

地理信息系统中双地图窗口的交互查询

程吉宽 周健 (北京航空航天大学 202 教研室 100083)

摘要:地理信息系统(GIS)在国民经济的各个领域得到了广泛应用。本文介绍了如何使用 VB 在 GIS 开发中实现两个地图窗口的交互查询,为用户提供方便的使用界面,并给出了相应的程序。

关键词:地理信息系统 GIS Visual Basic 主地图窗口 浏览窗口 交互式查询

一、问题的提出

在开发地理信息系统软件时,经常是一个大的地图窗口结合数据查询环境使用。很多情况下,用户需要快速地浏览整张地图,另外一些情况下,又需要放大后细致地查看某一部分,快速性和细致性之间是一对矛盾。另外,当用户查询细部时,只看到当前很小范围的地图信息,虽然可以通过数据来推测当前窗口在整张地图中的位置和相对大小,但是不够直观,难以形成一个整体印象,造成“只见树木不见森林”的情况,使得图形界面的优势不能完全发挥。因此,如何使用户能立即知道当前地图窗口在整个地图中的位置和相对大小并能快速、方便地浏览整个地图是非常重要的和实用的功能。

笔者在实际设计中采用两个地图窗口。一个是主地图窗口,另一个小的地图窗口称为浏览窗口。主地图窗口用来进行地图的实际查询和检索,浏览窗口用来显示主地图窗口的当前内容在整个地图中的相对大小和位置。两个窗口之间建立起连接,当在主地图窗口中进行漫游、放大或缩小操作时,浏览窗口中的指示框发生大小和位置的相应变化;反之,当用户在浏览窗口中改变指示框的大小和位置时,主地图窗口中的内容也相应变化。

二、实现方法

主地图窗口和浏览地图窗口的实现方法是不一样的,主地图采用 OLE 的方式与地图开发平台建立联系,在其上可以进行地图平台上所能进行的一切工作。浏览地图窗口内则放置一个位图图片,它和主地图窗口之间建立联系而与地图平台没有直接联系。

1. 用图像框 (picture box) 实现

首先做一个简单的练习,实现指示框的随意移动。

在 Form1 中加入一个时钟 Timer1,将其 Interval 设

置成 400, Visible 设置成 False。然后在 Form 的右边放置另一个 picture box,称为 picture1。在该 picture box 中再画入一小 picture box,称为 picture2。将其 Appearance 属性设置为 0 - flate。我们的目的是使 picture2 在 picture1 中移动以表示主地图的位置和相对大小。在 picture1 中加入如下程序:

```
Private Sub picture1-MouseUp (Button As Integer,
Shift As Integer, X As Single, Y As Single)
    picture2.Left = X - picture2.Width / 2
    picture2.Top = Y - picture2.Height / 2
End Sub
```

当在 picture1 中单击鼠标时, picture2 的中心会移动至鼠标处。

这样编程后,能够实现图像框的随意移动,但不能改变图像框的大小。每次移动的图像框大小都一样。下面我们作一些修改,实现图像框大小的随意改变。

在 Form1 的 declarations 中加入:

```
Dim x1 As Single
Dim x2 As Single
Dim y1 As Single
Dim y2 As Single
```

在 picture1 中加入:

```
Private Sub Picture1-MouseDown (Button As Integer,
Shift As Integer, X As Single, Y As Single)
    If Button = 2 Then
        x1 = X
        y1 = Y
    End If
End Sub
```

将 Picture1-MouseUp 中的程序修改为:

```
Private Sub Picture1-MouseUp (Button As Integer, Shift As
```

```
Integer, X As Single, Y As Single)
If Button = 1 Then '按下左键
    Picture2.Left = X - Picture2.Width / 2
    Picture2.Top = Y - Picture2.Height / 2
ElseIf Button = 2 Then '按下右键
    x2 = X
    y2 = Y
    On Error Resume Next
    Picture2.Top = y1
    Picture2.Left = x1
    Picture2.Height = y2 - y1
    Picture2.Width = y2 - y1 '将 picture2 的宽和高设
    为一样,得到一个正方形
End If
End Sub
```

