


第5章 人机交互的实现

Authorware最重要的贡献莫过于它能实现交互功能，交互也可以简单的理解为人机对话。有了交互，就可以在程序的允许过程中通过鼠标、键盘或触摸屏等来控制程序的流向，这种功能在教学软件的制作过程中尤为重要。

本章主要内容：

- 按钮响应。
- 文本输入响应。
- 热区响应。
- 目标区域响应。
- 键盘响应。
- 时间限制响应。
- 条件响应。
- 重试限制响应。
- 热对象响应。
- 下拉菜单响应。

5.1 交互前的准备工作

在第2章的示例中，我们已经见到过 （交互）图标，“交互”图标可以设置十一种响应类型，它们分别是：


- | | |
|----------------------------|------------------------|
| 1) Button (按钮响应) | 7) Text Entry (文本响应) |
| 2) Hot Spot (热区响应) | 8) Condition (条件响应) |
| 3) Hot Object (热对象响应) | 9) Key press (键盘响应) |
| 4) Target Area (目标区域响应) | 10) Tries Limit (重试限制) |
| 5) Pull-down Menu (下拉菜单响应) | 11) Time Limit (时间限制) |
| 6) Event (事件响应) | |

另外，在交互图标的右侧还可以设置许多分支，使程序的跳转显得更加复杂，下面将详细介绍交互图标的几种分支结构及交互图标的属性设置。

5.1.1 了解程序分支结构

在交互图标中共设置四种分支结构，它们分别是 Try Again(重试)、Continue(继续)、Return(返回)和Exit Interaction(退出交互)，下面分别加以介绍。

在流程线上放置一个“交互”图标，然后再在它的右侧放置一个其他的图标，如“显示”图标，当释放鼠标后，屏幕上将弹出一个 Response Type(响应类型)对话框，如图5-1所示。在响应类型对话框中显示了 Authorware “交互”所支持的十一种响应方式，可以单击它们对应的复选框来选择相应的响应类型，最后按 OK按钮确认。

 提示 如果再向交互图标的右侧拖动图标，响应类型对话框将不再显示，它会默认前一个分支的响应类型。

如图5-1的黑圈所示，在每一个响应类型的左侧都有对应的响应类型按钮，当在程序中设置了某一种响应类型后，在该分支的上方就会出现该类型对应的响应类型按钮。图 5-1中流程

线上显示的就是热区响应按钮。

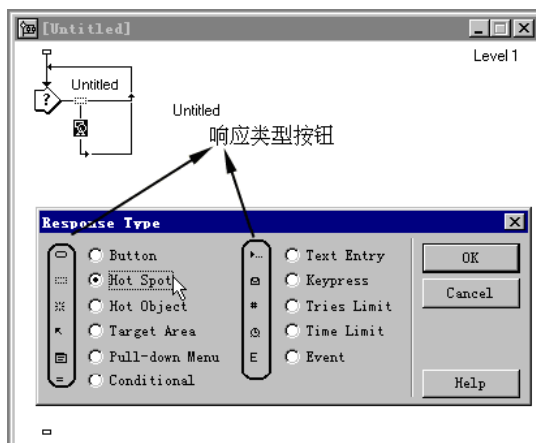


图5-1 响应类型对话框

如果单击流程线上的响应类型按钮,如图5-1中的“响应类型按钮”,Authorware就会弹出Properties: Response(响应属性)对话框,打开Type下拉列表框,也会看到 Authorware的十一种交互方式,此时可以单击其中的选项来改变响应类型,例如选择 Button响应类型,如图5-2所示。

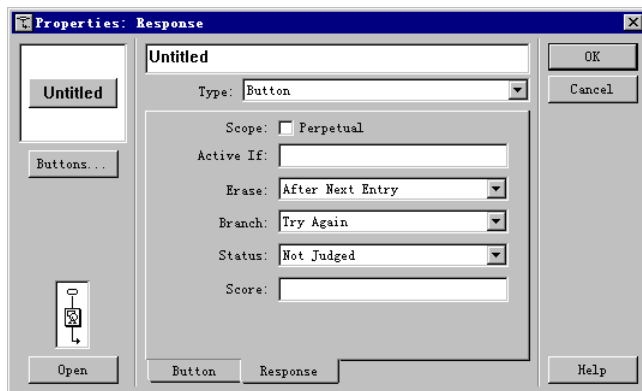



图5-2 响应属性对话框

打开Erase下拉列表框,其中有四个擦除条件。On Exit(退出):选择该条件, Authorware在退出交互图标后,将交互图标里的内容全部擦除; Before/After Next Entry(进入下一个响应之前或之后):用户给出响应后, Authorware在进入下一响应前或响应后擦除交互图标里的显示内容; Don't Erase(不擦除):选择该条件, Authorware将不擦除该交互图标中内容,除非使用擦除图标将其擦除。

 提示 1) 单击Perpetual(永久)复选框,用户所设置的响应类型(如按钮、区域、菜单等)将永远有效,直到退出该交互程序。2) 在Active If(条件)文本框中可输入控制响应产生的条件,如设置一个逻辑表达式。当条件满足时,该响应生效。3) 在Score(范围)文本框中可输入与响应有关的值。如果响应结果为 True(真), Score值为正;若响应结果是

False(假), Score值为负。也可以在文本框中使用表达式来表示相应的值。

打开Status(状态)下拉列表框,如图5-3所示,在其中有三种响应状态,它们分别代表响应结果的状态。Not Judged(不做判断), Correct Response(正确响应), Wrong Response(错误响应),对于它们的具体应用,我们将在本书的5.5节中讲解。

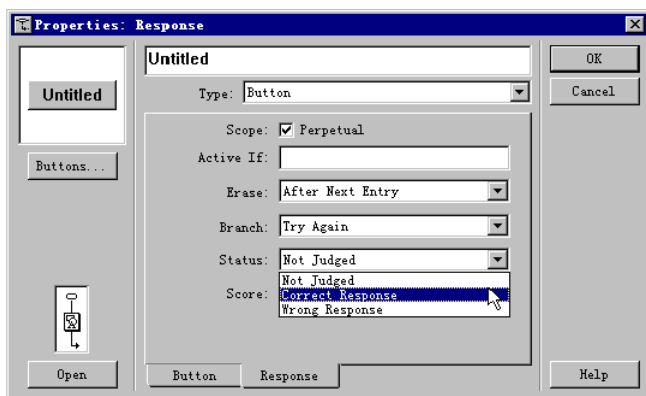


图5-3 响应状态

响应属性对话框还有另外一个标签,比如图5-3所示的Button,该标签的名称随响应类型的改变而改变,我们在下面各节的例子中将具体讲解。

另外,在Branch(分支)下拉列表框中有四个选项,它们代表了“交互”图标的四种分支结构,也将在下面分别进行介绍:

如图5-4所示,在“交互”图标的右侧可以放置多个图标,每个图标代表一路分支,在分支的右侧也可以输入分支标题,该标题就是各个图标的名称。如果分支较多,可以拖动滚动条来观察。

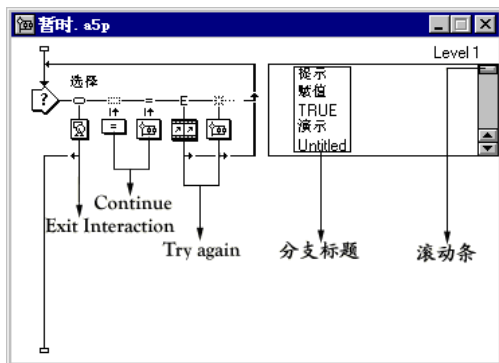



图5-4 交互分支

在各个图标下方的流程线上都有箭头指示,箭头的方向代表此反馈路径的方式,也就是前面所说的分支类型。当程序执行完 Try Again(重试)分支后,它将返回“交互”图标之前等待下一次交互输入;当程序执行完 Continue(继续)分支后,它将退回原处继续判断下一个响应;当程序执行完 Exit Interaction(退出交互)分支后,程序将直接退出交互,用户也可以直接从带箭

头的分支方向看出程序的流向。

 **注意** 在“交互”图标右侧不能放置其他“交互”图标、“框架”图标和“判断”图标，如果程序需要，可以先放置一个“群组”图标，然后在“群组”图标内再放置这些图标。

另外，“交互”图标还有一种分支结构 Return(返回)，该分支结构比较特殊，它需要将“交互”图标所有的响应分支都设为 Perpetual(永久性)，此时流程线将直接穿过“交互”图标，如图5-5所示。这种方式通常在制作下拉菜单响应时应用，我们将在 5.6节进行介绍。

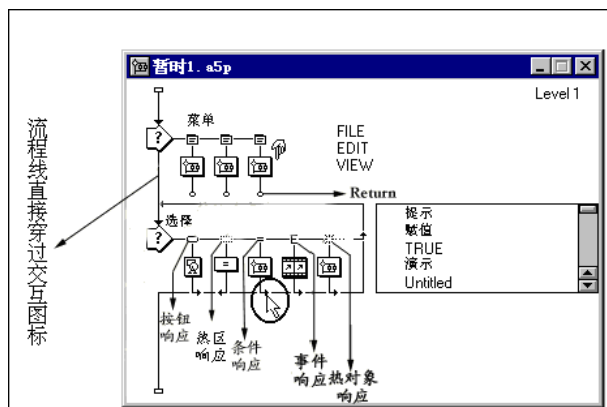



图5-5 Return分支



 **技巧** 按住Ctrl键，用鼠标在反馈箭头处单击一下就会改变反馈分支的方式，连续单击就可以在这几种方式中自由转换。

5.1.2 设置交互属性

在创建交互分支的过程中，不可避免地要进行“交互”图标的属性设置。打开其属性对话框，首先要选中“交互”图标，然后打开 Modify菜单，在 Icon子菜单中单击 Properties命令，将会弹出 Properties：Interaction Icon(交互图标属性)对话框。也可以选中该图标，然后按下 Ctrl+I快捷键。

如图5-6所示，对话框一共包括四个选项卡：Interaction(交互)、Display(显示)、Layout(布局)和CMI(管理)。另外，在对话框左上角演示窗口下还有一个 Text field(文本区域)按钮，单击该按钮将弹出 Properties：Interaction Text field(文本区域响应属性)对话框，该对话框的设置与文本响应有关，我们将在 5.6节中讲述。

在图5-6所示的 Interaction选项卡中，我们也会看到有一个 Erase下拉列表框，该列表框的设置与前面讲的比较类似，只不过它是代表擦除“交互”图标的设置。如果单击 Options选项组中 Pause Before Exit复选框，当程序执行完交互图标后，画面会暂停下来，以便让用户看清屏幕上的显示内容。看完后，按任意键可以继续执行。如果同时选中 Show Button复选框，在窗口上还会显示一个 Continue(继续)按钮，单击该按钮程序也能继续往下执行。

 **提示** 单击 Erase 旁的  按钮，弹出擦除过渡方式对话框，该对话框的设置与前面 4.8节讲的相同。

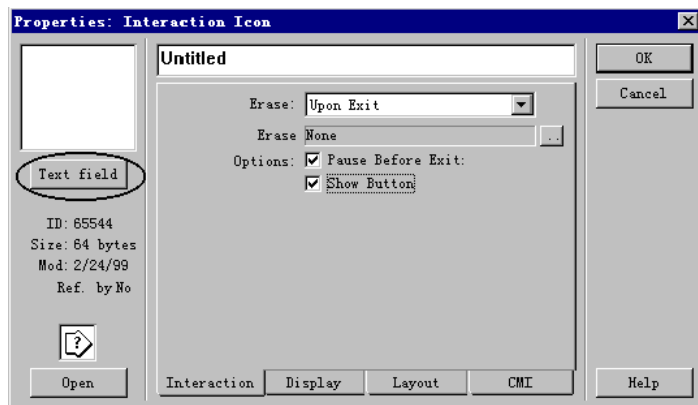


图5-6 交互图标属性对话框

单击Display标签或Layout标签，将打开显示或布局选项卡，对于它们的设置，可以参照前面讲的“显示”图标属性设置。

如图5-7所示，单击CMI选项卡可进行设置有关CMI(管理教学)的各选项。该功能是Authorware 5的新增功能，在系统中有23个崭新的简单易用的CMI函数和50个系统函数。

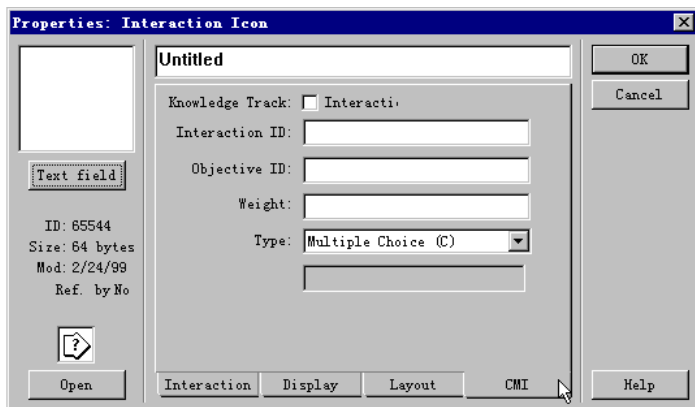


图5-7 CMI选项卡

1) 在Interaction ID文本框中可以输入交互图标的标识符，Authorware把用户在CMI Add Interaction函数中输入的ID号作为交互的标识符。

