

如何用 Delphi 开发多媒体应用程序

王敏 (上海长征医院 200003)

陶蓉 (上海长海医院 200003)

目前在开发多媒体软件的过程中,开发工具的选择可有两类方案:一类是采用创作系统(Authoring System),另一类是采用程序设计。一般来说程序设计的方法灵活性好、功能强大,能够开发范围广泛的多媒体软件。在此介绍如何以 Delphi 为工具在程序设计中进行多媒体开发。

一、图形与动画设计

在 Delphi 中可用两种方式来产生图形:一为图形控件(Graphical Controls);二为图形方法(Graphical Method)。

1. 图形控件

Delphi 中的图形控件包括三种:线条与边框控件(Bevel)、形状控件(Shape)、图像控件(Image)。使用图形控件与其他多媒体编辑软件很相似,可以从线条、形状或图像控件中选择一项置入或绘制图形。

线条与边框控件是绘直线或边框的工具。在控件当中可以决定线条的形状、3D 外形、位置以及长度等等。其中 Shape 属性可以决定其形状是箱状、边框、垂直线还是平行线;Style 属性可以决定其 3D 外形是凸起还是凹陷;此外 Top 与 Left 属性可以决定其位置;Width 与 Height 属性可以决定其长短。

形状控件可以直接在窗体上绘制各种几何图形,如圆形、椭圆形、矩形、正方形、圆角矩形及圆角正方形等。这些几何图形可以为空心或实心的,图形的边线可以显示也可以不显示出来,边线颜色、宽度、形态可以自行设置,图形内部可以为不同的图案、颜色等。

在图像区域可以输入照片或其他图像数据。这些图像数据的文件有三种格式。第一种为“位图”(Bitmap)文件,扩展名为“.BMP”或“.DIB”,它是以点状组合成的图形;第二种为“图标”(Icon)文件,它是一种特别的位图,只是它最大限度在 32×32 象素的范围,文件扩展名为“.ICO”;第三种为“元文件”(Metafile),它是一种综合的图像,以直线与形状共同组合而成,以“.WMF”为扩展名。图像控件一个最主要的属性是 Picture,通过设置

该属性即可将所选图像放置在图像控件中。图像控件还有一个主要的属性是 Stretch,将它的值设为 True 会使得装入图像控件的图像大小随图像控件的大小变化。如果它的值为 False 你将会发现图像控件也许只显示了所选图像的一个部分,这不是你所期待的结果。而另一个属性 AutoSize 若设为 True 则图像控件的大小随图像大小变化。

2. 图形方法

这是通过程序方法绘图。Delphi 中有一称为 TCanvas 的对象,它的主要功能是设计界面,但它的功能很强,其中包含了一些画图的方法。如 LineTo 为绘制线条的方法、Rectangle 为绘制矩形的方法、Ellipse 为绘制椭圆的方法;Pen 属性用于绘制边线、Brush 属性用于填充图形中心。可以利用这些方法和属性来绘制各种图形。图形方法比起图形控件来绘图较具弹性,且存储空间也较小,因为它只要程序码就可以了,不必再将图形本身的数据存储起来。

如下面程序将在窗体中绘制一个绿边红心的实心圆(花括号中为 Delphi 的注释语句):

```
Canvas.Pen.Color := clGreen; |设置绘边框的工具
Pen的颜色为绿色|
```

```
Canvas.Pen.Width := 4; |设置边框的宽度|
```

```
Canvas.Ellipse(4, 4, 100, 100); |绘制一个起始位置
为 4, 4 直径为 100 的圆|
```

```
Canvas.Brush.Color := clRed; |设置填充中心的工
具的颜色为红色|
```

```
Canvas.Brush.Style := bsSolid; |设置将圆填为实心|
```

```
Canvas.FloodFill(40, 40, clGreen, fsBorder); |从 40,
40 的位置开始填充圆,一旦遇到绿边则停止填充|
```

对于图像控件来说,它的图形方法很简单。图像控件的 Picture 属性有一 LoadFromFile 方法,用此方法则在程序运行过程中动态地为图像控件加入图像。

```
如 Image1.Picture.LoadFromFile(* c:\myPic-
ture\moon.bmp *);
```