作如上修改后,在 picture1 中任意点单击鼠标左键, picture2 都会移至该处。想修改 picture2 的大小时,按下鼠标右键,拖动至合适大小后松开,即改变了 picture2 的大小。

由于人眼对闪烁的物体更加敏感,在使用中常常希望 picture2 能够闪烁,以引起用户的注意。这一效果可以通过改变 picture box 的 visible 属性来获得。

首先将 picture2 的 BackColor 设置为红色。然后在 Timer1 中加入:

```
Private Sub Timer1-Timer( )
If Picture2.Visible = 0 Then
    Picture2.Visible = 1
Else
    Picture2.Visible = 0
End If
End Sub
```

运行该程序, picture2 在可见与不可见间交替变换。这样就获得了一个闪烁的红色正方形。

2. 用 shape 实现

在上面的方法中,虽然实现了图像框的移动,修改大小和闪烁。但当鼠标单击点在 picture2 中时,不一定能实现图像框的移动,有时需要单击多次才能移动 picture2。这是因为若单击时 picture2 恰好处于 visible = 1 状态,VB 认为单击点是在 picture2 上而不是在 picture1 上,此时不能实现 picture2 的移动。只有当单击时 picture2 的 visible 属性恰为 0,才能实现 picture2 的移动。而且

picture box 对象不能设置成透明,这样显示时无法维持一个稳定的边框,显示效果不理想。

解决这一问题的方法是把 picture1 中的 picture2 换成 shape 对象。shape 对象不会覆盖下面的 picture box 对象,即使 shape 显示在 picture box 对象上;单击其内部,VB 仍认为单击点是在 picture box 对象上。另外 shape 对象的外形种类多,其边界的颜色、宽度等属性都可以设置,能够获得比 picture box 更好的视觉效果。

具体做法是将上面例子中 picture1 内的 picture2 删除,在其中画入一 shape 对象 Shape1。设置 Shape1 的 BackColor 属性为红色, Border - Style 属性为 1, Border-Width 属性为 2,颜色为黄色(或其他颜色),如图 1 所示。

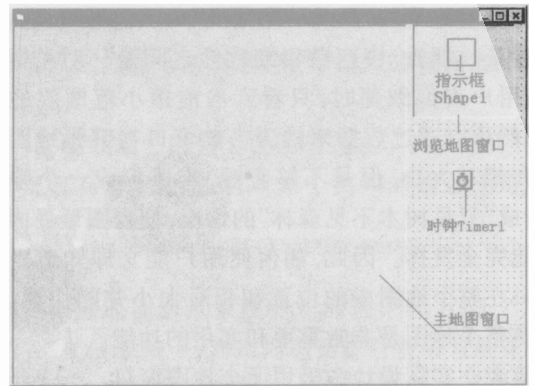


图 1

该方法程序与上例类似,但 Timer1 中的程序如下:

```
Private Sub Timer1-Timer( )
If Shape1.BackStyle = 0 Then
    Shape1.BackStyle = 1
Else
    Shape1.BackStyle = 0
End If
End Sub
```

经过修改后的程序运行会得到一个闪烁的红框,其边界为黄色。单击 picture1 中任意一点,包括 Shape1 中, Shape1 都会移至该处。

有兴趣的读者可以试验一下,比较两种方法的视觉效果哪个更好。

3. 主地图窗口与浏览图窗口的交互访问显示

上面介绍的方法只是实现了浏览图(本例中是 pic-

ture1)中图像框的移动和修改大小,并没有和主地图建立联系。此时主地图中显示的部分究竟在整个地图中处于什么样的位置,以及大小如何在浏览图中没有显示出来。

要想做到浏览图与主地图的对应,必须进一步编程。这里仅给出一个例子,其基本思想是由时钟(Timer1)控制对浏览图进行查询,当浏览图(picture1)中的指示框Shapel发生改变(包括移动位置,改变大小)时,将改变后的位置和大小信息传递给主地图,实现由浏览图到主地图的查询。反之,当用户改变主地图的大小或位置时,要将这一改变传递给浏览图,使得其中的Shapel随着改变。实现由主地图到浏览图的查询。

