

计算机编程技巧

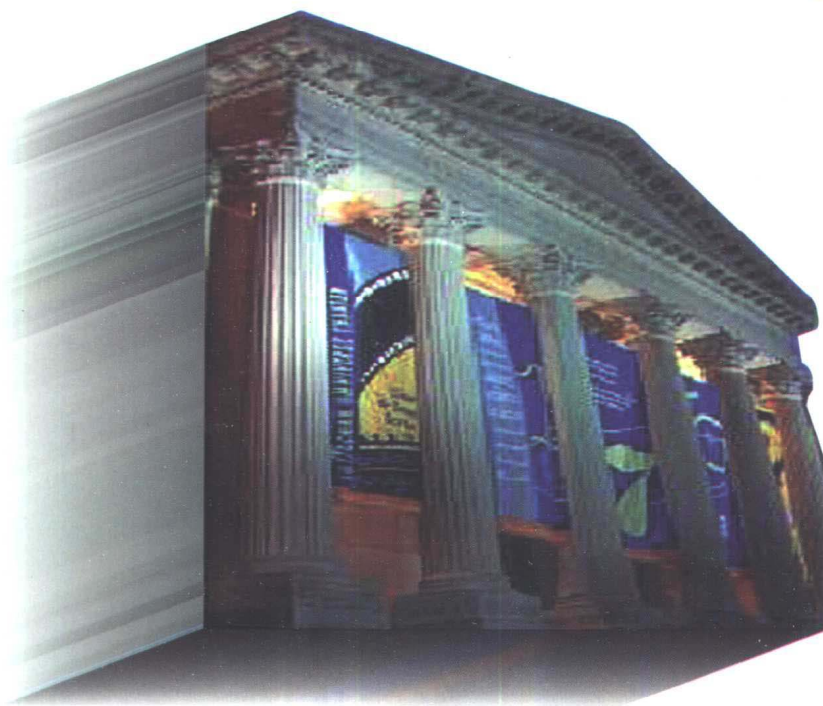
ABC

系列丛书

Delphi

编程技巧

多媒体与系统篇



清宏计算机工作室
编著



机械工业出版社
China Machine Press

计算机编程技巧 ABC 系列丛书

Delphi 编程技巧

(多媒体与系统篇)

清宏计算机工作室 编著



机械工业出版社

本书全面地介绍了 Delphi 在多媒体和系统方面的各种各样的使用技巧, 包括多媒体、注册表、打印、系统控制、进程和线程、图像处理、窗体、控件使用、剪贴板、图标、鼠标和键盘、文件和磁盘、字符串处理、文字和字体等内容。

本书介绍的技巧全面而又不失代表性, 书中详细介绍了各种技巧的关键所在、实现过程和专家点评。它既可以作为广大用户学习 Delphi 的指导书, 又可以作为程序员编程时的参考手册。

本书读者对象为计算机程序开发人员、大专院校计算机专业师生。

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 边 萌 封面设计: 姚 毅

责任印制: 路 琳

北京市密云县印刷厂印刷·新华书店北京发行所发行

2001 年 4 月第 1 版第 2 次印刷

787mm×1092mm 1/16·22.25 印张·548 千字

5 001—7 500 册

定价: 39.00 元 (1CD, 含配套书)

ISBN7-900043-47-0/TP·43

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、68326677-2527

前 言

随着计算机技术的飞速发展，计算机的开发迅速得到普及推广。当前有一大批计算机开发人员，虽然已经掌握了某种开发工具的基础知识、甚至达到了相当高的水平，但仍有很多盲区，或者急需更多的编程经验和技巧。为此，我们决定组织编写这套“计算机编程技巧 ABC 系列丛书”，以满足广大读者的需求，帮助读者快速成为计算机编程得行家里手。

这套丛书针对当前最流行的几种开发语言，介绍开发过程中的编程经验和技巧，并解决开发过程中的疑点和难点，主要突出“技巧”二字。针对某些开发语言，根据其内容相关性和独立性，我们编写了两个系列的书籍，即“网络与数据库篇”和“多媒体与系统篇”。这样可以使读者有更大的选择余地，并且在内容编排、读者阅读上都更为合理。