2) 在Objective ID文本框中可以输入目标标识符，Authorware把用户在CMI Add Interaction函数中输入的ID号(标识符)作为目标ID参数。若此文本框为空，Authorware将使用图标名称作为目标ID标识符。

3) 在Weight文本框中可以输入某些数值，Authorware也把用户在CMI Add Interaction函数中输入的值作为Weight的参数。

4) Type(类型)下拉菜单，Authorware把用户所设的下拉菜单选项或输入的CMI Add Interaction函数作为Type参数。

单击Knowledge Track复选框，当程序运行时，交互图标自动将交互结果传递给CMI系统。



提示 如果选择Type下拉列表框中的From Field选项，Authorware将把从文本框中输入

的参数作为Type参数。用户也可以在文本框内输入某个语句或表达式，若输入表达式，在表达式的前面还应加上一个“=”号。

下面我们再讲一下“交互”图标的内部设置：

同“显示”图标一样，双击“交互”图标便可进入演示窗口，在窗口中可以插入图片和文本，也可以利用图解工具箱的各种绘图工具来制作各种图形。假如我们在“交互”图标右侧设置按钮响应、热区响应、目标区域响应或文本响应，打开“交互”图标的演示窗口后，我们将看到图5-8所示的画面。

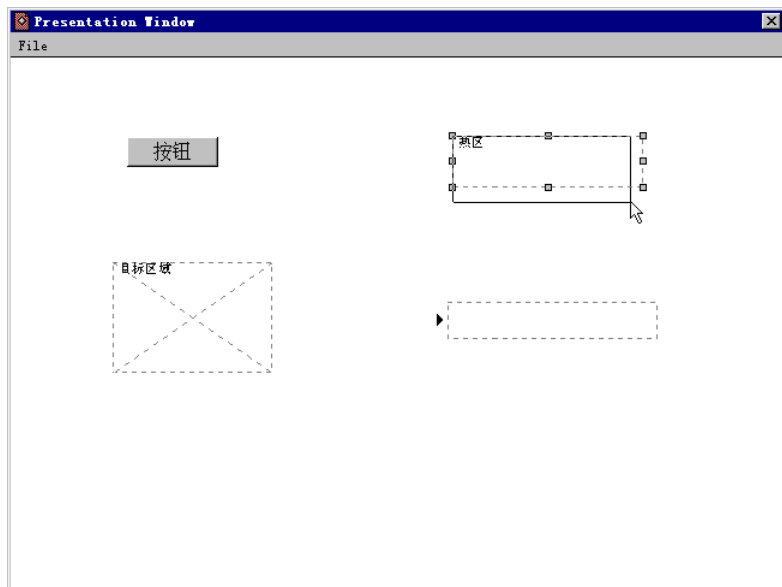


图5-8 改变区域

在各个区域或按钮上我们已经设置了相应的标题，这些区域标题的文字在程序运行过程中不会显示。如图5-8中鼠标所示，当单击其中对象时，按钮或区域的周围将出现调节方块，按住调节方块拖动会改变区域或按钮的大小。拖动按钮本身或区域线条，会改变它们在窗口中的位置。

双击按钮或区域线条，将会弹出对应的响应属性对话框。

5.2 使用按钮响应——实例“做数学题”

下面开始讲解交互的第一种方式，也就是按钮响应。按钮响应在交互程序中使用比较广泛，它的响应形式更为简单，当程序运行时，只要单击一下设置的按钮，程序就会执行该按钮对应的分支。

按钮的形状多种多样，在我们平时遇到的考卷中，选择题的选项前就有单选按钮，用鼠标单击单选按钮就会选择该选项。在下面就举了一个做数学选择题的例子。

先在设计窗口中创建如图5-9所示的流程线。在将“显示”图标放置在“交互”图标右侧时，Authorware会弹出响应类型对话框，选择Button单选按钮后确认。然后再从图标栏中拖放三个“显示”图标，它们的响应类型将会默认为按钮响应。如图5-9所示更改标题名称为A、

B、C、D。在流程线最上方的“显示”图标中我们将存放试卷画面，在“交互”图标中我们将输入试题和选项。

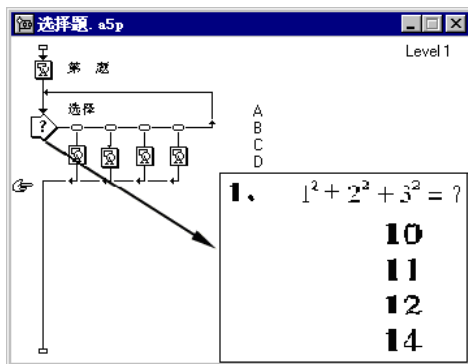



图5-9 流程线

 提示 可以不使用“显示”图标，将全部的内容都输入“交互”图标也可以达到同样的效果。

打开“第一题”图标，在演示窗口中输入如图 5-10所示的文字，然后再制作试卷背景，但要注意设置文字及背景的层次关系。打开 Properties : Display Icon(显示图标属性)对话框，如黑圈内所示，设置画面的过渡模式为 Vertical Blind。这样，当程序一进入显示图标时就会显示出画面的过渡效果，如此设置会丰富画面的显示效果。

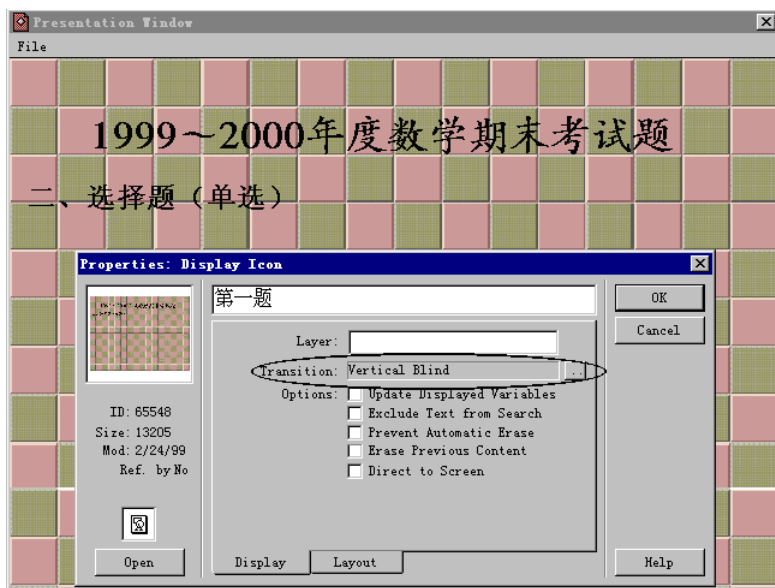


图5-10 设置过渡效果

双击流程线上的响应类型按钮，打开 Properties : Response(响应属性)对话框，如图5-11所示，打开Branch下拉列表框，在其中设置反馈分支为 Exit Interaction。其他选项取默认值。

Untitled按钮是 Authorware 默认的按钮样式，下面我们来自己设置单选按钮的形状。单击

对话框中的Button按钮，打开图 5-12所示的Buttons(按钮)对话框，拖动对话框右端的滚动条，在按钮列表框中选择图中所示的单选按钮，然后单击 OK按钮确认。

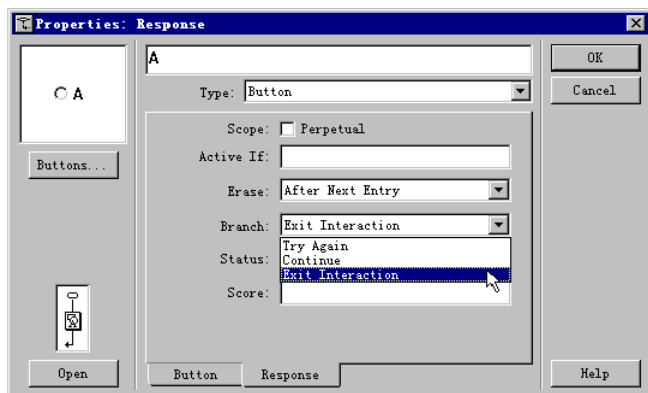


图5-11 设置反馈分支

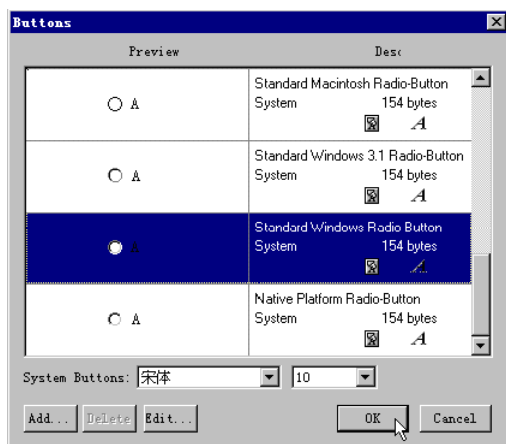




图5-12 按钮对话框

 **提示** 在System Buttons列表框中，可以选择按钮标题的文字字体。同时，也可以在旁边的列表框中选择标题文字的字号。

如果觉得按钮设置不太理想，Authorware也允许对其进行修改。选择按钮样式，然后单击对话框中的Edit(编辑)按钮，接下来会弹出图 5-13所示的Button Editor(按钮编辑)对话框。在State(状态)列表框中有两列按钮，单击它们可以用来预览按钮状态。如图中鼠标所示，在Checked列按钮上单击，在右侧的预览框中将显示按钮按下的状态。单击 Normal列中的按钮显示正常状态的按钮形状。

 **提示** 打开Label(标签)列表框，其中的选项主要用于设置按钮标签的显示状态，如选择Show Label，在按钮的右侧会显示标签名称。同时，在其右侧的列表框中设有有关的选项，它们主要是用于设置标签在按钮上的显示位置，如Align Left，设置标签显示在按钮左侧。如果单击Sound Import(输入声音)按钮，还会弹出一个对话框，要求加载该

按钮的伴随声音文件。

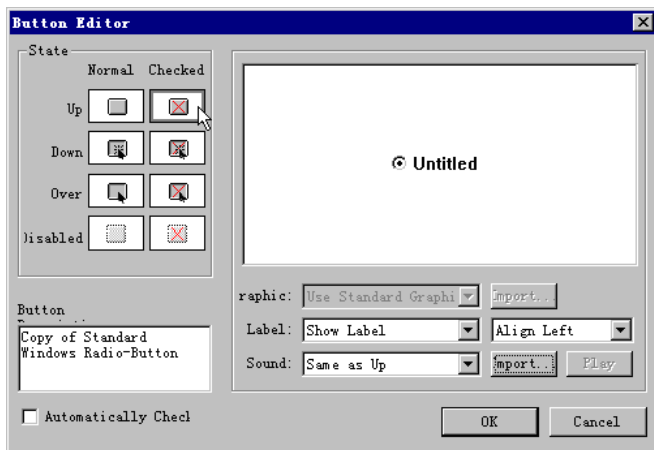


图5-13 按钮编辑对话框

如果想添加自己制作的按钮，在图 5-12所示的对话框中单击 Add按钮，Authorware同样会弹出 5-13所示的对话框，同时 Graphic列表框及 Import按钮被激活。单击 Import按钮，Authorware会弹出如图 5-14所示的对话框，在列表框中选择完图片文件后单击 Import按钮，所选的按钮就会出现在图 5-13的预览框中，Graphic列表框中也会显示 Use Imported(使用插入按钮)。

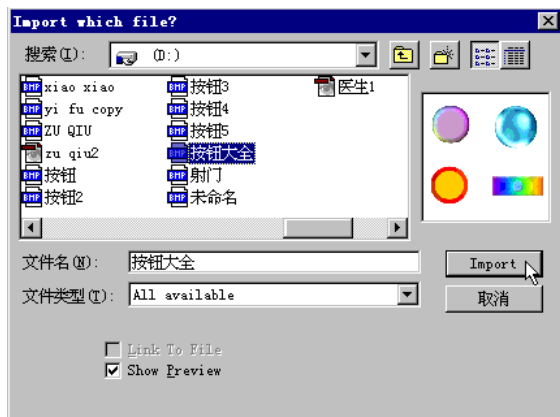


图5-14 插入按钮

设置完按钮后返回响应属性对话框，打开 Button选项卡，如图 5-15所示。在演示窗口中出现一个带标签的按钮，这就是设置的按钮形状。单击该按钮，周围会出现调节方块，拖动调节方块可改变它的大小，拖动该按钮可改变它在窗口中的位置。在 Label文本框中可输入按钮的标签。在 Key文本框中可输入此按钮响应的快捷键，如 a，按下字母a键和单击该按钮的效果相同。

如果按钮是系统提供的标准按钮，单击 Make Default复选框，在按钮的周围将显示黑框，表示该按钮是默认按钮，此时按下 Enter键同单击该按钮一样。若按钮是自己创建的，则此项