具体做法是先将picture1中加载一幅整个地图的位图图片。注意该位图的大小一定要精确,否则影响以后使用时的定位精度。因为Shapel所对应的主地图位置实际上是通过计算在picture1中的相对位置来实现的。

然后在picture1中画上一个shape对象,名为Shapel。

程序如下:

```
Private Sub Timer1-Timer( )
Dim n As Integer
Dim zom As Single
Dim X As Single
Dim Y As Single
'返回图形区域信息
On Error Resume Next
zom = Gistool.eval("mapperinfo(" & mapid-city &
", 1)") '此处返回主地图 '范围
X = Gistool.eval("mapperinfo(" & mapid-city &
", 3)") '主地图中心 X 坐标
Y = Gistool.eval("mapperinfo(" & mapid-city &
", 4)") '主地图中心 Y 坐标
'以下将获得 shapel 闪烁效果
If Shapel.BackStyle = 0 Then
Shapel.BackStyle = 1
Else
Shapel.BackStyle = 0
End If
Shapel.Visible = True
Shapel.Top = (24 - Y - zom / 2) * 200
Shapel.Left = (X - 15 - zom / 2) * 200
Shapel.Width = zom * 200
Shapel.Height = zom * 200
```

End Sub

注:程序中的Gistool是创建的一个地图工具对象,依用户使用的GIS工具不同而不同。Private Sub Picture1-MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

```
If Button = 2 Then '按下右键
```

```
x1 = X
```

```
y1 = Y
```

```
Else
```

```
End If
```

End Sub

Private Sub Picture1-MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)

```
Dim map-x As Single
```

```
Dim map-y As Single
```

```
Dim mapid As Long '主地图窗口标识号
```

```
If Button = 1 Then '按下左键
```

```
Gistool.do "set next document parent " & mainpict.
hWnd & "style 1"
```

```
Shapel.Left = X - Shapel.Width / 2
```

```
Shapel.Top = Y - Shapel.Height / 2
```

```
map-y = 15 + (1800 - Y) / 200 '地图坐标,单
```

位公里

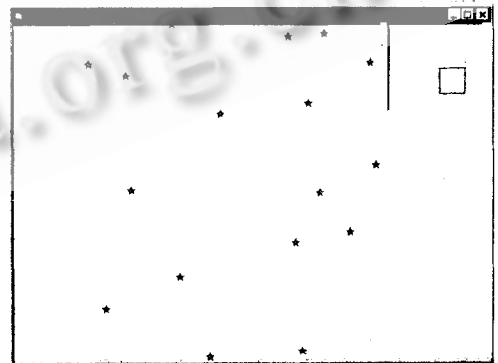


图 2

```
map-x = 15 + X / 200
```

```
Else 'button=2 按下右键
```

```
x2 = X
```

```
y2 = Y
```

```

Shape1.Left = x1
Shape1.Top = y1
Shape1.Width = Abs(x2 - x1)
Shape1.Height = Shape1.Width
map-y = 15 + (1800 - (y1 + y2) / 2) / 200 '地
图坐标,单位公里
map-x = 15 + ((x1 + x2) / 2) / 200
End If
On Error Resume Next
Gistool.do " Set Coordsys nonearth units" & Chr

```

```

$(34) & "km" & Chr$(34)
Gistool.do "set map window " & mapid & "center("
& map-x & "," & map-y & ")" & "zoom" & Shape1.
Width / 200
End Sub

```

采用上面的程序后,完全实现了我们前面提出的要求,浏览地图方便而且迅速,效果很好。用户使用后认为非常实用而有效,其效果如图 2 所示。

(来稿时间:1997 年 1 月)