

# Visual Basic 中的图形和文本交互技术

应 崎 严寒冰 (浙江丝绸工学院电子系 310033)

**摘要:**本文介绍了在 Visual Basic 中实现图形和文本的人机交互技术,介绍了多媒体软件平台 ZSMITS 的“电子小黑板”的实现方法。

**关键词:**交互技术 橡皮筋线 Visual Basic 多媒体 CAI

## 前言

Visual Basic 是一个理想的多媒体开发工具,利用它可以开发出交互性较强的应用程序。图形和文本的交互是一个基本的交互形式,VB 中可以用 picturebox 控件显示图片,但是不能直接用于文本交互;Textbox 控件是专门用于文本交互的,但是它不能用于图形交互。

在有些应用程序中,需要同时进行图形和文本的交互。比如说开发一块上课用的电子小黑板,教师在黑板上既要写板书,又要画示意图。那么,怎样使得黑板既能作图,又能写板书呢?本文介绍在我们自己开发的多媒体教学软件平台 ZSMITS 中实现电子小黑板的图形和文本交互技术。

## 一、图形交互技术

图形交互可以在窗体(Form)或 picturebox 控件上实现,其最基本的功能是画直线和二次曲线功能。画线的

基本交互方式有输入参数和橡皮筋线法两种:

### 1. 输入参数法

由 Textbox 接收从键盘输入的参数,然后执行画线命令。

这种方法对用户的限制较大,需要参数齐全、准确。而大多数用户并不知道,也不需要知道屏幕上的坐标,因而在输入上造成了困难和不便。

### 2. 橡皮筋线法

橡皮筋线是在屏幕上动态产生用户满意的直线段的方法。在屏幕上固定线段的一个端点 p1,线段的另一个端点 p2 随着位置的变化而清除前一根线段,产生一根新的 p1p2 线段。

在窗体的 mousemove 事件中填写代码之前,应该将其 drawmode 属性设为 7 - Xor Pen(异或笔),具体代码如下:

(设 brush 为窗体名, xstart, ystart 为直线起点坐标)

```
brush.DrawMode = 7
```

```
Line (xstart, ystart) - (X, Y)
Line (xstart, ystart) - (xold, yold)
xold = X
yold = Y
```

在上面的程序中使用了 xold 和 yold 两个变量,它们用来记录鼠标移动瞬间的坐标值,这是非常有用的。利用这个原理,我们还可以实现另一个图形交互操作——徒手画。

所谓徒手画,就是按住鼠标左键,画出鼠标移动的轨迹。在窗体的 mousemove 事件中添加如下代码:(设 brush 为窗体名)

```
If Button = 1 Then
    Line (oldx, oldy) - (X, Y), brush.ForeColor
End If
oldx = X
oldy = Y
```

## 二、文本交互技术

实现文本交互的基本方法有 Textbox 控件和 Print 方法等。

### 1. Textbox 控件

文本框控件的优点是简单、交互性强,无需添加代码,用户就可以通过键盘在文本框内输出文字。如果与 Comondialog 控件配合使用,可以改变文本框内的字体。如:

(设 Textbox 的 Name 属性为 txtbrush, Comondialog 的 Name 属性为 cdlbrush)

```
cdlbrush.action = 4
txtbrush.FontName = cdlbrush.FontName
txtbrush.FontSize = cdlbrush.FontSize
txtbrush.FontBold = cdlbrush.FontBold
txtbrush.FontItalic = cdlbrush.FontItalic
txtbrush.FontUnderline = cdlbrush.FontUnderline
txtbrush.FontTransparent = cdlbrush.FontStrikethru
```

但是,Textbox 控件的文本定位较为不便,而且它是为文本交互设计的,不能进行图形交互。

### 2. Print 方法

Print 方法可以在 (Currentx, Currenty) 处将文本打印到窗体或 picturebox 控件上。其语法结构如下:

```
object.Print [outputlist]
```

这里,object 是窗体名,outputlist 是输出的文本。

例如要输出文本“OK”,程序代码如下:(设窗体名为 brush)

```
brush.Print "OK"
```

Print 方法的优点是文本定位灵活,可以将文本输出在窗体或 Picutrebox 控件上,而图形的交互恰恰只能在窗体或 Picturebox 上实现。但是,它失去了交互性,输出的文本需要事先写入代码。

既然文本框与 Print 方法各有优缺点,那就可以结合 Print 方法定位灵活和文本框交互性强的优点,在窗体上同时实现图形和文本的交互。基于这个想法,可以设计出第三种方法。

### 3. Print 方法 + Textbox 控件

在窗体上添加一 Textbox 控件,将其 Visible 属性设为 false。要输入文本时,将文本框设为焦点。当键盘有字符输入时,通过 Print 方法将其输出到窗体上,在文本框控件的 Change 事件中添加如下代码:(设 txtbrush 为文本框控件名,brush 为窗体名)

```
txtbrush.SetFocus
brush.Print Right(txtbrush.Text, 1)
```

## 三、电子小黑板设计

电子小黑板模拟了板书、示意图、黑板擦等功能,它以窗体作为黑板,实现了图形和文本的交互。

在实现电子小黑板时,我们定义了一个名为 brush 的窗体,名为 cdlbrush 的 Comondialog 控件和一个名为 txtbrush 的文本框控件。

### 1. 界面设计

将窗体的背景色设为黑色, Autodraw 属性设为 True。其界面和效果如图 1 所示:

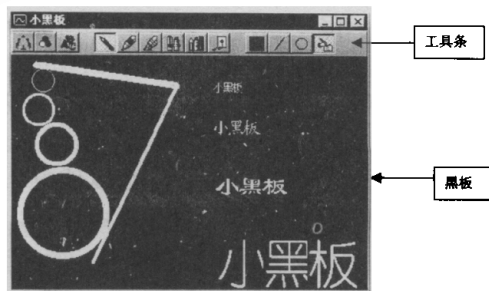


图 1

### 2. 功能设计

如图 2 所示,在窗体上添设一工具条。



图 2

①清屏:使用 Cls 方法。

```
Brush. cls
```

②颜色:添加一 CommonDialog 控件,将其 Action 属性设为 3。

```
cdlbrush. Action = 3
```

```
brush. ForeColor = cdlbrush. Color
```

③字体:将 CommonDialog 控件的 Action 属性设为 4。

```
cdlbrush. Action = 4
```

```
brush. FontName = cdlbrush. FontName
```

```
brush. FontSize = cdlbrush. FontSize
```

```
brush. FontBold = cdlbrush. FontBold
```

```
brush. FontItalic = cdlbrush. FontItalic
```

```
brush. FontUnderline = cdlbrush. FontUnderline
```

```
brush. FontTransparent = cdlbrush. FontStrikethru
```

④粉笔:改变窗体的 Drawwidth 属性。

```
Brush. drawwidth = x(x:1-9)
```

⑤黑板擦:将鼠标附近的区域涂为黑色,代码添加在窗体的 mousemove 事件中。

(设按下 Ctrl 键时可以擦除)

```
If Shift = 2 Then
```

```
brush. DrawMode = 13
```

```
brush. Line (X - 60, Y - 200) - (X + 60, Y + 200), QBColor(0), BF
```

```
End If
```

图形和文本的交互方法已经在上面介绍过,这里要注意几点:

①画橡皮筋线时,窗体的 Drawmode 属性为 7,即异或方式。但是,当退出画橡皮筋线的循环时,应将 Drawmode 属性设为 13,即拷贝方式。

②由于采用了 Print 方法,并且控制文本输出的代码发生在 txtbrush 的 Change 事件中,所以应该屏蔽 Backspace、Tab、Delete 和 Enter 键,以免产生错误。

在 txtbrush 的 keypress 事件中添加如下代码:

```
Select Case KeyAscii
```

```
Case 8 'Backspace 键
```

```
KeyAscii = 0
```

```
Case 9 'Tab 键
```

```
KeyAscii = 0
```

```
Case 13 'Enter 键
```

```
KeyAscii = 0
```

```
Case 83 'Del 键
```

```
KeyAscii = 0
```

```
End Select
```

③由于字符与汉字的大小不一致,斜体与正体的字间距不一致,在用 Print 方法输出时可能会出现文字的重叠现象,需要相对调整文本的输出坐标。在 txtbrush 的 Change 事件中添加调整坐标程序代码如下:

```
CurrentX = px
```

```
CurrentY = py
```

```
If Asc(Right(txtbrush. Text, 1)) > 0 And Asc(Right(txtbrush. Text, 1)) < 255 Then
```

```
    If brush. FontItalic = True Then
```

```
        px = px + brush. Font. Size * 13.7
```

```
    Else
```

```
        px = px + brush. Font. Size * 10
```

```
    End If
```

```
    Else
```

```
        If brush. FontItalic = True Then
```

```
            px = px + brush. Font. Size * 22
```

```
        Else
```

```
            px = px + brush. Font. Size * 17
```

```
        End If
```

```
    End If
```

## 结束语

本文介绍的程序代码均在 VB5.0 上通过,“电子小黑板”已在 CAI 软件——“多媒体教学平台 ZSMITS”中实现,经使用,效果良好,ZSMITS 获得了 98 年度浙江丝绸工学院优秀教学成果一等奖。

## 参考文献

- [1] 顾志远、洪国胜编著,《Visual Basic 入门与应用》,清华大学出版社,1996年6月出版

(来稿时间:1999年5月)