

第1章 入门

这一章首先介绍如何安装随书所带软盘,以及如何运行这些游戏。此外,还对如何使用书中的源代码清单,如何装入和修改软盘中的源代码文件作了说明。

1.1 安装附带软盘

与本书配套的软盘包括所有游戏和游戏的帮助文件,以及在可视 Basic 环境下生成游戏需要的所有源程序,因为软盘中的文件是压缩文件,所以还须运行安装程序来将它们拷贝到你的硬盘上。

运行安装程序,在 Windows 下:

1. 将软盘放在软盘驱动器内。
2. 激活 Windows 程序管理器或 Windows 文件管理器。
3. 从 File 菜单上选择 Run 命令,弹出 Run 对话框。
4. 在命令行框内键入下列内容:软盘驱动器的盘符、冒号、反斜杠(\)和 install。例如,如果软盘在 A 驱动器中,则键入:

a:\install

5. 单击 OK 按钮(安装程序开始运行)。
6. 选择安装程序窗口内的一个或两个安装选项:
 - 安装游戏和游戏帮助文件
 - 安装游戏程序源文件
7. 缺省情况下,安装程序将文件全部拷贝到 C:\VBGAMES 目录下。如果想指定一个另外的目录,须将一个完整的目录路径键入到目标目录框内。如果被指定的目录不存在,安装程序就创建相应的目录。
8. 单击 OK 按钮。

如果选择了“安装游戏和游戏帮助文件”,安装程序就将每个游戏的可执行文件(EXE)和帮助文件(HLP)拷贝到\VBGAMES 目录下(如果指定了不同的目录名,就用指定的名字替代\VBGAMES)。然后安装程序就会在程序管理窗口内产生名叫“VBGAMES”的新程序组。这组程序包括每一个游戏的图标,便于运行这些程序。

如果选择了“安装游戏程序源文件”,安装程序将每个游戏的源文件拷贝到\VBGAMES 目录下的一个单独子目录下。下表列出了每个游戏的可执行文件和帮助文件的名字,以及用

来存放源程序的子目录名。

游戏(章)	可执行文件和帮助文件	源文件目录
法国钉(第3章)	PS.EXE, PS.HLP	\VBGAMES\PS
列骰子(第4章)	DD.EXE DD.HLP	\VBGAMES\DD
字谜(第5章)	WS.EXE, WS.HLP	\VBGAMES\WS
皇后问题(第6章)	QP.EXE, QP.HLP	\VBGAMES\QP
拼板游戏(第7章)	FP.EXE ,FP.HLP	\VBGAMES\FP
拼板生成器(第7章)	FRACTL.EXE	\VBGAMES
井字棋(第8章)	TT.EXE, TT.HLP	\VBGAMES\TT
十字跳棋(第9章)	LD.EXE ,LD.HLP	\VBGAMES\LD
圆盘跳棋(第10章)	RG.EXE, RG.HLP	\VBGAMES\RG
三角积木(第11章)	TP.EXE ,TP.HLP	\VBGAMES\TP
海盗战(第12章)	GW.EXE, GW.HLP	\VBGAMES\GW
轮盘赌(第13章)	BL.EXE, BL.HLP	\VBGAMES\BL

如果只安装了游戏或游戏源文件,可以随时再次运行 Install 来安装另外一部分。

1.2 玩游戏

安装完毕后,只要在程序管理窗口内双击新生成的“VBGAMES”组内的某个游戏的图标,就可以启动一个游戏。每个游戏的完整说明将在相应游戏的章节中介绍。也可以进入联机帮助,既可以按 F1 键浏览帮助目录,也可以从 Help 菜单上选择某一特定专题。联机帮助还包括关于使用每一菜单命令的解释和说明信息。

1.3 源代码及有关说明

每一章的编码部分(例如“编码十字跳棋”)包括该章所介绍游戏的一套完整的源程序清单和图表。直接把这些源文件输入到可视 Basic 中就可以生成游戏可执行代码(直接把软盘中的源文件装入要容易得多)。书中包括源程序及有关说明的主要目的是帮助读者学习游戏程序的编写方法。

每个游戏的源代码及有关说明包括六个部分:

