都无法实现的功能。对于资深的程序员，使用知识对象可以使之更有效地、以一种自动重复地完成一项任务。事实上，知识对象就是希望从根本上解放创作过程。

15.1.1 知识对象窗口

当启动 Authorware 后，一般并不是直接进入 Authorware 的主界面，系统会弹出一个“新建文件”对话框，如图 15-1 所示。



图 15-1 “新建文件”对话框

- “请选取知识对象创建新文件”列表框：在该列表框中列出了可以为新文件添加的知识对象。在列表框中选择一个知识对象后，然后单击“确定”按钮，即可将该知识对象添加到应用程序中。
- “描述”文本框：该文本框中是上面列表框中选中的知识对象的描述信息。
- “启动时显示本对话框”复选框：选中该复选框时，下次启动 Authorware 时，依然会自动启动该对话框；如果取消该复选框，在下次启动 Authorware 时，不会再启动该对话框了。

在该对话框中，要求用户为新建的文件选择一个知识对象，我们常常是直接单击“取消”按钮，直接建立一个没有知识对象的文件，而如果我们希望添加一个知识对象时，可以直接从知识对象窗口中拖动一个知识对象到流程线上即可。

当进入到 Authorware 的主界面后，知识对象窗口已经被系统自动打开了，如图 15-2 所示。



图 15-2 知识对象窗口

15.1.2 使用知识对象

要使用一个知识对象，用户只需要从知识对象窗口拖动需要的知识对象到设计窗口的流程线上。此时系统会弹出一个提示框，要求您在使用知识对象前必须保存文件，如图 15-3 所示。

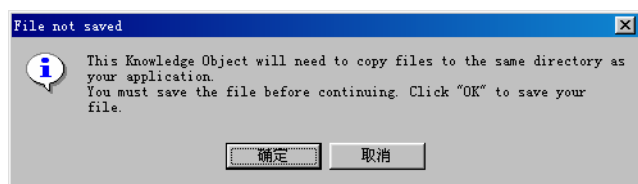


图 15-3 保存文件提示框

当您选择“确定”按钮，系统弹出一个保存文件对话框，用户只要选取合适路径，输入文件名，保存该文件即可启动该知识对象链接的向导。

如果选择“取消”按钮，Authorware 会暂时不启动该知识对象链接的向导。

15.2 系统提供的知识对象简介

Authorware 提供了共 42 个知识对象对象，分为 9 个类型。

15.2.1 评估——与评定操作有关的知识对象

该类型下面的知识对象都是一些与评定操作相关的知识对象，它包括 9 个知识对象。

- 拖放问题 用于创建一个拖放习题。当学生拖动图形对象到屏幕上指定的区域时，答案会自动显示出来。
- 热对象问题 用于创建一个热对象多项选择习题。当学生单击图形对象，答案就会显示出来。在图形对象被单击后，一个与该问题主题相关联的指定文件就会显示出来。
- 热点问题 用于创建一个热区多项选择习题。当学生在隐含的热区中做了单击操作，答案就会显示出来。在热区被单击后，一个与该问题主题相关联的指定文件就会显示出来。
- 登录 用于创建测试登陆过程以及选择测试成绩存储方式。
- 多选问题 用于创建一个多项选择习题。习题类型适合于有多于一个正确选项的习题。学生必须选中了所有正确的选项才可以得到本习题的分数。
- 得分 用于实现测试成绩的记录、统计和显示。
- 简略回答问题 用于创建简短回答习题。该问题类型适合于对学生信息输入做出反应的习题。通配符可以使用在允许细小拼写错误的习题。
- 单选问题 用于创建单选题。该习题类型只适用于习题要求学生只有选中了惟一的正确答案才能得分的情况。
- 真-假问题 用于创建真（True）或者（False）假类型习题。该习题类型适合于只有一个逻辑答案的习题。

15.2.2 文件——与文件操作相关的知识对象

该类型下面的知识对象都是一些与文件操作相关的知识对象，它包括 7 个知识对象。

- 增加-移除字体资源 用于添加或去掉计算机中某种字体，以使自己的应用程序可以使用该字体。
- 拷贝文件 用于将指定的一个或几个文件复制到一个指定目录下。
- 查找 CD 驱动器 用于查找当前计算机上的第一个 CD-ROM 盘符，并将该盘符字母或字符路径储存到一个指定的变量中，以供用户的应用程序使用。
- 读取 INI 值知识对象 用于该知识对象是 Authorware 6 新增加的，它可以从您指定的 INI 文件中读取值。

- 跳到指定 Authorware 文件 用于实现 Authorware 程序之间的跳转。
- 设置文件属性 用于设置一个或几个指定文件的属性。
- 写入 INI 值知识对象 该知识对象是 Authorware 6 新增加的，它可以向您指定的 INI 文件中写值。

15.2.3 Icon Palette Settings——与图标板设置相关的知识对象

该类型下面的知识对象都是一些与图标板设置相关的知识对象，在 Authorware 6 中，它还没有知识对象。

15.2.4 界面构成——与界面部件相关的知识对象

该类型下面的知识对象都是一些与界面部件相关的知识对象，它包括 13 个知识对象。浏览文件夹对话框 使用该知识对象，可以出现一个选择目录的目录对话框，通过用户在本地或网络磁盘驱动器上选择合适的目录，然后将用户选取的目录路径存放在一个变量中，供应用程序使用。

- 复选框 使用该知识对象，可以创建一个复选框，同时创建出该复选框的文本，最后将用户对该复选框的选择状态（选中或未选中）保存到一个变量中返回。
- 消息框 使用该知识对象，可以创建出多种样式的信息提示框，并将用户对信息提示框所做的操作保存到一个变量中，供应用程序使用。
- 移动指针 使用该知识对象，可以将鼠标光标移动到某个指定位置，而且移动可以设置成动态移动或直接跳转到指定位置。
- 电影控制 使用该知识对象，为播放的数字电影提供一个操作控制面板，可以播放的数字电影格式包括 AVI、DIR、MOV、MPEG 等几种。
- 打开文件时对话框 使用该知识对象，可以产生一个打开文件的对话框，用户可以通过它浏览本机或网络驱动器，并将用户对该对话框的选择，即选择的文件路径和名称保存到一个变量中，供应用程序使用。
- 收音机时按钮 使用该知识对象，可以创建出一组单选钮，同时建立该单选钮的文本，最后再将用户所做的选择保存到一个变量中，供应用程序使用。
- 保存文件时对话框 使用该知识对象，可以产生一个保存文件的对话框，用户可以通过它浏览本机或网络驱动器，并将用户对该对话框的选择，即保存的文件路径和名称存放到一个变量中，供应用程序使用。
- 设置窗口标题 使用该知识对象，可以设置当前 Authorware 应用程序的标题栏。如果在文件属性对话框中设置该应用程序无标题栏，则该知识对象无效。同时还可以将标题栏设置成一个变量，使得标题栏可以随着变量的变化而改变。
- 滑动条 使用该知识对象，可以建立一个指定的滑动条，其外观样式可以进行修改，同时将该滑动条所处的位置返回给一个变量，供应用程序使用。
- 窗口控制知识对象 该知识对象可以在展示窗口中显示一个 Windows 控件(Windows Control)（如：列表框控件或者组合框控件）。该知识对象可以很容易地使您创建用户输入表单（包括 Tab 操作）和/或者为一般的 Windows 应用程序创建仿真课件。
- 窗口控制—获取属性知识对象 使用该知识对象可以获取使用窗口控制知识对象创建的控件属性的当前值。该值会存放在您创建的或者是选择好的用户变量中。每个“窗口控制—获取属性”知识对象只能获取一个控件的一个属性值。
- 窗口控制—设置属性知识对象 使用该知识对象可以设置使用窗口控制知识对象创建的控件属性的值。每个“窗口控制—设置属性知识对象只能获取一个控件的一个属性值。



15.2.5 Internet——与网络相关的知识对象

该类型下面的知识对象都是一些与网络相关的知识对象，它包括 2 个知识对象。

- 运行默认浏览 使用该知识对象，可以使用系统默认的网络浏览器来执行用户指定的 URL 地址或其它 EXE 程序，我们可以使用它来调用外部的可执行文件。如果用户计算机上没有默认的浏览器，则系统会提示用户指定一个可执行文件作为浏览器，同时可以选择打开该网址时是否退出当前的 Authorware 应用程序。
- 发送 Email 使用该知识对象，可以通过 SMTP（简单邮件传输协议）向指定的 Email 地址发送一个电子邮件，同时将发送结果（成功或失败）保存到一个变量中，供应用程序使用。

15.2.6 Model Palette——与图标模板相关的知识对象

该类型的知识对象都是在图标模板中已有的图标信息，它下面的知识对象与图标模板中的图标功能一样。

15.2.7 新建文件——与创建新文件相关的知识对象

该类型下面的知识对象都是一些与创建新的 Authorware 应用程序相关的知识对象，它包括 2 个知识对象。

在新建一个作品工程，或者刚进入 Authorware 时，系统都会显示出一个启动对话框，如图 15-4 所示。



图 15-4 “新建”对话框

该对话框中的两个选项对应着“新建”类型知识对象中的两种类型。

- 应用程序 使用该知识对象，可以快速生成一个具有漂亮界面的多媒体教学、培训软件似的应用程序，其中包括大量的选项供用户进行选择，以适合自己的需要，其中主要包括学生登录、显示学生学习任务、习题、词汇表以及菜单等。
- 测验 使用该知识对象，可以产生一个测试性的应用程序，可以包括多种测试习题类型，如拖放测试题、热区测试题、单选题、多选题、文本输入测试题、判断题等。

15.2.8 RTF 对象——与 RTF 对象相关的知识对象

该类型的知识对象都是和 RTF 对象相关的知识对象，它包括了 6 个知识对象。

- 创建 RTF 对象 该知识对象可以在 Authorware 展示窗口中创建一个 RTF 对象。该知识对象

只能使用在 Win32 中。

- 获取 RTF 对象文本区 该知识对象可以返回一个已经存在的 RTF 对象中指定范围的文本。该知识对象只能使用在 Win32 中。
- 插入 RTF 对象热文本交互 该知识对象可以为 RTF 对象插入一个带有热区响应的交互图标。这些响应可以和 RTF 对象进行交互。该知识对象只能使用在 Win32 中。
- 保存 RTF 对象 该知识对象可以输出一个已经存在的 RTF 对象。该知识对象只能使用在 Win32 中。
- 查找 RTF 对象 该知识对象在一个已经存在的 RTF 对象中查找某些指定文本或者短语。该知识对象只能使用在 Win32 中。
- 显示或隐藏 RTF 对象 该知识对象可以使一个已经存在的 RTF 对象(该 RTF 对象是由前面创建 RTF 对象知识对象创建好的) 可见或者隐藏。该知识对象只能使用在 Win32 中。

15.2.9 指南——与教程相关的知识对象

该类型下面的知识对象都是一些与导航相关的知识对象，它只包括 2 个知识对象对象。

- 相机部件 使用该知识对象，可以在您的作品中使用 Authorware 教程——照相机部件说明。其实，该知识对象就是一个【群组】图标，其内部结构如图 15-17 所示。
- 拍照片 使用该知识对象，可以产生一些如“前一页”、“后一页”、“查找”等导航按钮。当拖动该知识对象到流程线上时，系统会弹出一个对话框，要求您定位该知识对象需要的一个库文件，如图 15-6 所示。



图 15-6 定位拍照片知识对象的库文件

15.3 知识对象应用实例

下面我们简单演示一下几个常用的知识对象对象的使用方法。

15.3.1 使用“查找 CD 驱动器”知识对象

可以查找到当前计算机上的第一个 CD-ROM 盘符，并将该盘符字母或字符路径储存到一个指定的变量中，以供用户的应用程序使用。

使用步骤：

- (1) 创建一个新文件。
- (2) 从知识对象窗口中拖动“查找 CD 驱动器”知识对象到流程线上(或者直接双击该知识对象)，此时出现提示对话框，要求用户保存应用程序。
- (3) 保存应用程序，此时启动了该知识对象的向导程序，如图 15-7 所示。

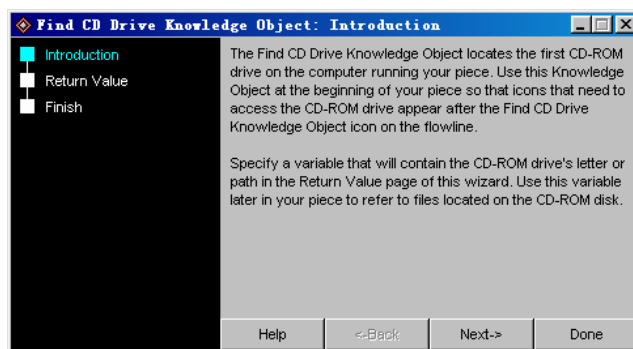


图 15-7 “查找 CD 驱动器”的向导程序

(4) 该界面中的文本是该知识对象的介绍信息。直接单击“Next”按钮即可。此时才出现“Return Value”对话框，如图 15-8 所示。

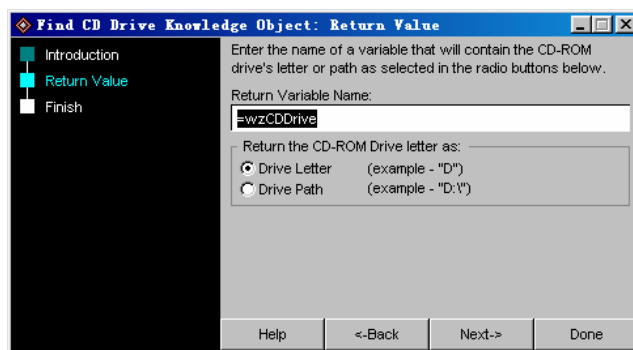


图 15-8 “Return Value”对话框

(5) 该对话框是用来设置返回信息的，其中：

“Return Variable Name”文本框：在该文本框中输入记录 CD-ROM 盘符的变量名。

“Return the CD-ROM Drive letter as”选项：设置返回类型，它包括两个选项：

“Drive Letter”单选钮：选择该选项，返回盘符形式，如“D”。

“Drive Path”单选钮：选择该选项，返回光盘路径形式，如“D:\”。

(6) 我们假设变量为系统默认变量，选择“Drive Letter”单选钮，然后单击“Next”按钮，进入“Finish”对话框，如图 15-9 所示。

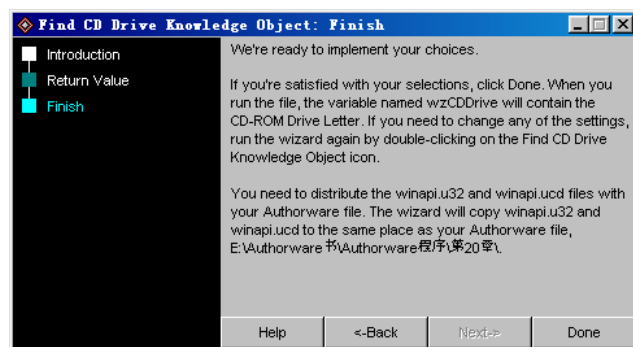


图 15-9 “Finish”对话框

(7) 该对话框中是一些提示信息，如果用户此时希望重新设置信息提示框，可以单击“Back”按

钮退回到原来地方重新设置。如果用户单击“Done”按钮完成设置后，希望重新设置，可以双击该知识对象图标，重新运行该向导，进行重新设置即可。

(8) 单击“Done”按钮完成设置。

当应用程序运行到该知识对象时，系统就会自动检测用户计算机上的第一个 CD-ROM 盘符，并将盘符字母保存到变量 wzCDDrive 中，这样就可以操作用户计算机上的 CD-ROM，对光盘中的文件进行操作。

15.3.2 使用“发送 Email”知识对象

使用该知识对象，可以通过 SMTP（简单邮件传输协议）向指定的 Email 地址发送一个电子邮件，同时将发送结果（成功或失败）保存到一个变量中，供应用程序使用。

使用步骤为：

(1) 创建一个新文件。

(2) 从知识对象窗口中拖动“发送 Email”知识对象到流程线上（或者直接双击该知识对象），此时出现提示对话框，要求用户保存应用程序。

(3) 保存应用程序，此时启动了该知识对象的向导程序，如图 15-10 所示。

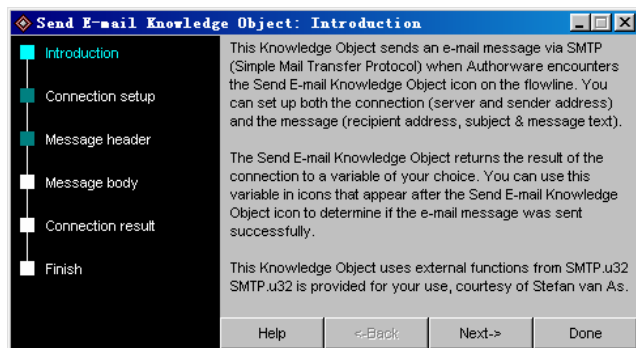


图 15-10 “发送 Email”知识对象的向导程序

(4) 该界面中的文本是该知识对象的介绍信息。直接单击“Next”按钮即可。此时才出现“Connection setup”对话框，如图 15-11 所示。

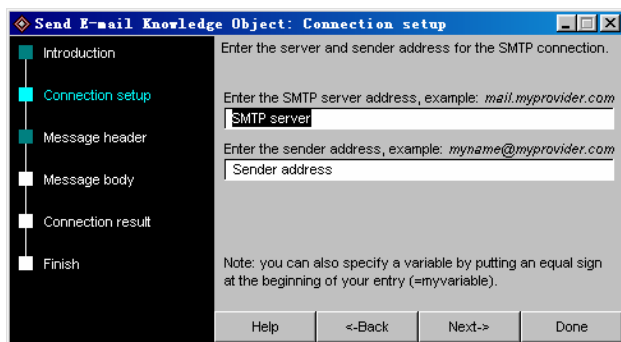


图 15-11 “Connection setup”对话框

(5) 该对话框用来设置发送邮件服务器和发送邮件信箱信息。其中：

“Enter the SMTP server address, example: mail.myprovider.com”文本框：在该文本框中输入发送

邮件的服务器地址，如 mail.myprovider.com。

“ Enter the sender address , example : myname@myprovider.com ” 文本框：在该文本框中输入邮件发送者的 Email 信箱地址。

(6) 在该对话框中输入服务器地址和发送者的地址，也可以使用变量来控制，然后单击 “ Next ” 按钮，进入 “ Message header ” 对话框，如图 15-12 所示。

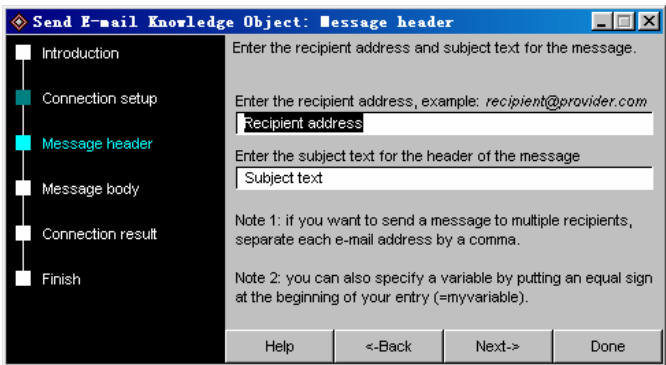


图 15-12 “ Message header ” 对话框

(7) 该对话框是用来设置所发邮件的主题，其中：

“ Enter the recipient address , example : recipient@provider.com ” 文本框：该文本框是用来设置收件人的信箱地址。

