用户名: 密码:
登陆 注册
加入收藏 | 设为首页 | 会员中心 | 我要投稿 | RSS



《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择器的实现2

时间：2009-04-19 01:54:07 来源：3SDN.Net 作者：3SDN原创 点击量： 2281

◆ 版权声明：
《ArcGIS Engine+C#实例开发教程》为3SDN（http://www.3sdn.net）原创教程，作者闲云野鹤，版权所有。禁止商业用途转载（如需请联系作者），非商业用途转载请注明出处并完整保留本声明。

◆ 读者对象：
使用C#开发ArcGIS Engine（以下简称AE）的初学者。

◆ 预备知识：
了解AE基本体系，了解C#基本语法，了解VS2005的基本使用方法。

◆ 预期学习效果：
进一步理解AE的体系结构与开发方法，掌握基本的GIS桌面应用程序的开发。

在第七讲 图层符号选择器的实现的第一阶段中，我们完成了符号选择器窗体的创建与调用。在第二阶段中，我们继续完成符号参数的调整与“更多符号”的加载。

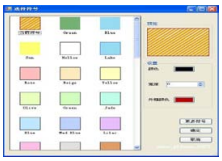
2.6 符号参数调整

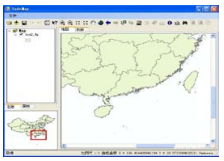
在地图整饰中，符号参数的调整是必须的功能。下面我们将实现符号颜色、外框颜色、线宽、角度等参数的调整。

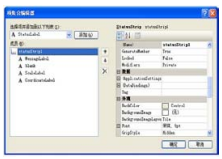
（1） 添加SymbologyControl的OnItemSelected事件，此事件在鼠标选中符号时触发，此时显示出选定符号的初始参数，事件响应函数代码如下：

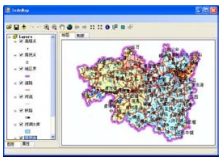
```
/// <summary>
/// 选中符号时触发的事件
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void axSymbologyControl_OnItemSelected(object sender, ESRI.ArcGIS.Controls.IS
        ymbologyControlEvents_OnItemSelectedEvent e)
{
pStyleGalleryItem = (IStyleGalleryItem)e.styleGalleryItem;
Color color;
switch (this.axSymbologyControl.StyleClass)
{
    //点符号
  case esriSymbologyStyleClass.esriStyleClassMarkerSymbols:
    color = this.ConvertIRgbColorToColor(((IMarkerSymbol)pStyleGalleryItem.Item).Colo
```

推荐资讯


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实

相关文章

- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》教程Bug及优化方案1
- 《ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加
- 《ArcGIS Engine+C#实例开发教程》第五讲 鹰眼的实现
- 《ArcGIS Engine+C#实例开发教程》第四讲 状态栏信息的
- 《ArcGIS Engine+C#实例开发教程》第三讲 MapControl与
- 《ArcGIS Engine+C#实例开发教程》第二讲 菜单的添加及
- 《ArcGIS Engine+C#实例开发教程》第一讲 桌面GIS应用
- AO开发客户化组件（DLL+ArcMap）入门实例
- 解决AO开发中的error C2011: 'enum' type redefini

栏目更新

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发

```
            r as IRgbColor);
        //设置点符号角度和大小初始值
        this.nudAngle.Value = (decimal)((IMarkerSymbol)this.pStyleGalleryItem.Item).Angle;
        this.nudSize.Value = (decimal)((IMarkerSymbol)this.pStyleGalleryItem.Item).Size;
        break;
        //线符号
      case esriSymbologyStyleClass.esriStyleClassLineSymbols:
        color = this.ConvertIRgbColorToColor(((ILineSymbol)pStyleGalleryItem.Item).Color a
          s IRgbColor);
        //设置线宽初始值
        this.nudWidth.Value = (decimal)((ILineSymbol)this.pStyleGalleryItem.Item).Width;
        break;
        //面符号
      case esriSymbologyStyleClass.esriStyleClassFillSymbols:
        color = this.ConvertIRgbColorToColor(((IFillSymbol)pStyleGalleryItem.Item).Color as
          IRgbColor);
        this.btnOutlineColor.BackColor = this.ConvertIRgbColorToColor(((IFillSymbol)pStyle
          GalleryItem.Item).Outline.Color as IRgbColor);
        //设置外框线宽度初始值
        this.nudWidth.Value = (decimal)((IFillSymbol)this.pStyleGalleryItem.Item).Outline.Wid
          th;
        break;
      default:
        color = Color.Black;
        break;
    }
    //设置按钮背景色
    this.btnColor.BackColor = color;
    //预览符号
    this.PreviewImage();
  }
```