- 设计时带标签的主表格,并给出该表格的名称和赋予主表格的所有控制的名称。
- 列出所有赋予主表格及其每一控制的非缺省(nondefault)属性表,该表只包括那些在设计时赋值的属性设置,不包括那些在运行时用程序赋值的属性。它也只包括那些直接用可视 Basic 属性杆设置的,或通过直接在屏幕上缩放或移动目标设置的属性,不包括可视 Basic 自动赋值的属性设置(例如,如果想改变高度(Height)属性设置,可视 Basic 将调整 Scale Height 属性到正确的相应值,这时表中只包括高度属性)。最后,请注意属性表中也包含表格的栅距(用 Edit 菜单上 Grid Settings 命令设置),尽管这些值不是实际的属性。
- 一个表格菜单设计描述表。通过选择表格并打开菜单设计的 Window 对话框就可以产生菜单(从可视 Basic 的 Window 菜单中选择菜单设计 Window 命令)。对于菜单的每一项,表中第一栏是必须在菜单设计的 Window 对话框内标题(Caption)框中键入

的标题,表中的第二栏是在菜单设计的 Window 对话框内的控制名(CtlName)文本框内键入的菜单项名称,表中的第三栏是给菜单项赋值的非缺省设置。例如,菜单项可赋值给一个加速键(通过加速器下拉(drop-down)列表框),它可被检验(通过选择 Checked 项),或者被禁止(通过关闭 Enable 选项)。

- 带标签的 ABOUT 框的图象,给出表格名称和赋给该表格的所有控制名(有关产生这种表格的信息,请参见第 2 章中“显示 ABOUT 对话框”一节)。
- 列有赋给 ABOUT 表格及其所有控制的非缺省属性表(该表与上述给主表格赋非缺省属性的表类似)。
- 程序的可视 Basic 源代码清单分为:全局模块(如果有),主表格的代码,ABOUT 表格的代码。每一部分内,代码都将以它在可视 Basic 代码窗口内出现的顺序列出;说明、以字母顺序编排的一般过程、以字母顺序编排的事件过程。

1.4 装载和修改源程序

玩游戏只是本书程序提供的娱乐和挑战的一部分。在已经玩了一段时间,并且通读了关于这些游戏编制解释的章节后,读者就可以着手增加一些特征或增强游戏。每一章的最后部分推荐了可以自己动手进行的一些增强方法,例如,如果觉得某一游戏太简单了,就可以自己动手改进使之变难一些。

因为附带软盘中提供了所有游戏单独的可执行文件,所以不必将游戏装入可视 Basic,只需单独运行就行了。然而,如果你想检查源文件代码,对游戏作一些改动或增强其功能,就必须首先将源文件装入。

如果还没有将源文件装入硬盘中,就按照本章前一部分所介绍的安装命令进行安装。要安装某一特定的源文件,从可视 Basic 的 File 菜单上选择 Open Project 命令,选出相应游戏的项目文件名。项目文件是用相应游戏的两个缩写字母来命名的,然后是文件扩展名 MAK。上述第 2 页表中列出每一游戏含有的源文件(项目文件)的目录名,如下表所示。

例如,要安装法国钉游戏的源文件,就要指定下列文件。

C:\VBGAMES\PS\PS.MAK

假设已在 C 盘的 VBGAMES 目录下安装了软盘文件,当打开项目文件时,可视 Basic 就装入程序的所有源文件,当源文件装载好后,就可以通过在 Project 窗口内选择文件进行检查和修改了。

下表简要说明了软盘中所提供的源文件。

源文件	说 明
GLOBAL???. BAS (GLOBAL. BAS、GLOBALFP. BAS 等)	全程模块,包括全程类型定义和变量说明。
???. FRM(BL. FRM、DD. FRM 等)	定义主表格及其所包含控制的文件
?? ABOUT. FRM (BLABOUT. FRM DDABOUT. FRM 等)	定义 ABOUT 对话框及其所含控制的文件
???. MAK(BL. MAK、DP. MAK 等)	程序的项目文件,一个项目存储了构成程序的源文件的清单

源文件	说 明
*.BMP(例如 BL.BMP 和 TILE.BMP)	在 Window Paintbrush(画笔)内产生的位图(Bitmap)图形文件,这个文件中的图形赋值给表格和图画盒的 Picture 属性。
*.ICO(例如 BL.ICO、DD.ICO 和 DD1.ICO)	图标图形文件,这些文件中的图形赋值给图画盒 Picture 属性(类似于 BMP 文件),或赋给表格的 Icon 属性(为产生程序的图标)

记住:不象其它的源文件,当打开项目文件时,BMP 和 ICO 图形文件是不装入可视 Basic 的,这是因为这些文件中的数据已经包含在表格中了(也就是说,图形数据已经存储在 FRM 文件中了)。这些文件也在磁盘内,以备编辑图像和替换表格中的原图。可以用 Window Paintbrush(画笔)来编辑 BMP 文件,用可视 Basic 中的 Icon Works 程序或其它图标设计软件来编辑 ICO 文件(将图形嵌入程序中去完整指令将在后续的章节中给出)。