设置无效。

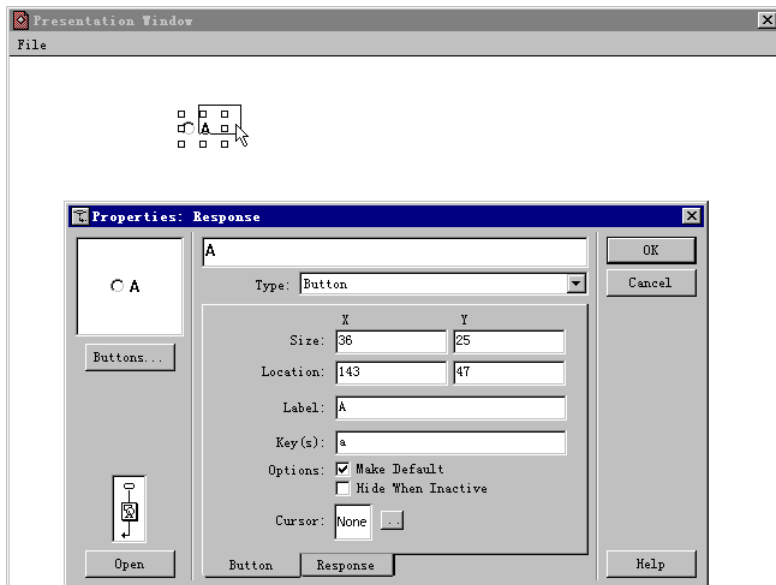




图5-15 设置按钮选项卡

单击Hide When Inactive复选框，则该按钮在不可用时被隐藏，在使用时自动显示。

技巧 如果确定了按钮在屏幕上的位置，可以直接在对话框的Location文本框中输入X、Y的坐标值。同样，也可以在Size文本框中输入按钮的长度和宽度。

如果想设置鼠标移至按钮上方时的形状，单击 Cursor旁的  按钮，弹出Cursors对话框，在鼠标列表框中可选择所需的鼠标形式，最后单击 OK按钮，该鼠标形式就会出现在对话框的Cursor预览框中。程序运行时，当将鼠标移至该按钮上方，鼠标指针就会变为刚才所设的样式，如  型。当然，您也可以单击 Add按钮，在随后弹出的对话框中选择插入外部的鼠标样式。

这样，第一个按钮就全部设置完了。用同样的方法可以设置另外三个“显示”图标，设置完毕，我们来看看按钮在演示窗口中的具体位置。打开“交互”图标，用鼠标将按钮一一拖至合适的位置。如果觉得按钮没有对齐，如图 5-16所示，按住 Shift键将按钮全部选中，然后调出 Align(排列)选择框，单击合适的对齐方式。

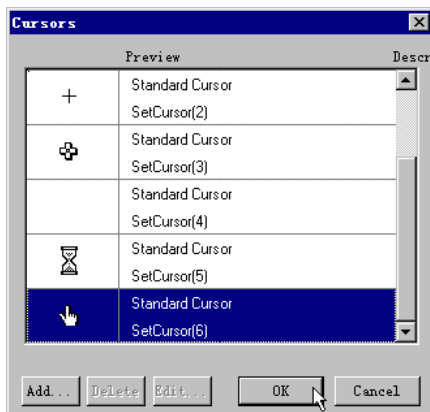


图5-16 鼠标列表框

提示 你也可以在程序运行过程中按 Ctrl+P快捷键终止程序，然后再调节它们的位置。

为了增加程序的趣味性，可以在 A、B、C、D 显示图标中放置不同的画面，程序如果设置 A、B、C为错误选项，在它们下面的“显示”图

标内可以存放一个正在哭泣的小姑娘的画面，如图 5-18所示。如D分支是正确选项，则在其下面的“显示”图标中放置一个兴高采烈的小姑娘画面。

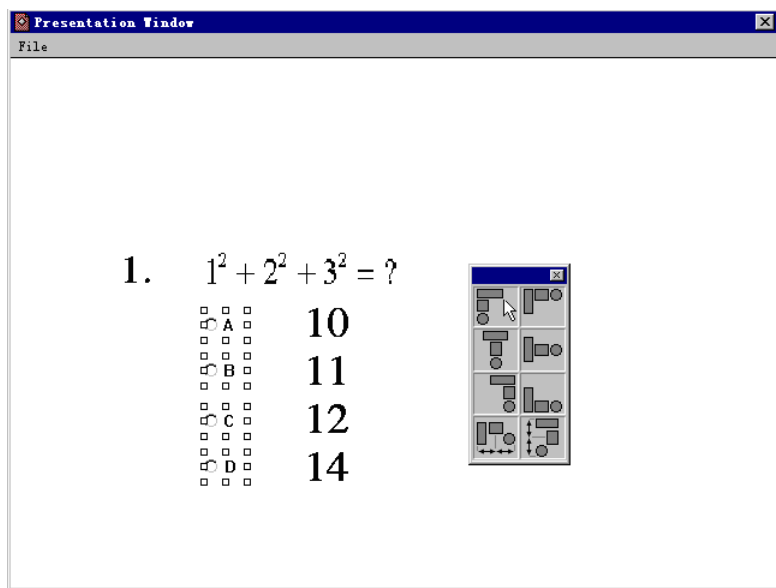


图5-17 对齐按钮

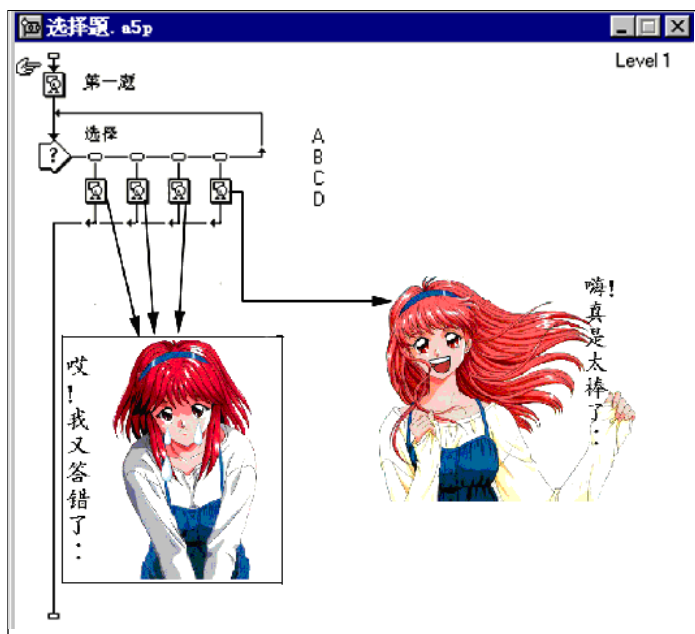



图5-18 设置显示图标

 **注意** 当运行程序时，用户会发现小姑娘身后的试卷画面没有消失，这可怎么办呢？没关系，可以打开“显示”图标属性对话框，然后单击Erase Previous Content (擦掉前一

幅画面)。小姑娘画面出现瞬间便消失,为此还应选择Prevent Automatic Erase (阻止画面自动擦除)复选框。当然,也可以在显示图标前放置一个“擦除”图标将前面的画面清除。

编完这个程序,你也许会想,如果要设置一套选择题该怎么办?答案很简单,依葫芦画瓢,在流程线上接着往下做第二道、第三道……。

5.3 使用热区响应——实例“提示向导”

在大多数的工具软件中,常常会看到工具栏、图标栏、工具箱等结构,如果我们将鼠标移至这些工具的上方,在鼠标的下面就会出现该工具的名称,这种功能非常方便,它会使我们更快捷地得到帮助信息。

“画图”对我们来说是最熟悉不过了,打开 Windows程序组中的附件,就能调用它。在“画图”的窗体中有一个工具栏,如果用鼠标移至某一工具按钮的上方,在鼠标下面就会出现它的名称。下面就这个名叫“提示向导”的例子介绍如何使用热区响应来实现这种功能,同时将它的功能加大,把有关它的帮助信息也显示出来。怎么样?是不是觉得有点不可思议?下面我们将此变为现实吧!

如图5-19所示,在设计窗口中创建流程线,当弹出 Response Type(响应类型)对话框时,选择Hot Spot(热区响应)单选按钮,同样放置其余的“显示”图标,然后将其一一更改为图中所示的标题名称。在“显示”图标中将存储对应工具按钮的名称和基本功能。

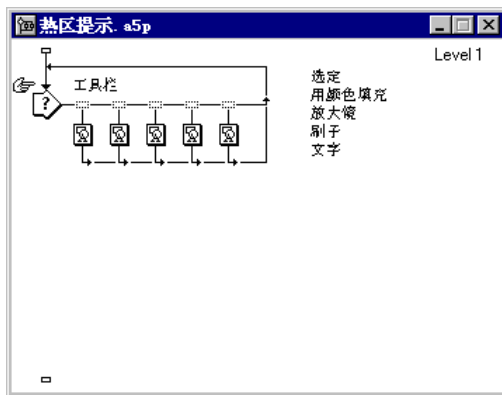


图5-19 流程线

单击流程线上的响应类型按钮,弹出如图 5-20所示的对话框。单击 Match下拉列表框,其中选项可设置与热区响应匹配的鼠标动作。这些选项分别是: Single-click:在热区内单击鼠标时产生响应; Double-click:在热区内双击鼠标时产生响应; Cursor in Area:当鼠标移至热区上方时产生响应。在此例中,设置最后一种响应方式。

单击Highlight on Match复选框,热区会产生如下变化:程序运行时,用鼠标单击/双击热区时,热区会以高亮显示,松开鼠标后热区状态复原。而单击 Mark on Match复选框,在演示窗口中的热区内会显示一个匹配标志,一般是一个白色小方块,当产生响应时,该匹配标志就被黑色填充,如果响应结束,黑色并不消失。

如图5-20的黑圈内所示，用户可以在 Key(快捷键)文本框中输入与此热区响应对应的快捷键，如可以输入1、2、3、4……等数字，也可以输入a、b、c等字母。如果需要使用组合键来代表快捷键，例如使用 Ctrl和Alt组合键，可以在它们后面直接加上该字母，图 5-20所示的 Ctrl15代表Ctrl+5快捷键。如果一个热区需要使用多个快捷键，那必须在字符中间插入运算符“|”，例如“B|b”，即按下B或b都能激活该热区响应。此快捷键的功能与 5.2节按钮响应的快捷键功能相同。

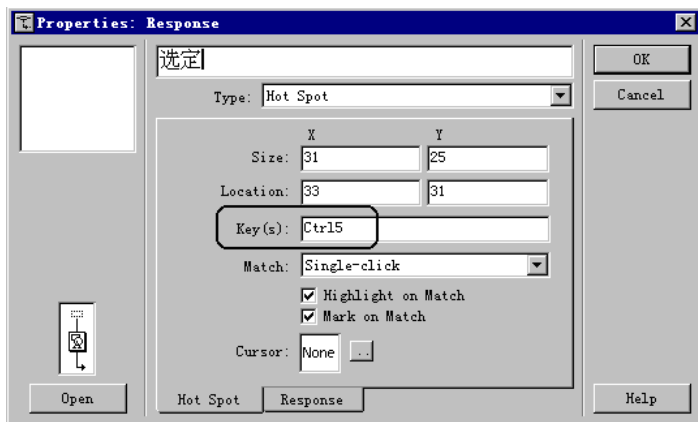





图5-20 设置快捷键

 **注意** 在键盘上的许多功能键可以直接使用，例如Tab、Enter、Backspace、Esc、Home、End、方向键等，可以直接在文本框中输入该键的名字。

在“交互”图标的右侧一共有五个热区响应，五条分支路径。因此在演示窗口中必有五个热区方块。打开“交互”图标，然后单击工具栏上的“插入”按钮，在图标中插入“画图”窗体的界面。然后如图 5-21所示，用鼠标将热区移至对应的工具按钮上方。如果热区大小与按钮大小不匹配，可以通过拖动热区的调节方块进行调整。热区的边框在程序运行时是不可见的，如果想将其变为可视，可以用鼠标在其周围画一个矩形的方框。

 **提示** 如果用户想马上就练习此程序，可以提供获得“画图”窗体的界面的方法：打开Windows 95/98附件中的“画图”程序，然后按下Print Screen(拷屏)键，然后单击编辑菜单中的“粘贴”命令，窗体的画面就被复制到当前的画面上了。

 **技巧** 如果觉得用鼠标操作不太方便，也可以通过键盘来完成。先选定热区，然后按键盘上的上下左右方向键来移动，每按一下对象移动一个象素点。

我们编辑的程序要达到如图 5-22所示的效果，移动鼠标到“刷子”按钮的上方时，在屏幕的右侧显示按钮的名称及相应的帮助信息，然后再将鼠标移到其他按钮的上方，原来的信息消失而显示新的信息。因此，可以在分支的每个“显示”图标中输入对应按钮的名称及相应的帮助信息。同时，在响应属性对话框的 Response选项卡中设置分支为 Try Again，当然，设为Continue也可以。在这个程序中，只做了工具按钮功能的一部分，如果用户有兴趣的话，可以仿照此例继续做完其他按钮的功能。

