

用 Delphi 开发多媒体应用程序

陈楠 (玉门石油管理局信息中心 735200)

摘要: 本文通过 Delphi 多媒体组件对音频、视频文件的处理阐述了用 Delphi 开发多媒体应用程序的方法,同时对 Delphi 的其他组件在多媒体程序设计中的应用进行了讨论。

关键词: 多媒体应用程序 Delphi 多媒体组件 MCI

一、概述

在 Delphi 中提供了功能完善的多媒体支持工具——MediaPlayer 组件,为用户提供了多媒体控制界面(MCI)驱动程序。使得用户在 Windows3. X/Windows 95 中建立的 MCI 驱动程序,都可以通过 MediaPlayer 组件进行播放。Delphi 所支持的多媒体文件包括音乐 CD、.wav、.mid 声音文件;.avi、.dat、.mpg 等格式的视频文件以及其他的多媒体文件。

本文首先简要介绍了 Delphi 的多媒体组件 MediaPlayer 及其功能,并在此基础上对 Delphi 的多媒体程序设计方法进行了介绍,最后对 Delphi 的其他组件在多媒体编程中的应用进行了阐述,包括动画设计,图像存取技术等,以期起到抛砖引玉的作用。

二、MediaPlayer 组件简介

Delphi 的多媒体播放器组件包含在 System 页中,在 System 所包含的 12 种组件中,MediaPlayer 组件位于从左至右的第 7 个组件。Media-Player 组件提供了媒体控制接口 MCI(Media Control Interface)的驱动程序,可以控制多种多样的媒体设备和文件。多媒体组件 MediaPlayer 中包含了一组控制音频和视频外围设备的与设备无关性的高级命令,用于管理在 MCI 设备上播放和记录多媒体文件。利用它可以控制 Windows/Windows95 提供的 MCI 设备,可驱动 CD-ROM、声卡、MIDI 音序器、激光唱机、VCD 视盘机等设备,支持 Autodesk Animation、CD Audio、AVI Video、MIDI 音乐和 WAVE 音效等媒体。

程序员利用 MediaPlayer 组件的方法和属性编写事件过程代码,就可以进行多媒体软件的开发和对多媒体设备进行控制。

三、用 Delphi 进行语音编程

1. 处理音频文件

Delphi 通过 MediaPlayer 组件可以处理 MIDI 和 WAVE 等音频文件。其编程方法如下:

(1) 建立一个项目文件,如 MYSOUND.DPR;

(2) 创建窗体,如 Form1,并在窗体中加入 MediaPlayer 组件;

(3) 在对象观察器(Object Inspector)中对 MediaPlayer 组件的各个属性进行必要的设置;

(4) 对媒体设备进行初始化,正确设置 Notify、Shareable、DeviceType 等属性;

(5) 指定要处理的音频文件名及其存取路径,为 FileName 属性赋值;

(6) 对各按钮设置方法;

(7) 编译运行该项目文件,生成可独立运行的 .EXE 文件。

下面的一段程序是一个简单的处理过程(建立窗体后双击):

```
procedure TForm1.formeCreat(Sender: TObject);
begin
    ...
    with MediaPlayer1 do
    begin
        AutoOpen := True; {自动打开媒体设备}
        DeviceType := dtWaveAudio; {指定媒体播放设备是音频设备}
        AutoRewind := True; {自动卷带}
        FileName := 'c:\pwin\abc.wav'; {指定打开的音频文件名及路径}
```

...

```
Play; |播放媒体设备|
Wait; = True; |控制权等待转移|
end;
end;
```

播放结束后,要关闭 MCI 程序,在 TForm1. FormDestroy() 事件中加入如下语句:

```
procedure TForm1. FormDestroy( Sender: TObject)
begin
  Form1. MediaPlayer1. Close; |关闭媒体设备|
  Halt;
end;
```

2. 处理视频文件

Delphi 的 MediaPlayer 组件对视频文件的处理也十分方便,在播放中只需指定相应的播放窗口,如一个按钮或是一个图形对象等。播放窗口可以通过 Display 属性在属性窗口或程序中设置。

下面的一段程序是处理视频文件的过程, BiPict1 是指定的播放窗口:

```
procedure TForm1. FormCreate( Sender: TObject);
begin
  BiPict1. Visible; = Ture; |显示区域可见|
  with MediaPlayer1 do
  begin
```

```
    DeviceType; = dtAVIVideo; |指定媒体设备为 AVI 视频设备|
```

```
    AutoRewind; = True;
```

```
    FileName; = 'c: \ video \ flowers. avi'; |指定打开的 AVI 文件及其路径|
```

```
    Open; |打开 AVI 文件|
```

```
    Display; = BiPict1; |指定播放窗口|
```

```
    Wait; = True; |控制权等待转移|
```

```
end;
```

```
end;
```