第 2 章 通用程序设计技巧

本章所介绍的三个基本程序设计技巧在本书介绍的所有游戏中都将会用到。这三个技巧是：

- 从可视 Basic 程序中显示 About 对话框
- 通过调用 Windows 应用程序接口函数来扩展可视 Basic。
- 产生用户联机帮助。

2.1 显示 About 对话框

About 对话框是 Windows 程序的一个标准特征,它可以用来显示程序作者的名字、版权通告、程序版本和序列号以及其它一些信息。当用户从游戏的 Help 菜单上选择 About 命令时,本书中所介绍的每一个游戏都显示一个简单的 About 对话框。

About 对话框包括游戏名、作者名、程序图标和一个用户单击就可关闭对话框的 OK 按钮。图 2-1 就是一个 About 对话框的示例(摘自第 3 章法国钉游戏)。

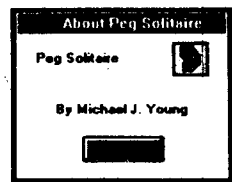


图 2-1 法国钉游戏的 About 对话框

将一个 About 对话框加到程序中去的第一步是从可视 Basic 的 File 菜单中选择 New Form 命令,执行了这一步后,可视 Basic 产生一个新表格,赋给它一个缺省名,然后可加上控制,设置表格和控制属性,并且用与定义主表格相同的方法加上代码。然而,为与 Windows 程序设计惯例相一致,对话框必须含有下表中所列的属性设置。

Property	Setting
BorderStyle	3—Fixed Double
ControlBox	True
Icon	(None)
MaxButton	False
MinButton	False
WindowState	0—Normal

对于每一游戏赋给 About 对话框的全部属性和控制将在介绍相应游戏的章节中给出。

2.1.1 装载 About 对话框

在本书介绍的所有程序中,About 菜单项命名为 MenuAbout(这个名字是在菜单设计的 Window 对话框内赋给的),About 表格命名为 FormAbout。因此,每当用户在运行时选择了 About 菜单命令,控制就传递给 MenuAbout Click 事件过程,这是加在主表格的代码的一部分,这个过程然后用如下的 Show 方法显示对话框:

```
Sub MenuAbout_Click()  
    FormAbout.Show 1  
End Sub
```

提示 通常必须在定义了主表格后产生对话框,如果首先产生对话框,就必须从 Run 菜单中选择了 Set Startup Form 命令,然后选择主表格以便可视 Basic 首先装载对话框。

传递给 Show 的属性值 1 表示对话框是模态的,即一旦显示对话框,直到单击对话框中的 OK 按钮关闭对话框之前,用户不能执行程序中的其它任何操作。

2.1.2 退出 About 对话框

当用户单击 About 对话框内的 OK 按钮,程序控制就传递给附属于对话框的 Command1 click 事件过程,这个过程通过如下的 Unload 方式关闭对话框:

```
Sub Command1_Click()  
    Unload FormAbout  
End Sub
```

2.2 调用 Windows 应用程序接口函数

通过直接调用 Windows 系统提供的函数,可以极大的扩展可视 Basic 的功能。这些函数统称为 windows 应用程序接口或 API。

因为应用程序接口函数不是可视 Basic 的组成部分,所以在调用它们之前必须作详细说明。说明必须放在它们被调用的模块的说明部分,或者放在全程模块内(这样,在程序中可以从任何表格或模块中调用它们)。在对函数作了说明后,就可以与调用自己编写的可视 Basic 函数、语句或过程一样的方式调用它们。

本书中介绍的所有游戏都调用了—个或多个应用程序接口函数。第一次使用一个函数时,相应的章节提供一幅图,该图列有该函数的说明,说明每一参数,并解释如何使用这个函数。例如 BitBlt 应用程序接口函数在第 7 章首次使用(在拼板游戏中);因此,第 7 章给出了解释这个函数的图。为便于参考,附录中按字母顺序列出了所有本书中用到的 Windows 应用程序接口函数。

注意: 可视 Basic 的手册中没有提供 Windows 应用程序接口函数的说明,也没有解释这些函数的应用。然而,当要用到本书中没有介绍的 Windows 应用程序接口函数时,买一本《Microsoft Windows 程序员参考手册》就可以得到需要的信息。(《Microsoft 可视 Basic 专门工具箱》也含有 Windows 应用程序接口函数的参考资料)