另外如果用户的图像是存放在数据库中的,则用

Delphi 的数据库图像控件(DBImage)也可以很方便地将其调用。这时就要涉及 Delphi 中数据库的使用方法,在此不再赘述。

3. 一个动画实例

在 Delphi 中实现动画比较容易,本例即实现用 Delphi 中的计时器(Timer)控件演示一个动画。这个动画由五幅图片组成,这五幅图片的文件名分别为 Cartoon1—Cartoon5,它们共同组成一组连续画面。计时器 Timer1 的 Interval 属性设为 1000,变量 i 的初值在 Form-Create 事件中设为 1。

```
procedure TForm1.Timer1Timer(Sender: TObject);
VAR CartoonName:string;
begin
  IF (i >= 9) OR (i < 1) THEN i := 1;
  i := i + 1;
  CASE i OF
    1:CartoonName := 'Cartoon1';
    2:CartoonName := 'Cartoon2';
    3:CartoonName := 'Cartoon3';
    4:CartoonName := 'Cartoon4';
    5:CartoonName := 'Cartoon5';
    6:CartoonName := 'Cartoon4';
    7:CartoonName := 'Cartoon3';
    8:CartoonName := 'Cartoon2';
    9:CartoonName := 'Cartoon1';
  END; {CASE}
  Image1.picture.LoadFromFile('c:\myPicture\' +
  CartoonName + '.bmp');
end;
```

二、OLE 载体控件

OLE(Object Linking Embedding,对象的连接与嵌入)协议是 Windows 环境中的概念。OLE 使应用程序间的数据交换更易于管理,且为多媒体数据对象的综合、集成提供了有力的工具。Delphi 中的 OLE 载体控件用于实现与 Delphi 外部某个对象的接口功能。这个功能可以为没有经验的用户提供进入其他应用程序及其所控制对象的捷径并允许 Delphi 外部对象实时变化。

下面通过一个例子来说明 OLE 载体控件的用法。本例演示了利用 OLE 功能,将 BMP 图形文件与 WAV 声音文件合为一体制作多媒体演示程序的方法。

1. 在窗体 Form1 上创建一个 OLE 对象 OleContainer1,在该对象上双击得到 Insert Object 对话框,选择

Create New 单选按钮,然后在 Object Type 中选择 Wave Sound,此时在 OLE 载体控件中出现了 Sound 图标。双击该图标,出现 Sound Object In OleContainer1 对话框,选择 Edit 菜单的 Insert File *,加载声音文件。本例加载声音文件为 c:\MyBmp\001.bmp。

2. 在窗体 Form1 上创建一个图片框 Image1。

3. 在窗体 Form1 上创建一个按钮 Button1。

4. 例子程序如下:

```
procedure TForm1.Button1Click(Sender: TObject)
begin
  Image1.Visible := True;
  Image1.Stretch := True;
  Image1.Picture.LoadFromFile('c:\MyBmp\001.
  bmp');

  OleContainer1.DoVerb(ovPrimary); {在 Delphi 中可用
  ovPrimary 代替 0}
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  OleContainer1.Visible := False;
  Image1.Visible := False;
end;
```

程序说明:

1. 程序第 1 步是在程序设计中静态地加载声音文件,还可以在程序中使用下列语句动态地加载声音文件:

```
VAR
  Info: Pointer;
begin
  IF InsertOLEObjectDlg(Form1, 0, Info) THEN
  BEGIN
    OleContainer1.PInitInfo := Info;
    ReleaseOLEInitInfo(Info);
  END;
end;
```

2. OleContainer1.DoVerb(ovPrimary)语句用于设置 OLE 对象的状态(还可写成 OleContainer1.DoVerb(0))。在 Delphi 中 ovPrimary 值代替 0,表示直接 Play。不同的 OLE 应用软件可能会有不同的 Verb 描述来代表不同的动作。一般而言,Verb = 0 代表“Play”、Verb = 1 代表“Edit”。