“ Enter the subject text for the header of the message ” 文本框：该文本框用来设置所发邮件的主题。

(8) 设置了收件人的邮箱和信件主题后，单击 “ Next ” 按钮进入 “ Message body ” 对话框中，如图 15-13 所示。

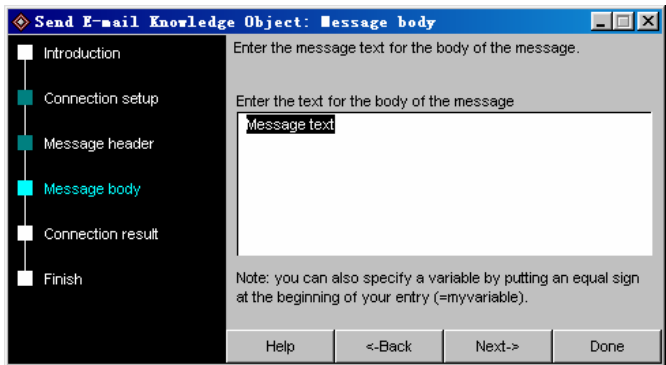


图 15-13 “ Message body ” 对话框

(9) 在该对话框中设置所要发送的邮件的正文，其中：

“ Enter the text for the body of the message ” 文本框：在该文本框中输入所要发送的邮件的正文。

(10) 在文本框中输入邮件正文后，单击 “ Next ” 按钮，进入 “ Connection result ” 对话框，如图 15-14 所示。

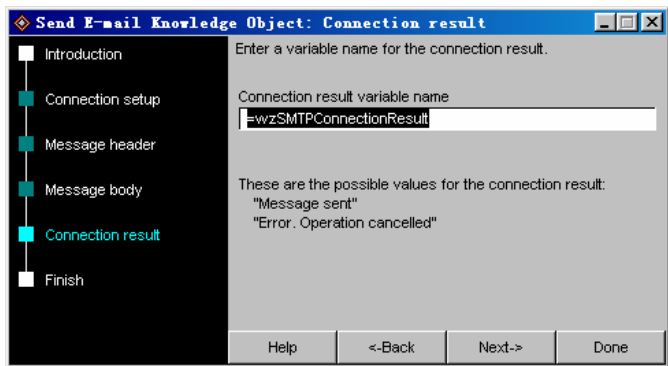


图 15-14 “Connection result”对话框

(11) 在该对话框中设置返回的结果，其中：

“Connection result variable name”文本框：在该文本框中输入一个变量，用来保存返回的信息。

(12) 在文本框中输入要保存结果的变量后，单击“Next”按钮，进入“Finish”对话框中，如图 15-15 所示。

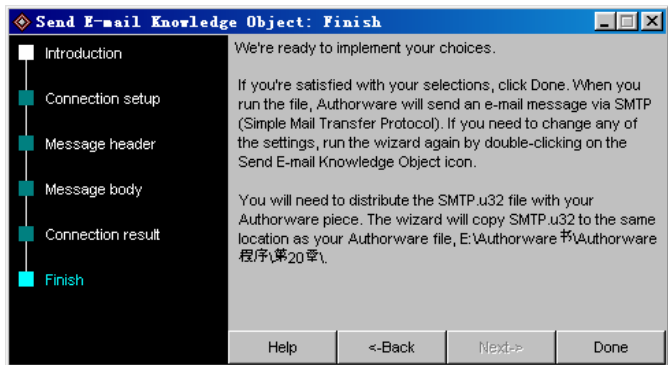


图 15-15 “Finish”对话框

(13) 该对话框中是一些提示信息，如果用户此时希望重新设置信息提示框，可以单击“Back”按钮退回到原来地方重新设置。如果用户单击“Done”按钮完成设置后，希望重新设置，可以双击该知识对象图标，重新运行该向导，进行重新设置即可。

(14) 单击“Done”按钮完成设置。此时该知识对象向导程序全部完成。

当应用程序执行到该知识对象时，将会向指定的 Email 信箱发送一封邮件。

第16章 Authorware 的高级功能

- 使用 OLE 对象
- ActiveX 控件的使用
- 数据库的调用——ODBC
- Flash 动画的使用
- QuickTime 电影的使用
- 动态 Gif 文件的使用

16.1 使用 OLE 对象

对于软件开发人员来说，OLE（Object Linking and Embedding，对象链接和嵌入）技术的引入是一个福音。OLE 的引入，可以减轻程序员的负担，在嵌入 OLE 的应用程序中使用可重复使用的、预先建好的功能模块，这样在解决相似问题时就可以大大缩短程序开发周期，削减开发费用。

OLE 技术的引入，使得在 Authorware 应用程序中可以访问 Windows 环境下的其他应用程序。因为在 Windows 环境下的有些对象不是 Authorware 所独有的，而是由其他应用程序所提供的，而且这些应用程序的某些对象特性是对外开放的，所以我们就可以通过 OLE 技术对这些对象的特性进行编程开发而实现在自己的应用程序中使用这些外部应用程序所提供的对象功能。

16.1.1 创建一个 OLE 对象

要创建一个 OLE 对象，必须先打开一个应用程序。可以插入 OLE 对象的图标包括【显示】图标、【交互】图标、【框架】图标等可以打开展示窗口的图标。双击要插入 OLE 对象的图标，打开展示窗口，然后选择菜单[插入]→[OLE 对象...]命令，打开“插入对象”对话框，如图 16-1 所示。



图 16-1 插入对象对话框

下面对该对话框的一些选项进行简单介绍：

- “新建”单选钮：该对话框中，默认选项为选中“新建”单选钮，此时对象类型列表框中列出可以新建的对象类型。用户可以从列表框中选择合适的类型，创建一个新的该类型的对象。
- “从文件创建”单选钮：选择该单选钮，对话框变为如图 16-2 所示。

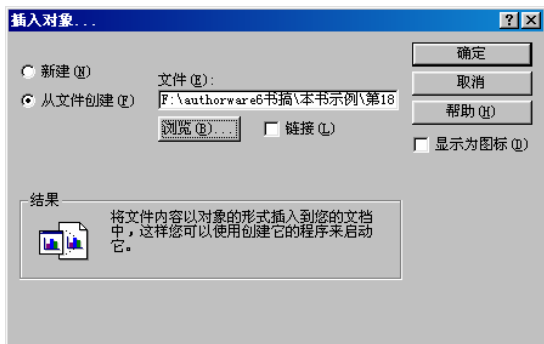


图 16-2 选中“从文件创建”单选钮后的对话框

此时可以在“文件”文本框中输入要嵌入的文件路径。当然，也可以直接单击“浏览”按钮，在弹出的“浏览”对话框中选择文件，然后单击“插入”按钮即可将该文件添加到“文件”文本框中。

如果选中“浏览”按钮右边的“链接”复选框，表示将该对象链接到应用程序中。

- “结果”选项：该选项是上述操作的描述信息。
- “显示为图标”复选框：选中该复选框时，嵌入或链接的对象在展示窗口中以一个图标形式显示，如图 16-3 所示。

16.1.2 嵌入一个 OLE 对象

在插入对象对话框中，如果没有选择“链接”复选框，则表示将当前对象嵌入到 Authorware 应用程序中。我们假设选择了“从文件创建”单选钮，并且选中了一个 Word 文档，然后单击插入对象对话框中的“确定”按钮将选中的 Word

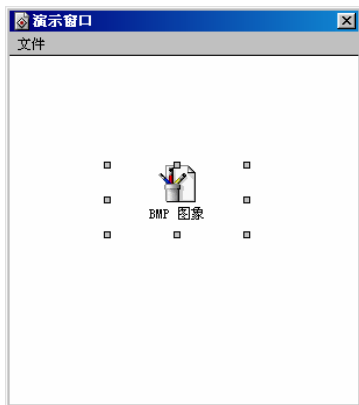


图 16-3 插入对象显示为图标

文档嵌入到应用程序中，如图 16-4 所示。

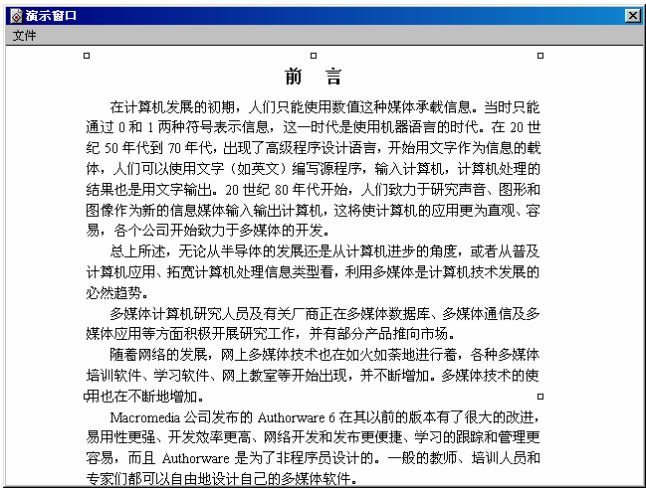


图 16-4 嵌入 Word 文档

此时只要双击该 Word 文档，即可进入该对象的编辑窗口中，如图 16-5 所示。

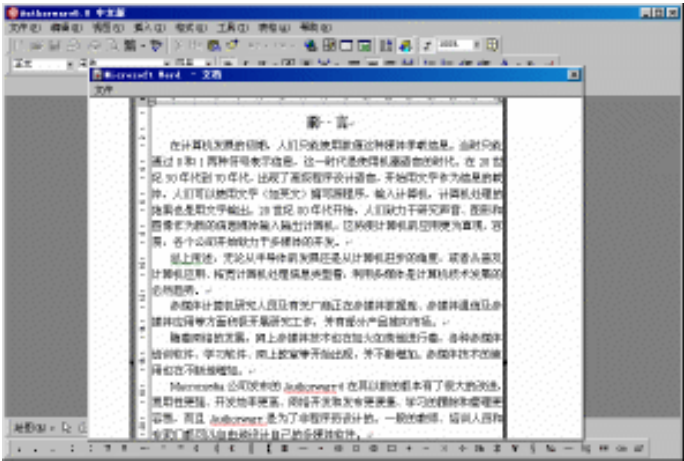


图 16-5 编辑嵌入的 Word 文档

屏幕上会出现 Word 的界面，此时就可以如象在 Word 中一样编辑该文档了。其实，OLE 就是调用 Windows 其他的应用程序为自己的应用程序服务。当在应用程序中要编辑该对象时，操作系统会自动调用其所在的应用程序，然后对其进行修改。这样就实现了该应用程序中的对象由其他应用程序来管理的现象。

如果要回到 Authorware 界面中，只要在该 Word 文档外部单击一下即可返回。

16.2 ActiveX 控件的使用

ActiveX 控件以前也被称为 OLE 控件。每一种 ActiveX 控件都有其单独的一节面向任务的概述文档，这些概述的目的并不是大体在联机帮助中提供的语言参考，而是为了给使用这些控件的新用户提供一些来龙去脉。

ActiveX 控件是 Authorware 的扩充部分，使用 ActiveX 控件与使用 OLE 对象有一些相似。在应用程序中加入 ActiveX 控件后，它将成为开发和运行环境的一部分，并为应用程序提供新的功能。

因为开发人员可以把经过测试的标准部件装配成应用程序，所以应用程序软件开发方法可以缩短编程时间，并开发出更强有力的产品。

ActiveX 控件是 Microsoft 的元素软件标准。ActiveX 虽然在以前也成为 OLE，但如今它已经囊括了 OLE 的所有技术和功能，并且具有了许多的新特性和新功能，以适应网络技术的发展。

16.2.1 添加一个 ActiveX 控件

要创建一个 OLE 对象，必须先打开一个应用程序。然后将粘贴指针指在要添加 ActiveX 控件的位置，然后选择菜单[插入]→[控件]→[ActiveX...]命令，打开“选择 ActiveX 控件”对话框，如图 16-6 所示。

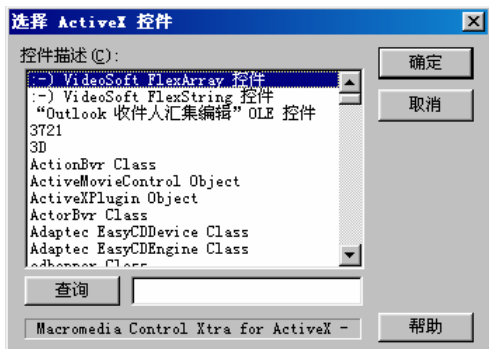


图 16-6 选择 ActiveX 控件对话框

在该对话框中，上面的“控件描述”列表框，可以在控件描述列表框中选择要插入的控件，然后单击“确定”按钮即可将选中控件添加到流程线上，此时系统将弹出“ActiveX 控件属性”对话框，如图 16-7 所示。



图 16-7 ActiveX 控件属性对话框

可以在“查询”按钮右边的文本框中输入控件关键字，然后单击“查询”按钮即可在上面的控件描述列表框中查找到指定关键字的 ActiveX 控件。

在 ActiveX 控件属性对话框中，包含有三个标签页：

- (1) “Properties (属性)”标签页：该 ActiveX 控件的属性。



该属性对话框与 Visual Basic 中的属性窗口极为相似，可以选择属性标签页中的一个属性，然后在上面的文本框中修改该属性的值，有时还会出现文本框变为灰显，表示该属性此时不能更改其属性值；如果该属性只有系统给定的几个值，则上面的文本框变为下拉框，可以从下拉框中选择选项；如果是一些需要其他选择的属性，该文本框右侧会出现一个按钮，单击该按钮会弹出一个对话框，让用户在该对话框中选择合适的值，这种属性有颜色（打开一个调色板）、文件（打开一个标准 Windows 对话框）等。

Authorware 提供了对应的系统函数来控制 ActiveX 控件的属性：

● SetSpriteProperty

语法：SetSpriteProperty (@"SpriteIconTitle", #Property, Value)

说明：该函数可以在一个【计算】图标中设置指定 ActiveX 控件的属性值。参数 SpriteIconTitle 是该 ActiveX 控件图标的图标名，参数 Property 是该 ActiveX 控件的要进行设置的属性，参数 Value 是要设置的属性值。

可以使用系统函数 SetIconProperty 设置 ActiveX 控件图标的属性。

● GetSpriteProperty

语法：Result := GetSpriteProperty (@"SpriteIconTitle", #Property)

说明：该函数是读取参数 SpriteIconTitle 指定的 ActiveX 控件图标的参数 Property 指定的属性值。

可以使用系统函数 GetIconProperty 获取 ActiveX 控件图标的属性。

（2）“Methods（方法）”标签页：该 ActiveX 控件的控制方法。

选择方法标签页，如图所示，出现该 ActiveX 控件提供的调用方法，如图 16-8 所示。



图 16-8 ActiveX 控件属性对话框的方法标签页

在该标签页中选择方法列表中的一个方法，在上面的的函数描述文本框中会显示出当前选中的方法的描述信息，其中包括该方法的有无返回值、返回值类型、参数个数、以及各个参数的类型等。

Authorware 提供了对应的系统函数来调用 ActiveX 控件的方法：

● CallSprite

语法：Result := CallSprite (@"SpriteIconTitle", #Method [, Argument...])

说明：该函数是调用参数 SpriteIconTitle 指定的 ActiveX 控件图标的的一个方法，参数 Method 用来指定要调用的方法，参数 Argument...是调用方法需要带的参数值，它可能有多个参数。该方法有几个参数，后面就应该带有几个参数值。如果该方法没有参数，可以不输入参数值。

（3）“Events（事件）”标签页：该 ActiveX 控件的事件列表。

选择事件标签页，出现该 ActiveX 控件提供的事件，如图 16-9 所示。

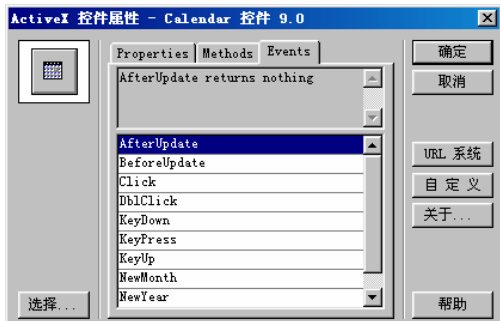


图 16-9 ActiveX 控件属性对话框的事件标签页

事件标签页提供了 Authorware 和该 ActiveX 控件之间传递信息的时间列表。可以在 Authorware 应用程序中的【计算】图标或属性对话框中来判断用户对该 ActiveX 控件做出了何种动作，Authorware 又该如何获取该事件并做出相应的响应动作。

(4) 在 ActiveX 控件属性对话框中还有几个选项：

- “选择”按钮：单击该按钮，可以打开选择 ActiveX 控件对话框，重新从对话框中选择新的 ActiveX 控件。
- “URL 系统”按钮：单击该按钮可以打开“ActiveX 控件 URL 系统”对话框，如图 16-10 所示。



图 16-10 ActiveX 控件 URL 系统对话框

在该对话框中，“Classid”文本框显示的是该 ActiveX 控件的 ClassID 号。“Download from URL”文本框是用来输入下载该 ActiveX 控件的 URL 网址，输入的 URL 必须完全匹配该 ActiveX 控件的 URL 网址，否则下载失败。在“Version to”文本框中输入该 ActiveX 控件版本号，如果要使用最近的版本，保持缺省的“-1, -1, -1, -1”。版本号也必须完全匹配该 ActiveX 控件的版本号，否则也会下载失败。

(5) “自定义”按钮：该按钮可以设置当前 ActiveX 控件的一般属性。单击该按钮，可以打开一个“Authorware 属性”对话框，如图 16-11 所示。



图 16-11 Authorware 属性对话框

在该对话框中可以设置当前 ActiveX 控件的一般属性。
“关于”按钮：单击该按钮，可以打开一个“关于日历控件”对话框，如图 16-12 所示。

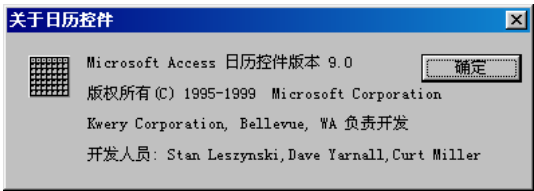


图 16-12 关于对话框

该对话框主要是描述该控件插件信息。
当在 ActiveX 控件属性对话框中的设置完成后，单击“确定”按钮即可，此时流程线上出现新添加的 ActiveX 控件图标，如图 16-13 所示。

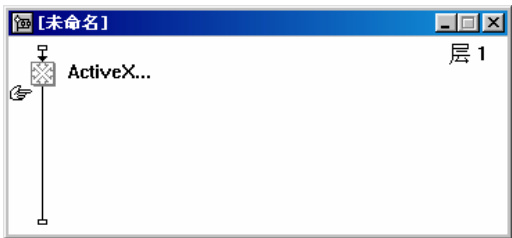


图 16-13 添加 ActiveX 控件图标后的流程线

16.2.2 编辑 ActiveX 控件图标

- 当在设计窗口的流程线上添加一个 ActiveX 控件后，可以对其进行编辑。
- 命名 ActiveX 控件图标
将该 ActiveX 控件图标重新命名。由于添加的 ActiveX 控件都使用一个缺省的名称“ActiveX...”，所以为 ActiveX 控件图标重新命名，最好命名的名称与该图标的功能相称。
 - 修改 ActiveX 控件图标属性
如果要修改 ActiveX 控件图标属性，双击该 ActiveX 控件图标，即可打开其属性对话框，如图 16-14 所示。