本书中的 Windows 应用程序接口函数描述与其它地方函数描述不同。首先是本书的描

述简化了,并重点介绍对可视 Basic 程序设计最有用的一些性能。另外,一些实际上带返回值的 Windows 应用程序接口函数被定义为可视 Basic 子过程(没有提供返回值),这是当返回值没有用时可以简化说明和函数调用(例如应用程序接口函数 SetCapture)。最后需说明的是,函数是用可视 Basic 的术语叙述的,而没有用标准 Windows 程序设计术语。例如,用术语“表格(form)”而没用“窗口(window)”,用“事件(event)”而没用“消息(message)”。

2.3 产生联机帮助系统

本书中所有的游戏都提供了包括有玩游戏和使用程序命令在内的全部指令的联机帮助,这些联机帮助系统利用了 Windows 提供的联机帮助工具软件。

用 Windows 的联机帮助工具软件开发一个联机帮助系统需要两个主要步骤。首先,必须产生一个含有想要显示的帮助信息的 HLP 文件,这个文件由用户程序创建。其次,须将代码加入用户程序以调用 Windows 联机帮助实用程序。

2.3.1 产生 HLP 文件

本书所带的软盘提供了所有游戏的 HLP 文件。然而,要产生自己的 HLP 文件,需要一些可视 Basic 本身没有的工具。除了必须有 Windows Help 编译程序,还必须有一个文字处理器,如 Windows 环境下的 Microsoft Word,它可以用“丰富的文本格式(RTF)”产生文档。

注意:Help 编译程序可以作为一个独立的产品从 Microsoft 公司买到。另外,Microsoft 可视 Basic 专门工具箱中包括有 Windows 3.0 Help 编译程序和创建系统设置程序(Setup Program)的工具。

产生 HLP 文件的主要步骤如下:

1. 使用文字处理器,产生一个包括帮助文字和图形的文档。通过应用适当的格式代码,可以在标题和连接字之间建立一个超文本链(hypertext link)来定义,通过加脚注,可以给标题赋予标识符(这样从用户程序就可参考这些标题),并为 Help 的 Search 命令指定使用的关键字。
2. 用 RTF 格式存储文档。
3. 产生一个联机帮助项目文件,这是一个标准文件加上一个 HPJ 扩展名。它将包括帮助文本在内的 RTF 文档的名字通知给 Help 编译程序(注意:可将几个文档组合在一个联机帮助系统中)。项目文件还提供正确编译联机帮助文件所须的其它一些信息。
4. 指定联机帮助项目文件的名字,运行 Help 编译程序。如果编译成功的话,Help 编译程序就产生 HLP 文件。例如,下面的命令运行编译程序并产生 PS.HLP 文件。

```
HC PS. HPJ
```

2.3.2 编码联机帮助系统

在运行时,通过调用 Windows 联机帮助工具软件,程序将显示 HLP 文件中包括的帮助信息。联机帮助工具软件可以通过一个简单的 Windows 应用程序接口函数 WinHelp 获得,WinHelp 的描述如图 2-2 所示。

Window 应用程序接口函数:WinHelp

目的:调用 Windows 联机帮助工具软件,既可以显示帮助标题,也可以在程序已结束显示求助时,撤消求助窗口。

说明: Declare Sub WinHelp Lib"USER"(ByVal hwnd As Integer,
ByVal lpHelpFile As String,ByVal wCommand As Integer,
ByVal dwData As Long)

参数	描述															
hWnd	主表格的 hWnd 属性															
lpHelpFile	包括有想要显示的帮助信息的 HLP 文件名的字符串,如果帮助文件既不在当前目录也不在 Windows 目录下,该字符串还必须包括完整的路径。															
wCommand	所需动作的代码,允许的值如下所示。 <table><tr><th>常数</th><th>值</th><th>意义</th></tr><tr><td>HELP_CONTEXT</td><td>1</td><td>显示用 dwData 参数表示的标题</td></tr><tr><td>HELP_HELPPONHELP</td><td>4</td><td>显示关于使用联机帮助的帮助</td></tr><tr><td>HELP_INDEX</td><td>3</td><td>显示帮助索引</td></tr><tr><td>HELP_QUIT</td><td>2</td><td>撤消帮助窗口</td></tr></table>	常数	值	意义	HELP_CONTEXT	1	显示用 dwData 参数表示的标题	HELP_HELPPONHELP	4	显示关于使用联机帮助的帮助	HELP_INDEX	3	显示帮助索引	HELP_QUIT	2	撤消帮助窗口
常数	值	意义														
HELP_CONTEXT	1	显示用 dwData 参数表示的标题														
HELP_HELPPONHELP	4	显示关于使用联机帮助的帮助														
HELP_INDEX	3	显示帮助索引														
HELP_QUIT	2	撤消帮助窗口														
dwData	如果 wCommand 属性设置成 HELP_CONTEXT,则为提供给用户要参考的特定帮助标题的标识符(当 HLP 文件产生时,每个帮助标题都赋予了标识符),如果 wCommand 没有设置成 HELP_CONTEXT,则为 0。															