每本书按章节编写，各章由既相对独立，又相互关联的技巧主题组成，并且在顺序编排上考虑了内容的前后连续性。

针对每一个技巧主题，我们将其分为“ABC”三个部分进行讲解。

A——关键所在（或原理方法） 以简单明了的语言指出实现技巧的关键所在或原理方法。

B——实现与应用 以一个简单、完整实例向读者介绍技巧的实现方法或步骤。

C——专家点评 对技巧加以适当说明，包括补充解释，强调注意事项等。比如简要介绍实现该技巧的其他方法，或者介绍关键技术的扩展应用等。

针对每一个技巧主题，力求在内容讲解清楚的前提下，篇幅短小，使读者在有限篇幅内学到更多的技巧，整套书充分体现强调短小精悍、高效率的特点。

读者在阅读本套丛书之前，应该对相应开发语言有了一定了解，但并不要求读者开发水平一定很高，在章节编排和内容讲解过程中，我们都尽量照顾了水平还比较低的读者，因此初学者还是可以阅读这套丛书的。

希望这套丛书能使你很快成为软件开发的高手。

恳请广大读者对书中不足之处提出宝贵意见。

清宏计算机工作室

编者的话

业界有这样一种说法：真正的程序员用 C；聪明的程序员用 Delphi。笔者编的第一个程序用的是 Turbo Pascal 5.0（Delphi 前 Borland 开发的同种语言的编程平台）；后来又经历了从 TC20 到使用 BC31 又到 VC5，在因项目开发需要而转向 Delphi 后才真正对这句话有所感触。Delphi 与 VC 相比，虽速度上没有优势，但语法约束性强，出错容易查找，可视化性更高。Delphi 同 VB、PB 等语言同为真正的可视化语言，基于窗口和对象的技术，与 Windows 平台紧密结合，对数据库及网络开发有强大的支持，编译快速高效；很多建模工具如 SA、Rose 均对其有支持，能够比较平滑的从分析和设计过渡到编码实现，从而在众多开发工具中独树一帜。

本书通过大量丰富的实例介绍了 Delphi 多媒体和系统的应用程序，它的特点是有大量的实例，这些实例都是笔者根据体会精心挑选出来的，比较全面和具有代表性。在编程中常常会遇到这些实例，比较适合于初级和中级程序员的需要。读者只要根据步骤一步一步地执行下去，就肯定可以得到正确的运行结果。读者还可以举一反三，全面地了解此书中的各个技巧。

在读者阅读本书之前要求掌握 Object Pascal 语法，有一定编程基础，对 Windows 可视化编程有所了解，对 Delphi 有一定的了解，有一定的 Delphi 编程知识。

本书的配套光盘中含有此书中的全部源程序，这些程序都是经过严格的调试和测试的。

本书的配套光盘

光盘的源代码中，有些 .bmp 文件没有提供，读者可采用尺寸相当的 .bmp 文件代替。只要按书中文档所给路径设置准确就可以运行程序。

与作者的联系方式

honghongzhong@263.net

此地址不会改变，当“263”不再提供邮件服务时除外。

由于编写时间较紧和水平有限，书中难免出错，希望读者谅解并批评指正。

目 录

前言

编者的话

第 1 章 多媒体	1
1.1 显示 AVI 第一帧	1
1.2 用 API 实现播放一个 WAV 文件	2
1.3 Delphi 编程实现 3DS 动画的播放	5
1.4 全屏显示 AVI	8
1.5 关闭和打开声道	9
1.6 在多媒体文件中批量抓取图像	12
1.7 循环播放音乐	19
1.8 实现文本显示与声音播放同步	21
1.9 利用 Delphi 编写屏幕保护程序	26
第 2 章 注册表	31
2.1 获得系统的默认目录	31
2.2 建立文件类型相关联	33
2.3 设定光驱的自动运行属性	36
2.4 如何将你的程序放在 Windows 启动中	38
2.5 获得时区信息	39
2.6 壁纸更换	41
2.7 添加字体	43
第 3 章 打印	45
3.1 文本和图像的打印	45
3.2 获得打印驱动、端口名	48
3.3 获得打印页边距	50
3.4 利用 API 直接打印	53
3.5 切换打印机的彩色模式	62
3.6 使用打印机内置字体打印	64
3.7 同时打印到多台打印机	65
3.8 在打印中间改变打印机的设置	72
3.9 改变默认的打印机	77
3.10 检测是否存在打印机	78
第 4 章 Windows 系统控制	80
4.1 获取环境变量	80
4.2 任务条上隐藏应用程序窗口	81