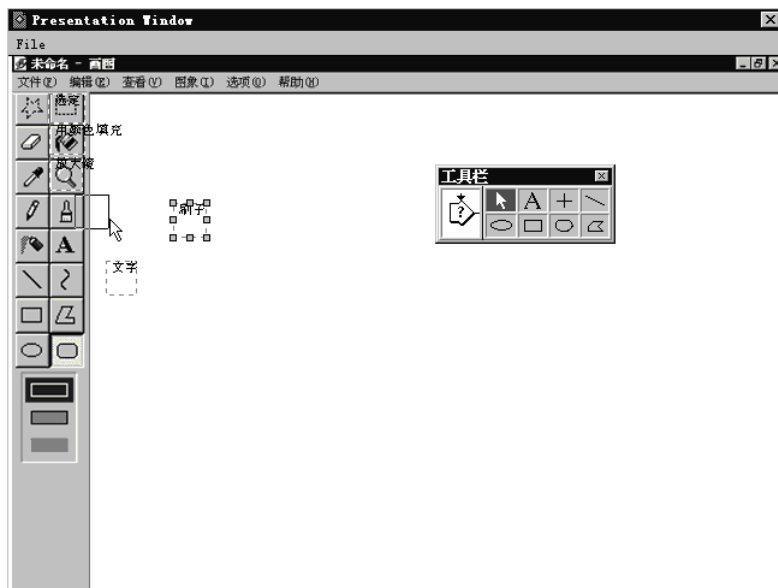


图5-21 移动热区

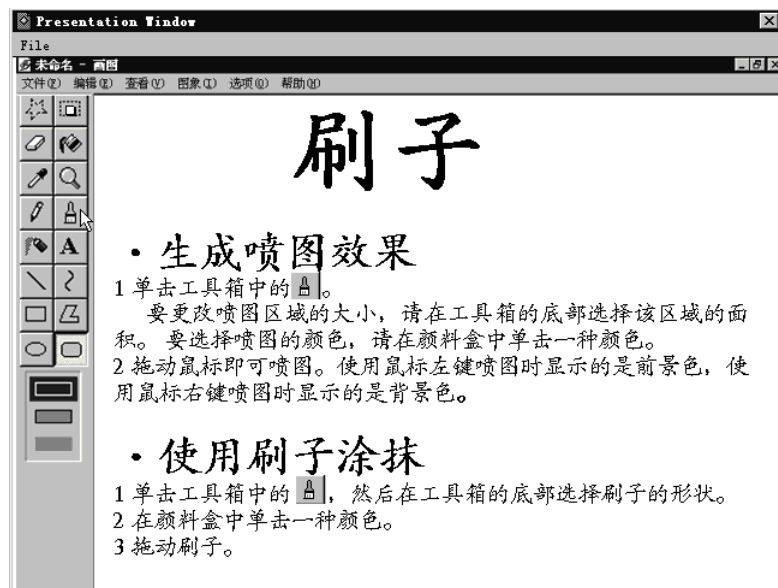


图5-22 程序运行

5.4 使用热对象响应——实例“动物乐园”

“东芝动物乐园”想必是大家喜爱看的节目，下面将把你带入这千奇百怪的乐园，在里面将会看到许多熟悉的和陌生的动物。

热对象响应和热区响应类似，它们的响应方式也几乎相同。因此，有了前面制作热区的

经验，这章内容可谓轻而易举。热区响应与热对象响应的唯一区别是：前者产生响应的对象是一个区域范围，而後者的对象是一些实实在在的物体。

下面所讲的程序要达到如下目的：程序运行时，在画面上显示各种动物，如果操作者不知道它们的名称，用鼠标单击某个动物，在该动物的旁边就会出现该动物的名称。根据题目的要求，我们可以编辑如图 5-23 所示的流程线。在弹出 Response Type(响应类型)对话框时，选择 Hot Object 单选按钮，然后将其他四个显示图标同样放置在“交互”图标的右侧。在驴、猫、狮子、狗、兔子和笼子“显示”图标中分别插入它们的图片，最后还要将各个图标进行命名。

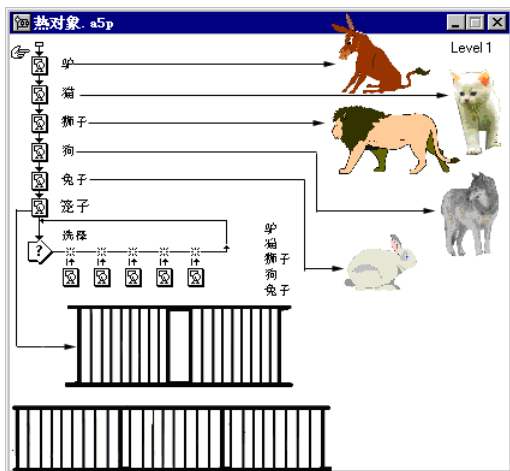



图5-23 流程线

 **注意** 在热区响应程序中，所有的画面可以放置在一幅图中，而在热对象响应的制作中却不可以，热对象中的每一个对象必须单独放置在一个“显示”图标中。

当所有的图片都插入图标后，运行程序时，会出现如下的问题，所有的动物图片出现后立即消失，而画面上最后只保留了一幅笼子的画面，实际上我们要将所有的动物都显示在笼子的后面。解决这个问题就要利用前面讲过的知识了，选择“笼子”显示图标，然后按 Ctrl+I 快捷键打开属性对话框，如图 5-24 中黑圈内所示，在 Layer 文本框中输入图片的层为“-1”，这样笼子就不会将所有的动物图片挡在后面了。

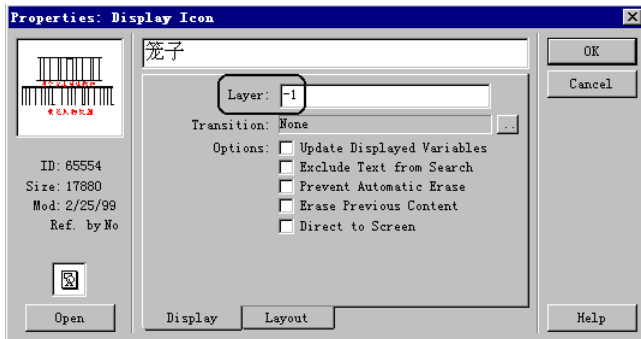


图5-24 设置层



提示 在Authorware的层设置中,如果Layer文本框内没有任何值,系统默认为“0”。因此,在本例中的“笼子”图标的层为 $-1 < 0$,所以该画面将显示在所有的动物图片的前面。

在设置图标层之前,如果想将动物图片和笼子图片显示在同一窗口中,可以这样操作:按住Shift键,然后如图5-25所示,用鼠标双击流程线上的“狗”“显示”图标,此时,在窗口中就会同时出现“狗”的图片。“狗”图片的背景是白色的,它会将笼子的一部分遮住,因此,还要打开重叠模式选择框,单击其中的Inverse(反显)模式,设置完毕,最好看一看效果怎样。

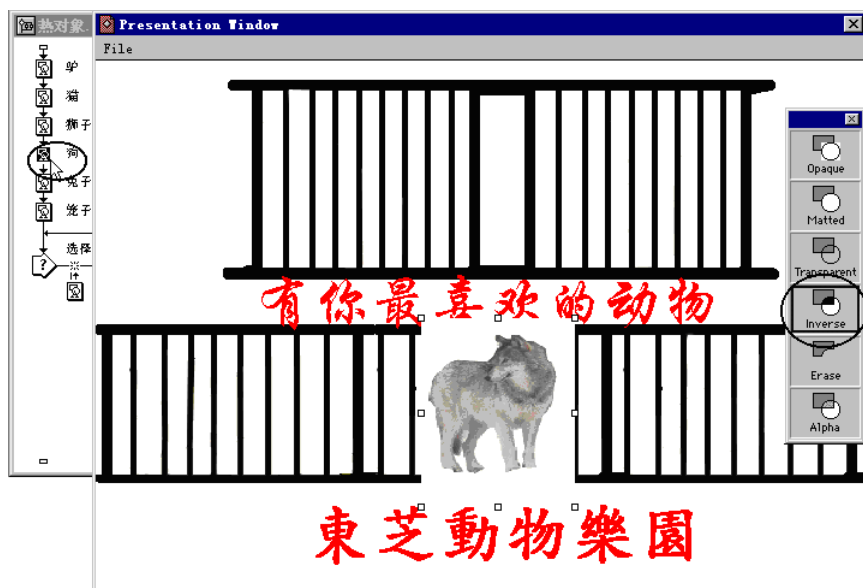


图5-25 设置重叠模式

将所有的动物图片都设为反显模式后,我们来运行一下程序,单击“运行”按钮后,屏幕上出现如图5-26所示的Properties: Response(响应属性)对话框。同时,该对话框有标题提示“Click an object to make it the hot object”,该提示的意思是让用户单击画面上的一个物体作为热对象,如图中鼠标所示,单击“驴”对象,在对话框的Hot Object(热对象)文本框内就会显示该对象图标的名称,如黑圈内所示。

单击对话框的OK按钮确认后,Authorware又会弹出下一个响应属性对话框,要求继续指定下一个热对象,用同样的方法将其他响应属性对话框设置完毕。然后,在交互分支中的各个显示图标中输入对应的动物名称,如图5-27所示。回到流程线后,如果用户觉得流程线有点太长,可以将它们进行归组。用鼠标将“交互”图标上方的所有“显示”图标都选中,然后,单击Modify菜单下的Group命令,所有的“显示”图标就会转化为一个“群组”图标,如图5-27所示,我们将它命名为“动物”。

至此,这个程序也就全部讲完了。因为,在响应属性对话框的Match下拉列表框中的默认选项Single-Click,因此在程序运行后,单击画面上的动物后,就会在屏幕上出现该对象的名称。当然,也可以设置双击等方式。

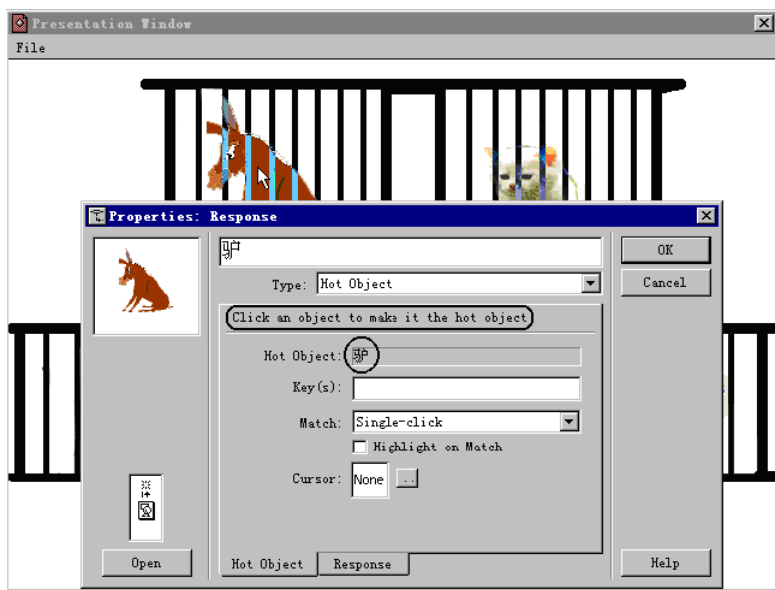


图5-26 选择热对象

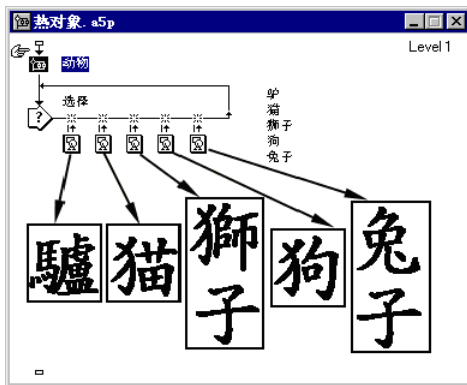


图5-27 图标归组

5.5 使用目标区域响应——实例“巧对唐诗”

平时，我们经常玩一种叫做华容道的游戏，在这个游戏中，可以用鼠标来拖动各种方块进行移动。如果用 Authorware 来制作这个游戏，设置的方块必须能够用鼠标进行移动，这时，我们便会想到前面曾经提到过“显示”图标的 Layout 选项卡，在这个选项卡中可以将图片设置为可移动的。

由上面的提示，还可以想到，在化学实验课和物理实验课上，老师常常教我们连接实验仪器，仪器一般都很昂贵，不小心弄坏了将得不偿失。如果我们将各种仪器的图片插入 Authorware，然后再将它们设置为在画面上可动，再设置正确的拼凑结果，一样可以判断是否真正掌握实验能力。

使用Authorware制作这类游戏并不难，下面就举一个类似的例子，如图 5-28所示，画面上有两首古诗，每首诗都缺少一句，要求用户来选择正确答案，并将答案拖至诗中正确的位置，如果选择错误，选择的诗句会返回原处，并且画面上会显示错误信息，如图 5-28中所示。如果选择正确，诗句会停留在目标标处，窗口中会显示鼓励性的文字。在这节要讲的目标区域响应就能达到这种效果。

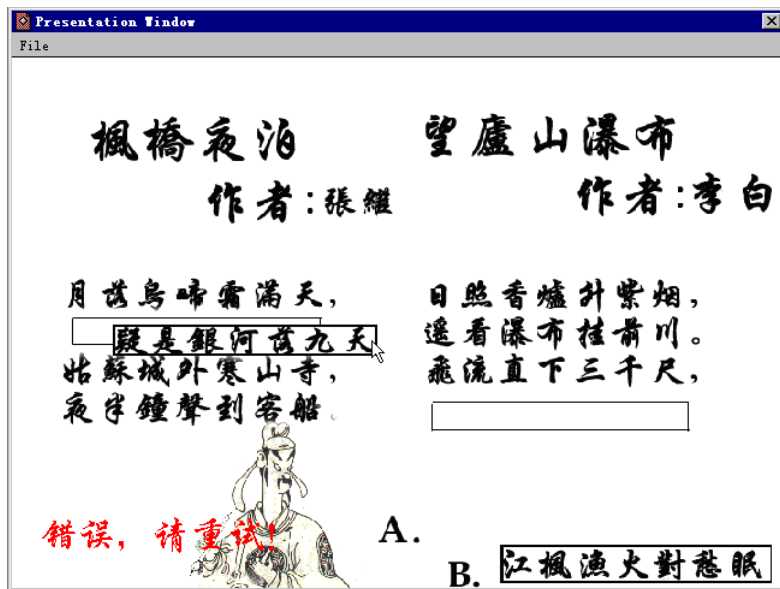


图5-28 运行画面

根据题目的要求，可以编辑如图 5-29所示的流程线。在“诗”图标中放置程序运行的主画面，在图标A、B中分别输入如图5-29中所示的诗句A和B。放置交互分支的“显示”图标时，在响应类型对话框中选择 Target Area(目标区域响应)。