3. 多媒体播放器

利用 Delphi 的 MediaPlayer 组件可以进行 CD 唱片播放、AVI 影片播放、MIDI 音乐文件播放和 WAV 音效文件播放等。

下面的一段程序简要地说明了 Delphi 对 CD-ROM 中的 CD 唱盘进行播放的过程(当然稍加改动就可以对

其他的多媒体文件进行播放):

(1)设计播放界面。首先选取 MediaPlayer1 置于窗体 Form1 上,创建 6 个图形按钮: OpenCDbtn、Playbtn、Pausebtn、Stopbtn、Ejectbtn 和 Exitbtn; 双击 Object Inspector 中 OpenFileDialog 的 Filter 属性右边的小框,在弹出的表格中输入要播放的多媒体文件类型及过滤器。完成上述操作后,多媒体播放器的播放界面就可以工作了。

(2)设计事件处理程序。

①打开媒体文件。双击 OpenCDbtn 按钮,在弹出的事件窗口中写入下列程序:

```
procedure TForm1. OpenCDbtnClick( Sender: TObject);
begin
```

```
  OpenFileDialog. Execute;
```

```
  with MediaPlayer1 do
```

```
  begin
```

```
    DeviceType; = dtAutoSelect;
```

```
    FileName; = OpenFileDialog. FileName; |指定文件来源|
```

```
    if FileName < > ' ' |如果文件存在|
```

```
    then Open; |打开文件|
```

```
  end;
```

```
end;
```

②播放媒体文件。双击 Playbtn 按钮,输入以下播放事件处理语句:

```
procedure TForm1. PlaybtnClick( Sender: TObject);
```

```
begin
```

```
  MediaPlayer1. DeviceType; = dtCDAudio; |指定媒体文件类型|
```

```
  MediaPlayer1. FileName; = ' '; |在播放前,先将文件名设为空字符串|
```

```
  MediaPlayer1. Play; |播放媒体文件|
```

```
end;
```

双击 Pausebtn 按钮,输入以下暂停事件处理语句:

```
procedure TForm1. PausebtnClick( Sender: TObject);
```

```
begin
```

```
  MediaPlayer1. Pause; |暂停播放 CD|
```

```
end;
```

③结束媒体播放。双击 Stopbtn 按钮,输入以下结束事件处理语句:

```
procedure TForm1. StopbtnClick( Sender: TObject);
```

```
begin
  MediaPlayer1.Stop; {停止播放}
  MediaPlayer1.Rewind; {退回到起始点}
end;
```

双击 Ejectbtn 按钮, 输入以下的弹出媒体事件处理语句:

```
procedure TForm1.EjectbtnClick(Sender: TObject);
begin
  MediaPlayer1.Eject; {CD 出盒}
end;
```

④退出媒体播放程序。双击 Exitbtn 按钮, 输入以下的退出事件处理语句:

```
procedure TForm1.ExitbtnClick(Sender: TObject);
begin
  MediaPlayer1.Stop;
  MediaPlayer1.Close; {关闭媒体设备}
end;
```

上述程序经编译后生成可执行的 .EXE 文件, 在 Windows 3.2/Windows95 环境下运行, 稍加修改, 用 Openbtn 按钮可以选择其他的多媒体文件进行播放, 如声音文件、视频文件、图象文件等。

四、用 Delphi 进行图像编程

1. 图像存取技术

以数据库应用为例来说明这一技术。Delphi 的数据库引擎(BDE)具有处理图像的能力, 在一个表中创建一个 Graphic 字段就可以对图像进行存取。

(1)保存图像数据。其基本思想是: 首先建立一个含有 Graphic 字段的数据库, 然后在窗体中创建一个显示图像的窗口, 接着打开要显示的图像文件(.bmp 文件), 最后把图像数据保存到相应的数据库中。

实现方法如下:

①建立数据库。在 Delphi 的 Database Desktop 组件中建立数据库文件 mydbfile.gdb:

```
create table mydbs
(name varchar(10) not null, age varchar(4) not null,
address varchar(30), telephone varchar(12), picture varchar(15), image blob);
```

库中, picture 字段用于保存图像的名称及其路径, image(是 Graphic 字符)用于存储图像数据, 其数据类型为“blob”。

②建立图像显示窗口。在 Form 中建立某人的资料输入界面, 姓名、年龄、地址和电话用 Tedit 组件创建, 图像(相片)显示窗口用 Timage 组件创建。主要控件的属性设置如下:

```
DataSource1.DataSet := Table1;
Table1.DatabaseName := mydbfile.gdb;
Table1.TableName := mydbs;
Table1.active := True;
...
```