4.3	禁止用 Alt+Tab 键切换任务	82
4.4	禁止拖动时显示窗口内容	83
4.5	获取除了任务条之外桌面的大小	84
4.6	使用资源管理器打开相应的目录	86
4.7	启动控制面板的各项控制功能	87
4.8	链接的视觉效果	90
4.9	设置输入法.....	91
4.10	响应即插即用变化	93
4.11	隐藏 Windows 的任务栏	96
4.12	用小执行程序实现 Win95 桌面的隐藏	97
4.13	得到计算机名	103
4.14	如何获得 Win/Dos 版本号	104
4.15	获取内存信息	105
4.16	修改系统颜色	107
4.17	辨别分辨率.....	108
4.18	获取全球标准标准时间	109
4.19	设置系统时间	111
4.20	Delphi 中实现系统状态栏图标	112
第 5 章	进程和线程	116
5.1	列出系统进程.....	116
5.2	关闭外部已开启的应用程序	117
5.3	防止程序运行多个例程	118
5.4	调用外部程序并知道它的运行状态	121
5.5	提高进程优先级	125
5.6	在线程中处理异常	126
5.7	编程中使用线程	129
第 6 章	图像处理	132
6.1	动态帮助功能.....	132
6.2	创建调色板.....	134
6.3	创建位图.....	136
6.4	动态多边形.....	140
6.5	画过滤色.....	143
6.6	基于图像漫游.....	149
6.7	屏幕拷贝.....	152
6.8	使用路径和几何笔	158
6.9	用鼠标选取范围	160
6.10	透明图像画刷	163
6.11	为组件加上边框	168

6.12	移动的字幕和图像	170
6.13	阴影或投影效果	173
6.14	用 API 处理位图	175
6.15	把 TColor 值转换成 RGB 值	177
6.16	RGB 颜色与 CMYK 颜色的转换	178
6.17	显示 Jpeg 图像文件	182
6.18	图像局部放大原理	184
6.19	图像渐隐技术	188
6.20	图像的镜像显示	191
6.21	实现窗体的渐变背景	197
6.22	实现图像的下拉效果	199
6.23	实现图像的百叶窗效果	200
第 7 章	窗体设计	202
7.1	显示启动画面	202
7.2	通过客户区移动窗体	204
7.3	窗体生成时的事件顺序	206
7.4	制作可移动的分割窗体	207
7.5	限制窗体的大小	209
7.6	窗体融合技术	212
7.7	给 MDI 主窗体加背景	214
7.8	在窗体的边框上作图	216
7.9	创建任意形状的窗体	219
7.10	查阅可视窗体的标题	223
7.11	枚举子窗口	225
7.12	确定当前鼠标所在的窗口	229
7.13	使其他程序成为当前窗口	231
7.14	闪烁标题栏	233
第 8 章	控件的使用	236
8.1	动态生成控件	236
8.2	防止编辑框的输入滚动	238
8.3	列表框项目的拖放操作	240
8.4	在窗体中拖动控件	244
8.5	在状态栏中插入可视控件	245
8.6	控件提示信息的自动显示	249
8.7	制作控件	250
8.8	限制窗口中某类控件的数目	259
8.9	列表框的水平滚动条	261
8.10	在列表框中加入图片	263

第 9 章 剪贴板	268
9.1 监视剪贴板内容	268
9.2 备份剪贴板格式列表	270
第 10 章 图标	275
10.1 获取同文件相关联的图标	275
10.2 获取 Windows 图标	276
10.3 获取动态图标	280
10.4 从文件中提取图标	283
10.5 把图标转换成位图	286
10.6 把位图转换成图标	287
第 11 章 键盘和鼠标	291
11.1 将鼠标限定在一定范围	291
11.2 判断 PrintScreen 键是否按下	292
11.3 若隐若现的鼠标	293
11.4 无标题栏窗口程序设计中鼠标拖动技巧	295
第 12 章 文件与磁盘	298
12.1 获取和修改文件的时间	298
12.2 把文件删除到回收站中	305
12.3 创建临时文件	308
12.4 获取同文件扩展名相关联的应用程序	310
12.5 复制文件	313
12.6 检测驱动器的类型	316
12.7 检测驱动器的容量	317
12.8 获取和设置驱动器卷标	320
12.9 打开和关闭光驱	323
第 13 章 字符串处理	325
13.1 中英文字数的统计	325
13.2 数值与字符串的相互转化	327
13.3 将小写金额转换成大写金额	330
第 14 章 文字和字体	334
14.1 改变文字的背景颜色	334
14.2 获得可用字体大小	335
14.3 枚举当前字体	338
14.4 判断是否有某种字体	341
14.5 透明的文字背景色	342
14.6 旋转字体	344