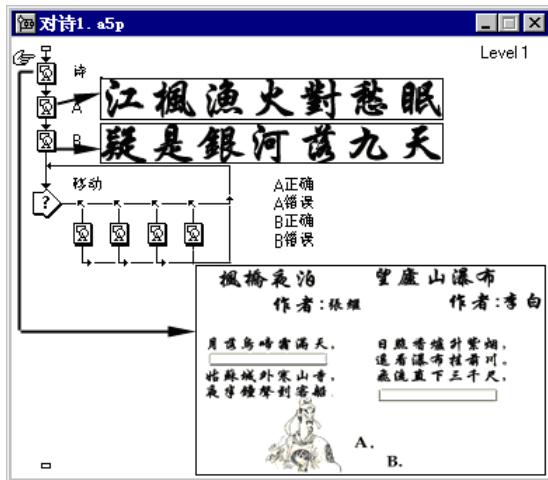



图5-29 流程图

 提示 与热对象响应的设置方式一样，在目标区域响应中的每一个移动对象也必须单独放置在一个“显示”图标中。


将流程线组建完毕后,单击工具栏上的“运行”按钮,窗口中就会弹出响应属性对话框。如图5-30所示,在窗口中会出现一个矩形区域,这个矩形的大小和位置就是我们所讲的目标区域。对话框的标题提示为 Select a target object, 提示用户选择一个目标对象,用鼠标单击画面上的“江枫渔火对愁眠”图片(该诗句是正确选择),如图5-31所示,在Target Object(目标对象)文本框中就会出现该图片对应的标题。



图5-30 选择目标对象

单击对象后,如图5-31所示,对话框的标题又改为 Drag object to the target position, 要求将对象拖至画面上的目标位置。在将对象移至目标位置后,目标区域也同时自动到达该位置,如图5-31中的鼠标所示,将目标区域调节至与填充区域同样大小。这样,所设的区域就成为 A 诗句的正确目标区域。

在On Drop列表框中有三个选项,程序运行时,在将对象拖动释放后,对象在窗口中的位置就由它们来控制。单击 Snap to Center选项,在程序运行时,当用户最终将对象拖到正确目标区域释放后, Authorware将自动把该对象放置在目标区域的中央。此选项常常是用来设置对象移动正确的选项。

 提示 1) 单击Leave at Destination选项,在程序运行时,当用户将对象拖到正确目标区域附近处释放,对象将停留在当前位置,但此位置并不一定是目标区域的中央。2) 单击Put Back 选项,在程序运行时,当用户拖放对象的位置不正确,对象将返回原处。这个选项通常在错误响应中设置为 Try Again分支,这是由于当你将对象移错位置时,该分支还会让你重试。

单击Accept Any Object 选项,设定的目标区域将会接收任何对象,而不管对象移动是否正确。

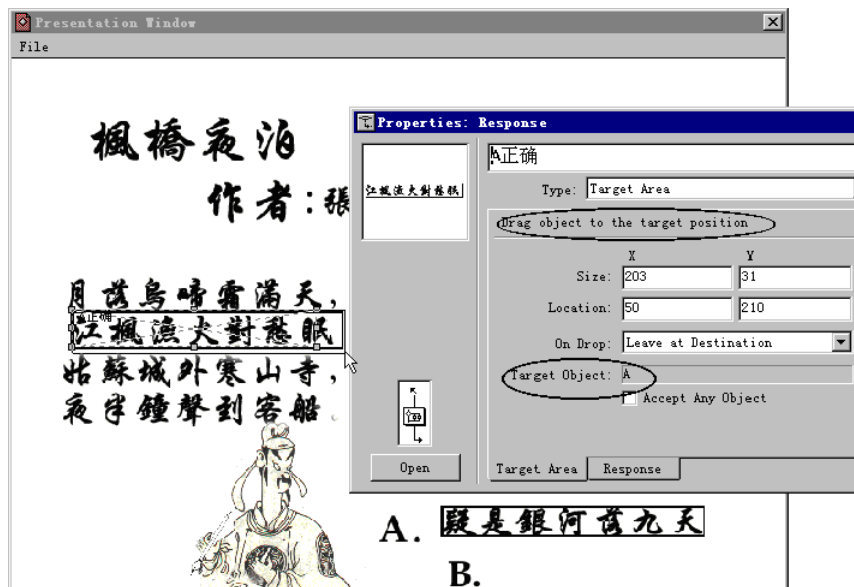


图5-31 调整目标区域

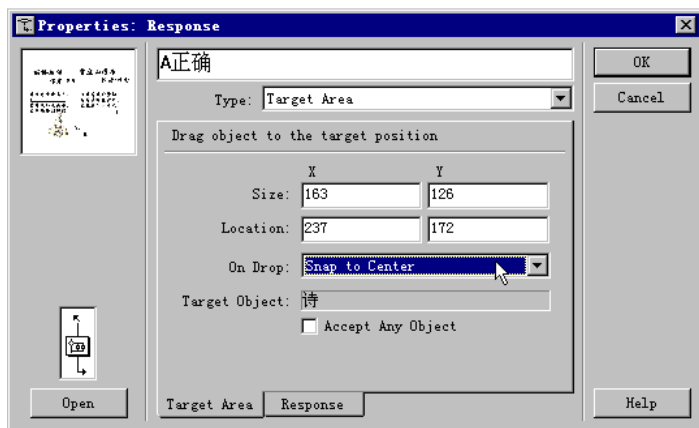


图5-32 响应属性对话框

设置完“A正确”对话框，单击OK按钮。Authorware将弹出下一个分支的响应对话框要求你设置错误区域。如图5-33所示，同样选择诗句A作为对象，然后用鼠标将目标区域调至全屏大小。由于此分支设置为错误选项，所以应设置Put Back选项，这样，当对象的拖动位置不正确，该对象会返回原位置要求你重试。

技巧 由于我们设置程序的运行窗口大小是640×480，因此可以直接在对话框的Size和Location文本框中输入相应的坐标。

单击工具栏上的“运行”按钮运行程序，当画面出现时，单击流程线上的“交互”图标，在窗口中就会出现如图5-34所示的所有的目标范围。可以根据对象来调整目标区域，如图5-34中鼠标所示。

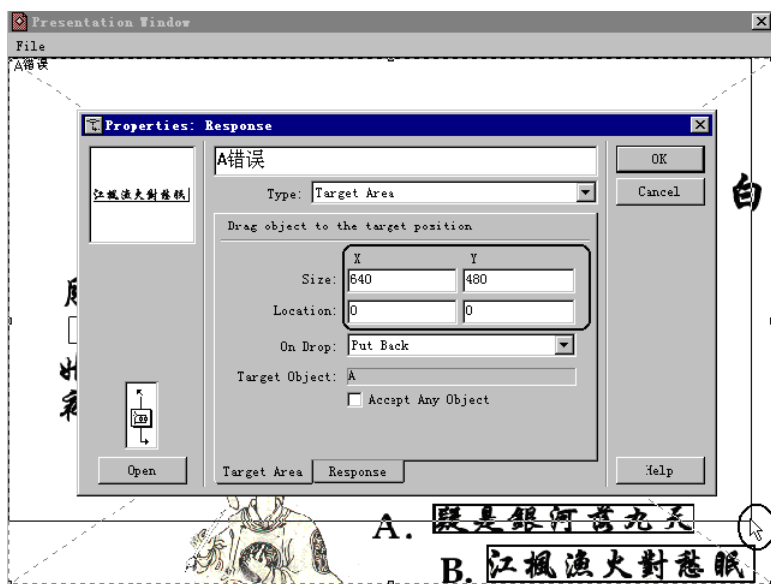


图5-33 设置错误区域

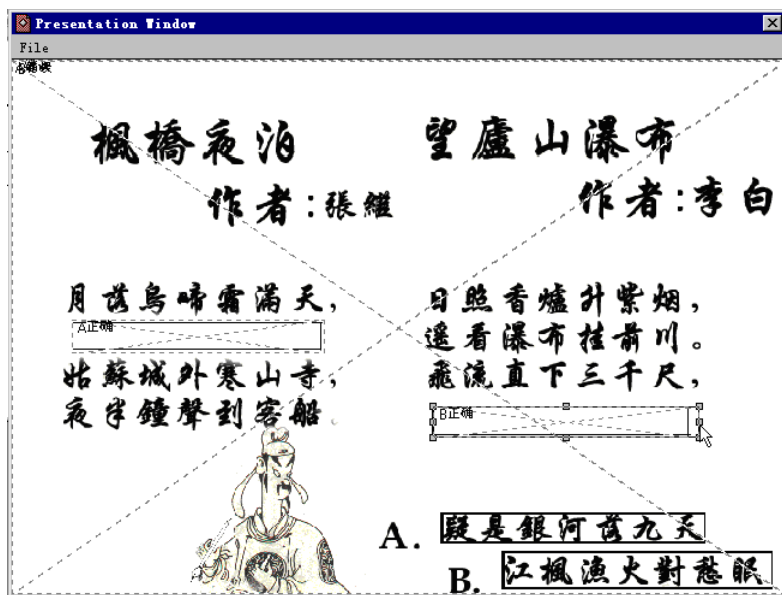


图5-34 目标区域

看到窗口中重叠的区域，用户也许会提出这样的疑问：窗口中的错误区域是将正确区域完全覆盖，在程序运行时，难道不会出现错误响应与正确响应同时执行的情况吗？当然不会！Authorware的响应优先级是根据分支距离“交互”图标的远近来确定，当执行了正确操作，Authorware就会执行正确的分支，然后返回“交互”图标等待下一次交互操作。如果觉得屏幕上的区域太乱，也可以先不运行程序而直接打开“诗”图标，然后双击“交互”图标，接下来就可以调整目标区域了。

运行程序后，程序并没有达到所要求的效果，不管移动对象是否正确，对象总会在释放位置停留，这是怎么回事啊？问题是出现在图片的移动属性上，前面我们给大家提到过“显示”图标的Layer选项卡，里面的选项就能实现图片各种类型的移动属性。打开显示图标属性对话框，如图5-35所示，选择Movable下拉列表框中的Anywhere(任何位置)选项，设置的图片对象就可以被移动了。

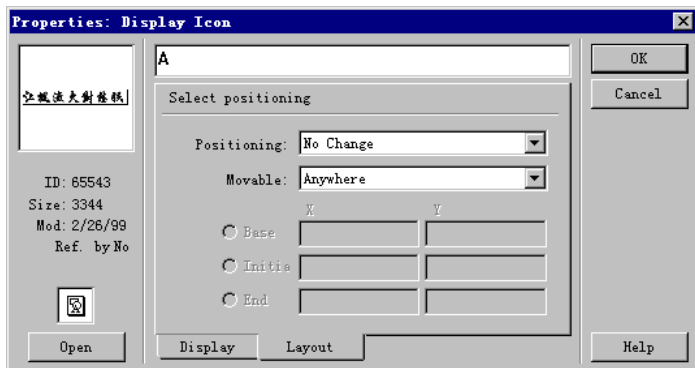


图5-35 设置对象移动属性

最后，再将“A正确”、“A错误”、“B正确”和“B错误”显示图标中分别输入“正确，请继续！”和“错误，请重试！”文字。运行程序时，画面上就会出现判断结果了。

5.6 使用文本输入响应——实例“常识问答”

在本节我们将介绍Text Entry(文本输入响应)的有关设置，其实，在前面我们已经接触过这类响应方式。在本书第2章的示例中，曾经使用该响应制作了一个“考试.a5P”程序，用户现在就可以打开此程序，让我们来具体了解一下文本输入响应的真正面目。

首先，在流程线上选择“交互”图标，然后打开其响应属性对话框，最后单击对话框中的Text field(文本区域)按钮，如图5-36所示，Authorware将会弹出Properties: Interaction Text Field对话框。同时，在对话框后面的窗口中出现一个黑色三角，当程序运行时，该三角符号后面将出现闪烁的光标，就可以在后面输入各种文字了。如图5-36中所示，用鼠标拖动文本输入区域的调整方块可以改变它的大小，它的设置与热区的设置基本相同。在Character Limit(字数限制)文本框中，可以输入某个数值，该数值用来限制用户输入响应字符的个数，比如，在图5-36的文本框中输入数字1，那么当程序运行时，在光标处1个字符后，将不能继续输入。

单击Auto-Entry at Limit(自动限制输入)复选框，在程序运行时，如果输入的字符个数达到限制的字符数，Authorware将自动结束输入。用户可以无须再按确认键(如回车键)程序就自动往下执行程序。