图 2-2 WinHelp Windows 应用程序接口函数

本书中的所有程序都有如图 2-3 所示的统一的帮助菜单。

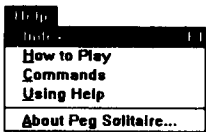


图 2-3 帮助菜单

当游戏者从帮助菜单中选择了 About 命令以外的一条命令,该命令的事件过程调出 WinHelp 来显示相应的帮助标题。

参考信息 处理某个特定菜单命令的事件过程是根据在菜单设计的 Window 对话框内的 CtlName 文本框中键入的标识符来命名的。

特别地,当用户选择了 Index 命令或按了 F1 键(这是赋给该菜单项的加速键), MenuIndex_Click 事件过程如下所示调用 WinHelp:

```
WinHelp hwnd HELP_FILE, HELP_INDEX, 0
```

第一参数 hwnd 提供了主表格的句柄,第二参数 HELP_FILE 提供了帮助文件的名字。

例如,在法国钉程序中,常量 `HELP_FILE` 定义如下:

```
Const HELP_FILE="PS. HLP"
```

提示 当运行程序时如果帮助文件包含在当前目录中或在 Windows 目录下,就不需要指定帮助文件的完整路径。

第三个参数的值, `HELP_INDEX`, 使得联机帮助工具软件显示索引, 产生帮助文件后, 用户应标识索引标题。

类似地, 当用户选择了 Using Help 命令时, `MenuUsingHelp_Click` 事件过程调用 `WinHelp` 来显示关于使用帮助系统的标准信息(由帮助设施自身提供)。

```
WinHelp hWnd, HELP_FILE, HELP_HELPPONHELP, 0
```

当用户选择了 How To Play 命令时, `MenuHowTo_Click` 事件过程调用 `WinHelp`:

```
WinHelp hWnd, HELP_FILE, HELP_CONTEXT, HelpHowToPlay
```

第三个参数传递的值 `HELP_CONTEXT` 使得联机帮助工具软件显示一个标题而不显示标准帮助标题之一(索引和有关使用帮助的信息都是标准的帮助标题), 在这种情况下, 第四个参数 `HelpHowToPlay` 提供用户想看的标题的标识符(用户建立帮助文件时赋值)。

当用户选择 Commands 菜单命令时, `MenuCommand_Click` 事件过程以相同的方式调用 `WinHelp`。

```
WinHelp hWnd, HELP_FILE, HELP_CONTEXT, HELP_COMMANDS
```

最后在程序退出之前, 必须再调用一次 `WinHelp` 来撤消窗口。如果程序没有做这一步, 求助窗口将显示在屏幕的左边, 直到用户人为撤消它。相应地, 当用户从 File 菜单中选择了 Exit 命令时, `MenuExit_Click` 事件过程将在结束之前立即调用 `WinHelp`:

```
Sub MenuExit_Click()  
    WinHelp hWnd, HELP_FILE, HELP_QUIT, 0  
End  
End Sub
```

值 `HELP_QUIT` 使帮助系统撤消窗口。实际上, 所有使用了联机帮助工具软件的应用程序将 `HELP_QUIT` 传递给 `WinHelp` 后, 窗口就撤消了。

如果用户通过控制菜单框(位于窗口的左上角)来结束程序, `Form_Unload` 过程也将调用 `WinHelp` 来撤消帮助窗口。

```
Sub Form_Unload( Cancel As Integer)  
    WinHelp hWnd, HELP_FILE, HELP_QUIT, 0  
End Sub
```

注意: 如果帮助窗口现在没有显示, 就没有必要调用 `WinHelp`, 但调用也无妨。

第3章 法国钉(PEG SOLITAIRE)

法国钉游戏是一个 18 世纪游戏的计算机版本。用户可以学到一些传统游戏的历史以及怎样玩这一新的版本,也可学习这个游戏是如何编码的。本章介绍的可视 Basic 编程技巧,有两方面非常重要,即用位图和图画盒产生游戏的图形,以及用可视 Basic 的拖放(drag-and-drop)特性来产生简单的动画。

3.1 关于法国钉游戏

法国钉,又名板钉(board Solitaire),是具有多种形式的著名游戏。它是用一帶有孔阵的板来玩,其中一些孔开始充满钉子(或弹子),游戏的目的是移去一些钉子直到剩下一个钉子或某一特定形式的钉子在板上,当某一钉子“跳”过另一钉子到对面的空孔中去时,被跳钉子就被移去。

这个游戏起源于法国,据称是巴士底狱中的一名囚犯发明的,后来在英国和世界各地非常流行。

传统的法国钉游戏板是八面形,有 37 个孔,如图 3-1 所示,通常用钉子玩。标准的英国板是圆的,有 33 个缺口,用与缺口相配的弹子来玩。

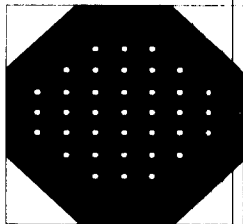


图 3-1 法国钉板

基本的游戏从除一个以外的孔都填满开始,游戏者选择初始空着的孔,然后就可以通过垂直或水平方向跳钉子试着移去除一颗事先设计好的钉子以外的所有钉子。

通过使用一个特定形式的钉阵(例如十字形),而不是整板开始来改变传统的游戏。也可以通过移去钉子来变换这个游戏,这样可选定某个形式的钉阵而不是一颗钉子留在板上的游戏模式。

3.2 玩法国钉游戏

本章介绍的法国钉游戏与该游戏的法国原形(37 个钉)最相近。它只用 17 颗钉子玩,而且,孔是以对角和竖直行而非水平排列的。

第1章解释了如何安装和运行本书中所介绍的软盘中所含的游戏。法国钉是包括在文件 PS.EXE 中的,当运行此文件时,游戏就开始了。任何时候从 Game 菜单上选择 New Game 命令就可以开始一个新的游戏(甚至在当前游戏完成之前)。程序显示一个如图 3-2 所示的窗口。

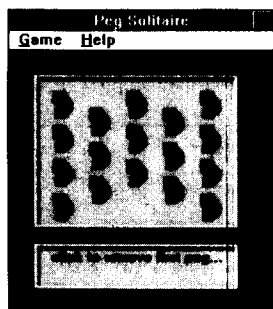


图 3-2 开始时的法国钉游戏

游戏开始时所有的孔中都填有钉子,游戏的目的就是移去除了游戏板顶部中间的那颗钉子以外的所有钉子。

开始玩时,首先要在某颗钉子上单击将它从板上移去,可以移去任何一颗钉子(每一步的不同选择都影响游戏赢的策略,改变游戏的玩法)。

移去第一颗钉子后,可以通过跳过钉子来移动。即用鼠标的单击和拖(Click-and-drag)的步骤将钉子从孔中移出来,让它跳过一颗邻近的钉子,然后停在一个想要移去的钉子另一边的空孔里。这样,计算机就移去被跳的钉子,如果这颗钉子停在不正确的位置上,程序将这颗钉子送回原孔中。

注意:可以用鼠标上的任何一个键来单击和拖。

当不能再移动时,计算机就结束游戏并显示留下的钉子数目。程序还将评价游戏者的水平,如果仅剩一颗钉子在顶部中间的位置,如图 3-3 所示,在游戏的消息区将出现“天才”的信息。仅有一颗钉子留在其它的位置被评价为“杰出”,其它的最终结果以稍差一些的祝贺词来评价。

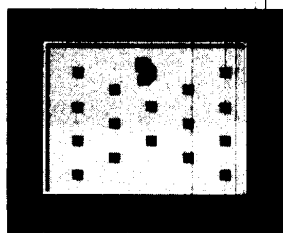


图 3-3 法国钉游戏获胜的结局位置

3.2.1 策略和提示

用户也许想要提高自己玩法国钉的策略。如果用户左脑发达(Strongly Left-brained),可

以通过事先计算两步或三步后选择一个最佳的移动方法。如果用户右脑发达(Strongly Right-brained),可以通过玩快点和观察钉阵的形式来玩得更好。通常应避免留孤立的钉子,一种方法是从一边开始动作到相对的一边,而不要在后边留下任何钉子。

3.3 编码法国钉

法国钉程序的源代码包含在下表所列的文件中。

文件	说明
GLOBALPS. BAS	包括有一个全程类型(RECT)
PS. FRM	定义主表格(FormMain)及其所包含的控制
PSABOUT. FRM	定义 ABOUT 表格(FormAbout)及其所包含的控制
PS. MAK	包括有法国钉的项目文件

所有这些文件随书附带的软盘中都有。第 1 章解释了怎样把文件安装到硬盘中,以及在需要检查和修改时怎样装载到可视 Basic 中。

图 3-4 至图 3-8 提供的是产生法国钉所需的所有信息。图 3-4 图解了在设计时出现的主表格(FormMain,在 PS. FRM 文件中定义),图 3-5 列出了在 PS. FRM 中定义的表格和控制属性设置,图 3-6 列出了通过可视 Basic 菜单设计的 Window 对话框赋给 PS. FRM 表格的所有菜单项,图 3-7 图解了在设计时出现的 ABOUT 表格,图 3-8 列出了赋给 PSABOUT. FRM 的属性,该表格没有菜单。

最后,图 3-9 列出了 PS. FRM 和 PSABOUT. FRM 中所有的源代码。

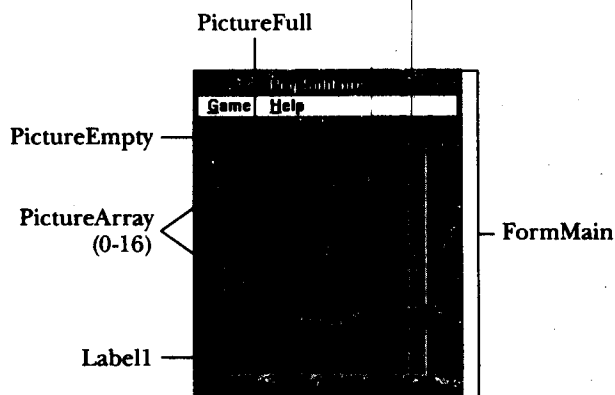


图 3-4 设计时的 PS. FRM

对象名	对象类型	特 性	设 置
FormMaain	Form	BackColor	&H00800000[dark blue]
		BorderStyle	1-Fixed Single
		Caption	PegSolitaire

图 3-5 PS. FRM 表格和控制属性

对象名	对象类型	特 性	设 置
Label1	Label	FormName	FormMain
		Grid Height	8[pixels]
		Grid Width	8[pixels]
		Height	4335[twips]
		Icon	(Icon) [PS. ICO]
		Left	990[twips]
		MaxButton	[False]
		Picture	(Bitmap) [PS. BMP]
		ScaleMode	3-Pixel
		Top	1080[twips]
		Width	3480[twips]
		Alignment	2-Center
		BackColor	&H0000FFFF&.[yellow]
		BorderStyle	0-None
		ForeColor	&H00000000&.[black]
PictureArray	Array of picture boxes	Height	34[pixels]
		Left	27[pixels]
		Top	187[pixels]
		Width	168[pixels]
		BackColor	&H0000FFFF&.[yellow]
		CtlName	PictureArray
		Height	33[pixels]
		Index	0-16
		Left	[varies]
		Top	[varies]
PictureEmpty	Picture box	Width	33[pixels]
		CtlName	PictureEmpty
		Height	33[Pixels]
		Left	0[pixels]
		Picture	(Icon) [PS. ICO]
		Top	0[pixels]
		Visible	False
		Width	33[pixels]
PictureFull	Picture box	CtlName	PictureFull
		DragIcon	(Icon) [PS3. ICO]
		Height	[pixels]
		Left	32[pixels]
		Picture	(Icon) [PS2. ICO]
		Top	0[pixels]
		Visible	False
		Width	33[pixels]

图 3-5 PS. FRM 表格和控制属性(续)

菜单选项/标题	控制名	其它特性
&Game	MenuGame	
....&New Game	MenuNew	
....-	MenuSep1	
....E&xit	MenuExit	
&Help	MenuHelp	
....&Index	MenuIndex	Accelerator=F1
....&How to Play	MenuHowTo	
....&Commands	MenuCommands	
....&Using Help	MenuUsingHelp	
....-	MenuSep2	
....&About Peg Solitaire...	MenuAbout	

图 3-6 PS. FRM 菜单设计

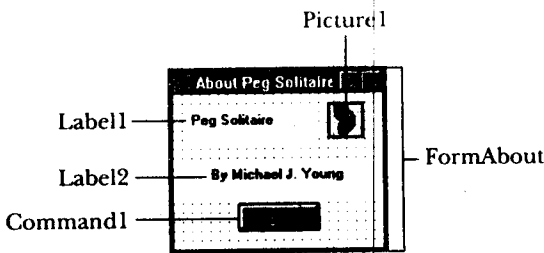


图 3-7 设计时的 PSABOUT. FRM

对象名	对象类型	特 性	设 置
Command1	Command button	Caption	OK
		Default	True
		Height	375[twips]
		Left	840[twips]
		Top	1440[twips]
		Width	1095[twips]
FormAbout	Form	BorderStyle	3-Fixed Double
		Caption	About Peg Solitaire
		FormName	FormAbout
		Grid Height	120[twips]
		Grid Width	120[twips]
		Height	2445[twips]
		Left	1140[twips]
		Width	2880[twips]
		MaxButton	False
		MinButton	False

图 3-8 PSABOUT. FRM 表格和控制属性

对象名	对象类型	特 性	设 置
Label1	Label	Top	1440[twips]
		Caption	Peg Solitaire
		Height	375[twips]
		Left	240[twips]
		Top	240[twips]
Label2	Label	Width	1575[twips]
		Alignment	2--Center
		Caption	By Michael J. Young
		Height	375[twips]
		Left	0[twips]
Picture1	Picture box	Top	960[twips]
		Width	2745[twips]
		AutoSize	True
		BorderStyle	0--None
		Height	480[twips]
		Left	2040[twips]
		Picture	(Icon)[PS.ICO]
		Top	120[twips]
		Width	480[twips]

图 3-8 PSABOUT.FRM 表格和控制属性(续)

GLOBALPS.BAS code

```

Type RECT
    Left As Integer
    Top As Integer
    Right As Integer
    Bottom As Integer
End Type

```

PS.FRM code

```

'Help Section:
Const HELP_FILE = "PS.HLP"
'WinHelp wCommand values:
Const HELP_CONTEXT = &H1
Const HELP_HELPONHELP = &H4
Const HELP_INDEX = &H3
Const HELP_QUIT = &H2
'WinHelp dwData values (help topics):
Const HELP_HOWTOPLAY = 10&
Const HELP_COMMANDS = 20&
'Windows help API:
Declare Sub WinHelp Lib "USER" (ByVal hWnd As Integer, ByVal
lpHelpFile As String, ByVal wCommand As Integer, ByVal dwData
As Long)

```

图 3-9 法国钉程序的源代码(注:长程序行折行后缩进一格)

图 3-9(续)

```

Const CLIENTBOTTOM = 242
Const CLIENTRIGHT = 223

Dim DragSource As Integer
Dim GameStarted As Integer
Dim I As Integer

Const WS_CAPTION = &HC00000
Const WS_THICKFRAME = &H40000
Declare Sub AdjustWindowRect Lib "USER" (lpRect As RECT, ByVal dwStyle&, ByVal bMenu%)

Const LOGPIXELSX = 88
Const LOGPIXELSY = 90
Declare Function GetDeviceCaps Lib "GDI" (ByVal hdc%, ByVal Index%) As Integer

Sub InitNewGame ()
    For I = 0 To 16
        PictureArray(I).Picture = PictureFull.Picture
    Next I
    Label1.Caption = "click to remove first peg..."
    GameStarted = 0
End Sub

Function JumpOK (Idx1 As Integer, Idx2 As Integer, Idx3 As Integer) As Integer
    If PictureArray(Idx1).Picture = PictureEmpty.Picture Then
        JumpOK = 0
    ElseIf PictureArray(Idx2).Picture = PictureEmpty.Picture Then
        JumpOK = 0
    ElseIf PictureArray(Idx3).Picture = PictureFull.Picture Then
        JumpOK = 0
    Else
        JumpOK = -1
    End If
End Function

Function MovePossible () As Integer
    If JumpOK(0, 5, 10) Or JumpOK(0, 3, 6) Then
        MovePossible = -1
    ElseIf JumpOK(1, 3, 5) Or JumpOK(1, 6, 11) Or JumpOK(1, 4, 7) Then
        MovePossible = -1
    ElseIf JumpOK(2, 4, 6) Or JumpOK(2, 7, 12) Then
        MovePossible = -1
    ElseIf JumpOK(3, 8, 13) Or JumpOK(3, 6, 9) Then
        MovePossible = -1
    ElseIf JumpOK(4, 6, 8) Or JumpOK(4, 9, 14) Then
        MovePossible = -1
    End If
End Function

```