第 1 章 多 媒 体

1.1 显示 AVI 第一帧



关键所在

通过调用一组 TApplication 对象和 TMultiMedia 组件的一些方法，再恰当的设置 TMediaPlayer 组件的一些属性。即不难实现显示 AVI 第一帧。这组方法如下。

1. TApplication 对象的 ProcessMessages 方法

中断一个应用程序的执行，使 Windows 可以对事件做出响应。

2. TMediaPlayer 组件

(1) Open 方法 此方法用于打开一个多媒体设备。在调用该方法之前，必须确保在 MediaPlayer 组件的 DeviceType(设备类型)属性中设定正确的多媒体设备类型。

(2) Step 方法 在当前装载的媒体中向前移动一帧。

(3) Previous 方法 把当前的位置设置为前一个音轨的开始处。

(4) DeviceType 此属性用来指定要打开设备的类型。

在这里面还有许多选项，主要指明 MCI 的各类设备，如：

- DtAviVideo 动画设备
- Dtdat 数字音频磁带机
- Dtother 其他未定义的 mci 设备
- DtwaveAudio 数字化波形文件
- Dtscanner 图像扫描仪
- DtsepQuenceMiDi 音讯器
- Dtvcr 盒式录像机
- DToverlay 在小窗口中播放模拟电视
- DtmmmorvieMMM 动画文件

(5) FileName 属性 文件名,用于存储或打开的文件名。



实现与应用

实现显示 AVI 第一帧的步骤如下。

- (1) 首先，创建一个应用程序。
- (2) 在应用程序的窗体上加上一个 TButton 组件和一个 TMediaPlayer 组件。
- (3) 设置此 TButton 组件的 Caption 属性为“显示 AVI 第一帧”。
- (4) 设置此 TMediaPlayer 组件的 DeviceType 属性设置为 DtAviVideo。把 FileName 属

性设置为 D:\private\speedis.avi。speedis.avi 是 Delphi 自带的一个 AVI 文件。用户可以把她拷贝到 D:\private 目录下。

(5) 编写此 TButton 组件的 OnClick 事件如下:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Application.ProcessMessages;
  MediaPlayer1.Open;
  Application.ProcessMessages;
  MediaPlayer1.Step;
  Application.ProcessMessages;
  MediaPlayer1.Previous;
end;
```

在此事件的句柄中, 首先调用 TApplication 的 ProcessMessages 方法中断应用程序的执行, 使应用程序打开一个 AVI 设备; 然后又调用 TApplication 的 ProcessMessages 方法中断应用程序的执行, 使应用程序在 speedis.avi 中向前移动一帧; 最后调用 TApplication 的 ProcessMessages 方法中断应用程序的执行, 使当前位置移到 speedis.avi 的前一个音轨。

读者可以通过单击此 TButton 按钮来实现这个功能。



专家点评

上面仅介绍了实现显示 AVI 第一帧的方法, 读者完全可以参照其实现方法, 实现如“显示 AVI 最后一帧”、“显示 AVI 第几帧”等功能。此外, 上面仅是实现了显示 AVI 文件的方法, 读者还可以调整用于实现其他类型的多媒体文件。

1.2 用 API 实现播放一个 WAV 文件



关键所在

通过调用一组 API 函数, 是不难实现这个功能的。这组 API 函数如下。

(1) sndPlaySound 函数

声明: BOOL sndPlaySound(LPCSTR lpszSound, UINT fuSound);

此函数用来播放一个 Wav 格式的声音。如果成功返回 True, 否则返回 False。

参数 lpszSound 表示一个规定要播放的声音的字符串, 可以是一个 Wave 格式的文件名或注册表的一个入口或 Win.ini 文件中标识一个系统声音的入口。

参数 fuSound 表示播放声音的标记。它可取的值得有 SND_LOOP, SND_MEMORY, SND_NODEFAULT, SND_NOSTOP, SND_SYNC, SND_ASYNC。

- (2) FindResource 函数 用于说明在指定的模板中含有指定名称和指定类型的一个资源的位置。
- (3) LoadResource 函数 把指定的资源调入内存中。
- (4) LockResource 函数 锁定内存中指定的资源。
- (5) UnlockResource 函数 解开内存中指定资源的锁
- (6) FreeResource 函数 释放占有此资源的内存。



实现与应用

下面的程序使用了四种实现调用 API 播放一个 Wave 文件的方法，步骤如下。

- (1) 首先，创建一个应用程序。
- (2) 在应用程序的窗体上加上四个 TButton 组件。
- (3) 分别编写此四个 TButton 组件的 OnClick 事件，读者可以通过单击每一个按钮来实现此四种方法。

源代码如下：

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils,
```

```
Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Button4: TButton;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
procedure Button4Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
    Form1: TForm1;

implementation
uses mmsystem;
//不要忘了把这个 use 语句加入程序中
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
    sndPlaySound('d:\private\chimes.wav',SND_FILENAME or SND_SYNC);
end;

procedure TForm1.Button2Click(Sender: TObject);
var
    f: file;
    p: pointer;
    fs: integer;
begin
    AssignFile(f, 'D:\private\chimes.wav');
    //把'D:\private\chimes.wav'与变量 f 相关联
    //chimes.wav 文件是 Windows 自带的一个 Wav 文件,可以把它拷贝到 D:\private 目录下

    Reset(f,1);
    //打开一个存在的文件
    fs := FileSize(f);
    //获取文件长度
    GetMem(p, fs);
    //为此文件分配内存
    BlockRead(f, p^, fs);
    //把整个文件的记录读到缓冲区 P 中
    CloseFile(f);
    //关闭此文件
    sndPlaySound(p,
    SND_MEMORY or SND_SYNC);
    FreeMem(p, fs);
    //释放内存
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  PlaySound('D:\private\chimes.wav',
    hInstance,
    SND_RESOURCE or SND_SYNC);

end;

procedure TForm1.Button4Click(Sender: TObject);
var
  h: THandle;
  p: pointer;
begin
  h := FindResource(hInstance, 'D:\private\chimes', 'WAVE');
  h := LoadResource(hInstance, h);
  p := LockResource(h);
  sndPlaySound(p, SND_MEMORY or SND_SYNC);
  UnlockResource(h);
  FreeResource(h);
end;

end.
```