提示 在文本框中输入的空格也将被算作字符。

打开Interaction选项卡，如图5-37所示，用户可以在Action Key文本框中输入文本输入结束的确认键，如前面提到过的回车键Enter。也可以定义其他的功能键，如End，这样，当用户在文本区域输入框内输完字符后，按键盘上的End键就可以确认。



图5-36 改变文本输入区域

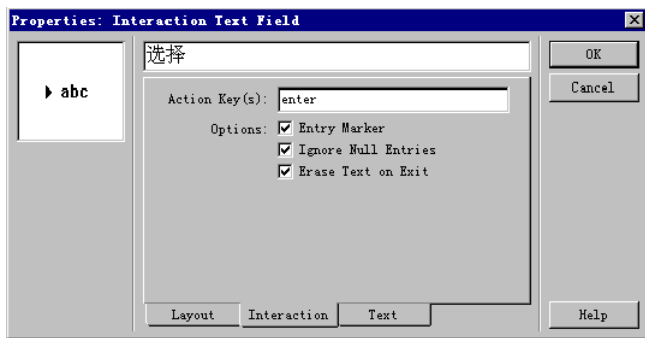




图5-37 文本区域响应对话框

 **提示** 1) 单击Entry Marker(输入标志)复选框, 在程序运行时, 文本输入区域前将显示文本输入标记“|”。2) 单击Ignore Null Entries(忽略空字符输入), 如果在文本输入区域内未键入任何字符就按下确认键, Authorware会要求用户重新输入。3) 单击Erase Text On Exit(退出时擦掉文本)复选框, 则在退出交互时擦掉输入的字符。

打开Text选项卡, 其中的选项可用来设置输入文字的字体、字号、字型、颜色及字体模式。例如, 在Font下拉列表框内可设置输入字符的字体; 在Size列表框中可选择输入字符的字号; 单击Style各选项的复选框可以设置字符的字型; Text颜色按钮或Background颜色按钮将弹出Color调色板, 单击颜色方块来设置字符颜色或字符背景颜色; 在Mode下拉列表框中可以选择输入文本对象的重叠模式。设置完毕, 还可以根据预览框中显示的示例来决定各个选项, 如图5-38中黑圈内所示。

 **技巧** 双击文本输入区域也能打开Properties: Interaction Text Field(文本区域响应属性)

对话框。

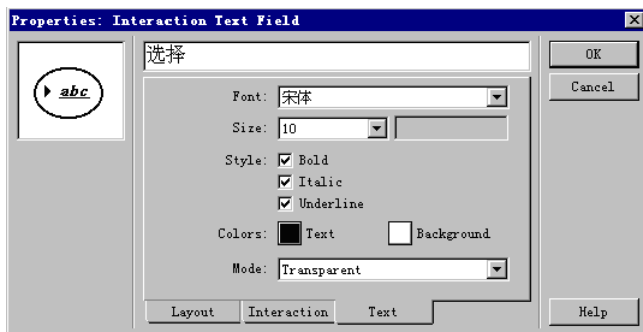


图5-38 文字设置

返回流程线，用鼠标双击响应类型按钮，打开 Properties：Response(响应属性)对话框，如图5-39所示。应该引起注意的是，在 Pattern文本框内的字符与该响应的分支标题相同，这样“交互”图标就会根据用户输入的字符来判断是否产生响应。例如：在 Pattern文本框内输入The word，当程序运行时，在文本输入区内输入此词组程序就会得到响应。如果想使用多组字符来匹配此响应，那么在這些字符之间要加上“|”。例如，在 Pattern文本框内输入Boy|Girl，则程序运行时，输入Boy或Girl都会产生响应。

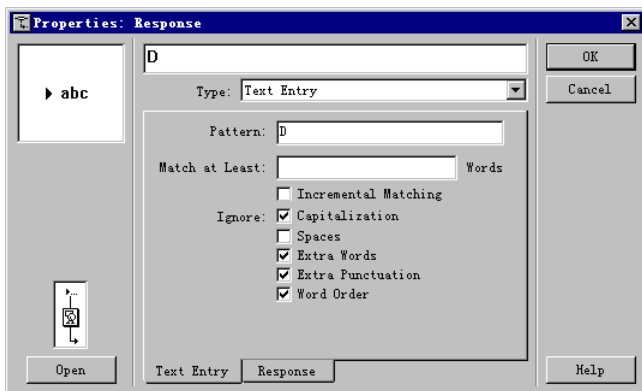


图5-39 文本输入选项卡

可以在Match at least(至少匹配几个字符)文本框内输入数字，该数字决定输入字符与分支标题至少应匹配的字符数目。例如在此处输入数字 2，如果分支标题是 Boy，那么只要输入Bo即可产生响应。

单击Incremental Matching(增加匹配)复选框，该选项可以达到如下效果：如果分支标题为A Boy，你可以先输入 A，然后按确认键，随后再接着输入 Boy，按下确认键后也能产生响应。




提示 1) 单击Capitalization(忽略大小写)复选框，Authorware在判断文本是否匹配时将忽略大小写。2) 单击Spaces(忽略空格)复选框，将忽略空格。3) 单击Extra Words(忽略多余单词)复选框，Authorware 将忽略输入的多余单词。4) 单击Extra Punctuation(忽略

标点)复选框, Authorware 将忽略标点符号。5) 单击 Word Order(忽略单词顺序)复选框, Authorware 会忽略单词的输入顺序。

关于通配符的补充说明:

如右图交互流程, 当输入字符串含有字母 P 时, 响应第一分支; 输入以 a 打头的两个字符时响应第二分支; 连续输入 3 次 f 时, 响应第三分支; 输入 a、b、c 中任一字母时响应第四分支; 若响应标题为 “|” 或 “*” 时, 输入任何字符都能响应。

 注意 响应优先级取决于该分支距离“交互”图标的距离, 近者优先。

5.7 使用条件响应——实例“健康咨询”


学完前面的几种响应后, 估计用户对 Authorware 的制作交互程序的思想也有所了解, 当然也可以这样来说, 当程序执行到“交互”图标时, Authorware 便等待程序发出命令, 一旦符合某种响应方式, 程序就自动进入相应的分支来执行, 其中 Conditional(条件)响应无疑是这种说法的最典型的代表。在条件响应程序的执行中, 当某一变量的数值满足交互分支的要求时, 程序便开始执行这一分支中的内容, 下面所讲的例子的具体思路就是这样。

在一个条件响应程序中, 允许设置多个条件来满足变量的各种变化范围。例如下面举的一个“健康咨询”的例子, 在这个例子中我们先做如下假设: 若体重(斤)与身高(厘米)的比值大于 0.8 时, 略显肥胖; 比值小于 0.6 时略显消瘦; 当比值在 0.6 与 0.8 之间时, 则肥瘦适中。在这个例子中, 我们就要定义一个变量来代表体重与身高的比值, 如下文所用的 Result。

如图 5-40 所示, 这是程序运行后的画面, 画面上病人在向医生询问病情, 如果在画面上的光标处输入你的体重和身高值, 按回车键后, 医生就会给出正确的答复。



图5-40 运行画面

 注意 在输入的体重值和身高值之间要有空格或 “/”, 如 100 180 或 100/180。

此程序设置的流程线如下,如图 5-41所示,在黑圈内的流程用来实现文本输入,即病人往程序中输入的体重值和身高值。在“计算”图标中,通过设置一对变量来获取这组值。由于屏幕输入的值要赋予“计算”图标中的变量,所以该分支标题要虚设,即在响应属性对话框的Pattern文本框内不输入任何字符。同时,此处的交互分支类型应设为 Exit Interaction,这样,在输入数值后,程序将继续往下运行。

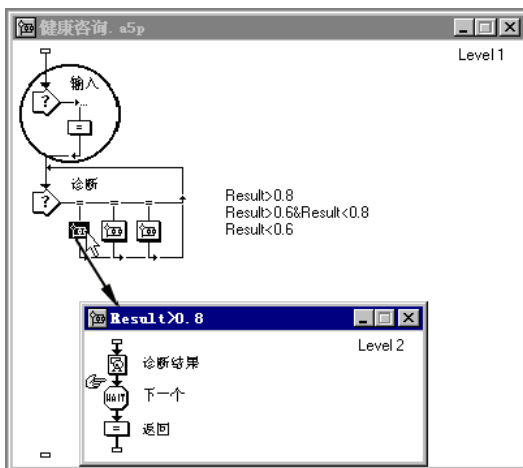


图5-41 流程线



提示 可以打开此处的“交互”图标,然后在其中输入图5-40所示的程序画面。

双击“计算”图标,打开计算图标文本框,如图 5-42所示。在文本框内输入“a:=NumEntry”和“b:=NumEntry2”,其中NumEntry是一个系统变量,它会获取键盘的输入值。因此,这两条语句的功能是分别将键盘输入的第一个值和第二个值赋予变量a和b。而“Result:=a/b”语句的功能是将a和b的比值(也就是体重与身高的比值)赋予变量Result。



图5-42 设置计算图标

注意 在输入时一定要加冒号,在“=”号前面要加上“:”,“:=”符号具有赋值功能。这与C语言和Basic语言不同。

单击文本框上角的“关闭”按钮, Authorware会弹出一个New Variable(新变量)对话框,该对话框要求给新变量赋初值及进行变量描述,如图 5-43所示。在Initial Value文本框中输入变量a的初值“0”,然后关闭对话框。接下来, Authorware还会弹出b和Result的新变量对话框,同样可以按上述方法设置它们。有关变量的具体信息,最好参考一下本书的第8章。

返回流程线,双击响应类型按钮,打开 Properties : Response(响应属性)对话框,如图 5-44所示。在Condition文本框中输入Result>0.8,这样,当变量Result的值为大于0.8

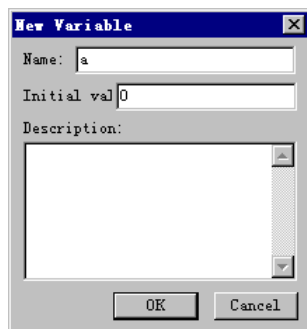


图5-43 变量赋初值

的数值时满足该条件，Authorware将执行该分支。打开 Automatic 下拉列表框，如果选择 When True 选项，在程序运行时，Authorware 就会根据条件变量的值来判断响应是 True 还是 False，一旦符合条件，Authorware 将执行此响应分支。若选择 On False to True 选项，则只有在条件由假变为真时，Authorware 才会执行交互响应。如果选择 Off 选项，Authorware 将关闭判断功能，因此，只有在指明条件是正确时才能产生响应。

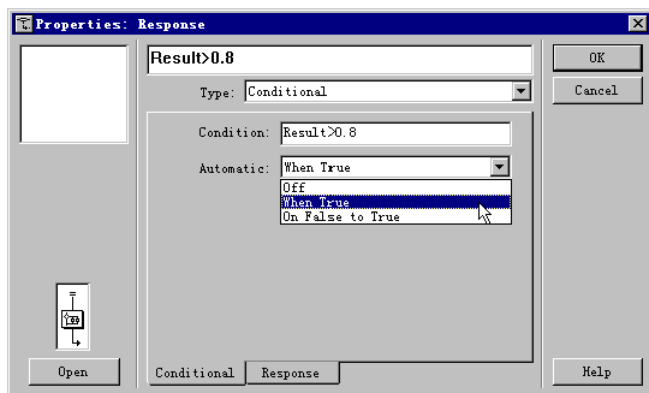


图5-44 设置条件

在 Automatic 列表框中选择 When True 选项后，单击 OK 按钮。打开交互分支中的 Result>0.8 群组图标，如图 5-45 所示，在流程线上放置“显示”图标用于存储诊断结果的画面。打开“返回”计算图标，在文本框中输入“GoTo(IconID@"输入")”语句，这样当程序执行到“计算”图标时，流程将跳转到输入主流线的“输入”显示图标处。我们还要在流程线设置一个“等待”图标，该图标的作用主要是使“诊断结果”画面暂停，同时，在屏幕上显示一个标签为“下一个”的等待按钮。有关等待按钮的样式的设置，请参考本书 10.1.4 节。



图5-45 设置程序跳转

下面再来看一下程序流程是怎样执行的：如图 5-46 所示，假如在光标处输入 100/180 这一组数值，当它进入“计算”图标，变量进行赋值。即 $a=100$ ， $b=180$ 程序， $Result=a/b=0.55$ 。

当变量 Result 进入“交互”图标后,由于 $\text{Result}=0.55<0.6$,所以它将执行第三个分支 $\text{Result}<0.6$,如图所示,当执行分支中的群组图标时,首先显示“诊断结果”显示图标,画面如图所示。然后,执行“等待”图标,即显示画面左上角的“下一个”按钮。单击此按钮后,程序执行“计算”图标中的 GoTo 语句,程序将再次返回流程线的最开始端,等待再次输入下一组值。