（2） 调整点符号的大小

添加nudSize控件的ValueChanged事件，即在控件的值改变时响应此事件，然后重新设置点符号的大小。代码如下：

```
  /// <summary>
    /// 调整符号大小-点符号
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void nudSize_ValueChanged(object sender, EventArgs e)
    {
      ((IMarkerSymbol)this.pStyleGalleryItem.Item).Size = (double)this.nudSize.Value;
      this.PreviewImage();
    }
```

（3） 调整点符号的角度

添加nudAngle控件的ValueChanged事件，以重新设置点符号的角度。代码如下：

```
  /// <summary>
    /// 调整符号角度-点符号
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void nudAngle_ValueChanged(object sender, EventArgs e)
    {
      ((IMarkerSymbol)this.pStyleGalleryItem.Item).Angle = (double)this.nudAngle.Value;
      this.PreviewImage();
    }
```

（4） 调整线符号和面符号的线宽

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- （连载三）VC、OpenGL、ArcG
- （连载二）VC、OpenGL、ArcG
- （连载一）VC、OpenGL、ArcG

栏目热门

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- ArcGIS Engine中的数据访问
- （连载一）VC、OpenGL、ArcG
- ArcGIS Engine的license部署

添加nudWidth控件的ValueChanged事件，以重新设置线符号的线宽和面符号的外框线的线宽。代码如下：

```
        /// <summary>
        /// 调整符号宽度-限于线符号和面符号
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void nudWidth_ValueChanged(object sender, EventArgs e)
        {
            switch (this.axSymbologyControl.StyleClass)
            {
                case esriSymbologyStyleClass.esriStyleClassLineSymbols:
                    ((ILineSymbol)this.pStyleGalleryItem.Item).Width = Convert.ToDouble(this.nud
Width.Value);
                    break;
                case esriSymbologyStyleClass.esriStyleClassFillSymbols:
                    //取得面符号的轮廓线符号
                    ILineSymbol pLineSymbol = ((IFillSymbol)this.pStyleGalleryItem.Item).Outline;
                    pLineSymbol.Width = Convert.ToDouble(this.nudWidth.Value);
                    ((IFillSymbol)this.pStyleGalleryItem.Item).Outline = pLineSymbol;
                    break;
            }
            this.PreviewImage();
        }
```

（5） 颜色转换

在ArcGIS Engine中，颜色由IRgbColor接口实现，而在.NET框架中，颜色则由Color结构表示。故在调整颜色参数之前，我们必须完成以上两种不同颜色表示方式的转换。关于这两种颜色结构的具体信息，请大家自行查阅相关资料。下面添加两个颜色转换函数。

ArcGIS Engine中的IRgbColor接口转换至.NET中的Color结构的函数：

```
        /// <summary>
        /// 将ArcGIS Engine中的IRgbColor接口转换至.NET中的Color结构
        /// </summary>
        /// <param name="pRgbColor">IRgbColor</param>
        /// <returns>.NET中的System.Drawing.Color结构表示ARGB颜色</returns>
        public Color ConvertIRgbColorToColor(IRgbColor pRgbColor)
        {
            return ColorTranslator.FromOle(pRgbColor.RGB);
        }
```