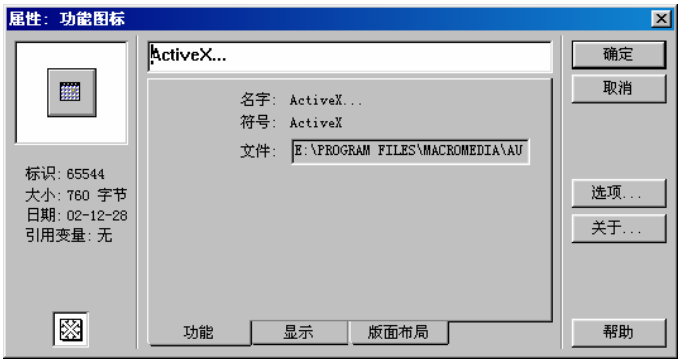


图 16-14 ActiveX 控件图标属性对话框

该对话框有三个标签页，“功能”标签页显示的是该 ActiveX 控件的一些信息，而“显示”标签页和“版面布局”标签页与【显示】图标的对应标签页相同，所以不再一一赘述。

单击属性对话框中的“选项”按钮，可以打开 ActiveX 控件属性对话框，如图 16-13 所示。此时就可以在 ActiveX 控件属性对话框中对该 ActiveX 控件的属性进行修改和编辑。

单击“关于”按钮，就打开了如图 16-21 所示的关于对话框。

● 编辑 ActiveX 控件在展示窗口中的显示范围

由于双击该 ActiveX 控件图标只会打开该控件属性对话框，虽然可以在属性对话框后面的展示窗口中移动 ActiveX 控件的位置，但不能编辑该 ActiveX 控件的大小。如果要编辑 ActiveX 控件在展示窗口中的显示范围，必须首先选择该 ActiveX 控件图标，然后选择菜单[窗口]→[演示窗口]命令或者使用快捷键<Ctrl>+1 打开展示窗口，然后选中该 ActiveX 控件，此时该 ActiveX 控件会出现八个句柄，单击句柄并拖动可以改变该 ActiveX 控件的显示范围；同样单击并拖动句柄以外的部分，可以移动该 ActiveX 控件，这与一般的显示对象的编辑完全相同，有疑问可以参考前面的内容。

16.2.3 使用 ActiveX 控件实例——媒体播放器

下面我们使用 Windows Media Player ActiveX 控件来设计一个可播放 Video 文件(*.dat)、Wave 文件(*.wav)、AVI 文件(*.avi)、Movie 文件(*.mov)、MP3 文件(*.mp3, *.m3u)、Media 文件(*.mmm)、Midi 文件(*.mid, *.rmi)、MPEG 文件(*.mpeg)等媒体文件，如图 16-15 所示

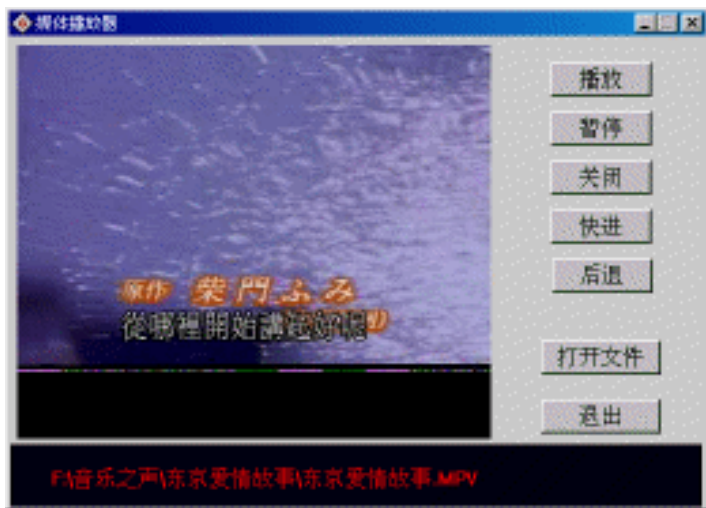


图 16-15 播放 VCD 影片

操作步骤为：

- (1) 新建一个文件，将其命名为“媒体播放器.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性]命令，打开文件属性对话框，设置应用程序的背景色为灰色，不显示菜单栏。单击“确定”按钮确定。
- (3) 拖动一个【计算】图标到设计窗口中流程线上，将其命名为“初始化”。
- (4) 双击【计算】图标，打开计算编辑窗口，在计算编辑窗口中输入下面语句：

```
ResizeWindow(500, 350)
File_Name:=""
PlayFlags:=FALSE
```

PauseFlags:=FALSE
StopFlags:=FALSE
NextFlags:=FALSE
BehindFlags:=FALSE

- (5) 关闭计算编辑窗口，系统弹出新建变量对话框，分别单击“确定”按钮即可。
- (6) 选择菜单[插入]→[控件]→[ActiveX...]命令，打开“选择 ActiveX 控件”对话框。
- (7) 从控件描述列表框中选择 Windows Media Player 控件单击“确定”按钮。
- (8) 直接单击弹出的 ActiveX 控件属性对话框中单击“确定”按钮确定。
- (9) 将该 ActiveX 控件图标命名为“媒体播放器”。
- (10) 拖动一个【交互】图标到流程线上，将其命名为“控制”。
- (11) 依次拖动七个【计算】图标到【交互】图标的右侧，分别将其命名为“播放”、“暂停”、“关闭”、“快进”、“后退”、“打开文件”和“退出”，响应类型为按钮响应。
- (12) 单击“播放”按钮响应类型图标，打开响应类型属性对话框，在“交互”标签页的“激活条件”文本框中输入“PlayFlags”。
- (13) 同样，在“暂停”、“关闭”、“快进”和“后退”响应类型属性对话框“交互”标签页的“激活条件”文本框中分别输入“PauseFlags”、“StopFlags”、“NextFlags”和“BehindFlags”。
- (14) 双击“控制”【交互】图标，打开展示窗口，将各个按钮进行对其排列。并在窗口的下面绘制一个黑色的矩形，并在矩形的中间输入红色文本“{File_Name}”，如图 16-16 所示。

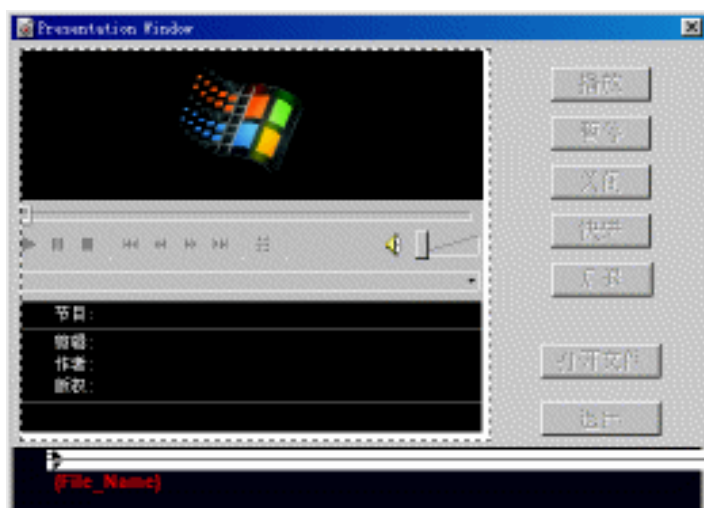


图 16-16 排列界面

- (15) 选择 ActiveX 控件图标，按下<Ctrl>+I 键打开展示窗口，在窗口中调整 ActiveX 控件的大小，使之正好位于按钮的左侧，如图 22-21 所示。

- (16) 双击“打开文件”【计算】图标，输入下面语句：

```
File_Name := OpenFileDialog ("", "Music01.dat", "", "", "媒体文件 (所有类型) |*.dat;*.wav;*.avi;*.mov;*.mp3;*.m3u;*.mmm;*.mid;*.rmi;*.mpeg|Video 文件 (*.dat) |*.dat|Wave 文件 (*.wav) |*.wav|AVI 文件 (*.avi) |*.avi|Movie 文件 (*.mov) |*.mov|MP3 文件 (*.mp3;*.m3u) |*.mp3;*.m3u|Media 文件 (*.mmm) |*.mmm|Midi 文件 (*.mid;*.rmi) |*.mid;*.rmi|MPEG 文件 (*.mpeg) |*.mpeg|所有文件 (*.*) |*.*")
```

```

if File_Name <> "" then
    PlayFlags:=TRUE
    PauseFlags:=TRUE
    StopFlags:=TRUE
    NextFlags:=TRUE
    BehindFlags:=TRUE
    SetSpriteProperty(@"媒体播放器", #filename, File_Name)
else
    File_Name:="打开文件错误！"
end if

```

(17) 双击“播放”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #play)
```

(18) 双击“暂停”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #Pause)
```

(19) 双击“关闭”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #Stop)
```

```
File_Name:=""
```

```
SetSpriteProperty(@"媒体播放器", #filename, File_Name)
```

```
PlayFlags:=FALSE
```

```
PauseFlags:=FALSE
```

```
StopFlags:=FALSE
```

(20) 双击“快进”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #FastForward)
```

(21) 双击“后退”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #FastReverse)
```

(22) 双击“退出”【计算】图标，输入下面语句：

```
CallSprite(@"媒体播放器", #Stop)
```

```
Quit()
```

(23) 此时程序已经设计完成，流程如图 16-17 所示。

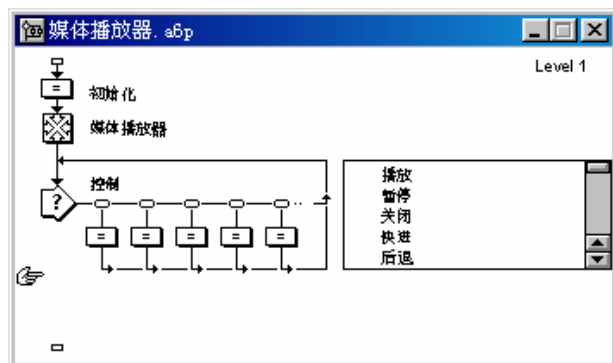


图 16-17 设计好流程

16.3 数据库的调用——ODBC

ODBC (Open DataBase Connectivity, 开放式数据库互连) 是一种建立数据库驱动程序的开放标准, 应用程序是通过一系列标准命令访问 ODBC (即 ODBC API) 的。一种专用的 ODBC 驱动程序原来匹配所需要被访问的数据源的类型。可以编写一个应用程序以访问标准化的 ODBC 命令和访问任何具有可用的 ODBC 驱动程序的数据源。一个程序被编写成为调用 ODBC API 之后, 不同的数据源在没有任何代码修改的情况下能够被此程序访问。

ODBC 的框架有下面的四个组件构成:

- 应用程序 (Application): 是用程序开发语言编写的应用程序, 它通过调用 ODBC API 函数及 SQL 语句来和数据库通信, 而 Authorware 是使用自身提供了几个 ODBC 的函数。
- 驱动程序管理器: 在逻辑数据源名称和实际驱动程序之间起链接器的作用。根据数据库引用的驱动程序, 驱动程序管理器装载并初始化该指定的 ODBC 驱动程序。
- 驱动程序: 是一个含有供应商专门制作的 ODBC API 执行代码的动态链接库。驱动程序负责和数据源接口、处理 SQL 语句、返回结构集合并执行错误检查。
- 数据源: 数据源有多个部分组成, 这些部分在 ODBC.INI 和系统注册表中定义的一个入口。

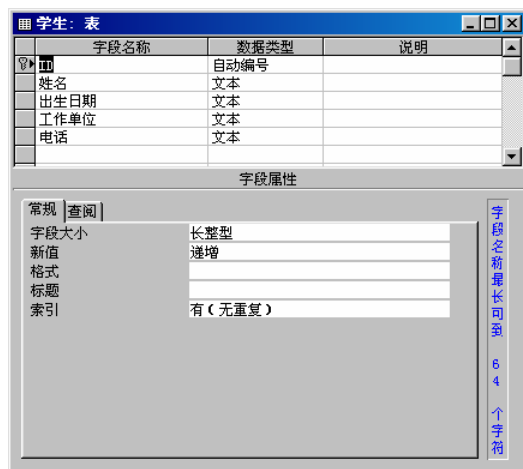
关于 ODBC 的其他功能和介绍, 请参考其他书籍。

下面我们以一个实例来说明如何使用 ODBC。

16.3.1 创建一个数据库

我们使用 Access 来创建一个小数据库。

(1) 打开 Access, 新建数据库, 将其命名为“通讯录.mdb”保存起来。在打开的“通讯录: 数据库”窗口中选择“使用设计器创建表”, 然后单击上面的“设计”按钮, 打开表设计窗口, 如图 16-18 所示。



字段名称	数据类型	说明
姓名	文本	
出生日期	文本	
工作单位	文本	
电话	文本	
自动编号	自动编号	

字段属性	
常规	字段大小: 长整型
查阅	新值: 递增
	格式:
	标题:
	索引: 有 (无重复)

图 16-18 设计表

(2) 在字段名称一栏中输入各个字段, 在数据类型一栏中选择合适的数据类型, 并在下面的“常规”标签页中设置属性。

(3) 设置完成后, 关闭该窗口, 弹出一个保存表的对话框, 如图 16-19 所示。



图 16-19 保存表

(4) 在文本框中输入“学生”，单击“确定”按钮。此时，在数据库窗口中添加了一个名为“学生”的表，如图 16-20 所示。

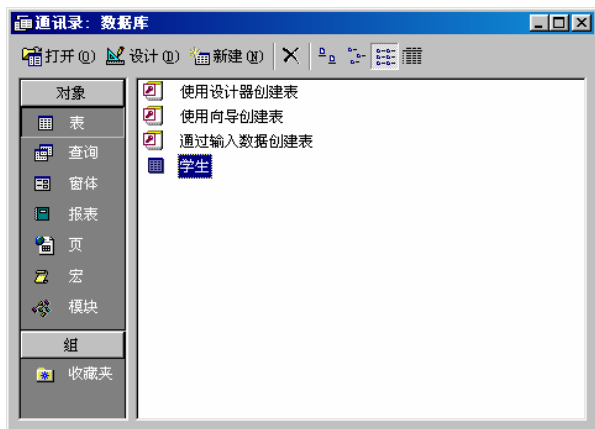


图 16-20 为数据库创建表

下面我们为表添加记录。

(5) 双击“学生”表，打开该表，然后在表中输入记录，如图 16-21 所示。

学生: 表						  	
	ID	姓名	出生日期	工作单位	电话		
	22	李青	76-7-3	西安电力机械厂	029-3784983		
	23	刘丽	77-12-10	成都托普集团	028-7600123		
	24	张师方	74-2-27	北京北大方正电子有限公司	010-45566767		
	25	王鑫鑫	77-5-4	济南浪潮集团	0531-6578787		
	* (自动编号)						
记录:   4   ** 共有记录数: 4							

图 16-21 添加记录

(6) 然后保存对表的修改。

此时就设计好了数据库。

16.3.2 使用 ODBC 链接数据库

要将数据库与 Authorware 链接起来，必须首先配置数据源和使用 ODBC。

首先打开控制面板，双击其中的“ODBC 数据源 (32 位)”图标，打开 ODBC 数据源管理器，如图 16-22 所示。



图 16-22 ODBC 数据源管理器

然后单击 ODBC 数据源管理器中的“添加”按钮，添加一个新的用户数据源，出现创建新数据源对话框，如图 16-23 所示。



图 16-23 创建新数据源对话框

选择第一项“Microsoft Access Driver (*.mdb)”，将 Access 数据库驱动作为新的数据源的驱动。单击“完成”按钮，出现 ODBC Microsoft Access 安装对话框，如图 16-24 所示。

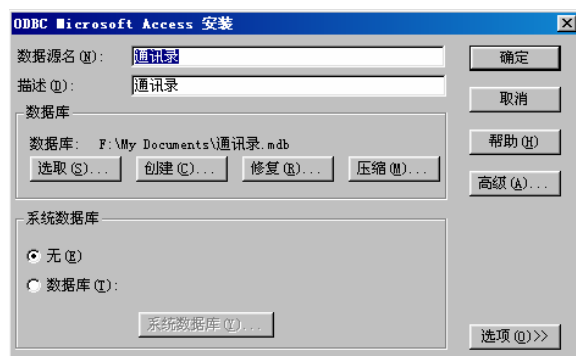


图 16-24 ODBC Microsoft Access 安装对话框

在“数据源名”文本框中输入新的数据源名称，在“描述”文本框中输入该数据源的描述信息。单击“选取”按钮，在弹出的“选定数据库”对话框中选取我们创建好的数据库“通讯录.mdb”。

完成后的对话框如图 16-22 所示。

单击“确定”即可将新数据源建成。

16.3.3 使用 Authorware 调用外部数据库

现在，我们就可以打开 Authorware 来调用外部的数据库了。

Authorware 提供了三个函数来对数据库进行操作。

- ODBCclose

语法：ODBCclose (ODBCHandle)

说明：该函数是将 ODBCHandle（数据源句柄）指定的 ODBC 数据源关闭。该 ODBCHandle 是由函数 ODBCOpen 返回的。

- ODBCExecute

语法：Data := ODBCExecute (ODBCHandle, SQLString)

说明：该函数使用 SQLString 参数指定的 SQL 查询语句对 ODBCHandle 指定的数据库进行查询操作。SQLString 必须是一个合法的 SQL 命令。查询得到的记录返回保存在变量 Data 中，这些记录之间由 Return 键隔开，不同的域之间由 Tab 键隔开。当使用 ODBCOpen 函数时会将错误信息返回到用户变量中。

该 ODBCHandle 是由函数 ODBCOpen 返回的。

- ODBCOpen

语法：ODBCHandle := ODBCOpen (WindowHandle, ErrorVar, Database, User, Password)

说明：该函数是打开参数 Database 指定的数据源。其中参数 WindowHandle 是该数据库窗口的句柄，可以直接添入该参数即可；参数 ErrorVar 保存打开该数据库时执行的错误信息；参数 Database 要打开数据库的数据源名；参数 User 和 Password 分别是该数据源的使用用户名和密码，在 ODBC 驱动程序中这两个参数不是必须的。

下面我们就使用这三个函数来调用设计好的数据库“通讯录.mdb”。

设计的通讯录可以翻看编辑好的数据库中的各个记录，如图 16-25 所示。



图 16-25 通讯录

操作步骤如下：

- (1) 新建一个文件，将其命名为“通讯录.a6p”。
- (2) 选择菜单[修改]→[文件]→[属性]命令，打开文件属性对话框，设置应用程序的窗口是可变的，不显示菜单项。单击“确定”按钮确定。
- (3) 拖动一个【显示】图标到设计窗口中流程线上，将其命名为“背景”。
- (4) 双击打开“背景”图标，导入一幅图片作为背景。并且输入一些提示文本，如图 16-26 所示。



图 16-26 设置背景

- (5) 拖动一个【计算】图标到流程线上，命名为“打开数据库”。
- (6) 在该【计算】图标中输入如下语句：

```
DB_DatabaseName:="通讯录"
--数据源名称
DB_ODBCError:=""
DB_ODBCHandle := ODBCOpen (WindowHandle,"DB_ODBCError", DB_DatabaseName,
"admin","")
--获取数据库句柄
```
- (7) 拖动一个【计算】图标到流程线上，命名为“查询数据”。
- (8) 在该【计算】图标中输入如下语句：

```
DB_SQLString:="SELECT * FROM 学生"
--SQL 语句，从“学生”表中选取所有信息
DB_ODBCData := ODBCExecute(DB_ODBCHandle, DB_SQLString)
--执行该 SQL 语句
ODBCClose(DB_ODBCHandle)
--关闭数据库
n:=1
string:=GetLine(DB_ODBCData,n)
PersonName:=GetLine(string,2,2,Tab)
PersonBir:=GetLine(string,3,3,Tab)
```