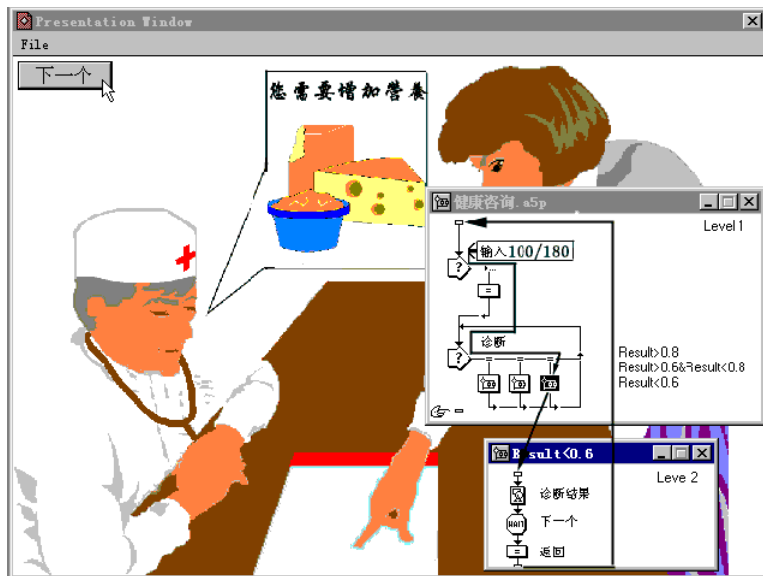



图5-46 程序流向

 注意 使用GoTo语句还有另外一个作用:如果把“返回”计算图标去掉,当输入一组数值后,程序将永久不断地执行一条分支,使用该图标将避免这一现象,不信可以试一试。

5.8 使用键盘响应——实例“纸牌游戏”

在前面讲的各种响应方式中,几乎都是采用鼠标来进行操纵控制,本节我们所讲的内容是Keypress键盘响应,即当程序运行时,可以通过键盘操作来使程序产生响应。

在下面的例子中,我们将使用键盘响应来编辑一个纸牌的小游戏。游戏开始后,当用户按下Continue按钮,画面上就会提示你出牌,而只要按下数字 1、2、3...J、Q、K等字母键,对应的纸牌就会弹出来。

如图5-47所示,根据题目要求编辑下面的流程线。在“交互”图标的右侧放置十三个“群组”图标,然后分别将它们设置为键盘响应,最后分别进行命名。打开“群组”图标,然后在第二级窗口中放置“显示”图标和一个“动画”图标,“显示”图标用于存放纸牌图片,“动画”图标用来控制出牌。

有关键盘响应的设置最为简单,双击响应类型按钮,打开如图 5-48所示的响应属性对话框。你可以在Key文本框中输入控制响应的字母键(a、b、c、d...)、数字键(1、2、3、4...)、标点符号和功能键(Enter、Esc、Insert、Home、End、PageUp、PageDown、F1~F12等)。

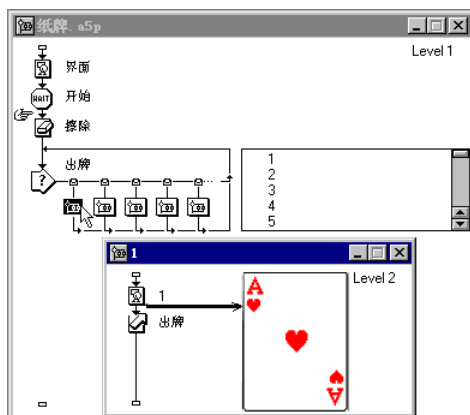


图5-47 流程图

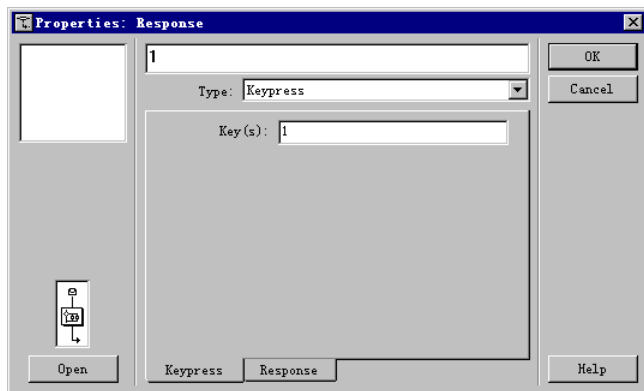




图5-48 响应属性对话框

 **注意** 在Key文本框中一定不要输入字符串，但功能键除外。

返回流程线，进入“群组”图标。先选择“显示”图标，然后按住 Shift键双击“动画”图标，打开Properties：Motion Icon(动画图标属性)对话框，如图5-49所示，设置动画方式为点到直线的动画。最后，将交互分支中的所有“群组”图标都如此设置。如果忘记了有关Authorware动画方式的制作过程，可以参考本书4.3节。

程序编辑完毕，下面再给小游戏添加一个启动画面。返回到前面，看一下流程图。在主流线上还设置了一个“等待”图标和一个“擦除”图标，它们的作用是暂停游戏界面和设置擦除效果。如图5-50所示，在画面左下角有一个Continue按钮，此按钮就是等待图标所做的设置，通过选择等待图标属性对话框中的Show Button复选框可以显示这个按钮。如图5-47中的流程线所示，单击Continue按钮后，程序将向下执行“擦除”图标，在“擦除”图标中我们设置擦除对象为界面，擦除过渡模式为Bar Door Open。图5-50所示的界面左右拉开的画面就是该过渡模式的效果。

 **提示** Authorware提供了百余种过渡模式，每一种模式都有其独特的效果，不必死记硬

背这些模式，只要在应用中多加观察、灵活运用即可。

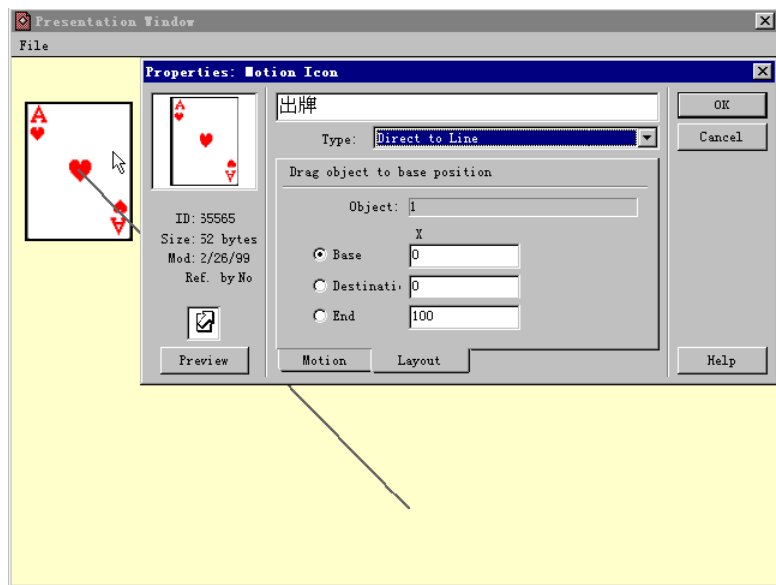


图5-49 设置动画



图5-50 设置界面拉开效果

5.9 使用重试限制响应——实例“开密码箱”

保险箱、保险柜、密码锁是日常生活中经常使用的防盗设施，有了它们就可以放心大胆地将金钱、首饰、密件以及其他一些隐密的东西放在里面，然后在加上一个自己熟悉的密码，这样别人就不会轻举妄动了。

在本节的内容中，将介绍 Authorware 的重试限制响应 Tries Limit，这种响应方式通常与文本输入响应相结合，它可以提供给用户重试密码的次数，如果在限制的次数内没有打开密码，Authorware 就会进入重试限制响应的分支，而不会进入文本响应分支。

下面我们举一个开密码箱的例子，在这个例子中我们设置密码的重试次数为 3 次，如果输入密码的次数超过三次，程序就会弹出一个报警的画面。如果在规定的次数内输入正确的密码，就可以打开密码箱了。

这个例子的流程图非常简单，如图 5-51 所示。我们在“交互”图标后面放置一个文本输入响应，第二条分支设为重试限制响应，更改分支标题为“限制次数”。在“重试限制”群组图标内设置第二级流程线，如图 5-51 中所示，在流程线上使用一个“显示”图标来设置报警画面。

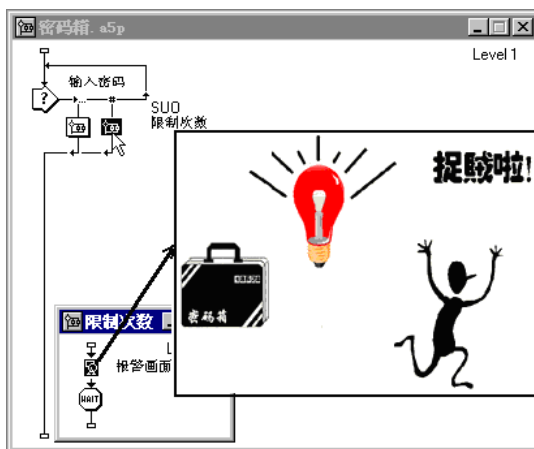


图5-51 流程图

双击重试限制响应类型按钮，打开 Properties : Response(响应属性)对话框。如图5-52所示，该对话框的设置同样简单。只需要在 Maximum Tries 文本框中输入重试响应的最大次数即可。如果在此文本框中不输入任何值，Authorware 将默认为最大次数是 0 或 1。在程序运行时，如果输入的超过此数值，“交互”图标就会执行该分支。



图5-52 响应属性对话框

单击OK按钮，返回流程线。如图 5-53所示，双击第一分支中的“群组”图标，在第二级窗口中设置流程线。在“显示”图标中存储一个打开密码箱后的界面图片，如图 5-53中箭头所示。在流程线最后的“秘密”群组图标中就可以放置需要保密的东西了。



图5-53 流程线

返回主流线，双击第一分支的响应类型按钮，打开文本响应属性对话框。如图 5-54的黑圈内选项所示，将所有选择的复选框都取消，也就是取消了 Authorware的一些默认的忽略选项。这样，取消忽略选项自然就增加了密码的安全性。

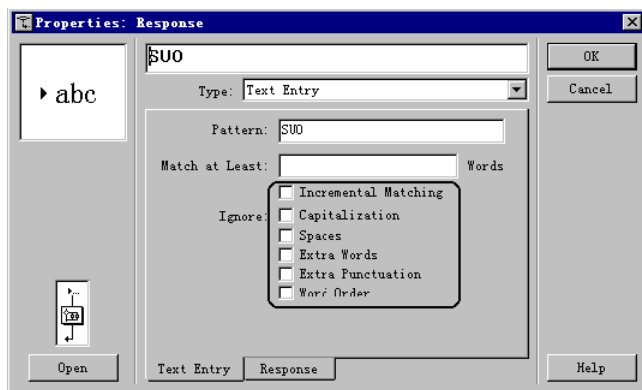


图5-54 设置响应属性

程序到此就设置完了，现在返回程序，试一下，看看是否象书上讲的这样。如果想把自己制作的多媒体作品加一层保护措施，可以在程序的最前面加上一个与此类似的重试响应，有了密码保护，就不会再会担心被别人偷着使用了。

5.10 使用菜单响应——实例“家庭影院”

与命令按钮一样，菜单也是各种软件普遍采用的一种命令控制方式。在 Authorware中，

菜单被用做一种交互响应的方式，因此我们可以为多媒体作品制作一个或几个菜单，然后用菜单里的命令来响应各个分支。一般来说，菜单里的每一个命令都要对应一条交互分支，所以，设置每一个命令必须对应创建一条分支。


在默认设置时，Authorware的演示窗口中有一个File菜单，菜单内只有一个Quit命令，该命令是用来控制程序退出的，另外，该命令还有一个快捷键 Ctrl+Q，按该快捷键也能使程序退出。下面我们讲的例子是在 File 菜单的基础上扩展 Authorware 的菜单条，然后在各个菜单中再添加一些相关的命令。对于菜单中的各条命令我们不能一一进行详细地讲解，只是想通过这个例子来了解制作菜单响应的思路及一般菜单命令的制作方式。

如图5-55所示，我们制作的这个作品一共有三个菜单，它们分别是 File菜单、“控制”菜单和“帮助”菜单。



图5-55 作品画面

对应图5-55所示的三个菜单，在流程线上要使用三个“交互”图标来分别创建。如图 5-56所示，“File”交互图标与图 5-55的File菜单对应，“控制”图标与图 5-55“控制”菜单对应，“帮助”图标与图 5-55的“帮助”菜单对应。“交互”图标的各个分支标题就是菜单中的各个命令。除此之外，Authorware也可以制作子菜单，而子菜单的制作过程比较复杂，在一般的多媒体作品中使用菜单方式对初学者来说已经足够了。

 **注意** Authorware演示窗口的菜单File不能被取消，更不能更改为中文的“文件”菜单，但可以在File菜单中添加一些命令来丰富它。

打开作品中的File菜单，如图5-57所示，在两个菜单命令之间有一条分隔线，在我们平时所接触的菜单中也会经常看到，这条分隔线的制作非常简单，如图 5-56所示，只需在“打开”和“退出”分支之间加一条响应分支，然后将分支标题命名为“—”或“(—”即可。

另外，从图5-57中还可以看到，在菜单命令的右面都对应它的快捷键，例如，“打开”命令的快捷键是Ctrl+O，单击“打开”命令和按Ctrl+O快捷键都能执行“打开”命令。单击响应

类型按钮，打开响应属性对话框，如图 5-58所示，可以在 Menu(菜单)选项卡的 Key 文本框中输入快捷键。快捷键的格式是功能键再加上字母键，如图中的 Ctrl O 就代表快捷键 Ctrl+O。