③设计图像处理程序。打开图像文件:

```
procedure TForm1.PictOpenbtnClick(Sender: TObject);
begin
  OpenFileDialog.execute;
  Image1.Picture.LoadFromFile := OpenFileDialog.FileName;
end;
```

将输入信息及图像数据保存到相应的数据库(Graphic1)中:

```
procedure TForm1.SavebtnClick(Sender: TObject);
var Graphic1: TGraphic;
begin
  Graphic1 := TGraphic.Create;
  Graphic1.LoadFromFile := OpenFileDialog.FileName;
  Table1.Insert;
  Table1.FieldByName('Name').AsString := Edit1.Text;
  Table1.FieldByName('Age').AsString := Edit2.Text;
  Table1.FieldByName('Address').AsFloat := (Edit3.Text);
  Table1.FieldByName('Telephone').AsFloat := (Edit4.Text);
  Table1.Fields[4].Assign(Graphic1);
  Table1.Post;
  Graphic1.Free;
end;
```

(2)读取图像数据。基本思想是: 建立一个信息显示(包括图像信息)的界面, 图像窗口用 TDBImage 创建, 其他窗口用 TDBEdit 创建。将控件 TDBEdit 和 TDBImage 的 DataField 属性设置为与保存时相对应的域,

TDBNavigator 的 DataSource 属性设置为 DataSource1 即可。

其实现方法基本与保存时类似。

2. 动画设计

在 Delphi 中有一个图象组件 Timage, 利用它与其他的组件配合使用先可以实现图像动画。

(1) 连续显示多幅图像。其基本思想是: 单个 Timage 组件可以显示一幅静态的图像, 利用多个 Timage 组件, 事先对要连续显示的图像进行预装, 在显示时复制组件的图像属性, 就可以实现多幅图像的动画显示。

其实现方法是: 首先在 Form 窗体中创建 4 个 Timage 组件(以显示 4 幅图像为例)和一个时钟组件 Timer。Timage1 为图像显示组件, 其 Visible 属性为 True, 其他 3 个 Timage 组件为图像预装组件, 它们的 Visible 属性为 False, 在 Timage2 - - Timage4 中分别装入 3 幅不同的图像, 然后双击 Timer 组件, 输入如下的 OnTimer 事件处理语句:

```
procedure TForm1.Timer1Timer(Sender: TObject);
const i: integer = 2;
begin
  case i of {按顺序装入后续图像}
    2: image1.picture := image2.picture;
    3: image1.picture := image3.picture;
    4: image1.picture := image4.picture;
    5: begin
        image1.picture := image2.picture; {更换图像}
        i := 2;
      end;
  end;
  i := i + 1; {显示下一幅}
end;
```

(2) 图像的快速移动。其基本思想是: Delphi 中的 Tpanel 组件可以实现图像的平滑移动, 用 Timage 组件和 Tpanel 组件相配合, 通过改变图像组件的显示位置属性 AutoSize, 可以实现图像的快速移动。既可以按固定路线移动, 也可以任意随机移动。

其实现方法是: 在窗体中创建 1 个 Tpanel 组件, 1 个 Timage 组件和 1 个 Timer 组件。Timage 组件须置于 Tpanel 组件中。在 Timage 组件中装入一幅图像, 其 AutoSize 属性为 True。

①按固定路线移动(以在 4 个点移动为例)。双击

Timer 组件, 输入以下的 OnTimer 事件处理语句:

```
procedure TForm1.Timer1.Timer(Sender: TObject);
begin
  i := i + 1;
  image1.left := arrposi[i, 1];
  image1.top := arrposi[i, 2];
  if i = 4 then i := 0;
end;
```

再激活 TForm 中的 OnCreate 事件, 输入以下的处理程序:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  i := 0
  {指定移动路线}
  arrposi[1, 1] := 5;
  arrposi[1, 2] := 156;
  arrposi[2, 1] := 287;
  arrposi[2, 2] := 10;
  arrposi[3, 1] := 587;
  arrposi[3, 2] := 156;
  arrposi[4, 1] := 287;
  arrposi[4, 2] := 315;
end;
```

②按随机路线移动。双击 Timer 组件, 输入以下的 OnTimer 事件处理语句:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  image1.left := random(80);
  image1.top := random(30);
end;
```

语句中的 80 和 30 分别为 Timage 组件在 Tpanel 组件中移动的边界值, 超过此值, 图像将不能显示。

上述图像移动的速度可以通过改变 Timer 组件的 Interval 属性的值来改变。

参考文献

- [1] 张育荣、曾彦贺, “Delphi - 从入门到精通”, 北京: 清华大学出版社, 1996.7.
- [2] [美] A. Wozniwicz & N. Shamma, 岳丽华、杨寿保译, “Delphi 程序员指南”, 北京: 科学出版社, 1997, SAMS Publishing USA. 1995.

(来稿时间: 1997年10月)