```
WorkPlace:=GetLine(string,4,4,Tab)
```

```
PhotoNum:=GetLine(string,5,5,Tab)
```

--获取第一个记录

(9) 拖动一个【交互】图标到流程线上，将其命名为“控制”。

(10) 拖动五个【计算】图标到【交互】图标的右侧，分别命名为“第一个”、“上一个”、“下一个”、“最后一个”和“退出”，响应类型为按钮响应。

(11) 单击“第一个”按钮响应类型图标，打开响应类型属性对话框，在“交互”标签页的“激活条件”文本框中输入“FirstFlags”。

(12) 同样，在“上一个”、“下一个”和“最后一个”响应类型属性对话框“交互”标签页的“激活条件”文本框中分别输入“PreFlags”、“NextFlags”和“LastFlags”。

(13) 新建这些变量，其初始值都设置成 TRUE。

(14) 按下<Shift>键双击【交互】图标，打开展示窗口，将五个按钮进行排列对齐在展示窗口的下方。

(15) 分别在文字“姓名”后面输入“{PersonName}”，在文字“出生日期”后面输入“{PersonBir}”，在“工作单位”后面输入“{WorkPlace}”，在“电话”后面输入“{PhotoNum}”，并且将它们与前面的文字对齐，如图 16-27 所示。



图 16-27 设置显示界面

(16) 打开“第一个”【计算】图标，输入下面语句：

```
n:=1
string:=GetLine(DB_ODBCData,n)
if n>=1 then
    PersonName:=GetLine(string,2,2,Tab)
    PersonBir:=GetLine(string,3,3,Tab)
    WorkPlace:=GetLine(string,4,4,Tab)
    PhotoNum:=GetLine(string,5,5,Tab)
    NextFlags:=TRUE
    LastFlags:=TRUE
```

```

n:=n-1
else
  PreFlags:=FALSE
  FirstFlags:=FALSE
end if

```

(17) 打开 “ 上一个 ” 【 计算 】 图标，输入下面语句：

```

n:=n-1
string:=GetLine(DB_ODBCData,n)
if n>=1 then
  PersonName:=GetLine(string,2,2,Tab)
  PersonBir:=GetLine(string,3,3,Tab)
  WorkPlace:=GetLine(string,4,4,Tab)
  PhotoNum:=GetLine(string,5,5,Tab)
  NextFlags:=TRUE
  LastFlags:=TRUE
else
  PreFlags:=FALSE
  FirstFlags:=FALSE
end if

```

(18) 打开 “ 下一个 ” 【 计算 】 图标，输入下面语句：

```

n:=n+1
string:=GetLine(DB_ODBCData,n)
if n<=LineCount(DB_ODBCData) then
  PersonName:=GetLine(string,2,2,Tab)
  PersonBir:=GetLine(string,3,3,Tab)
  WorkPlace:=GetLine(string,4,4,Tab)
  PhotoNum:=GetLine(string,5,5,Tab)
  PreFlags:=TRUE
  FirstFlags:=TRUE
else
  NextFlags:=FALSE
  LastFlags:=FALSE
end if

```

(19) 打开 “ 最后一个 ” 【 计算 】 图标，输入下面语句：

```

n:=LineCount(DB_ODBCData)
string:=GetLine(DB_ODBCData,n)
if n>=1 then
  PersonName:=GetLine(string,2,2,Tab)
  PersonBir:=GetLine(string,3,3,Tab)
  WorkPlace:=GetLine(string,4,4,Tab)
  PhotoNum:=GetLine(string,5,5,Tab)
  PreFlags:=TRUE

```

```
FirstFlags:=TRUE
n:=n+1
else
NextFlags:=FALSE
LastFlags:=FALSE
end if
```

(20) 打开“退出”【计算】图标，输入下面语句：

```
Quit()
```

(21) 此时程序已经设计完成，流程如图 16-28 所示。

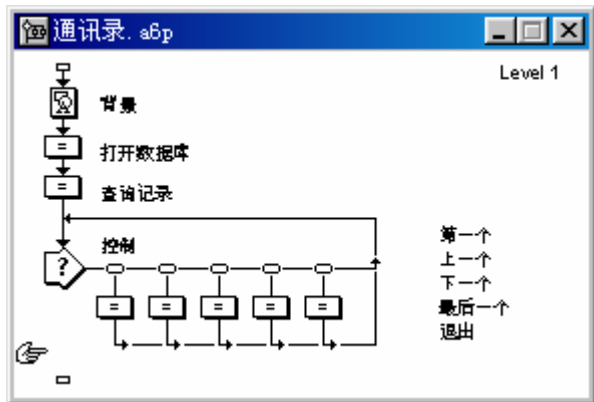


图 16-28 设计好的流程

16.3.4 插入 Flash 动画

Flash 是美国 Macromedia 公司出品的矢量图形编辑和动画创作专业软件，主要应用于网页设计和多媒体创作等领域，功能十分强大和独特。利用该软件制作的矢量图和动画具有文件尺寸小、交互性强、可无损放大、可带音效和兼容性好等特点（目前广泛使用的 GIFs、JPEGs 动画均不支持交互操作和音效、且色彩深度最高只能达到 256 色；一些高级的网络技术如 JAVA、DHTML 又往往存在可靠性和兼容性的问题），可创作出效果细腻而独特的网页和多媒体作品。另外，Flash 以 Shockwave 的 Stream（数据流）技术打破了网带宽的限制，通过 Flash 生成 Shockwave 的矢量作品，对于网络的带宽要求非常低。在播放时，用户可以任意放大动画和图形，而不损失其显示质量。

现在我们在 Authorware 应用程序中插入 Flash 动画。

选择[插入]→[媒体]→[Flash...]命令，出现“Flash Asset Properties (Flash 动画设置属性)”对话框，如图 16-29 所示。

下面对该对话框的各个选项命令予以简单介绍。

- “链接”文本框：该文本框中显示的是当前指定要播放的 Flash 文件。

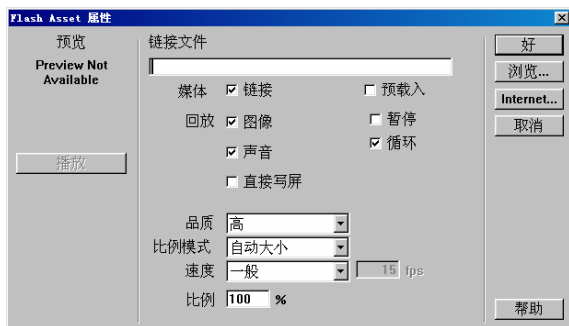


图 16-29Flash 动画设置属性对话框



- “媒体”选项：指定该 Flash 动画的加载方式。它包括两个选项：
 - “链接”复选框：作为外部媒体播放。
 - “预载入”复选框：在 Authorware 作品中对 Flash 动画预先装载。
- “回放”选项：对该 Flash 动画的回放方式。它包括五个选项：
 - “图像”复选框：显示图形。
 - “暂停”复选框：装载该 Flash 动画后先暂停。
 - “声音”复选框：播放该 Flash 动画附带的声音。
 - “循环”复选框：循环播放该 Flash 动画。
 - “直接写屏”复选框：将该 Flash 动画放在展示窗口的最上层。
- “品质”下拉框：设置该 Flash 动画的播放质量。它有四个选项：
 - “自动—高”选项：若播放系统的性能较高，可以高质量进行播放时，则自动调整以高质量画面播放。
 - “自动—低”选项：若播放系统的性能较低，无法高质量进行播放时，则自动调整以低质量画面播放。
 - “高”选项：强制以高质量画面播放（如果播放系统的性能较低，则播放速度较慢）。
 - “低”选项：强制以低质量画面播放，保证该 Flash 动画能够正常速度播放。
- “比例”下拉框：设置该 Flash 动画的缩放显示方式。它有三个选项：
 - “显示全部”选项：全部显示。
 - “无边界”选项：不显示边框。
 - “精确适配”选项：直接缩放到指定比例。
 - “自动大小”选项：自动调整大小，默认值。
 - “无比例”选项：不进行缩放。
- “速度”下拉框：设置该 Flash 动画的播放速度。它有三个选项：
 - “一般”选项：一般速度播放。
 - “固定”选项：选择该选项时，右侧的文本框被激活。可以在右侧的文本框中输入数值来指定播放速度。
 - “锁步”选项：始终以默认的播放速度播放。
- “比例”文本框：在该文本框中输入数值，控制该 Flash 动画的缩放比例。
- “播放”按钮：当引入一个 Flash 动画文件后，该按钮被激活。单击该按钮，可以预览动画的播放。
- “浏览...”按钮：单击该按钮，弹出一个选择要播放的 Flash 文件，如图 16-30 所示。

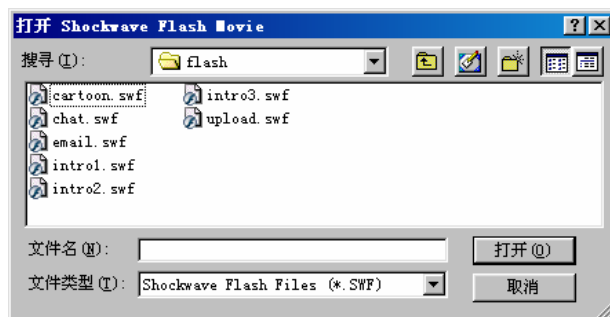


图 16-30 指定要播放的 Flash 文件

在该对话框中选择 Flash 动画文件，单击“打开”即可将该文件添加到 Flash 动画设置属性对话框。
当我们选择好一个 Flash 动画文件后，并在 Flash 动画设置属性对话框中设置了它的属性，单击“确定”按钮，即可将该文件添加到应用程序中。此时在流程线上显示出一个名为“Flash...”的图标，如图 16-31 所示。

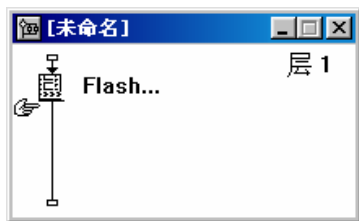


图 16-31 插入 Flash 动画

此时我们可以更改图标的标题，命名成一个合适的名称。
双击该“Flash...”图标，可以打开该图标的“功能图标”属性对话框，如图 16-32 所示，可以在该对话框中设置该 Flash 动画的显示方式和位置信息。其中，“显示”标签页和“版面布局”与一般图标的对应标签页相同，不再赘述。

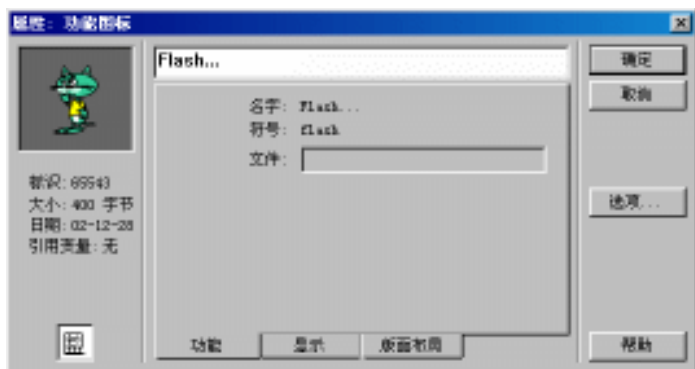


图 16-32 “功能图标”属性对话框

单击“功能图标”属性对话框中的“选项...”按钮，可以打开“Flash Asset 属性”对话框，然后就可以在该对话框中对其属性进行修改。

控制 Flash 动画的函数依然是 SetIconProperty、SetSpriteProperty、GetSpriteProperty 等
我们也可以使用 ActiveX 控件来插入 Flash 动画。

16.3.5 插入 QuickTime 电影

QuickTime 是 Apple 公司制作的数字电影格式，不但可以播放普通的 Mov 数字电影，而且还支持 Mpeg 数字电影文件、Wave、Mp3 等音乐文件的支持，同时还可以播放带有交互性的 QuickTime VR 虚拟现实影片等新文件格式。

选择[插入]→[媒体]→[QuickTime...]命令，出现“QuickTime Xtra Properties(QuickTime 插件属性)”对话框，如图 16-33 所示。



图 16-33 QuickTime 插件属性对话框

下面对该对话框的各个选项命令予以简单介绍。

- “回放”选项：控制 QuickTime 影片的回放方式。它包括四个选项：
 - “视频”复选框：选择该选项，显示影像。
 - “暂停”复选框：选择该选项时，加载该 QuickTime 影片文件后先暂停。
 - “声音”复选框：选择该选项，在播放 QuickTime 影片时附带播放声音。
 - “循环”复选框：选择该选项时，系统将循环播放该 QuickTime 影片。
- “取景”选项：控制该 QuickTime 影片在展示窗口中的显示方式。它包括两个选项：
 - “修剪”单选钮：选择该选项时，如果该 QuickTime 影片的尺寸比 Authorware 的展示窗口大，则 Authorware 会将多出的部分剪切掉。当选择该选项时，后面的“居中”复选框被激活，此时可以选中该复选框，使得该 QuickTime 影片在展示窗口中居中播放。
 - “比例”单选钮：选择该选项时，Authorware 会自动将该 QuickTime 影片全屏显示。
- “选项”选项：设置播放选项。它包括两个选项：
 - “直接写屏”复选框：选择该选项时，Authorware 会将该 QuickTime 影片显示在展示窗口的最上层。
 - “显示控制器”复选框：选择该选项，播放该 QuickTime 影片时会显示出播放控制条。
- “视频”下拉框：设置播放方式。它包括两个选项：
 - “与音轨同步”选项：选择该选项时，该 QuickTime 影片与声音同步播放。
 - “播放每一帧”选项：选择该选项时，Authorware 会播放每一帧。当可能会影响到播放速率以及与声音的同步。
- “比率”下拉框：设置播放速率。它包括三个选项：
 - “一般”选项：选择该选项时，该 QuickTime 影片会以普通播放速率播放。
 - “最大”选项：选择该选项时，该 QuickTime 影片将以最大播放速率播放。
 - “固定”选项：选择该选项时，右侧的文本框被激活，可以在文本框中输入该 QuickTime 影片的播放速率。
- “允许预先载入”复选框：选中该选项，允许进行预装载。
- “不加载”下拉框：设置卸载该 QuickTime 影片方式，它包括四个选项：
 - “3-一般1”选项：一般卸载。
 - “2-下一步”选项：执行完该 QuickTime 影片后卸载。
 - “1-最后”选项：执行完该应用程序后卸载。
 - “0-从不”选项：不卸载。

- “浏览...”按钮:单击该按钮,弹出一个选择文件对话框,在对话框中选择要播放的 QuickTime 影片文件。

当选择好 QuickTime 影片文件后,单击对话框中的“确定”按钮即可,此时流程线上出现 QuickTime 图标,如图 16-34 所示。

此时我们可以更改图标的标题,命名成一个合适的名称。

双击该“QuickTime...”图标,可以打开该图标的“功能图标”属性对话框,如图 16-35 所示,可以在该对话框中设置该 QuickTime 影片的显示方式和位置信息。其中,“显示”标签页和“版面布局”与一般图标的对应标签页相同,不再赘述。



图 16-34 插入 QuickTime 影片图标

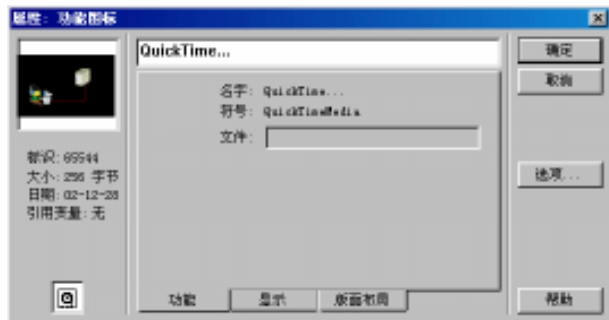


图 16-35 “功能图标”属性对话框

单击“功能图标”属性对话框中的“选项...”按钮,可以打开“QuickTime Xtra 属性”对话框,然后在对话框中对其属性进行修改。

控制 QuickTime 影片的函数依然是 SetIconProperty、SetSpriteProperty、GetSpriteProperty 等

16.3.6 插入动态 Gif 文件

我们可以通过 Authorware 内置的 Xtras 外挂直接插入动态 Gif 文件。

选择[插入]→[媒体]→[Animated GIF...]命令,出现“Animated GIF Asset 属性”对话框,如图 16-36 所示。



图 16-36 动态 GIF 文件属性对话框

下面对该对话框的各个选项命令予以简单介绍。

- “导入”文本框:显示或者输入要插入的动态 Gif 文件的文件名,包括该文件的地址,可以是绝对地址,也可以相对地址。
- “媒体”选项:该动态 Gif 文件的加载方式。它包括一个选项:
“链接”复选框:选择该选项时,该动态 Gif 文件作为外部媒体播放;如果不选择该选项,该动态

Gif 文件将作为程序的一部分，这样会增加程序的大小。

- “回放”选项：设置播放的回放方式。它包括一个选项：
“直接写屏”复选框：选择该选项时，Authorware 会将该动态 Gif 文件直接显示在展示窗口的最上层。
- “速度”下拉框：设置该动态 Gif 动画文件的播放速度。它包括三个选项：
“一般”选项：普通速度播放。
“固定”选项：当选择该选项时，右侧的文本框被激活，这时可以在该文本框中输入数值来调整播放速度。
“锁步”选项：始终按照默认的播放速度播放。
- “浏览...”按钮：单击该按钮，弹出一个选择文件对话框，在对话框中选择要播放的动态 Gif 文件。
- “Internet...”按钮：单击该按钮，弹出一个“Open URL”对话框，在该对话框的“File URL”文本框中输入要打开的动态 Gif 文件的网址即可。
- “取消”按钮：取消对该动态 Gif 文件设置。

当选择好动态 Gif 文件后，单击对话框中的“确定”按钮即可，此时流程线上出现动态 Gif 图标，如图 16-37 所示。



图 16-37 插入动态 Gif 图标

此时我们可以更改图标的标题，命名成一个合适的名称。

双击该图标，可以打开该图标的“功能图标”属性对话框，如图 16-38 所示，可以在该对话框中设置该动态 Gif 的显示方式和位置信息。其中，“显示”标签页和“版面布局”与一般图标的对应标签页相同，不再赘述。



图 16-38 “功能图标”属性对话框

单击“功能图标”属性对话框中的“选项...”按钮，可以打开“Animated GIF Asset 属性”对话框，然后在对话框中对其属性进行修改。

控制动态 GIF 图标 的函数依然是 SetIconProperty、SetSpriteProperty、GetSpriteProperty 等
在 Authorware 应用程序中还可以使用 UCD 函数插入 Director 文件等等。由于这些操作步骤与前面讲解的相似，所以就不再一一赘述，请读者自己亲自实践一下。

16.4 “影片欣赏”的制作

16.4.1 “影片欣赏”简介

在该实例中，我们设计的“影片欣赏”程序可以播放 Flash 动画、QuickTime 数字影片、动态 Gif 文件以及 VCD 影片等。

实例中，我们用到了 Authorware 6 的新增功能，Flash 动画的插入、QuickTime 插件的使用、动态 Gif 文件的插入、ActiveX 控件的使用以及图标模板的使用等。

16.4.2 “影片欣赏”的使用说明

进入该程序的主界面，如图 16-39 所示。



图 16-39 “影片欣赏”主界面

当您鼠标移动到左侧上方的五个方格上时，程序会自动移动出各个功能信息。其功能分别介绍如下：
最上方的一个按钮为播放 Flash 动画。当单击该方格时，程序会自动播放 Flash 动画。
第 2 个按钮为播放 QuickTime 影片。当单击该方格时，程序会自动播放安装 QuickTime 文件。
第 3 个按钮为播放动态 Gif 文件。当单击该方格时，程序会自动播放设置好的一个动态 Gif 文件。
第 4 个按钮可以播放 Video 文件 (*.dat)、Wave 文件 (*.wav)、AVI 文件 (*.avi)、Movie 文件 (*.mov)、MP3 文件 (*.mp3, *.m3u)、Media 文件 (*.mmm)、Midi 文件 (*.mid, *.rmi)、MPEG 文件 (*.mpeg) 等媒体文件。当单击该方格时，程序会显示播放时的控制按钮。
控制按钮从左到右分别是播放、暂停、关闭、打开文件、以及返回等。单击“打开文件”按钮，弹出“打开文件”对话框，选择要播放的文件，然后单击“打开”即可播放。
最后一个按钮是退出程序按钮，单击弹出一个对话框，如图 16-40 所示，选择“是”按钮退出；否则选择“否”按钮继续执行该程序。



图 16-40 播放各种 Video 影片

16.5 设计“影片欣赏”程序

其操作步骤如下：

- (1) 新建一个文件，将其命名为“影片播放”。
- (2) 选择[修改]→[文件]→[属性]命令，打开文件属性对话框，选择“大小”下拉框中的“根据变量”选项，在“属性”标签页中取消“菜单栏”复选框和“标题栏”复选框，然后单击“确定”按钮确定。
- (3) 拖动一个【计算】图标到设计窗口的流程线上，并将其命名为“初始化”。
- (4) 双击该【计算】图标，打开其计算编辑窗口，然后在窗口中输入语句：
 ResizeWindow(640,400)
 --初始化屏幕大小
- (5) 拖动一个【显示】图标到流程线上，将其命名为“背景”，从外部导入主界面图形。
- (6) 拖动一个【交互】图标到流程线上，将其命名为“选择影片”。
- (7) 在【交互】图标左侧拖动两个【群组】图标，分别命名为“ShowTips1”和“PlayFlash”，其中选择交互类型为热区交互。
- (8) 双击“ShowTips1”【群组】图标，打开二级设计窗口。
- (9) 在二级设计窗口中分别拖入一个【显示】图标、一个【擦除】图标和另一个【显示】图标，分别命名为“Tips”、“EraseTips”和“ShowTips”。
- (10) 按住<Shift>键并双击“ShowTips”【显示】图标，打开展示窗口，然后导入设计好的文字图片（由于该文字设计起来比较容易，所以我们就不再占用篇幅说明如何设计，读者可以从光盘中查看），并调整其位置正好在第1个方格右侧，如图16-41所示。



图 16-41 导入设计好的文字

(11) 按住<Ctrl>键双击“ShowTips”【显示】图标，打开其属性对话框，单击“显示”标签页中“特效”右侧的按钮，打开“特效方式”对话框，选择“填充方式/像素填充”，并在“周期”文本框中输入“0.5”，然后单击两次“确定”按钮确定。

(12) 按住<Shift>键并双击“Tips”【显示】图标，打开展示窗口，选择矩形工具，在导入的“残影效果的小球运动动画”图片的每一个字下绘制一个小方格，方格的大小与左侧的响应方格大小相近，如图 16-42 所示。



图 16-42 绘制几个方格

(11) 按住<Ctrl>键双击“Tips”【显示】图标，打开其属性对话框，单击“显示”标签页中“特效”右侧的按钮，打开“特效方式”对话框，选择“推出方式/从左往右推出展示”，并在“周期”文本框中输入“0.5”，然后单击两次“确定”按钮确定。

(12) 双击“Erase Tips”，打开其属性对话框。单击“擦除”标签页中“特效”右侧的按钮，打开“特效方式”对话框，选择“填充方式/像素填充”，并在“周期”文本框中输入“0.5”，然后单击两次“确定”按钮确定。

(13) 单击“Tips”【显示】图标中的方格，表示要擦除该【显示】图标中的内容。

(14) 选择[窗口]→[面板]→[模型调色板]命令或者直接按下快捷键<Ctrl>+3，打开图标模板，将主流程线上的“ShowTips1”【群组】图标拖动到图标模板上，如图 16-43 所示。



图 16-43 使用图标模板

(15) 双击“PlayFlash”【群组】图标，打开其二级设计窗口。

(16) 选择[插入]→[媒体]→[Flash...]命令，打开“Flash Asset 属性”对话框，单击对话框中的“浏览...”按钮，在打开的选择文件对话框中选中使用 Flash 设计好的“残影效果小球.swf”文件，然后单击“打开”确定。

(17) 双击该 Flash 图标，打开“功能图标”属性对话框，同时也打开了展示窗口。然后关闭。

(18) 按下快捷键<Ctrl>+1，打开展示窗口，在展示窗口中调整 Flash 的大小与位置，如图 16-44

所示。



图 16-44 调整导入的 Flash 动画的位置和大小

- (19) 关闭该二级设计窗口。
- (20) 从图标模板上将“ ShowTips1 ”【群组】图标图标拖动到“ 影片选择 ”【交互】图标的最右侧，并将其更名为“ ShowTips2 ”。
- (21) 双击“ ShowTips2 ”【群组】图标，打开其二级设计窗口。
- (22) 按住<Shift>键并双击“ ShowTips ”【显示】图标，将原来的“ 残影效果的小球运动动画 ”图片删除，然后导入“ QuickTime 数字电影的播放 ”图片，并调整其位置正好在第 2 个方格右侧。
- (23) 按住<Shift>键并双击“ Tips ”【显示】图标，修改绘制的方格的位置和个数。
- (24) 关闭该二级设计窗口。
- (25) 拖动一个【群组】图标图标拖动到“ 影片选择 ”【交互】图标的最右侧，并将其命名为“ PlayQuickTime ”。
- (26) 双击“ PlayQuickTime ”【群组】图标，打开其二级设计窗口。
- (27) 选择[插入]→[媒体]→[QuickTime...]命令，打开“ QuickTime Xtra 属性 ”对话框，单击对话框中的“ 浏览 ”按钮，在打开的选择文件对话框中选中安装 QuickTime 时的示例影片 Example.mov，然后单击“ 打开 ”确定。
- (28) 双击该 QuickTime 图标，打开“ 功能图标 ”属性对话框，同时也打开了展示窗口。然后关闭。
- (29) 按下快捷键<Ctrl>+1，打开展示窗口，在展示窗口中调整 QuickTime 影片的大小与位置。
- (30) 关闭该二级设计窗口。
- (31) 从图标模板上将“ ShowTips1 ”【群组】图标图标拖动到“ 影片选择 ”【交互】图标的最右侧，并将其更名为“ ShowTips3 ”。
- (32) 双击“ ShowTips3 ”【群组】图标，打开其二级设计窗口。
- (33) 按住<Shift>键并双击“ ShowTips ”【显示】图标，将原来的“ 残影效果的小球运动动画 ”图片删除，然后导入“ 动态 Gif 动画的播放 ”图片，并调整其位置正好在第 3 个方格右侧。
- (34) 按住<Shift>键并双击“ Tips ”【显示】图标，修改绘制的方格的位置和个数。
- (35) 关闭该二级设计窗口。
- (36) 拖动一个【群组】图标图标拖动到“ 影片选择 ”【交互】图标的最右侧，并将其命名为“ PlayGif ”。
- (37) 双击“ PlayGif ”【群组】图标，打开其二级设计窗口。

(38) 选择[插入]→[媒体]→[Animated GIF...]命令,打开“Animated GIF Asset 属性”对话框,单击对话框中的“浏览...”按钮,在打开的选择文件对话框中选择一个 Gif 文件,然后单击“打开”确定。

(39) 双击该 Gif 图标,打开“功能图标”属性对话框,同时也打开了展示窗口。然后关闭。

(40) 按下快捷键<Ctrl>+I,打开展示窗口,在展示窗口中调整 Gif 动画的大小与位置。

(41) 关闭该二级设计窗口。

(42) 从图标模板上将“ShowTips1”【群组】图标图标拖动到“影片选择”【交互】图标的最右侧,并将其更名为“ShowTips4”。

(43) 双击“ShowTips4”【群组】图标,打开其二级设计窗口。

(44) 按住<Shift>键并双击“ShowTips”【显示】图标,将原来的“残影效果的小球运动动画”图片删除,然后导入“VCD 影片的播放”图片,并调整其位置正好在第 4 个方格右侧。

(45) 按住<Shift>键并双击“Tips”【显示】图标,修改绘制的方格的位置和个数。

(46) 关闭该二级设计窗口。

(47) 拖动一个【群组】图标图标拖动到“影片选择”【交互】图标的最右侧,并将其命名为“PlayVCD”。

(48) 双击“PlayVCD”【群组】图标,打开其二级设计窗口。

(49) 拖动一个【计算】图标到设计窗口中流程线上,将其命名为“初始化”。

(50) 双击【计算】图标,打开计算编辑窗口,在计算编辑窗口中输入下面语句:

```
ResizeWindow(500, 350)
```

```
File_Name:=“”
```

```
PlayFlags:=FALSE
```

```
PauseFlags:=FALSE
```

```
StopFlags:=FALSE
```

```
NextFlags:=FALSE
```

```
BehindFlags:=FALSE
```

(51) 关闭计算编辑窗口,系统弹出新建变量对话框,分别单击“确定”按钮即可。

(52) 选择菜单[插入]→[控件]→[ActiveX...]命令,打开“选择 ActiveX 控件”对话框。

(53) 从控件描述列表框中选择 Windows Media Player 控件单击“确定”按钮。

(54) 直接单击弹出的 ActiveX 控件属性对话框中单击“确定”按钮确定。

(55) 将该 ActiveX 控件图标命名为“VCD 播放器”。

(56) 拖动一个【交互】图标到流程线上,将其命名为“控制”。

(57) 依次拖动五个【计算】图标到【交互】图标的右侧,分别将其命名为“播放”、“暂停”、“关闭”、“打开文件”和“返回”,响应类型为按钮响应。

(58) 单击“播放”按钮响应类型图标,打开响应类型属性对话框,在“交互”标签页的“激活条件”文本框中输入“PlayFlags”。

(59) 同样,在“暂停”和“关闭”响应类型属性对话框“交互”标签页的“激活条件”文本框中分别输入“PauseFlags”、“StopFlags”、“NextFlags”和“BehindFlags”。

(60) 使用按钮编辑器编辑各个按钮,可以直接使用第 8 章“媒体播放器”中的按钮图形。

(61) 双击“控制”【交互】图标,打开展示窗口,将各个按钮进行对其排列。并在窗口的下面绘制一个黑色的矩形,并在矩形的中间输入红色文本“{File_Name}”,如图 16-45 所示。

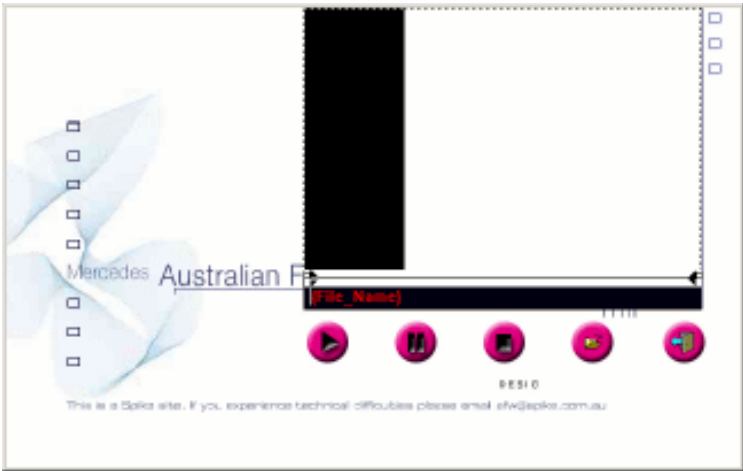


图 16-45 排列界面

(62) 选择 ActiveX 控件图标，按下<Ctrl>+1 键打开展示窗口，在窗口中调整 ActiveX 控件的大小，使之正好位于按钮的正上方。

(63) 打开“函数”窗口，从“分类”下拉框中选取“影片欣赏.a5p”，然后单击“载入”按钮，选择 UCD 函数“FielDlgs.U32”，如图 16-46 所示，单击“打开”按钮。

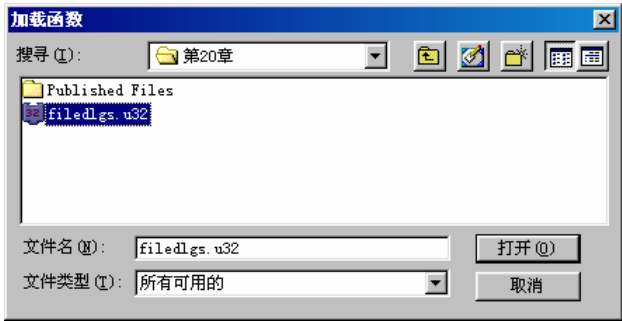


图 16-46 导入外部 UCD 函数

(64) 在显示的“FielDlgs.U32 中的自定义函数”对话框的函数列表中选择“OpenFileDialog”函数，然后单击“载入”按钮导入该函数。

(65) 双击“打开文件”【计算】图标，输入下面语句：

```
File_Name := OpenFileDialog ("", "Music01.dat", "", "", " 媒体文件 ( 所有类型 )  
|*.dat;*.wav;*.avi;*.mov;*.mp3;*.m3u;*.mmm;*.mid;*.rmi;*.mpeg|Video 文件 (*.dat)|*.dat|Wave  
文件 (*.wav)|*.wav|AVI 文件 (*.avi)|*.avi|Movie 文件 (*.mov)|*.mov|MP3 文件 (*.mp3;*.m3u)  
|*.mp3;*.m3u|Media 文件 (*.mmm)|*.mmm|Midi 文件 (*.mid;*.rmi)|*.mid;*.rmi|MPEG 文  
件 (*.mpeg)|*.mpeg|所有文件 (*.*)|*.*|")  
if File_Name <> "" then  
    PlayFlags:=TRUE  
    PauseFlags:=TRUE  
    StopFlags:=TRUE
```

```
NextFlags:=TRUE
BehindFlags:=TRUE
SetSpriteProperty(@"VCD 播放器", #filename, File_Name)
else
    File_Name="打开文件错误！"
end if
```

(66) 双击“播放”【计算】图标，输入下面语句：

```
CallSprite(@"VCD 播放器", #play)
```

(67) 双击“暂停”【计算】图标，输入下面语句：

```
CallSprite(@"VCD 播放器", #Pause)
```

(68) 双击“关闭”【计算】图标，输入下面语句：

```
CallSprite(@"VCD 播放器", #Stop)
```

```
File_Name:=""
```

```
SetSpriteProperty(@"VCD 播放器", #filename, File_Name)
```

```
PlayFlags:=FALSE
```

```
PauseFlags:=FALSE
```

```
StopFlags:=FALSE
```

(69) 双击“返回”【计算】图标，输入下面语句：

```
CallSprite(@"VCD 播放器", #Stop)
```

(70) 将“返回”按钮的响应方式改为“退出交互”方式，而其他四个按钮响应都为“重试”方式。

(71) 关闭该二级设计窗口。

(72) 从图标模板上将“ShowTips1”【群组】图标图标拖动到“影片选择”【交互】图标的最右侧，并将其更名为“ShowTips5”。

(73) 双击“ShowTips5”【群组】图标，打开其二级设计窗口。

(74) 按住<Shift>键并双击“ShowTips”【显示】图标，将原来的“残影效果的小球运动动画”图片删除，然后导入“退出程序”图片，并调整其位置正好在第5个方格右侧。

(75) 按住<Shift>键并双击“Tips”【显示】图标，修改绘制的方格的位置和个数。

(76) 关闭该二级设计窗口。

(77) 拖动一个【群组】图标图标拖动到“影片选择”【交互】图标的最右侧，并将其命名为“Exit”。

(78) 双击“Exit”【群组】图标，打开其二级设计窗口。

(79) 从“知识对象”窗口中拖动一个“消息框”知识对象到流程线上，设计好的信息提示框如图12-37所示。

(78) 拖动一个【计算】图标和【群组】图标到该【交互】图标右侧，响应类型为条件响应，分别命名为“x=6”和“x=7”，“x=7”的分支方式为“Exit Interaction”，“x=6”的分支方式任意，设置它们的匹配交互方式都为“When True”。

(79) “x=7”【群组】图标中无内容。

(80) 双击“x=6”【计算】图标，打开计算编辑窗口，在窗口中输入如下语句：