.NET中的Color结构转换至于ArcGIS Engine中的IColor接口的函数：

```
        /// <summary>
        /// 将.NET中的Color结构转换至于ArcGIS Engine中的IColor接口
        /// </summary>
        /// <param name="color">.NET中的System.Drawing.Color结构表示ARGB颜色</param
>
        /// <returns>IColor</returns>
        public IColor ConvertColorToIColor(Color color)
        {
            IColor pColor = new RgbColorClass();
            pColor.RGB = color.B * 65536 + color.G * 256 + color.R;
            return pColor;
        }
```

（6） 调整所有符号的颜色

选择颜色时，我们调用.NET的颜色对话框ColorDialog，选定颜色后，修改颜色按钮的背景色为选定的颜色，以方便预览。双击btnColor按钮，添加如下代码：

```
        /// <summary>
        /// 颜色按钮
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnColor_Click(object sender, EventArgs e)
        {
            //调用系统颜色对话框
            if (this.colorDialog.ShowDialog() == DialogResult.OK)
            {
                //将颜色按钮的背景颜色设置为用户选定的颜色
                this.btnColor.BackColor = this.colorDialog.Color;
                //设置符号颜色为用户选定的颜色
                switch (this.axSymbologyControl.StyleClass)
                {
                    //点符号
                    case esriSymbologyStyleClass.esriStyleClassMarkerSymbols:
                        ((IMarkerSymbol)this.pStyleGalleryItem.Item).Color = this.ConvertColorToIColor(this.colorDialog.Color);
                        break;
                    //线符号
                    case esriSymbologyStyleClass.esriStyleClassLineSymbols:
                        ((ILineSymbol)this.pStyleGalleryItem.Item).Color = this.ConvertColorToIColor(this.colorDialog.Color);
                        break;
                    //面符号
                    case esriSymbologyStyleClass.esriStyleClassFillSymbols:
                        ((IFillSymbol)this.pStyleGalleryItem.Item).Color = this.ConvertColorToIColor(this.colorDialog.Color);
                        break;
                }
                //更新符号预览
                this.PreviewImage();
            }
        }
```

（7） 调整面符号的外框线颜色

同上一步类似，双击btnOutlineColor按钮，添加如下代码：

```
        /// <summary>
        /// 外框颜色按钮
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnOutlineColor_Click(object sender, EventArgs e)
        {
            if (this.colorDialog.ShowDialog() == DialogResult.OK)
            {
                //取得面符号中的外框线符号
                ILineSymbol pLineSymbol = ((IFillSymbol)this.pStyleGalleryItem.Item).Outline;
                //设置外框线颜色
                pLineSymbol.Color = this.ConvertColorToIColor(this.colorDialog.Color);
                //重新设置面符号中的外框线符号
                ((IFillSymbol)this.pStyleGalleryItem.Item).Outline = pLineSymbol;
                //设置按钮背景颜色
                this.btnOutlineColor.BackColor = this.colorDialog.Color;
                //更新符号预览
                this.PreviewImage();
            }
        }
```

至此，你可以编译运行程序，看看效果如何，是不是感觉很不错了？我们已经能够修改

符号的参数，自定义符号了。

但是，SymbologyControl默认加载的是ESRI.ServerStyle文件的样式，用过ArcMap的你可能已经注意到，ArcMap中的Symbol Selector有一个“More Symbols”按钮，可以加载其它的符号和ServerStyle文件。3sdnMap当然“一个都不能少”。

1/2 1 2 下一页 尾页

6 来顶一下 返回首页

发表评论 共有 3条评论

用户名: 密码:

验证码: 3028 匿名发表

提交留言

站内搜索: 文章 搜索 高级搜索

网站首页 | 关于我们 | 服务条款 | 广告服务 | 联系我们 | 网站地图 | 免责声明 | 返回顶部

Copyright ? 2008-2009 3sdn.net, All rights reserved.