使用 OLE 进行多媒体设计的好处是简单易用,但是相对的代价是忍受较频繁的磁盘 I/O 操作。幸好

Delphi 中可以使用 Media Player 媒体播放器。

三、媒体播放控件(Media Player)

Media Player 是 Delphi 中提供的一个 MCI 控件,用于管理 MCI 设备上播放多媒体文件。通过对该控件属性的设置和事件的编写,可以完成多媒体控制。Media Player 处理多媒体信息的能力非常强。用户通知它播放哪个文件,无论该文件是何种类型(.WAV、.MID、.CD、.AVI、.FLC、.FLI、.MMM...),媒体播放器都会播放。

Media Player 控件在 Delphi 的控制板上的 System 页上。它的外观形式是一组彩色按钮,如图 1 所示。

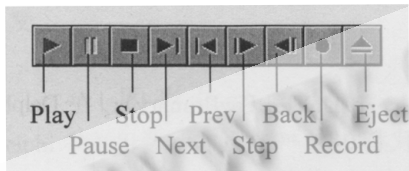


图 1 媒体播放器 Media Player

使用一个 Media Player 可以相应地控制一个 Windows 提供的 MCI 设备。这里的设备不仅指硬件设备,也指应用软件。这些按钮的功能也可以在程序中用相应的方法代替。

媒体播放器的主要属性如下:

- FileName: 指明媒体文件的文件名。
- DeviceType: 指明要打开的 MCI 设备的类型。若将值设为 dtAutoSelect 自动根据 FileName 中所指定的文件进行自动媒体类型检测。如果设置 DeviceType 为其他值,则自动关闭媒体类型检测,使用指定的设备。
- Wait: 是否等待命令执行完毕再执行下一条。
- Notify: 决定下一个 MCI 命令是否使用 MCI 通知服务。当 Notify 设置为真时,下一个完成的媒体控制方法将产生一个 OnNotify 事件。

如果预计下一个控制方法占用大量时间,就可能需要使用 Wait 和 Notify 属性。

- Shareable: 决定可否与其他应用程序共用同一个设备。
- AutoRewind: 是否自动卷带。

下面通过具体实例来介绍 Media Player 的使用方法。

1. 播放波形声音

(1) 选择 Media Player 控件放入窗体中。设置其

Visible 属性为 False; 设置 Media Player 控件的 Device Type 属性为 dtWaveAudio; AutoOpen 属性为 True。

(2) 在调用处编写如下程序:

```
WITH MediaPlayer1 DO
BEGIN
    IF Active THEN Close; {假如设备状态为打开,必须先关闭。}
    FileName := 'c:\MyWav\kick8.wav'; {设置声音文件名}
    Open; {打开 DeviceType 属性指定的设备}
    Position := 0;
    Play; {进行播放}
    Close; {关闭 DeviceType 属性指定的设备}
END; // WITH
```

(3) 如果想重复播放可在 WITH 中用如下语句:

```
Open;
Wait := True; {等待声音文件播放完成}
Play; {进行播放}
Play; {重新播放}
Close; {关闭 DeviceType 属性指定的设备}
```

2. 播放 MPG 文件

(1) 选择一个 Media Player 控件放入窗体中。设置 Media Player 控件的 Device Type 属性为 dtAutoSelect; AutoOpen 属性为 True。

(2) 在调用处编写如下程序:

```
WITH MediaPlayer1 DO
BEGIN
    IF Active THEN Close; {假如设备状态为打开,必须先关闭。}
    FileName := 'c:\MyMpg\001.mpg'; {设置媒体文件名}
    Open; {打开 DeviceType 属性指定的设备}
    Position := 0;
    Play; {进行播放}
    Close; {关闭 DeviceType 属性指定的设备}
END; // WITH
```

通过以上两个例子可见,在 Delphi 中使用 Media Player 控件进行多媒体播放是十分简单且方便。无论播放何种类型的文件,只要选择不同的 DeviceType,设置不同的 FileName 即可。其他方法都是通用的。

(来稿时间:1997 年 8 月)