```
Quit()
```

——退出系统

(81) 按住<Shift>键并双击“选择影片”【交互】图标，打开展示窗口，调整各个热区的位置，其中“ShowTips1”和“PlayFlash”热区覆盖第1个方格，“ShowTips2”和“PlayQuickTime”热区覆盖第2个方格，“ShowTips3”和“PlayGif”热区覆盖第3个方格，“ShowTips4”和“PlayVCD”热区覆盖

第 4 个方格，而 “ShowTips5” 和 “Exit” 热区覆盖第 5 个方格，如图 16-47 所示。



图 16-47 调整各个热区的位置和大小

(82) 此时整个系统设计完成，程序流程如图 16-48 所示。

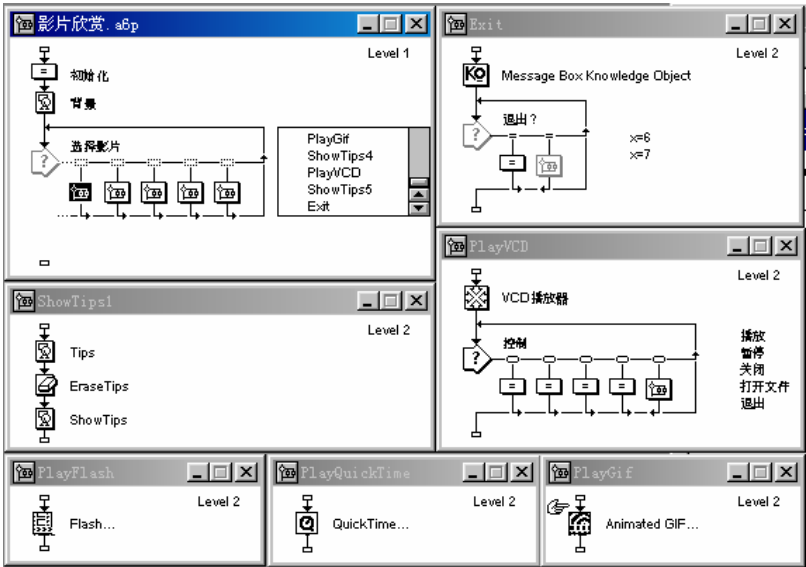


图 16-48 完成后的 “影片欣赏” 程序流程

第17章 Authorware 其他高级功能

- 输出媒体 (Export Media)
- 将.WAV 文件转换成.SWA 文件
- 查找 Xtras 文件
- RTF 对象编辑器

17.1 输出媒体

选择菜单[文件]▶[输出媒体]命令，打开“输出媒体”对话框，如图 17-1 所示。

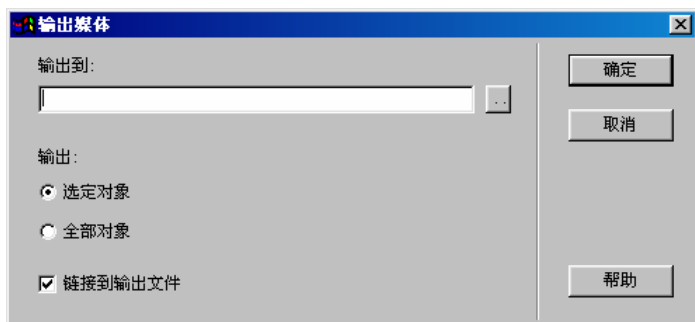


图 17-1 “输出媒体”对话框

在“输出到”文本框中输入要导出媒体的目标路径，也可以单击右侧的  按钮，打开“浏览文件夹”对话框，然后在该对话框中定位目标路径，如图 17-2 所示。



图 17-2 “浏览文件夹”对话框

选择“选定对象”单选钮，将只导出那些您已经选择的图标中嵌入的媒体；如果选择“全部对象”单选钮，则导出作品中或者库中所有嵌入的媒体。

选中“链接到输出文件”复选框，则要求系统移除所有您导出的嵌入媒体，代之的是使用一个到 Authorware 导出时刚创建的媒体文件的链接。如果您不选中该复选框，Authorware 则输出媒体到外部媒体文件中，但源媒体依然嵌入在作品中或者库中。

17.2 将.WAV 文件转换成.SWA 文件

选择菜单[Xtras]→[其他]→[转换 WAV 到 SWA...]命令，打开“转换 WAV 文件到 SWA 文件”对话框，如图 17-3 所示。



图 17-3 “转换 WAV 文件到 SWA 文件”对话框

对话框中各选项的含义如下：

- “文件转换”列表框：该列表框显示的是要进行转换的文件列表。
- “压缩设定”选项：压缩选项设置。它包括如下选项：

“位速率”下拉框：设置压缩位速率，该下拉框包括如下数值：160、128、112、80、64、56、48、32、24、16 和 8。

“精确度”选项：设置压缩精确性。如果选择“正常”单选钮，表示压缩精确性设置为一般精确性；

如果选择“精密”单选按钮，表示压缩精确性设置为高精确性，该选项为默认选项。

“立体声转换为单声道”复选框：选中该选项，将使立体声转换为单声道。

- “在覆盖文件前提示”复选框：选中该选项，当转换文件时，如果该文件已经存在，将显示出提示信息；否则将不提示信息直接覆盖已有文件。
- “转换文件的目标文件夹”选项：设置转换文件的目标文件夹。默认转换文件夹为当前作品路径。单击此按钮，打开“选择输出 SWA 文件的文件夹”对话框，如图 17-4 所示。

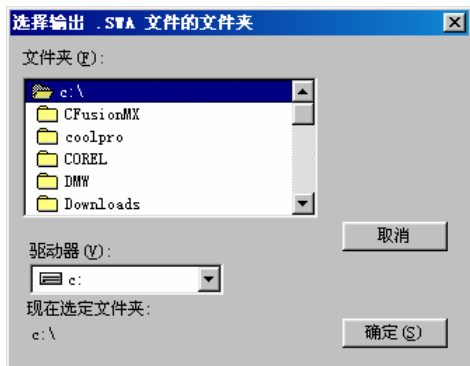


图 17-4 “选择输出 SWA 文件的文件夹”对话框

您可以从“驱动器”下拉框中选择当前计算机的盘符，然后在“文件夹”树状图中选择该盘符的目标文件夹。

- “添加文件”按钮：单击该按钮，系统弹出“选择 WAV 文件转换到 SWA 文件”对话框，如图 17-5 所示



图 17-5 “选择 WAV 文件转换到 SWA 文件”对话框

您可以从该对话框中选择要转换的 WAV 文件。选中后单击“打开”按钮确定。选中的 WAV 文件将显示在“文件转换”列表框。

- “删除”按钮：选中“文件转换”列表框中显示的 WAV 文件，然后单击该按钮可将该文件删除。
- “转换”按钮：单击该按钮，将会将“文件转换”列表框中的 WAV 文件全部转换。转换后的文件名与原文件名相同，只是文件扩展名改变为“.SWA”。
- “关闭”按钮：关闭对话框。

17.3 查找 Xtras 文件

选择菜单[命令]→[查找 Xtras]命令，打开“Find Xtras 文件”对话框，如图 17-6 所示。

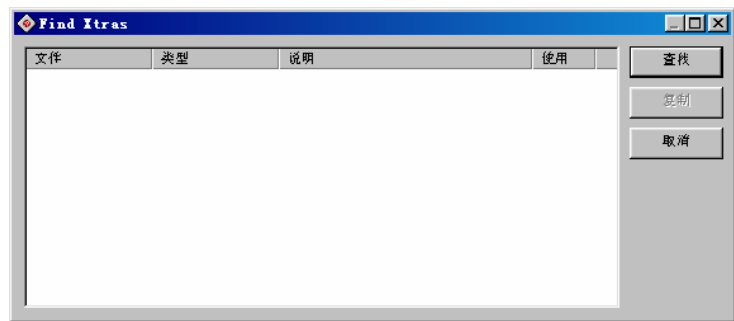


图 17-6 “Find Xtras”对话框

单击“查找”按钮，“Find Xtras”对话框开始查找当前作品中的所有 Xtras 文件，如图 17-7 所示。

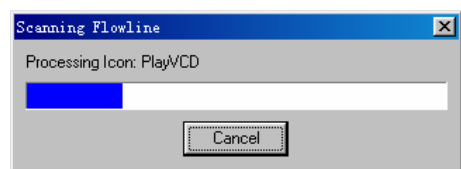


图 17-7 查找当前作品中的所有 Xtras 文件

当查找结束后，“Find Xtras”对话框将显示出作品中的所有 Xtras 文件，如图 17-8 所示。

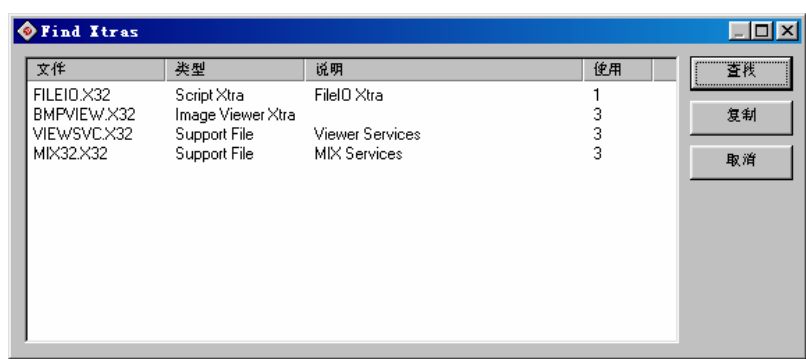


图 17-8 查找到的作品中所有的 Xtras 文件

在 Xtras 文件列表中，列出了查找到的作品中所有的 Xtras 文件，其中“文件”列显示的是 Xtras 文件名称，“类型”列显示的是该 Xtras 文件的文件类型，“说明”列显示的是该 Xtras 文件的说明信息，而“使用”列则显示的是作品中引用该 Xtras 文件的引用次数。

单击“复制”按钮，系统弹出“浏览文件夹”对话框，对查找到的作品中所有的 Xtras 文件进行保存，如图 17-9 所示。



图 17-9 “浏览文件夹”对话框

当您选定指定的文件夹后，系统会在该目标文件夹中创建一个“Xtras”子文件夹，然后将查找到的 Xtras 文件全部保存在“Xtras”子文件夹中，其中对于 Quicktime、Flash、Animate Gif 的 Xtras 文件，系统又会在“Xtras”子文件夹中分别创建三个子文件夹“QUICKTIMEASSET”、“FLASHASSET”和“ANIMATEDGIFASSET”，然后将 Quicktime 的 Xtras 文件（Qtasset.x32）、Flash 的 Xtras 文件（Flashast.x32）和 Animate Gif 的 Xtras 文件（Animgif.x32）保存在这三个子文件夹中。

“Find Xtras”对话框在打包发行作品时是十分有用的，您可以将作品需要的所有 Xtras 文件全部保存在打包发行的文件目录中，这样在发行的作品就可以单独运行了。

17.4 RTF 对象编辑器

选择菜单[命令]♦[RTF 对象编辑]命令，打开“RTF 对象编辑器”对话框，如图 17-10 所示。

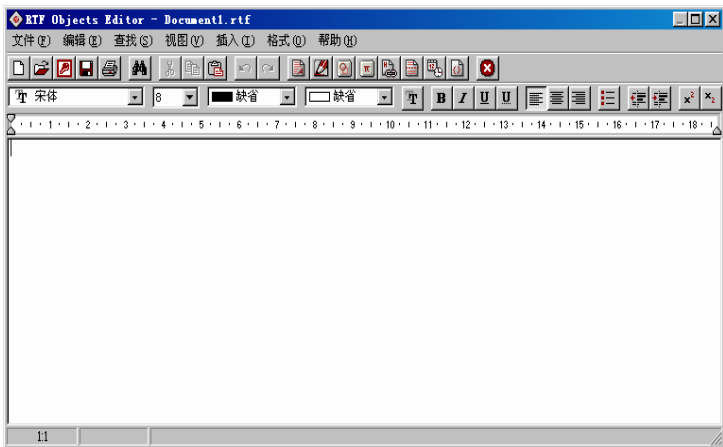


图 17-10 RTF 对象编辑器

RTF 对象编辑器是一个很强大的工具，它可以让您在 Authorware 之外创建全特性 RTF 象，然后将它们显示在 Authorware 展示窗口中。为了能够使用 RTF 对象编辑器提供的特性，您需要使用“创建 RTF 对象”知识对象来代替直接导入 RTF 文档到【显示】图标中（请参考“创建 RTF 对象”知识对象）。

为了理解 RTF 对象编辑器和 RTF 知识对象是如何工作的，清楚如下术语是非常重要的。

1 . RTF 文档 (RTF Document)

一个外部 RTF 文档是由 RTF 对象编辑器创建的。一个 Authorware 变量可以包含 RTF 代码，或者一个数据库的备忘录域也可以包含 RTF 代码。

2 . RTF 对象编辑器

一个可以从 Authorware [命令]菜单中运行的应用程序，它可以创作 RTF 文档。

3 . RTF 对象 (RTF Object)

一个由创建 RTF 知识对象或者 rtfCreate()函数创建的并可以在 Authorware 中显示的对象。RTF 对象是透明的，并可以对 Authorware 【交互】图标中的擦除选项（如在下次输入之前、在下次输入之后和在退出时）做出反应。Authorware 支持导入多个 RTF 实例。您也可以使用 Authorware 的【擦除】图标和【移动】图标来作用于 RTF 对象，而且您也可以将一个 RTF 对象导出为一个图片。

一个 RTF 文档在它成为一个 RTF 对象前需要做一些预先处理。在这个欲处理阶段，RTFObj.U32 取代嵌入的变量和表达式，并且导入外部的文本和图形。

4 . RTF 知识对象 (RTF Knowledge Objects)

该知识对象允许您创建和操作 Authorware 展示窗口中的 RTF 对象。不要将一个 RTF 对象和 RTF 知识对象相混淆。一个 RTF 对象是一个显示对象，而 RTF 知识对象是一个可以创建和操作 RTF 对象的知识对象。

第18章 Authorware 作品的发行与打包

- 应用程序的调试
- 作品的打包和发行
- 网络作品的打包和发行
- 网络发行的 Map 文件的编辑
- 如何制作自动运行的光盘
- 如何改变可执行文件的图标

18.1 应用程序的调试

在我们设计好应用程序时，我们需要对程序进行调试，然后再进行修改，再调试、修改，直到应用程序的功能完善。

Authorware 为开发人员提供了丰富的调试工具和检测工具。

18.1.1 使用“开始旗”和“终止旗”

我们可以使用“开始旗”和“终止旗”来调试应用程序的某个程序段。在要调试的程序段的开始放置“开始旗”，然后在该程序段的结束处放置“终止旗”，当此时运行程序时，程序将从“开始旗”处开始运行，遇到“终止旗”时结束，如图 18-1 所示，这样就可以方便地调试该程序段。

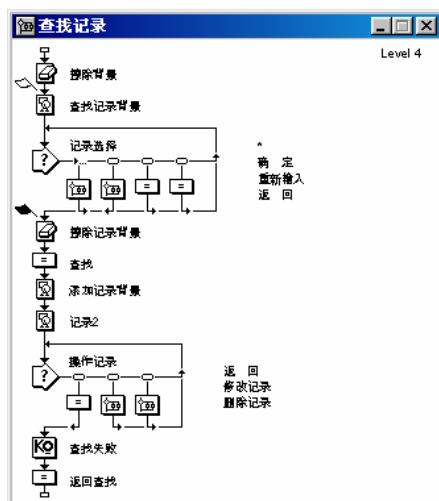


图 18-1 使用“开始旗”和“终止旗”

关于“开始旗”和“终止旗”的使用方法，请参考“开始旗和终止旗”一章。

18.1.2 使用控制面板

Authorware 为调试程序提供了一个有效的调试工具——控制面板。

选择菜单[窗口]→[面板]→[控制面板]命令，或者直接单击工具栏中的“控制面板”命令按钮，打开控制面板，如图 18-2 所示。



图 18-2 控制面板

控制面板可以控制程序的显示，并可以跟踪或调试应用程序。

单击控制面板中的“显示/隐藏跟踪”按钮，打开完整的控制面板，如图 18-3 所示。



图 18-3 完整的控制面板

图标类型和缩写如表 18-1 所示。

表18-1 图标类型和缩写对应表

图标类型	缩写	图标类型	缩写
【显示】图标 (DISPLAY)	DIS	【交互】图标 (INTERACTION)	INT
【移动】图标 (MOTION)	MTN	【计算】图标 (CALCULATION)	CLC
【擦除】图标 (ERASE)	ERS	【群组】图标 (MAP)	MAP
【等待】图标 (WAIT)	WAT	【数字电影】图标 (DIGITAL MOVIE)	MOV
【导航】图标 (NAVIGATE)	NAV	【声音】图标 (SOUND)	SND
【框架】图标 (FRAMEWORK)	FRM	【视频】图标 (VIDEO)	VDO
【决策】图标 (DECISION)	DES	-	-

我们可以使用函数 Trace 来跟踪变量、管理变量。我们先介绍一下 Trace 函数。

● Trace

语法：Trace (“String”)

说明：该函数用于调试 Authorware 程序。当程序遇到 Trace 函数时，就将其跟踪的 String 显示到 Authorware 的控制面板的跟踪窗口中。

示例：

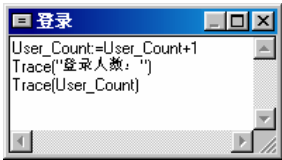
Trace (“开始交互！”)

当程序运行时遇到该语句，就会在跟踪窗口中显示出字符串“ 开始交互！”，提示用户开始交互了。

Trace (User_Count)

当程序运行时遇到该语句，就会在跟踪窗口中显示出变量 User_Count 的值，即显示出当前用户人数。

如果希望在程序的某个位置查看某个变量的值，您可以拖动一个【计算】图标到该处，然后在【计算】图标中使用 Trace 函数，参数为该变量，则在程序运行时，跟踪窗口中就会显示出该变量的值，如图 18-4 所示。



(a) 【计算】图标中使用 Trace 函数



(b) 跟踪窗口中显示出变量的值

图 18-4

使用 Trace 函数我们也可以直接在展示窗口中显示出某个变量的值，就是使用大括号将变量括起，如：登录人数：{User_Count}，等程序运行到该图标时，展示窗口中会自动显示出该变量的数值。

18.1.3 中断正在运行的程序

我们可以中断正在运行的程序，来查看此时各个变量的数值。

要中断正在运行的程序，可以选择菜单[调试]♦[暂停]命令，或者使用快捷键<Ctrl>+P，在程序中中断时，查看各个变量的值，编辑和修改部分程序段，然后在选择菜单[调试]♦[播放]命令，或者使用快捷键<Ctrl>+P，让程序继续执行。

当然，我们也可以使用控制面板中的按钮来中断和继续执行程序。

18.2 作品的打包和发行

当我们制作好 Authorware 的多媒体应用程序后，需要将该程序进行打包。根据当前媒体的传送方式，分为盘介质发行和网络发行。其中盘介质发行有包括磁盘发行和光盘发行。磁盘从存储量为 1.4M 的 3 寸软盘到几百兆的 ZIP 软驱，都可以用来发行作品。对于磁盘发行，如果一张磁盘装不下的话，可以将作品分成几个部分，分别装在几个磁盘中发行。由于现在使用光盘安装程序比较多，而且当前的程序都比较庞大，所以使用光盘发行是一个理想的发行方法。而网络发行又分为几种情况，如局域网发行、网间网发行等。

18.2.1 发行作品时所包含的文件

一个应用程序不但包括自身的许多内容，有时候还包括大量的外部文件（如库文件、链接文件、Xtras 文件等）。由于这些文件无法与应用程序一起打包，所以在发行的时候必须也要将这些文件一起发行。

一个应用程序所需要的外部文件，跟具体的应用程序有关，但是一般情况下，外部文件有以下几个。

- Runa6w32.exe 或者 Runa6w16.exe

使用这两个文件来运行打包后的应用程序文件。

- 应用程序中引用过的库文件

在发行作品的时候，要包括所有引用过的库文件。

- 所有链接的外部文件

在发行作品的时候，要包括所有链接的外部文件，如，图形文件、声音文件、数字电影文件、视频电影文件等媒体信息。

- 播放特殊类型的媒体文件的驱动程序

如果作品中包括一些特殊类型的媒体文件，还必须将播放其文件的驱动程序一起发行。

- 字体

为了确保作品能够在用户的计算机上正确运行，还要将作品中不常使用的字体一起发行。

- 用户的记录文件

- 压缩与解压软件

对于压缩的打包文件，需要为用户提供安装和解压的软件。

- 应用程序中使用的外部函数 UCD、DLL 文件

- 应用程序调用的 ActiveX 控件

- 应用程序中使用的 Xtra 外挂文件



18.2.2 外部文件的搜索路径

当交互式应用程序运行时，程序将按照一定的搜索路径搜索外部文件。

Authorware 默认的搜索路径顺序如下：

- 初次装载该文件时，该文件所处的目录。

当程序打包发行或者文件被移动后，Authorware 就不能找到该文件了，此时用户必须指定该文件的存放位置。

- 执行的应用程序所在的目录。
- Windows 目录。
- Windows\System 目录。
- Authorware 应用程序中指定的文件目录。
- Windows 目录下的 A6W.INI 文件。

在程序设置过程中，最好设定好文件搜索路径，有两种方法：

- (1) 在文件属性对话框“交互作用”标签页的“路径搜索”中输入各个搜索路径。
- (2) 使用系统变量 SearchPath 来定义搜索路径。如：SearchPath := SearchPath ^ “.media; \ucd”。

18.2.3 非网络作品的打包和发行

当应用程序调试成功后，需要将其打包发行。

1. 应用程序和库文件一起打包

选择菜单[文件]→[发布]→[打包]命令，打开“打包文件”对话框，如图 18-5 所示。

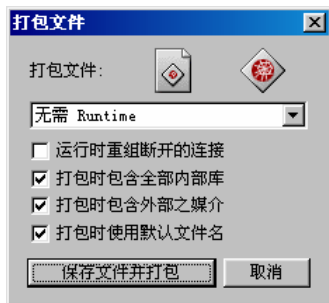


图 18-5 “打包文件”对话框

下面我们对该对话框的一些选项进行简单介绍。

- “打包文件”下拉框：设置文件的打包方式，它包括三个选项：

“无需 Runtime”选项：选择该选项，打包后的文件不是可执行的 EXE 文件，而是 A5R 文件，需要通过 Runa6w16.exe 或者 Runa6w32.exe 调用执行。

“应用平台 Windows 3.1”选项：选择该选项，打包后的文件是可以在 Windows 3.1 操作系统下直接运行的 EXE 文件。在 Windows 95/98/NT 下也可以执行，不过性能较低。

“应用平台 Windows 9x and NT variants”选项：选择该选项，打包后的文件是可以在 Windows 95/98/NT（包括 Windows 2000）下直接运行的 EXE 文件，不能在 16 位的操作系统下运行。

- “运行时重组断开的连接”复选框：选中该复选框时，在运行程序时会自动恢复断开的链接，这样会使程序的启动时间加长。如果您确认应用程序中的所有链接都是正常的，可以不选择该项。
- “打包时包含全部内部库”复选框：选择该选项时，所有与应用程序有链接关系的库文件将打



入打包文件中。如果不选中该选项，必须将这些库文件单独打包。

- “打包时包含外部之媒介”复选框：选中该选项时，所有应用程序中使用到的外部媒体信息都打入打包文件中。
- “打包使用默认文件名”复选框：选中该选项时，打包后的打包文件将与当前应用程序的文件名相同。
- “保存文件并打包”按钮：单击该按钮，系统弹出一个保存打包文件对话框，如图 18-6 所示。



图 18-6 保存打包文件对话框

在对话框中输入打包文件名，单击“保存”按钮，即可将当前应用程序按照上面的选择情况打包。

- “取消”按钮：单击该按钮，取消打包。

如果应用程序中调用了外部库文件，单击“保存文件并打包”按钮进行打包时，系统会自动弹出一个“打包库”对话框，如图 18-7 所示。

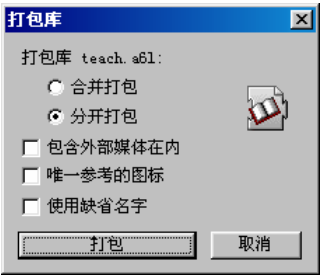


图 18-7 “打包库”对话框

下面我们对该对话框中的各个选项进行简单的介绍。

- “打包库”选项：该选项包括两个单选钮：
 - “合并打包”单选钮：将库文件与应用程序文件打包成一个文件。
 - “分开打包”单选钮：将库文件单独打包成一个文件。
- “惟一参考的图标”复选框：选中该选项时，只将与当前应用程序有链接关系的库图标进行打包。
- “使用缺省名字”复选框：选中该选项时，打包文件名与原库文件名相同。
- “包含外部媒体在内”复选框：选中该选项时，所有库文件中使用到的外部媒体信息都打入打包文件中。
- “打包”按钮：单击该按钮，系统打开“打开”对话框，先要求用户定位原库文件，如图 18-8 所示。

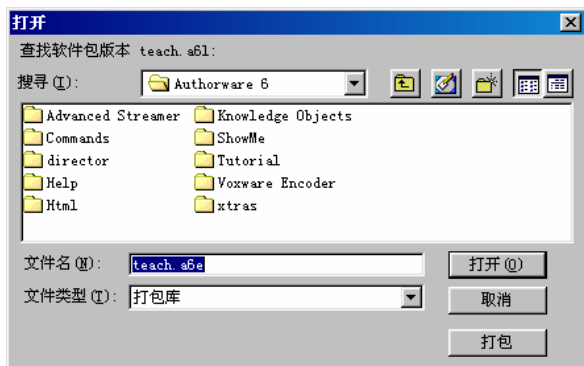


图 18-8 打包库文件对话框

在该对话框中，单击右下角的“打包”按钮，先将库文件打包，然后再打包主应用程序文件。

2. 库文件单独打包

如果希望单独打包库文件，可以首先将库窗口激活，然后选择菜单[文件]→[发布]→[打包]命令，打开“打包库”对话框，如图 18-9 所示。

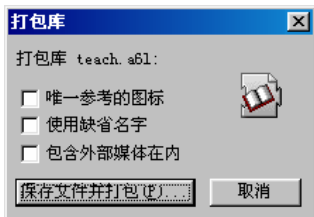


图 18-9 “打包库”对话框

该对话框包括三个选项：

- “唯一参考的图标”复选框：选中该选项时，只将与当前应用程序有链接关系的库图标进行打包。
- “使用缺省的名字”复选框：选中该选项时，打包文件名与原库文件名相同。
- “包含外部媒体在内”复选框：选中该选项时，所有库文件中使用到的外部媒体信息都打入打包文件中。
- “保存文件并打包”按钮：单击该按钮，单击该按钮，如果该库没有保存的话，系统会首先打开保存该库文件的对话框，如图 18-10 所示。



图 18-10 保存库文件对话框

保存该库文件后，系统又会弹出打包库文件对话框，如图 18-11 所示。

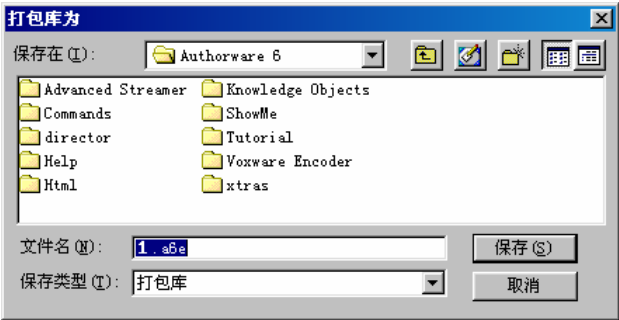


图 18-11 打包库文件对话框

- “取消”按钮：单击该按钮，取消打包。

3. 一键发布

选择菜单[文件]→[发布]→[发布设置]命令，打开“一键发布”对话框，如图 18-12 所示。



图 18-12 “一键发布”对话框

在发行过程中，一键发布结合和自动化了所有步骤。它使只是单击一下按钮就可以将应用程序发布到 Web、CD-ROW 或者企业网络中成为可能。

- 可以在同一时刻不同方式地打包和发行产品。例如，你可以仅仅在一个步骤中就可以将产品打包为非运行时文件（A6R 文件）、Web Player 文件（AAM 文件）和一个 Web 页面（HTM 文件）。
- 自定义发行方式，您可以重复使用设置好的发布设置。
- 自动识别和收集您发布产品中所需要的很多支持文件，如 Xtras 文件，DLL 文件和 UCD 文件等。
- 将你设置好的 WEB 文件 FTP 到远服务器上。

如果你需要发布多样的 Authorware 产品，你可以使用批发布特性在同一时间内将多个产品以不同方式发行，并且你可以跟踪发布的所有产品。

尽管一键发布可以帮助你定位文件中需要使用的 Xtras ,并可以在作品发布过程中将 Xtras 放置到合适的文件夹中。你也可以在发布产品前使用新增加的查找 Xtras 命令来定位 Xtras。

“一键发布”对话框包括了五个标签页，分别介绍如下：

(1) “指针或库”文本框

该文本框显示了当前要发行的文件名，该文件是 Authorware a6p 文件。如果希望发行其他文件，可以单击文本框右侧的  按钮，系统弹出一个弹出菜单，如图 18-13 所示。

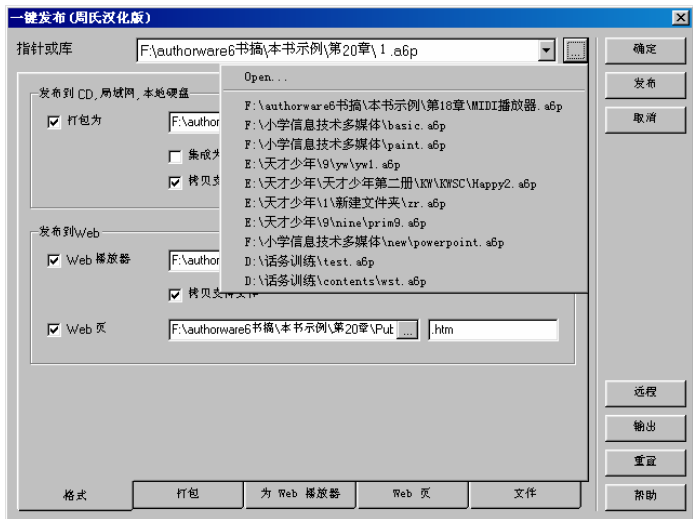


图 18-13 打开要发行的文件

该弹出菜单中列出了最近的文件，如果这些文件中没有要发行的文件，则可以单击“Open...”选项，打开“打开文件”对话框，如图 18-14 所示。

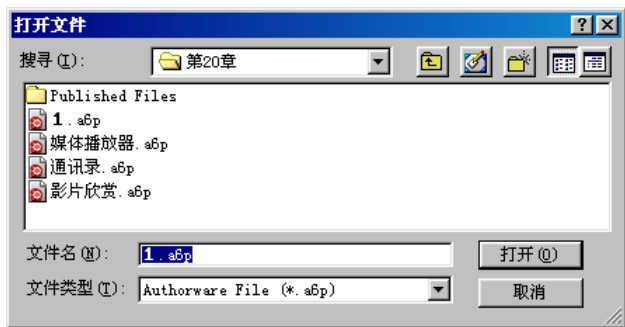


图 18-14 打开文件对话框

在该对话框中查找到要发行的文件，然后单击“打开”按钮将其添加到“指针或库”文本框中进行发行。

(2) “格式”标签页

该标签页主要是设置打包文件、网络播放器文件、以及网页文件的格式。

- “发布到 CD，局域网，本地硬盘”选项：设置为 CD、局域网、本地硬盘打包发行选项
- “打包为”复选框：将要发行的文件打包成的目标打包文件，该文件可以在右侧的文本框中输入，

包括文件名和路径；也可以单击文本框最右侧的...按钮，系统弹出“打包文件为”对话框，要求您定位打包文件位置，如图 18-15 所示。

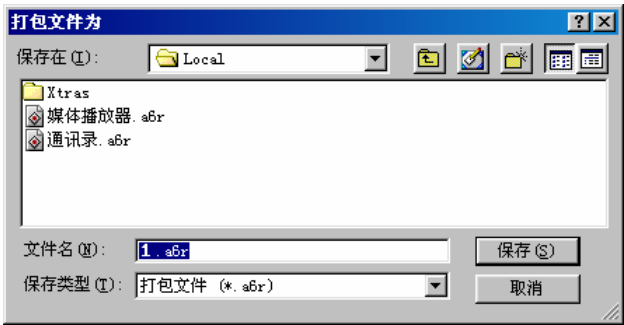


图 18-15 “打包文件为”对话框

“集成 Windows 9x and NT 变量的 Runtime”复选框：选择该选项，打包后的文件是可以在 Windows 9x/NT（包括 Windows 2000）下直接运行的 EXE 文件，不能在 16 位的操作系统下运行。

“拷贝支持文件”复选框：选中该选项，在打包时会将该文件所支持的所有文件一起打包。

- “发布到 Web”选项：设置网络打包发行选项

“Web 播放器”复选框：设置为网络播放器打包发行选项；在右侧的文本框中输入要网络发行的文件所在的文件路径，也可以不使用默认的文件路径，单击文本框右侧的...按钮，打开“浏览文件夹”对话框，在该对话框中定位文件路径，如图 18-16 所示。



图 18-16 “浏览文件夹”对话框

最右侧的文本框...aam 是网络播放器支持文件的扩展名，您可以修改该文件的扩展名，建议使用默认的文件扩展名。

“拷贝支持文件”复选框：选中该选项，在打包发行时会将该文件所支持的所有文件一起打包发行。

“Web 页”复选框：设置为网页打包发行选项；在右侧的文本框中输入要网页发行的文件所在的文件路径，也可以不使用默认的文件路径，单击文本框右侧的...按钮，打开“浏览文件夹”对话框，在该对话框中定位文件路径；同样可以在最右侧的文本框中修改网页文件的扩展名，如“.asp”。

(3) “打包”标签页

该标签页设置打包时的各个选项信息，如图 18-17 所示。



图 18-17 一键发布对话框的“打包”标签页

“打包所有库在内”复选框：将所有库打包到打包文件中。

“打包外部媒体在内”复选框：将所有外部媒体打包到打包文件中。

“仅为引用图标”复选框：只打包被库引用图标。

“重新断开的连接在 Runtime”复选框：在运行时解决断开的链接。

(4) “为 Web 播放器”标签页

该标签页设置为网络播放器文件 (*.aam) 打包时的各个选项信息，如图 18-18 所示。



图 18-18 一键发布对话框的“ For Web Player ”标签页

- “映射文件”选项：设置打包发行网络作品选项映射文件信息。映射文件是由 Authorware Web 打包创建的文件，使用它来引导 Authorware Web Player 如何下载和运行一个网络打包的作品。



“ 片段前缀名 ” 文本框：该文本框中是数据文件的文件头名称。可以在该文本框中输入新的文件头名称进行修改。

“ 片段大小 ” 下拉框：该文本框是记录每个数据文件的大小，默认为将源程序压缩为 16K 大小的多个数据文件包，此时是 56kbps Modem 方式。

该下拉框包括如下选项，如图 18-19 所示。

您可以选择不同的选项，以决定压缩后数据文件包的大小。当选择选项时，右侧的文本框是灰显的，表示依照不同的传输介质会有固定的数据文件包大小。

如果选择 “ Custom (自定义) ” 选项，则右侧的文本框被激活，您可以在文本框中输入数值进行修改文件大小，也可以调整微调按钮来调整数值。一般来说，该数值不应太大，否则客户端下载一个数据包的时间就会过长。

“ 显示安全对话框 ” 复选框：是否显示安全对话框。

“ Advanced Streamer ” 选项：设置打包发行是否使用高级流选项。

“ Use Advanced Streamer ” 复选框：设置打包发行是否使用高级流选项。如果选中该选项，下侧的各个选项都被激活。

“ CGI-BIN URL ” 文本框：设置要使用高级流所对应服务器 CGI-BIN 的 URL 地址，默认是本机 IIS 默认路径 scripts 路径下，即 http://localhost/scripts。

“ 输入 URL ” 选项：设置 Authorware Web Player 要使用的作品 Segments 文件位置。

“ 输出 URL ” 选项：指定您希望保存的作品文件的位置。

(5) “ Web 页 ” 标签页

该标签页设置为网页文件 (*.htm) 打包时的各个选项信息，如图 18-20 所示。

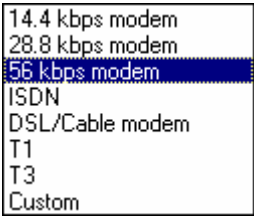


图 18-19 “ 片段大小 ” 下拉框选项

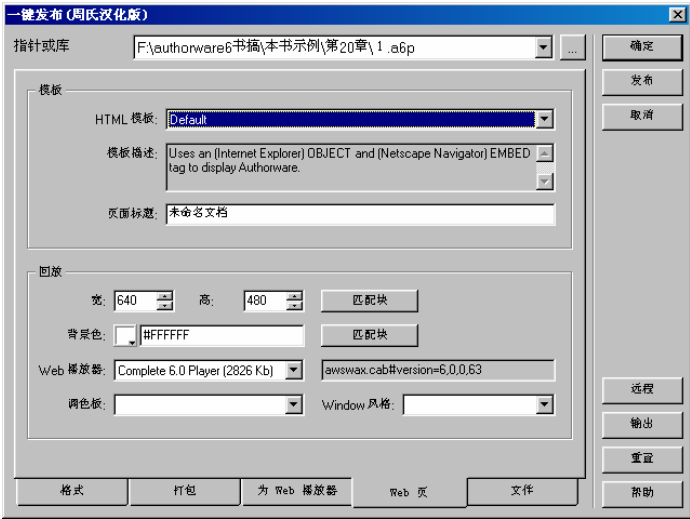


图 18-20 一键发布对话框的 “ Web 页 ” 标签页

- “ 模板 ” 选项：设置打包发行网页选项模板信息。
- “ 模板描述 ” 文本框：描述选中的 HTML 模板描述信息。
- “ 页面标题 ” 文本框：设置网页 Title 名称，默认名为 “ Untitled Document ”。
- “ 回放 ” 选项：设置播放作品时的播放信息。

(6) “文件”标签页

使用“文件”标签页可以查看和改变源文件和目标文件。该标签页设置打包时的各个文件信息，如图 18-21 所示。

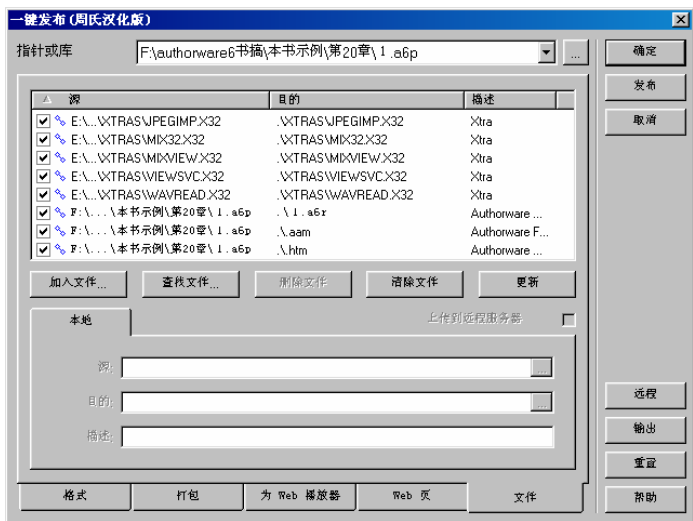


图 18-21 一键发布对话框的“文件”标签页

- “源”列：指定发行时需要的文件的文件名和路径。
- “目的”列：指明要发行文件的发行目标文件名和路径。
- “描述”列：显示要发行文件的描述信息。
- “加入文件”按钮：该按钮可以添加另外的、没有列出的文件到包中，如对 Flash、QuickTime、或者 ActiveX 的引用。
- “查找文件”按钮：该按钮是用于查找另外的、没有列出的文件，并将它们添加到包中。
- “删除文件”按钮：一般用于移去用户手工添加的文件。您可以从文件列表中选中手工添加的文件，然后单击该按钮将其删除。
- “清除文件”按钮：一般用于移去作品中所有支持的文件。单击该按钮，可以将文件列表中的所有文件删除。
- “刷新”按钮：为了重新定位移去文件和恢复断开的链接，可以单击该按钮。
- “上传到远程服务器”复选框：一般用于指明选中的文件是否上传到远方的服务器上。它在您重新发行作品而且您只想导入该服务器上的文件时很有用。
- “本地”标签页：该标签页主要显示选中文件的“源”、“目的”和“描述”信息。
- “Web”标签页：如果你选中了一个添加到文件列表中将要发行的 Xtra、U32、或者影片文件，该标签页就会显示在“本地”标签页的下一页，如图 18-22 所示。

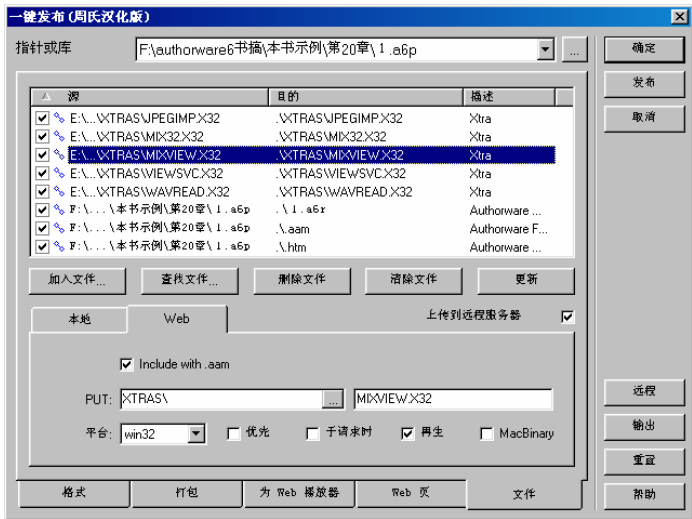


图 18-22 一键发布对话框“文件”标签页的“Web”标签页

一键发布对话框中的其他按钮含义

- “重置”按钮：单击该按钮，系统会弹出一个“重置设置”对话框，如图 18-23 所示。

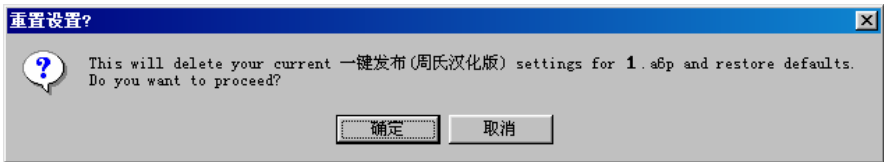


图 18-23 “重置设置”对话框

- “输出”按钮：单击该按钮，系统会弹出“输出设置为”对话框，如图 18-24 所示。

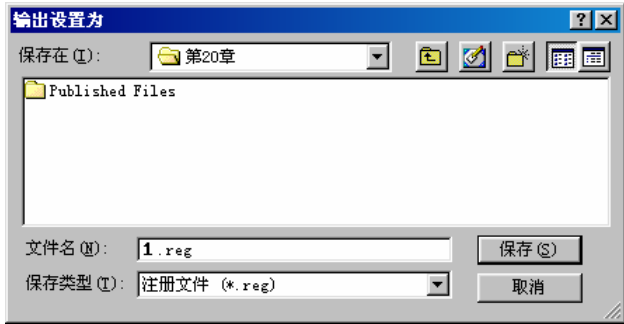


图 18-24 “输出设置为”对话框

- “远程”按钮：您也可以使用一键发布将作品发行到远端的 FTP 站点上。
- “发布”按钮：单击该按钮，可以按照设置好的打包发行信息对作品进行打包发行。

从上面可以看出，Authorware 6 新增的一键发布功能是十分强大的，用户需要认真学习，从实践中理解，不明白之处需要查看 Authorware 帮助。

4. 发布

选择菜单[文件]→[发布]→[发布]命令，或者直接按下<F12>键对作品进行发行。在第一次发行作品时，系统会显示出“一键发布”对话框，如图 18-25 所示。

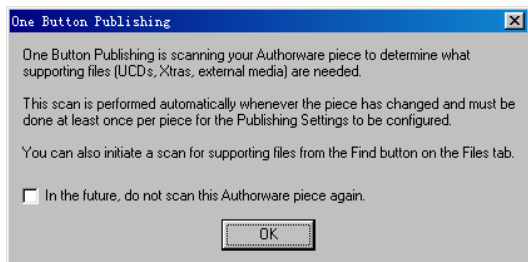


图 18-25 “一键发布”对话框

该对话框显示出一键发布功能信息。当您再次发行作品时，该对话框不会显示。单击“OK”按钮，就开始进行作品发行，其中发行的设置由“一键发布”中的“发布设置”来决定发行属性。

5. 批量发布

选择菜单[文件]→[发布]→[批量发布...]命令，或者直接按下<Shift>+<F12>键对作品进行批量发布。当选择该命令时，系统会显示出“批量发布”对话框，如图 18-26 所示。



图 18-26 “批量发布”对话框

批量发布命令可以对多个作品进行同时打包发行。

单击“增加”按钮，添加要发行的作品，系统弹出“选择文件”对话框，要求您选择要打包发行的作品，如图 18-27 所示。

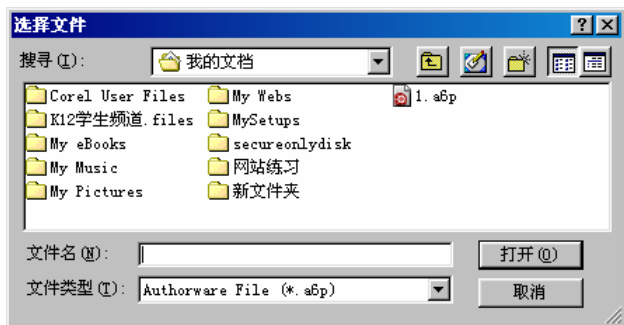


图 18-27 “选择文件”对话框

当选择好要发行的作品文件后，单击“打开”按钮将其加入到上侧的发行文件列表中，如图 18-28

所示是加入了两个作品文件。



图 18-28 在发行文件列表中加入多个发行作品文件

在发行文件列表中选中一个文件，然后单击“删除”按钮，可以将该文件从发行文件列表中移除。
“更新”按钮（或者直接按下快捷键<F5>键）可以刷新已经有所更改的作品文件。

在发行文件列表中，如果有文件不希望此次发行，可以取消文件前方的复选框，这样在发行时就不会将该文件一起发行。

在发行文件列表中，如果希望选择多个文件，可以选中一个文件，然后按住<Shift>键单击另一个文件，则两个文件之间的所有文件都被选中；如果选中一个文件，然后按住<Ctrl>键单击另一个文件，则将该文件一起选中。如果希望选中所有的文件，则可以之间按下快捷键<Ctrl>+A 键。

单击“发布”按钮，可以将发行文件列表中复选框选中的文件全部打包发行，发行属性设置还是由“一键发布”中的“发布设置”来决定发行属性。

当设置好所有的发行文件后，您将该批发布信息保存到一个文件中。选择菜单[文件]→[保存]命令，系统显示“批保存为”对话框，如图 18-29 所示。



图 18-29 “批保存为”对话框

在文本框中输入要保存的文件名，然后单击“确定”按钮保存，单击“取消”按钮取消保存。

18.2.4 网络作品的打包和发行

随着网络的飞速发展，计算机的网络应用已经成为信息世界发展的主潮流。Authorware 可以将应用程序进行网络发行。Macromedia 公司采用的 ShockWave 流技术应用到了 Authorware 中，使用 Authorware 6 Web 打包 r 对应用程序进行网络作品的打包和发行。

要在网络上发布 Authorware 应用程序，首先要将其打包成网络作品。

可以选择菜单[文件]→[发布]→[打包]命令，或者选择[开始]→[程序]→[Macromedia Authorware 6]→[Authorware 6 Web Packager]命令，打开其工具界面。

在打开 Authorware Web 打包窗口的同时，系统也打开一个对话框，要求用户选择要打包成网络作品的文件，如图 18-30 所示。



图 18-30 选择源文件对话框

在该对话框中选择要网络打包的源文件，然后单击“打开”按钮，系统又会弹出一个保存打包文件的对话框，如图 18-31 所示。

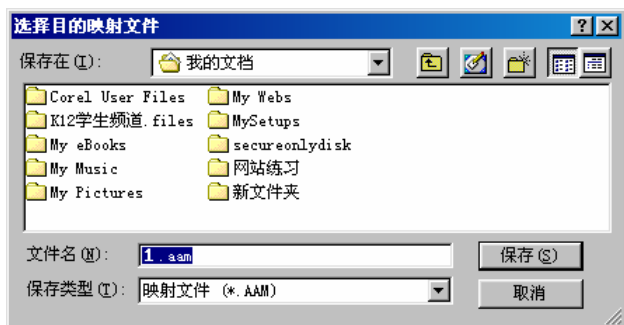


图 18-31 保存打包文件对话框

在该对话框中选择要保存文件的路径和文件名，然后单击“保存”按钮，系统弹出“分片设置”对话框，如图 18-32 所示。



图 18-32 “分片设置”对话框

在该对话框中，其中：

- “分片前缀”文本框：该文本框中是数据文件的文件头名称。可以在该文本框中输入新的文件头名称进行修改。
- “分片大小”文本框：该文本框是记录每个数据文件的大小，默认为将源程序压缩为 16K 大小的多个数据文件包。文件大小可以在文本框中输入数值进行修改。一般来说，该数值不应太大，否则客户端下载一个数据包的时间就会过长。

单击“确定”按钮，Authorware Web 打包会根据用户的配置信息进行打包压缩，压缩完成后，

在 Authorware Web 打包 r 窗口中显示出打包后的 Map 文件信息，如图 18-33 所示。

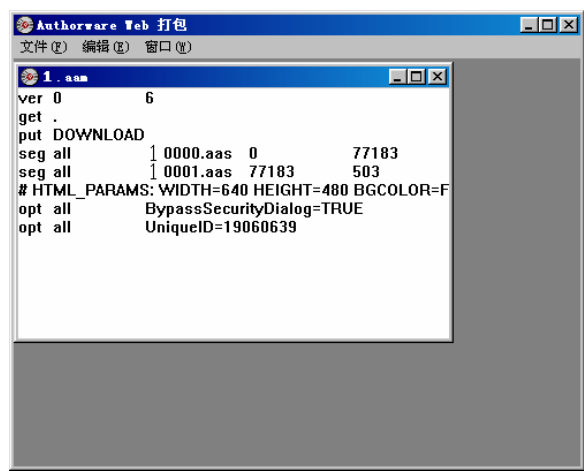


图 18-33 打包后的 Map 文件信息

18.2.5 编辑网络发行的 Map 文件

Map 文件是由 Authorware Web Packager 创建的文件，使用它来引导 Authorware Web Player 如何下载和运行一个网络打包的作品。

一个 Map 文件是一个文本格式的文件。

如：

```
ver    0    6
get    .
put    DOWNLOAD
seg    all  通讯 0000.aas  0    78324
seg    all  通讯 0001.aas  78324  4656
put    .
bin    win16    "odbc.ucd" "odbc.ucd"
bin    win32    "odbc.u32" "odbc.u32"
# HTML_PARAMS: WIDTH=503 HEIGHT=364 BGCOLOR=FFFFFF
opt    all  BypassSecurityDialog=FALSE
opt    all  UniqueID=4174329
```

在 Map 文件中，所有的行分为八类，下面我们对这八类不同的行分别进行介绍。

1．Ver 行

在 Map 文件中的第一行是 Ver 行，该行指定了作品将要运行的 Authorware Web Player 版本。播放器会忽略 Map 文件中与声明版本号不匹配的所有信息，例如，如果播放器的版本比 6.0 要低，则它会忽略 Map 文件中 Ver 行下面的所有版本要低的行数。但是，播放器会继续扫描整个 Map 文件，直到找到一个合适版本的行。

2．Get 行

在 Map 文件中的第二行是 Get 行，该行是定义 Map 文件、DLL 文件以及库文件在 HTTP 服务器上的位置。另外，在 Map 文件中出现的 Get 行还可以指定外置内容在 HTTP 服务器、网络服务器、CD 驱动器等上面的位置。Get 行后面的语句标识着作品与 HTML 页是在同一个位置，如果存储的作品不再

同一个路径,则需要使用一个相对 URL 路径(相对于 Map 文件的地址),后者使用一个绝对 URL 地址,来指明作品在 HTTP 服务器上的位置。

3. Put 行

在 Map 文件中的第三行是 Put 行,该行是定义外部文件,如影片文件,下载到用户的计算机上的位置。当在 Map 文件中使用 put DOWNLOAD,则在下载时,Authorware 会自动创建一个 Download 目录,然后将下载的文件下载到该目录下。

4. Seg 行

在 Map 文件中的第四行是 Seg 行,该行是显示 Authorware Web 打包 r 对文件网络打包的分段情况。该行是由四个部分组成的,如:

```
seg    all        媒体 0000.aas        0        4525
```

- (1) 说明该作品所能运行的操作平台,它有“all”、“mac”、“win”、“win16”、“win32”五个不同的选项,分别对应不同的操作平台。
- (2) 该列是数据文件在 HTTP 服务器上的名称,一般是数据文件的文件头加上四位数字组成,数字一般从 0000 开始计算,每增加一个数据文件段,该数字加 1。
- (3) 该列是该数据文件段的起始位置,单位是字节。
- (4) 该列是该数据文件段的长度大小,单位是字节。

5. Lib 行

Lib 行为 Authorware Web Player 播放器提供网络打包后与 Authorware 文件相链接的库文件信息。

一个 Lib 行一般包括三个部分,如:

```
lib    all    "graphiclib.aam" "GraphicLib.a5e"
```

- (1) 说明该作品所能运行的操作平台,它有“all”、“mac”、“win”、“win16”、“win32”五个不同的选项,分别对应不同的操作平台。
- (2) 打包后库文件的 Map 文件名称。
- (3) 库文件打包后的名称。

6. Bin 行

Bin 行是提供插件文件的信息。

一个 Bin 行一般包括五个部分,如:

```
bin    win32    "filedlgs.u32"    "filedlgs.u32"    ondemand, length=14336
```

- (1) 指定作品运行的操作平台。
- (2) 在 HTTP 服务器上插件的文件名。
- (3) 网络打包后的文件名。当该文件下载到用户的计算机上时,依然使用该文件名。
- (4) 该外部文件下载过程中的控制方式,它包括“Preempt”、“OnDemand”、“Recycle”和“MacBinary”四种方式。
- (5) 其他属性,包括文件长度等。

7. Opt 行

Opt 行是设置网络运行时的相关参数。

网络运行时一般包括四个参数,如:

```
optall    BypassSecurityDialog=FALSE
```

```
opt all InputPredictiveURL = http://www.myserver.com/scripts /sstrd.exe?mypiece.aab
```

```
opt all OutputPredictiveURL = http://www.myserver.com/scripts /sswtr.exe?mypiece.aab
```

```
optall    UniqueID=992884
```

8 . Comment 行

该行是提供其他各行的注释信息。注释行是以#号开头的，后面的信息即是注释信息。如：

```
# HTML_PARAMS: WIDTH=512 HEIGHT=345 BGCOLOR=CCCCC
```

18.2.6 如何制作自动运行的光盘

要想让你的光盘插入光驱后可以自动播放其实也很简单，只要在光盘的根目录下建立一个 autorun.inf 文件即可，autorun.inf 文件的内容和格式如下：

```
[aurotun]
open=*.exe
icon=*.ico
```

其中*.exe 是可执行文件名，*.ico 是光盘的图标文件。

18.2.7 如何改变可执行文件的图标

我们知道打包后的程序图标存放 Authorware 安装目录下的 runa6w32.exe 文件中，所以只要我们将其默认图标换成起他图标就可以了，这样打包后的图标都将使用换后的图标，我们也可以只对打包后的程序进行修改，所用方法也一样。

更换图标的方法有很多种，一般常用的有 eXeScope、Microangelo 等工具，这里以 Microangelo 修改 runa6w32.exe 为例来说明更换图标的方法。

(1) 打开 Microangelo 程序，如图 18-34 所示。

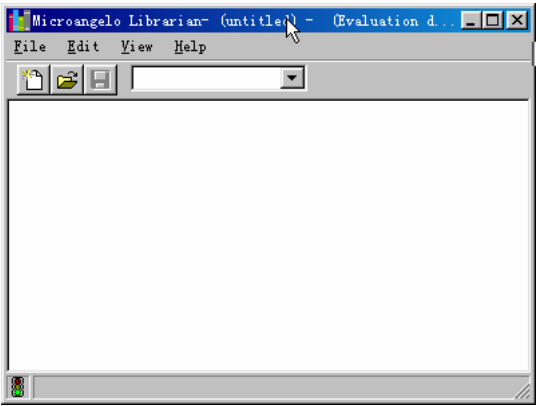


图 18-34 Microangelo 的程序窗口

(2) 在如图 18-34 所示的窗口中，单击“打开”按钮，打开如图 18-35 所示的“Open File”对话框，在该窗口中找到 Authorware 文件夹中的 runa6w32.exe。

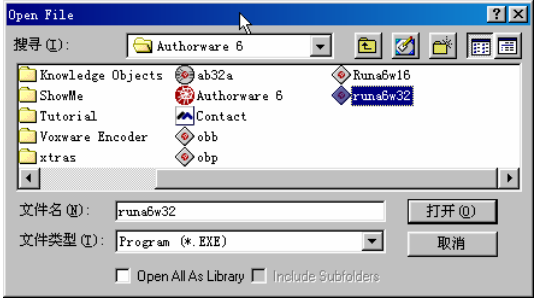


图 18-35 “Open file”对话框

(3) 在如图 18-35 所示的“Open File”对话框中单击“打开”按钮，弹出如图 18-36 所示的编辑对话框。

(4) 双击“runa6w32004”图标，弹出如图 18-37 所示窗口。

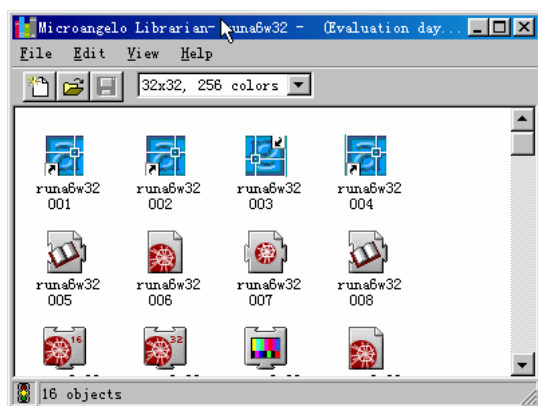


图 18-36

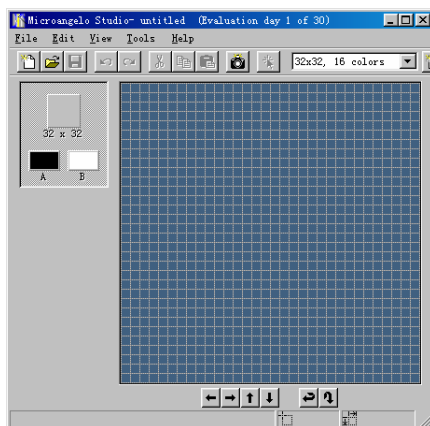


图 18-37

(5) 用如图 18-38 所示的绘画工具或单击工具栏中的“捕捉”按钮捕捉图案。



图 18-38 工具栏

(6) 粘贴图像，如图 18-39 所示。

(7) 关闭窗口将所做的修改保存到 runa6w32.exe。如图 18-40 所示。

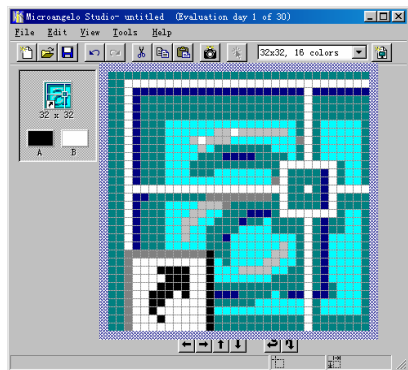


图 18-39

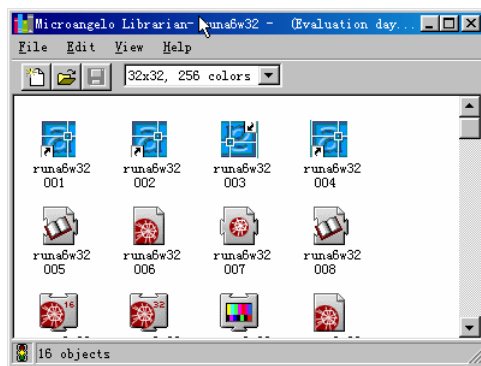


图 18-40

(8) 打包一个 authorware 程序，如图 18-41 所示。



图 18-41