专家点评

上面仅介绍了用 API 播放 WAV 文件的四种方法，实际上是对 `sndPlaySound` 函数做了一个全面的阐述，读者通过参见此四种方法，应该对 `sndPlaySound` 函数有了一个大致的了解，在遇到相似的问题时，完全可以参照其实现方法。

1.3 Delphi 编程实现 3DS 动画的播放



关键所在

Delphi 是一种具有强大功能的编程语言，用它可以轻松创建任何一种数据库应用程序。将 3DS 动画文件。FLC 分解后，按先后顺序存于 Delphi 提供的一个表的 Graphics 数据类型中，通过使用 Timer 组件来读取 Graphics 类型的图形数据，并在 TDBImage 组件中显示出来，从而在窗口中实现动画效果。

Ttable 组件

(1) DataBaseName 属性 这个属性指定要访问的数据库,通常是数据库的别名,也可以设为数据库文件所在的路径。

(2) TableName 属性 这个属性用于指定要访问的数据库表。

(3) Active 属性 如果设为 True,相当于把 TTable 组件变成活动状态。

(4) Eof 属性 如果当前记录是最后一个记录,就返回 True。

(5) First 方法 这个方法使第一条记录变成当前记录。

(6) Next 方法 这个方法使下一条记录变成当前记录。

(7) FieldByName 方法 这个方法用于让用户通过字段名访问当前记录的某个字段,返回一个 Tfield 对象。

此程序调用了 TTimer 组件的 Interval 属性,用来设置触发 OnTimer 事件的时间间隔,以 ms 为单位,如果设为 0 表示禁止定时器。



实现与应用

Delphi 编程实现 3DS 动画的播放的步骤如下:

(1) 将 3DS 分解的图片存于数据库中,进入 3DS 的 KeyFrame 模块,在 Output 选项中选择.bmp 类型的文件。将图形文件全部存于同一目录中(例如目录 D:\private,文件可为 f01.bmp, ..., f04.bmp)。

利用 Delphi 的数据工具 DeskTop 建立一个图形数据库 Picture.db,其中包括图形文件名 Filename 和图片 Picture 两个字段。

(2) 创建一个新的应用程序,并加入一个 Timage 组件。

(3) 从 DataAccess 页中选择 Table 组件,并将其放入窗体 Form1 中,其属性为:

Name 属性 table1

DataBase 属性 DBDEMOS

TableName 属性 Picture.db

从 DataAccess 页中选择 DataSource 组件,放入 Form1 窗体中,设置属性为:

Name 属性 DataSource1

DataSet 属性 table1

从 DataControl 页中选择 DBImage 选件,放入 Form1 窗体中,设置属性为:

Name 属性 image1

DataSource 属性 DataSource1

DataField 属性 Picture

Stretch 属性 True

(4) 为 Form1 窗体的 FormCreate 事件填写如下代码:

```
procedure TForm1.FormCreate(Sender:TObject);
begin
table1.open;
with table1 do
```

```
begin
while not eof do
begin
image1.Picture.LoadFromFile('d:\private\' + fieldbyname('filename').asString + '.bmp');
edit;
fieldbyname('picture').assign(image1.picture.graphics);
next;
end;
end;
end;
```

(5) 使用 Timer 组件实现动画演播, 从 System 页中选择 Timer 组件放置到窗体 Form1 中, 设置属性如下:

```
Name   Timer1
Enabled False
Interval 250
```

(6) Timer 组件的 OnTimer 事件定期自动发生。例如设置 Timer1 组件的 Interval 属性为 250, 那么每隔 250ms 都将会自动触发 OnTimer 事件。为 OnTimer 事件添加的代码为:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
table1.next;
if table1.eof then
table1.first;
end;
```

建立好的窗体如图 1-1 所示。

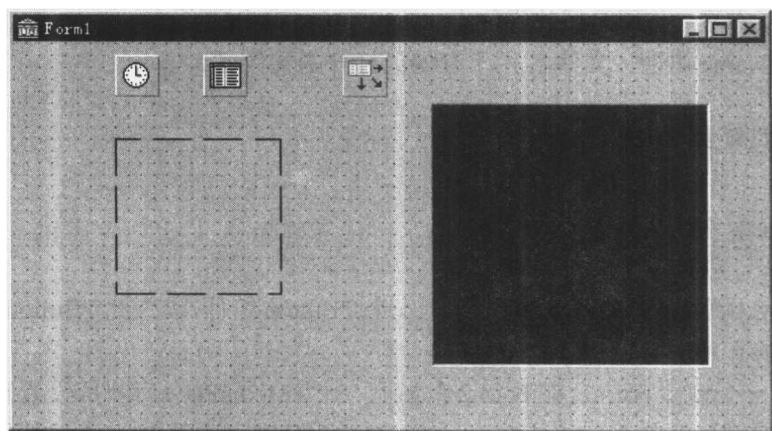


图 1-1 程序的主窗体



专家点评

上面介绍了用 Delphi 实现 3DS 动画的播放。读者可以从中了解 TTimer 组件的大概用法,读者也可以利用这个方法实现几幅平面图拼凑的动画效果。

1.4 全屏显示 AVI



关键所在

通过调用一组 TMultiMedia 组件的一些方法和属性,再恰当的设置 TMediaPlayer 组件的一些属性即不难实现全屏显示 AVI 的。这组方法和属性如下。

(1) Open 方法 此方法用于打开一个多媒体设备。在调用该方法之前,必须确认 MediaPlayer 组件的 DeviceType(设备类型)属性中设定正确的多媒体设备类型。

(2) Notify 属性 如果这个属性设为 True,设置后的第一个媒体控制方法(如 Back、Eject、Next、Open 等)执行结束后,将会触发 OnNotify 事件表示操作已结束,结束时的状态可以通过 NotifyValue 属性访问。如果这个属性设为 False,执行结束是不会触发 OnNotify 事件。

(3) NotifyValue 属性 这个属性返回调用媒体控制方法的执行情况,可以是如下值: NvSuccessful(执行成功)、nvSuperseded(被其他调用覆盖)、nvAborted(被用户中止)和 nvFailure(执行失败)。

(4) Previous 方法 把当前的位置设置为前一个音轨的开始处。

(5) DeviceType 属性 此属性用来指定要打开设备的类型。

(6) DisplayRect 属性 可以调整显示区域位于该控制控件和位置。

(7) Display 属性 决定当媒体播放时,显示的区域在哪个控制控件上。

(8) Play 方法 作用是播放媒体。

(9) FileName 属性 文件名,用于存储或打开的文件名。



实现与应用

实现全屏显示 AVI 步骤如下。

(1) 首先,创建一个应用程序。

(2) 在应用程序的窗体 Form1 上加上一个 TButton 组件,设置此 TButton 组件的 Caption 属性为“全屏显示 AVI”。

(3) 增加一个窗体 Form2,在 Form2 上加上一个 TMediaPlayer 组件,设置它的 DeviceType 属性为 dtAVIVideo,再在 Form2 上加上一个 TPanel 组件,设置 Tpanel 组件的大小为整个屏幕。

(4) 编写此 TButton 组件的 OnClick 事件如下: