

目 录

第 0 章 介绍	(1)
0.1 概述	(1)
0.2 应用对象	(1)
0.3 本书的内容	(1)
0.4 期待你的回音	(3)
第一章 内容匹配(Memory match)	(4)
1.1 概述	(4)
1.1.1 选择游戏	(4)
1.1.2 高分	(4)
1.1.3 About 对话框	(5)
1.2 程序操作	(5)
1.2.1 首先考虑的任务	(6)
1.2.2 全局模块	(8)
1.2.3 Board 窗体	(9)
1.2.4 HSENTRY 窗体	(20)
1.2.5 High Scores 窗体	(21)
1.2.6 About 窗体	(25)
第二章 内存资料信息(VBMem)	(28)
2.1 概述.....	(28)
2.2 程序操作.....	(28)
2.3 VBMem 的源代码	(30)
第三章 浏览字体(FontView)	(33)
3.1 概述.....	(33)
3.2 程序操作.....	(34)
3.2.1 各部分的综合:窗体和菜单设计	(34)
3.2.2 FontView 窗体代码分析	(37)
3.3 FontView 的源代码	(40)
第四章 附加的按钮条(ButtonBar Plus)	(47)
4.1 概述.....	(47)
4.2 程序操作.....	(47)
4.2.1 全局模块(BBARINT. BAS).....	(48)
4.2.2 ButtonBar 窗体(BUTTONBA. FRM)	(50)
4.2.3 条配置窗体(BARCONFL. FRM)	(64)
4.2.4 BrowseBox 窗体(BROWSEBOX. FRM)	(71)
4.2.5 IconDisplay 窗体(ICONDISP. FRM)	(77)
4.2.6 全局例程模块(BBAR. BAS)	(78)
4.2.7 License 窗体(LICENSE. FRM)	(88)

4.3 总结	(89)
第五章 打印剪贴板(PrintClip)	(91)
5.1 概述	(91)
5.2 程序操作	(92)
5.2.1 Load 事件	(92)
5.2.2 Resize 事件	(93)
5.2.3 结论	(96)
5.3 PrintClip 的源代码	(96)
第六章 任务调度(Job Scheduler)	(98)
6.1 概述	(98)
6.1.1 工具设计	(98)
6.1.2 任务调度的特征	(99)
6.2 程序运行	(99)
6.2.1 设计主屏幕	(99)
6.2.2 设计修改任务窗体	(110)
6.2.3 应用程序保护	(116)
6.2.4 可做的修改	(119)
第七章 时间设置(SetTime)	(120)
7.1 概述	(120)
7.2 程序操作	(121)
7.2.1 SETTIME.BAS 模块	(121)
7.2.2 OPTFORM.FRM 模块	(124)
7.2.3 SETTIME.FRM 模块	(125)
7.2.4 SETTIME.INI 文件	(128)
7.3 SetTime 的源代码	(129)
7.3.1 SETTIME.BAS 文件	(129)
7.3.2 OPTFORM.FRM 文件	(131)
7.3.3 SETTIME.FRM 文件	(134)
7.3.4 SETTIME.INI 文件	(140)
第八章 成组操作器(GroupWorker)	(141)
8.1 概述	(141)
8.2 程序操作	(141)
8.2.1 SetUp 窗体	(142)
8.2.2 处理群组	(143)
8.2.3 文件及帮助菜单	(144)
8.2.4 其它程序注解	(144)
8.2.5 最后的活动及一些思考	(145)
8.3 GroupWorker 的全部源代码	(145)
8.3.1 GLOBAL.BAS——全局声明	(145)
8.3.2 MAIN.FRM——Form2(启动窗体)	(151)

8.3.3	SETUP.FRM——Form1	(155)
8.3.4	ABOUT.FCM——Form5	(155)
8.3.5	HELPTXT.FRM——Form6	(155)
8.3.6	NAGBX.FRM——Form4	(156)
8.3.7	NAGBOX2.FRM——Form3	(157)
第九章	VB 时钟(VBClock)	(159)
9.1	概述	(159)
9.2	选项	(159)
9.3	程序操作	(160)
9.3.1	基本概念	(160)
9.3.2	保持 VBClock 处于顶部	(161)
9.3.3	使用活动标题栏悬浮 VBClock	(161)
9.3.4	存储显示选项	(162)
9.3.5	使用 VBClock.INI 文件工作	(162)
9.3.6	选择字体和颜色	(163)
9.3.7	在没有标题栏的情况下移动窗体	(164)
9.3.8	操纵弹出式菜单	(164)
9.3.9	节省存储	(164)
9.3.10	使用标签在窗体上显示	(165)
9.3.11	将图片框用作按钮栏	(165)
9.3.12	下一步工作的想法	(166)
9.3.13	运行良好的程序的通用规则	(166)
9.4	VBCLOCK 的源代码	(167)
9.4.1	全局列表文件	(167)
9.4.2	FRM_Clock.FRM	(170)
9.4.3	FRM_Displ.FRM	(175)
9.4.4	FRM_Opts.FRM	(176)
9.4.5	VBCLOCK.BAS	(181)
第十章	留言条(Blink Blank!)	(183)
10.1	概述	(183)
10.1.1	定制消息	(183)
10.1.2	多种消息	(184)
10.1.3	午餐消息	(184)
10.1.4	留下消息	(185)
10.1.5	检索消息	(185)
10.1.6	设置显示选项	(185)
10.2	程序操作	(185)
10.2.1	Windows API 函数	(186)
10.2.2	匈牙利命名法	(186)
10.2.3	组织屏幕布局	(187)

10.2.4	全局模块	(187)
10.2.5	窗体	(187)
10.2.6	MODULE1.BAS 例程	(193)
第十一章	州名程序(Name the States)	(195)
11.1	概述	(195)
11.2	程序操作	(195)
11.2.1	Windows API	(196)
11.2.2	GLOBAL.BAS 声明	(197)
11.2.3	过程操作	(197)
11.2.4	注解与建议	(201)
第十二章	视点(ViewPoint)Jr.	(203)
12.1	概述	(203)
12.2	程序操作	(204)
12.2.1	开始编程	(205)
12.2.2	装载窗体时初始化	(206)
12.2.3	在 ViewPoint Jr. 中使用的 API 调用	(209)
12.2.4	SendMessage 与目录列表框	(210)
12.2.5	消息由谁接收	(212)
12.2.6	最后的图形处理	(216)
12.2.7	其它优点	(217)
12.2.8	简单的帮助系统	(221)
第十三章	生命游戏(Life WorkShop)	(224)
13.1	概述	(224)
13.2	自动化细胞	(224)
13.3	使用生命游戏	(226)
13.3.1	传统的生命实验	(226)
13.3.2	六边形的生命实验	(227)
13.3.3	大型的实验	(228)
13.4	对生命游戏程序的理解	(231)
13.4.1	程序的结构	(232)
13.4.2	程序的启动事件	(232)
13.4.3	编写 Workbench 的程序	(232)
13.4.4	大型的生命	(234)
13.4.5	帮助及其它	(234)
13.5	生命游戏的源代码	(234)
13.6	结论	(269)
附录	附带软盘的安装	(270)

第0章 介绍

0.1 概述

欢迎您进入《Visual Basic 趣味程序设计》世界。我们之所以强调趣味,是因为这是整本书所采用的方法。在这里,你不会发现任何有关 Visual Basic 关键字的乏味的解释,相反,你会发现许多有趣、以前未发表的 Visual Basic 的编程工具和技巧。

《Visual Basic 趣味程序设计》的应用程序来源广泛,包括有关 CompuServe 的 Visual Basic 研讨会(MSBASIC),还有全国大量的电子广告板系统(BBSs)。每一章都阐述一个不同的 Visual Basic 应用程序的用法和程序结构,并且有编程人员的注释。

0.2 应用对象

《Visual Basic 趣味程序设计》是面向由初级到中级的 Visual Basic 程序员。尽管你不必对 Visual Basic 语言(VB)有透彻的了解就可以理解本书中的材料,但是你需要对 VB 的功能有一个好的理解。例如,你必须熟悉 Visual Basic 语言及其开发环境。

如果你刚开始接触 Visual Basic,但是熟悉其它 BASIC 语言(如 QBasic, GW-BASIC 等),那么你应当使自己熟悉 VB 开发环境和事件驱动的程序结构(可以通过一些 Visual Basic 文档中讨论的应用程序例子做到这一点)。

不是只有程序员才能从这套书/磁盘集中获得益处。本集包括了运行应用程序所需的一切,既使你并不拥有 Visual Basic 备份!按照本书后面的安装指示,你就可以在不必知道程序设计和 Visual Basic 的情况下,使用这些 Windows 游戏/工具!

0.3 本书的内容

本书包含十三章,作为磁盘上程序的详细文档,每一章都以解释如何应用该程序开始,然后进入不同层次的细节说明,阐明怎样用 Visual Basic 编写该程序。

我们说“不同层次的细节”,是因为有的程序很短,作者可以逐行解释程序结构,然而有的程序却太长,以至于 25~30 页都不能解释清楚。对这些长程序,作者只在他们程序源代码的最复杂的部分做了详细说明。尽管并非这些长程序的每一个地方都有详细解释,但是从书后面的磁盘中却可以获得以供你检查的详细注释的源代码。

现在我们浏览一下《Visual Basic 趣味程序设计》的每一章:

第一章 内容匹配(Memory Match),解释由 Lou Marino 编写的 Concentration-like 游戏的用法和结构,内容匹配的目的在于翻开许多牌并找到匹配的部分。世界国旗图标用于匹配,如果用户找到一匹配,被匹配的对仍然面朝上。如果一对不匹配,就把牌翻面。程序一直运行到所有的对都匹配。

第二章 内存资料信息(VBMem),解释一个简单的 Visual Basic 工具的结构,该工具实时显示内存的占用情况,显示随系统内存占用的改变而改变。该程序由 Charles Snider 编写,它显示下列系统信息:窗口操作模式(标准的或增强的)、空闲资源(以百分数表示)、空闲内存(以 KB 为单位)。

第三章 浏览字体(FontView),同样由 Charles Snider 编写。FontView 是一个有用的 Visual Basic 工具,你可以用它查看和打印安装在你的系统上的字体。它全兼容 Windows 3.x 和所有的 Windows 字体,包括 True Type 和 Adobe Type1。

第四章 附加的按钮条(ButtonBar Plus),详述一个吸引人的 Windows 程序激活工具。该程序由 Mark J. DiBiasio 编写,并用到浮动工具杆概念。该工具杆包含有 36 个按钮,这些按钮被安排在用户选择的矩阵内。用户可以设置一按钮以激活程序。每一个程序都可由用户所钟爱的图标表示。ButtonBar Plus 给用户提供了在窗口程序管理上的便利,包括程序按钮的连续位置。

第五章 打印剪贴板(PrintClip),由 Art Krumsee 编写。PrintClip 是本书中包含的最简单的程序之一(但却是功能强大的)。PrintClip 提供了打印剪贴板内容的简单直观的方法——只要连续两次按动 PrintClip 图标。

第六章 任务调度(Job Scheduler),解释一个任务调度程序,该程序可以使你安排一个任务(或应用程序)在一周内的具体某天和具体的时间内运行。Job Scheduler 由 Michael Kosten 编写,它提供给你在远离 PC 时自动运行程序的简便方法。

第七章 设置时间(SetTime),由 Stephen Pruitt 编写。SetTime 有两个目的,这主要是展示使用 Windows API 通讯和私有 .INI 文件。它也是一个调准计算机时钟的工具。

第八章 成组操作器(GroupWorker),Dr. Jerry Miller 开发,它是一个可使 Windows 用户创建应用程序组的有趣的工具,其中每一个程序只需按一次鼠标就可以激活。

第九章 VB 时钟(VBClock),Sarah Holland 开发,它可显示系统日期、时间、星期几、空闲系统资源和空闲内存。VBClock 还有一简单的闹钟和消息特性。

第十章 留言条(Blink Blank!),由 Max Burgsther 编写,Blink blank! 是保存屏幕的一种新方法,它很有用,Blink Blank! 不以无意义的符号填充你的屏幕,实际上它保持你的 PC 在你离开的时候仍在工作。如果你激活 Blink Blank!,你就给予了你的 PC 在你离开的时候显示信息的权利。如果有人希望在你离开的时候给你留个信息,他只需按你的 PC 的空格键并输入他想留下的话。Blink Blank! 可以保留 5 条信息(不管多长),并在你回到机器身边时显示给你。你甚至还可以打印这些消息,它们自动地被加上时间。

第十一章 州的名字(Name the States),由 Dan Lewczyk 编写。Name the States 是一个教你 50 个州的名字、地点和州府的 Visual Basic 程序。

第十二章 视点(ViewPoint)Jr. 由 Barry Seymour 编写,此程序可让你查看大量的图形文件,包括几种 Paintbrush 不支持的格式。

第十三章 生命游戏(Life Workshop),来自 Ivory Tower Software,允许你用二维的细胞状自动装置(two-Dimensional cellular automata)实验。你可以修改规则,然后观察有关的大量的人工生命(artificial life),或者你可以在工作平台中给指定的单元赋与生命,然后观察不同的规则集所涉及的不同的工作方式。

0.4 期待你的回音

Que Development Team 花费了大量的时间创建 Visual Basic 趣味程序设计。如果你愿意配本书或其它 Que 编程教材提出意见,请写信给:

Publishing Manager
Que Programming Books
11711 North College Avenue, Suite 141
Carmel, IN 46032

如果你是 CompuServe 客户,欢迎你们随时通过下列 CIS 地址:76226,632 向我们提出你的意见。

如果你写了一个有趣的 Visual Basic 程序,欢迎写信寄给我们,如果合适,我们会在本书的下一版中把它包括进去。请把应用程序寄到我们的通信地址或是我们的 CompuServe 地址。

第一章 内容匹配(Memory match)

1.1 概述

Memory Match 的目的是翻开成对的牌并找到匹配的对。包含全世界各类旗帜的图标被用于匹配。如果用户找到一对匹配,被匹配的对仍保留面朝上。如果不匹配,程序将翻转牌。程序运行至所有的对都已匹配。

1.1.1 选择游戏

从 Games 菜单中,选择 8 个游戏格中的任何一个。每一次程序都将自动发牌,创建不同的游戏。图 1.1 显示了进行中的 Memory Match 游戏。

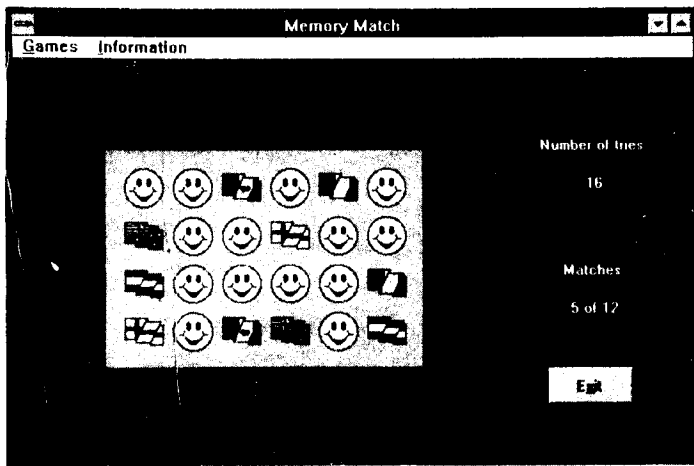


图 1-1 进行中的游戏

1.1.2 高分

象大多数游戏一样,Memory Match 提供了存储功能,即查看和输入高分的工具。实际上,“高分”意味着游戏中的最少匹配次数。这就象打高尔夫球,击球最少次数的人获得最高分。

查看高分

从 Information 菜单,选择 High Score。一个包含获得高分的列表窗口将弹出,同时还包含高分和日期。图 1-2 显示了高分窗口。

记录高分

游戏玩至新的最高分,屏幕上将出现一个数据输入窗口,操作者可在此输入其姓名,程序将自动记下分数和日期。图 1-3 显示了 New High Score 窗口。

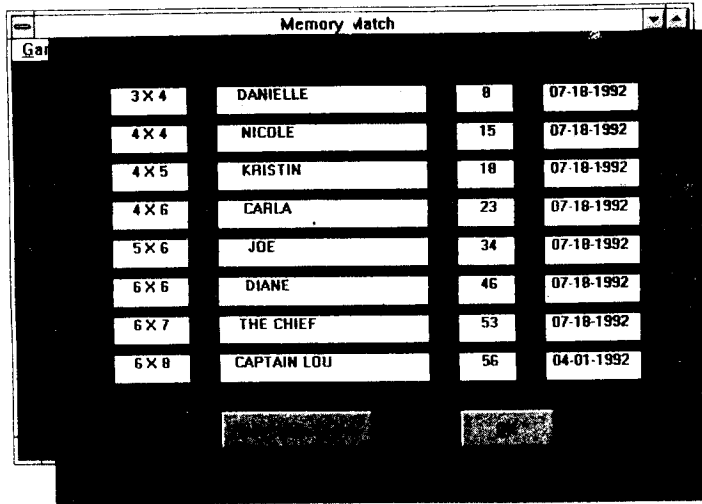


图 1-2 高分窗口

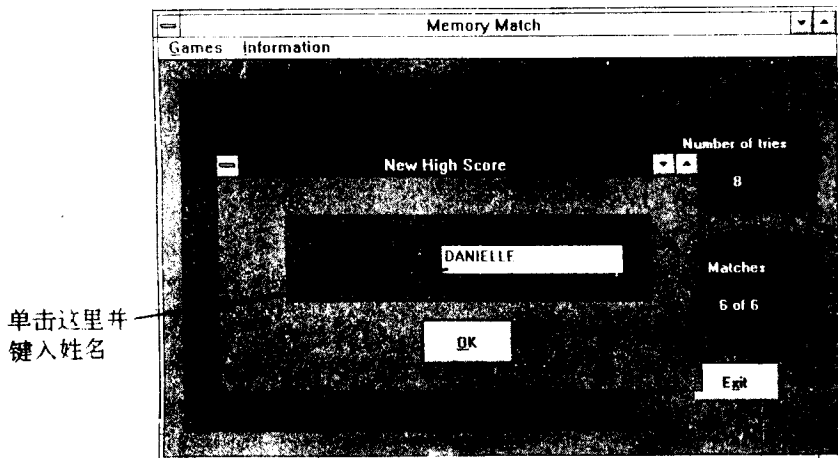


图 1-3 输入新的最高分

1.1.3 About 对话框

从 Information 菜单中选择 About 命令, 屏幕上将出现 About 对话框。图 1-4 显示了 About 对话框。

1.2 程序操作

把一个大的编程项目分成小部分或子程序是很有帮助的。通过把程序的每一个主要操作看作一个单独的实体, 你可以集中精力进行细节处理。让我们看一下构成 Memory Match 问题的各个部分。

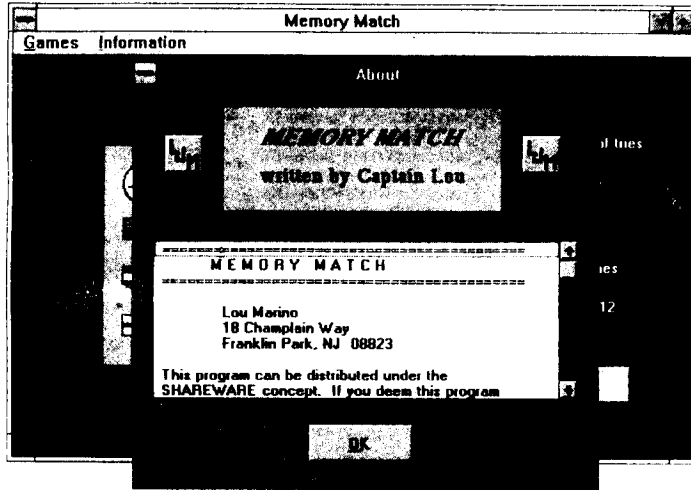


图 1-4 About 对话框

1.2.1 首先考虑的任务

首先,你应决定游戏的类型和规模。勾勒出游戏的大致规则之后,你理应预测所需的窗体数。如果对用户来说有很多选择项,你的游戏则需要菜单系统。

接着,考虑 Visual Basic 提供的节省时间和代码性能,这些性能会对你的应用程序有用。例如,在该程序的两个重要部分用到了控件数组(Control arrays)。

决定游戏的规模

每个游戏都包含一个由各类旗帜填充的格子。该骗局是所有的旗帜背放,就象一幅牌面朝下放着。选手必须翻开牌并匹配这些旗帜。

重构游戏,以便选手可以选择从易到难的游戏。下列是 Memory Match 包含的 8 种游戏格子:

游戏类型	匹配数
3×4	6
4×4	8
4×5	10
4×6	12
5×6	15
6×6	18
6×7	21
6×8	24

游戏需要多少窗体

处理游戏需要有三个窗体。你知道至少需要一个窗体去容纳实际的游戏,但是除了游戏窗体外,还需要其它什么项呢?大多数的游戏都保留达到高分的选手的记录;Memory Match 也不例外。你可以记录 8 个游戏中每一个的记录。游戏需要第二个窗体以便在选手达到高分时输入他们的名字,为了显示高分表还需要第三个窗体。

上面提到的三个窗体将处理游戏的需要。另外还包括一个附加的窗体——About 窗体,这也是所有的 Windows 程序的一部分。这最后的一个窗体包含有关程序的附加信息,如版本、作者等。

该游戏需要菜单系统吗

从 8 个游戏中选择一个,我们有足够的理由考虑用一个下拉式(pull-down)菜单来实现。另外加上显示高分和查看 About 窗体等,如果不用一菜单系统,屏幕将变成乱七八糟的命令按钮。

必须用控件数组吗

好的程序员总是想着减少内存占用量和缩小程序大小的方法。在任何时候,Memory Match 都需要提供给选手 8 个不同的游戏。为每一游戏创建一窗体并且只显示选手选中的游戏窗体是很容易的。而这样的方式构造程序却不是有效地利用资源的方法。

让我们看一下用单独的窗体分别创建每一游戏所需的图片控件(Picture Controls)数目:

游戏类型	图形控件数
3×4	12
4×4	16
4×5	20
4×6	24
5×6	30
6×6	36
6×7	42
6×8	48
Total:	228

向一应用程序中增加 228 个画形控件无疑会大大增加程序的大小。利用内存资源最有效的方法是只把对当时所选游戏有必要的控件变为可用的。

象窗体一样,控件可以被装入(LOADED)和卸下(UNLOADED)。然而,控制却有一个很大的优点。它们可以在运行的时候动态地创建和删除。这可以用一个控件数组实现。

控件数组很象 Visual Basic 中应用的其它数组。如字符串数组或整数数组等。主要的区别是这样的数组包含的是控件而不是变量类型。控件数组必须是同样的控件类型,有相同的

CtlName 和唯一的索引号(Index number)。

图 1-5 是一控件数组的例子。该数组包含有四个命令按钮。注意命令按钮间唯一的区别是它们不同的索引。

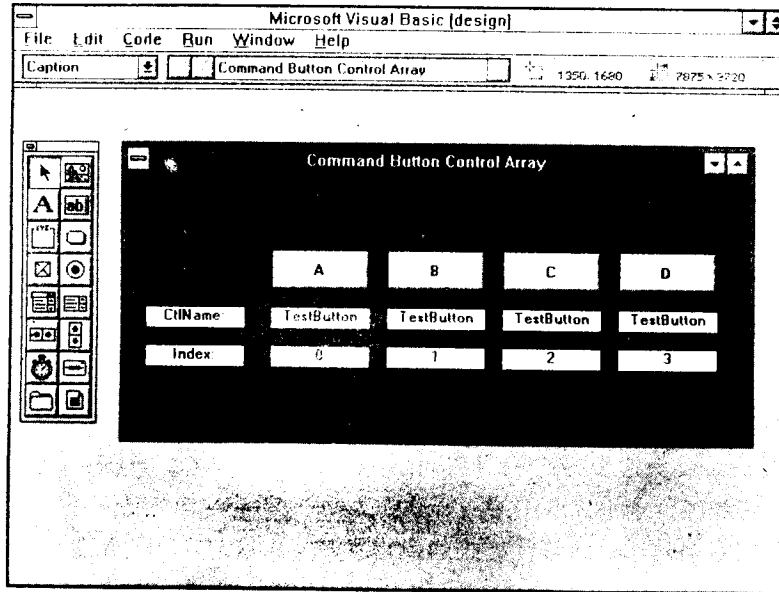


图 1-5 命令按钮控件数组

有关控件数组的附加信息,请查阅 Visual Basic By Example(由 Que 出版)。

1.2.2 全局模块

启动 Visual Basic 并从 File 菜单中选择 New Project。从 Project 窗口醒目显示 GLOBAL.BAS。从 File 菜单选择 Save File As...。当信息框(information box)出现时,键入 MEMORY.GBL,然后单击 OK 命令按钮接受该输入。至此全局模块命名为 MEMORY.GBL。

启动 Visual Basic 之前,设置 startup 目录到你的工作目录。你可以通过醒目 VB 图标,从 File 菜单选择 Properties,并在 Working Directory 域内输入你的工作目录的名字(如 C:\VB\MMATCH)做到这一点。在开发过程中,Visual Basic 搜寻 Startup 目录查找文件。对 Memory Match,你可以用几个图标和正文文件。你还需把这些文件移到缺省目录下。图1-6 显示了 MEMORY.GBL 的内容。

下面是对 MEMORY.GBL 中的每一变量如何在程序中应用的解释:

- CARD\$ —— 含有 48 个元素的字符串数组。该数组存有旗帜图标的名字。因为有 24 个不同的标记,故需要 48 个元素存放匹配的对。
- FLAG\$ —— 包含有 24 个旗帜图标的名字。这些名字被随机地放到 CARD\$ 数组中,这样确保每次创建不同的游戏。
- CARDBACK\$ —— 包含有前一张牌面朝下时用到的图标的名字,就象下一幅牌的背面。
- NOMFLAGS —— 选中的游戏所需的卡片的数目。这也是动态创建控件数组时所用

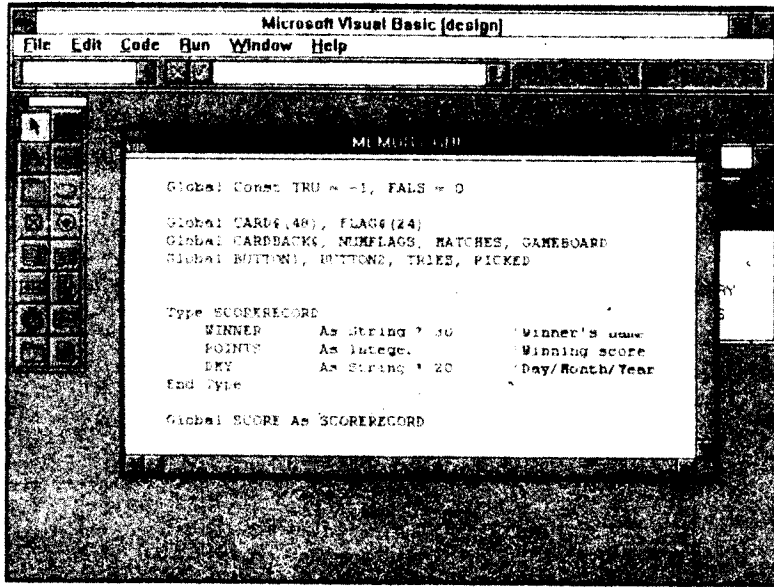


图 1-6 MEMORY.GBL 模块的内容

到的数字(如,对于 4×6 游戏,NOMFLAGS 将为 24)。

- MATCHES——当前游戏已成功配对的数目。
- GAMEBOARD——用于检查选手是否平或超过当前选中游戏的最高分。
- BUTTON1——一对中的第一张卡片的索引号。
- BUTTON2——一对中的第二张卡片的索引号。
- TRIES——标记匹配中已经试过的次数。
- PICKED——报告当前选择是选项一或选项二。
- SCORERECORD——一个 TYPE 结构,用于向高分文件中读或写记录:

Type SCORERECORD

WINNER	As String * 30	' Winner's name
POINTS	As Integer	' Winning score
DMY	As String * 20	' Day/Month/Year

End Type

Global SCORE As SCORERECORD

1.2.3 Board 窗体

Board 是该游戏中用于工作区(Playing area)的窗体。让我们一步一步地创建该窗体。

- (1)拉长该窗体以便有足够的屏幕区域可用。
- (2)放置一图片控件(PICTURE1)到窗体中并定尺寸为5655×4455。(创建该应用程序的控件尺寸是建议的,当然你可以用自己喜欢的尺寸)。
- (3)从工具框中选择另一图片控件(PPLAYFIELD)并且在 PICTURE1的顶部画一975×

975的正方形。不要双击选择这一控件。通过在一控件顶部画另一控件,建立了一父母——孩子关系(这一关系的必要性后面有解释)。双击选择一控件创建和其他控件无关系的独立对象。

(4)象前面的步骤一样,选择一图片控件(MEMORY)并在 PLAYFIELD 的中心画一 495×495的正方形。现在 PLAYFIELD 是 PICTURE1的孩子,MEMORY 的父母。

(5)在 PICTURE1的右边,放置另外两个画出控件(PICTURE2和 PICTURE3),每个的尺寸都为2055×1215。

(6)在 PICTURE2中画两个标签控件(LABEL2和 NUMTRIES)。

(7)同样,在 PICTURE3中画两个标签控件(LABEL2和 NUMMATCH)。

(8)在窗体中放置一命令按钮。

图1-7显示了开发过程中 Board 窗体的情况。

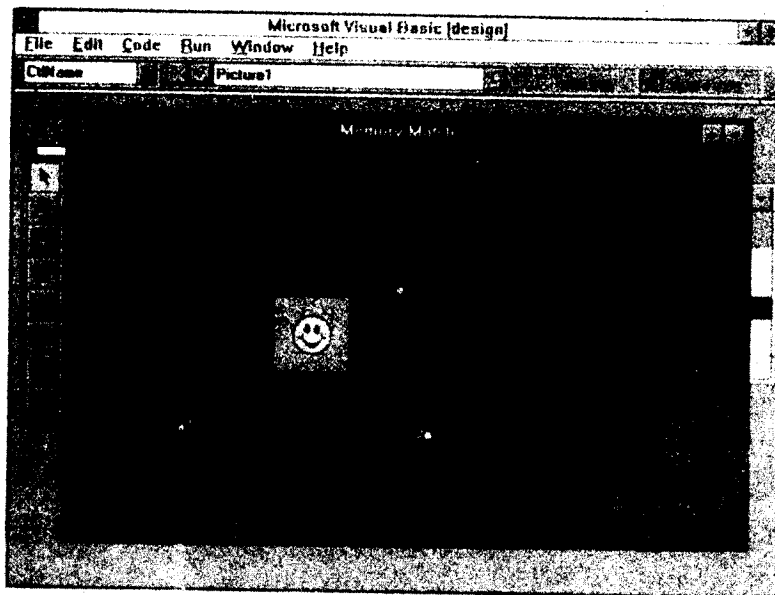


图1-7 开发 Board 窗体

表1-1是创建 Borard 窗体所需的控件列表。

表1-1 创建 Board 窗体所需的控件

窗体/控件	属性	设置
Form:	BackColor	&H00808080&
	Caption	Memory Match
	FormName	BOARD
Picture:	BackColor	&H00800000&
	CtlName	Picture1Pictue:
	BackColor	&H00C0C0C0&
	Ct1Name	PLAYFIELD

(续表)

窗体/控件	属性	设置
Picture:	ctlName	MEMORY
	Index	1
	Picture	FACE03.ICO
Picture:	BackColor	&H00000080&
	CtlName	PICTURE2Label
	Alignment	2-Center
	BackColor	&H00000080&
	Caption	Number of tries
	CtlName	LABEL1
	ForeColor	&00FFFFFF&
Label:	Alignment	2-Center
	BackColor	&H00000080&
	Caption	Number of tries
	CtlName	NUMRIES
	ForeColor	&00FFFFFF&
Picture:	BackColor	&H00000080&
	CtlName	PICTURE3
Label:	Alignment	2-Center
	BackColor	&H00000080&
	Caption	Matches
	CtlName	LABEL2
	ForeColor	&00FFFFFF&
	Label:	Alignment
Label:	BackColor	&H00000080&
	Caption	Numbwr of tries
	CtlName	NUMMATCH
	ForeColor	&00FFFFFF&
	Command:	Caption
	CtlName	COMMAND1

图1-8显示了控件之后的 Board 窗体。

创建菜单条

利用菜单设计窗口(Menu Design Window)为 Board 窗体创建一下拉式菜单系统。从 Window 菜单选择 Menu Design Window 开始对菜单工作。表1-2包含了构造下拉式菜单系

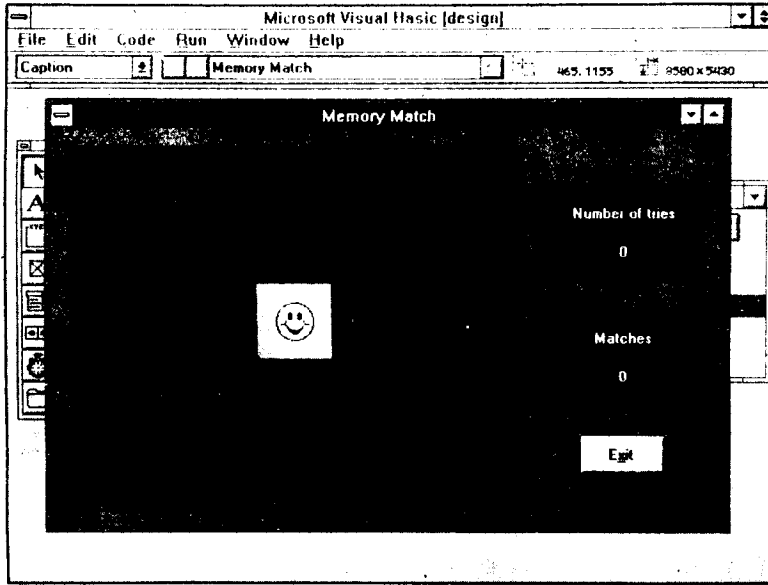


图1-8 放置控件后的 Borad 窗体

统所需的元素。

表1-2 创建下拉式菜单系统所需的元素

标题	控件名	缩进	索引
&Games	GAMES	0	
2×2(2 matches)	JOKE	1	
3×4(6 matches)	GAME	1	1
4×4(8 matches)	GAME	1	2
4×5(10 matches)	GAME	1	3
4×6(12 matches)	GAME	1	4
5×6(15 matches)	GAME	1	5
6×6(18 matches)	GAME	1	6
6×7(21 matches)	GAME	1	7
6×8(24 matches)	GAME	1	8
—	dash1	1	
&Quit	QUIT	1	
&INFORMATION	INFORMATION	0	
&High scores	HISCORE	1	
—	dash2	1	
&About	PROGINO	1	