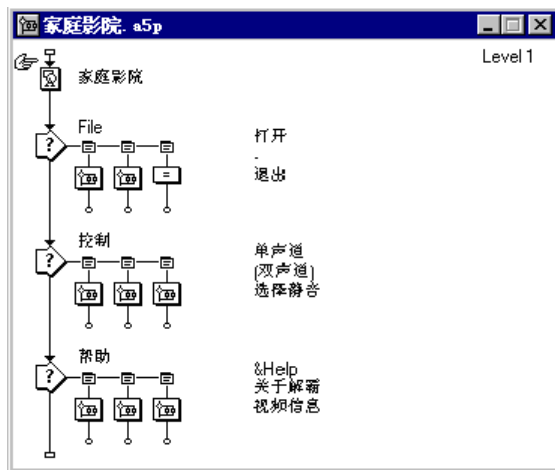


图5-56 流程图

如图5-59所示，在“控制”菜单中有一个灰色命令，该命令在程序执行时不可用。要制作此命令效果，用户可以在图 5-56的“控制”交互图标的“双声道”标题上加“()”或“()”。实际上，这样制作的灰色命令没有多大的实用价值，而真正有用的灰色命令在使用时会被激活，通过设置 Active If 文本框中的变量可以完成这一效果。单击 Response 标签，打开 Response 选项卡，在相应文本框中设置即可。例如，可以在 Active If 文本框中输入一个变量表达式 “Use=1”，这样，当 “Use = 1” 时，命令变灰色；当 “Use=1” 时，命令就变为黑色。不信现在就可以试一下。

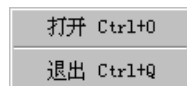


图5-57 File下拉菜单

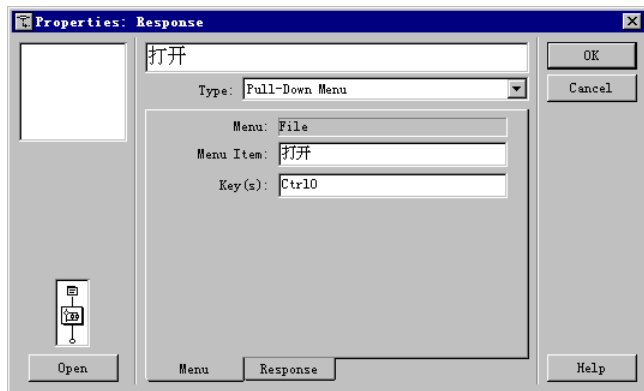


图5-58 设置快捷键

打开“帮助”菜单，如图 5-60所示，菜单中的 Help 命令下面有一条下划线，这也代表一种快捷键的方式。即打开 Help 菜单，然后按下 H 键也能执行该命令。要实现上述的命令方式，可以在快捷键字母前加上“&”字符。若要使“&”也作为快捷键，可用“&&”来表示。



 提示 如果快捷键的字母是单词的第二个字母，还可以将&加在第二字母前面，依次类推。

图5-56中流程线的形状在前面曾经提到过，用户可以打开 Response 选项卡，如图5-61所示，选中Perpetual(持续性响应)选项，流程线就会直接穿过“交互”图标。当一个响应被设置为持续性交互响应时，在整个程序的执行过程中可以随时匹配这个响应。

 提示 选中Perpetual复选框后，Branch下拉列表框中增加了Return选项。由于本程序的设置为持续性菜单响应，因此可以随时使用菜单中的命令。

在File菜单下制作了一个“退出”命令，该命令的功能是退出“家庭影院.a5P”，即在程序运行时单击“退出”命令，程序就会终止。要实现该功能，可以双击图5-56中的“退出”计算图标，在打开的文本框中输入 Quit(0)命令，如图5-62所示。

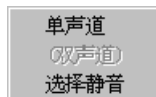


图5-59 灰色命令

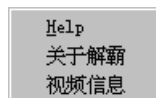


图5-60 下划线快捷键

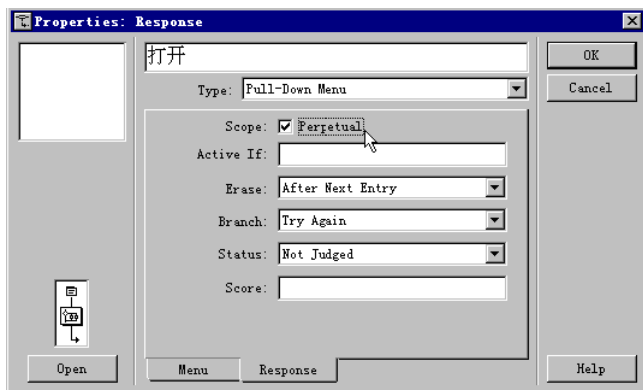


图5-61 设置永久属性

用同样的方法，用户也可以设置帮助信息，这涉及到使用Authorware的JumpFile函数，可以参考本书第8章进行函数设置。

5.11 使用时间限制响应——实例“冒险岛”

在Authorware中，还提供了一种时间限制响应的交互方式，只要用户在规定的时间内没有做出选择，“交互”图标就会执行时间限制响应的分支。在下面的例子中，我们也举了一个类似的例子，如图5-63所示，在画面上有一个四面环海的小岛，岛上有一颗硕大的钻石，现在就由你来决定是走左面的桥还是右面的桥，如果走错，岛上的炸弹就会把画中人炸死；如果走对了，就可以发一笔横财。另外，设定的时间限制是3秒，超过此时间，岛上的炸弹就会自动引爆，将钻石炸飞。应该知道，这不是在考判断能力而是看运气好坏，怎么样？来试一试吧！

看完上面的内容，首先要考虑一下制作的思路，程序中涉及到对象的移动，因此当然要选择目标区域响应的交互方式，如图5-63所示，我们要设置两个热区，以便你用鼠标移动图

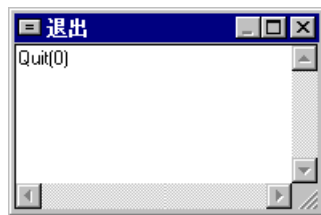


图5-62 退出函数

中的小人。另外，还要使用本节要讲的时间限制响应。

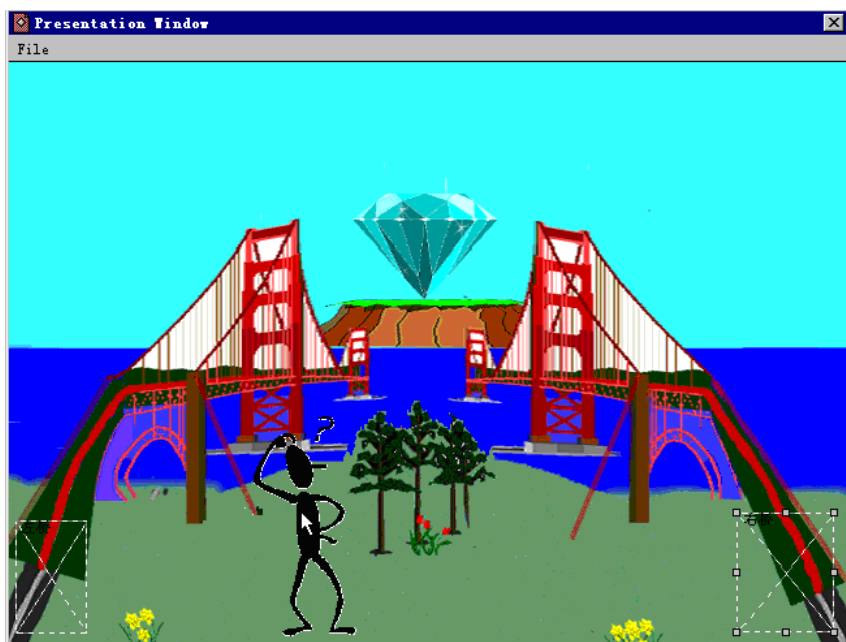


图5-63 程序运行画面

如图5-64所示，先来创建程序所要求的流程图，在“人”显示图标中放置移动的对象人，如图中的箭头所示。在“交互”图标的右侧设置三条分支，前两条分支用于设置目标区域响应，最后的分支设置时间限制响应。当在规定的时间内没有进行移动操作时，“交互”图标就执行时间限制响应分支。

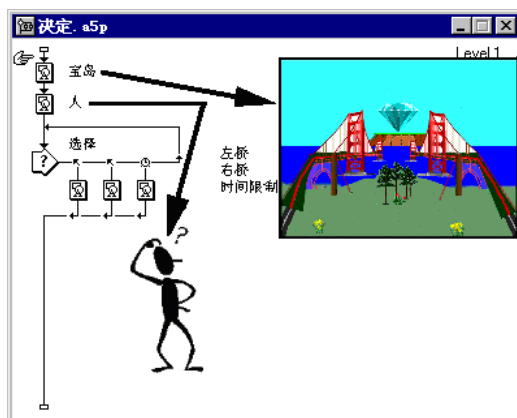


图5-64 流程图

双击第一分支的响应类型按钮，弹出的对话框如图 5-65所示，根据对话框的提示，用鼠标将“人”图片移至窗口的左下角，即画面上左面的大桥路口。然后将 On Drop 下拉列表框中的选项设为 Snap to center。最后，用同样的方式将流程线上的第二条分支设置完毕。

设计完目标区域响应后，返回主流程线。双击时间限制响应类型按钮，弹出的对话框如图5-66所示，在Time Limit(限制时间)文本框中输入交互响应的限制时间为3秒。



提示 1) 若单击show Time Remaining复选框，在程序执行时，屏幕上会出现一个时钟，用来显示剩余的时间。2) 若单击Restart for Each Try复选框，在程序执行时，如果同时又在执行交互中的另一个响应时，系统将重新计时。

打开Interruption(中断)下拉列表框，其中设有四个选项，它们是用来设置响应执行过程中同时又执行其他内容时的计时方式。各选项意义如下：

- 1) Continue Timing：时钟不暂停，继续计时。
- 2) Pause, Resume On Return：暂停后，响应过后继续计时。

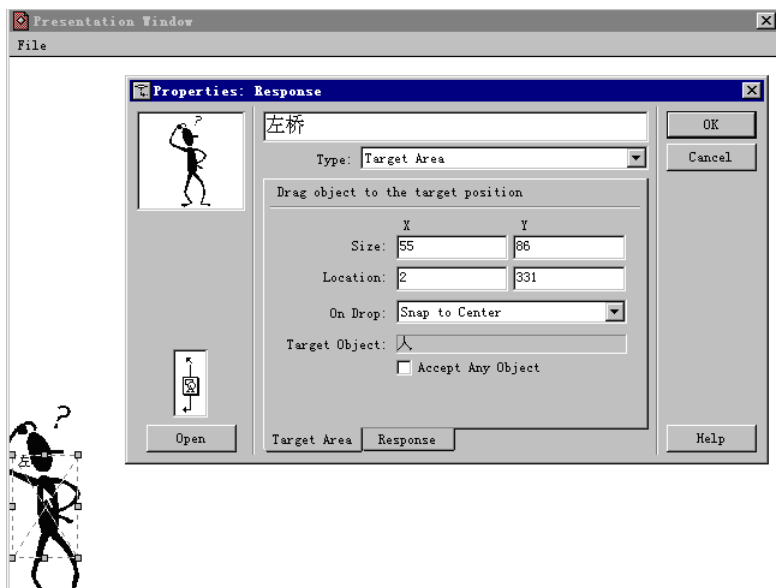


图5-65 设置目标位置

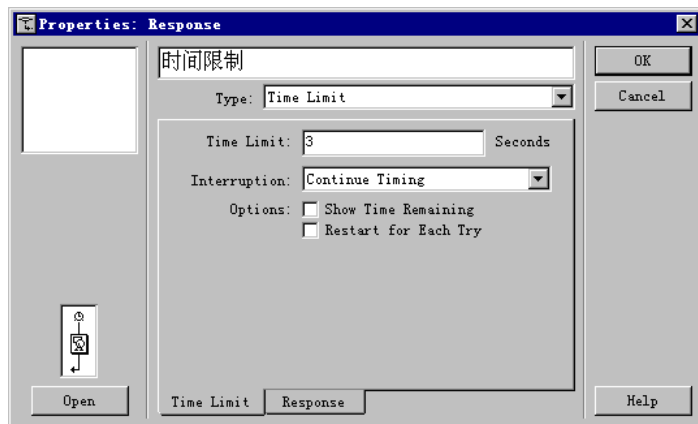


图5-66 设置时间限制

3) Pause, Restart On Return: 暂停后, 重新计时, 允许交互时间超出限制时间。

4) Pause, Restart If Running: 暂停后, 若未超时则重新计时。

最后, 在分支中还有三个“显示”图标没有设置, 在“左桥”图标中可以绘制一幅庆祝发财的画面, 在“右桥”图标中绘制一幅炸弹爆炸的画面, 然后在“时间限制”显示图标中拷贝同样一幅炸弹爆炸的画面即可。

到这里为止, 有关交互响应的例子就都讲完了。除此之外, Authorware还新增了一种更为复杂的响应方式——“Event”事件响应, 与其他响应方式不同, 事件响应是实现计算机同Xtra文件之间的交互, 其中最主要是同Active X控制的交互。

事件响应主要涉及到有关Windows的Active X标准问题。原理比较复杂, 对它的理解需要有关高级语言的基础, 一般来说只适用Windows的高级用户, 对于初级用户来说, 掌握了上面所讲的十种交互方式, 就已经能够达到制作优秀的多媒体作品的要求了。