3sdnnet@Gmail.com

用户名: 密码:
登陆 注册
加入收藏 | 设为首页 | 会员中心 | 我要投稿 | RSS



《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择器的实现2

时间：2009-04-19 01:54:07 来源：3SDN.Net 作者：3SDN原创 点击量： 2283

2.7 添加更多符号菜单

还记得我们在开始的时候添加了ContextMenuStrip控件吗？现在它终于派上用场了。我们要实现的功能是：单击“更多符号”弹出菜单（ContextMenu），菜单中列出了ArcGIS自带的其它符号，勾选相应的菜单项就可以在SymbologyControl中增加相应的符号。在菜单的最后一项是“添加符号”，选择这一项时，将弹出打开文件对话框，我们可以由此选择其它的ServerStyle文件，以加载更多的符号。

（1） 定义全局变量

在SymbolSelectorFrm中定义如下全局变量，用于判断菜单是否已经初始化。

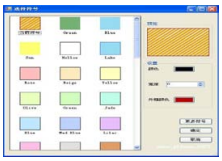
```
//菜单是否已经初始化标志
bool contextMenuMoreSymbolInitiated = false;
```

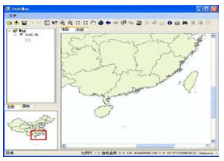
（2） 双击“更多符号”按钮，添加Click事件。

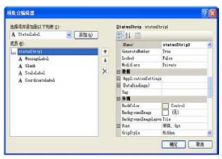
在此事件响应函数中，我们要完成ServerStyle文件的读取，将其文件名作为菜单项名称生成菜单并显示菜单。代码如下：

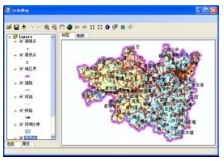
```
        /// <summary>
        /// “更多符号”按下时触发的事件
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnMoreSymbols_Click(object sender, EventArgs e)
        {
            if (this.contextMenuMoreSymbolInitiated == false)
            {
                string sInstall = ReadRegistry("SOFTWARE\\ESRI\\CoreRuntime");
                string path = System.IO.Path.Combine(sInstall, "Styles");
                //取得菜单项数量
                string[] styleNames = System.IO.Directory.GetFiles(path, "*.ServerStyle");
                ToolStripMenuItem[] symbolContextMenuItem = new ToolStripMenuItem[styleNames.L
ength + 1];
                //循环添加其它符号菜单项到菜单
                for (int i = 0; i < styleNames.Length; i++)
                {
                    symbolContextMenuItem[i] = new ToolStripMenuItem();
```

推荐资讯


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实

相关文章

- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》教程Bug及优化方案1
- 《ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加
- 《ArcGIS Engine+C#实例开发教程》第五讲 鹰眼的实现
- 《ArcGIS Engine+C#实例开发教程》第四讲 状态栏信息的
- 《ArcGIS Engine+C#实例开发教程》第三讲 MapControl与
- 《ArcGIS Engine+C#实例开发教程》第二讲 菜单的添加及
- 《ArcGIS Engine+C#实例开发教程》第一讲 桌面GIS应用
- AO开发客户化组件（DLL+ArcMap）入门实例
- 解决AO开发中的error C2011: 'enum' type redefini

栏目更新

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发

```
                symbolContextMenuItem[i].CheckOnClick = true;
                symbolContextMenuItem[i].Text = System.IO.Path.GetFileNameWithoutExtension(sty
leNames[i]);
                if (symbolContextMenuItem[i].Text == "ESRI")
                {
                    symbolContextMenuItem[i].Checked = true;
                }
                symbolContextMenuItem[i].Name = styleNames[i];
            }
            //添加“更多符号”菜单项到菜单最后一项
            symbolContextMenuItem[styleNames.Length] = new ToolStripMenuItem();
            symbolContextMenuItem[styleNames.Length].Text = "添加符号";
            symbolContextMenuItem[styleNames.Length].Name = "AddMoreSymbol";

            //添加所有的菜单项到菜单
            this.contextMenuStripMoreSymbol.Items.AddRange(symbolContextMenuItem);
            this.contextMenuMoreSymbolInitiated = true;
        }
        //显示菜单
        this.contextMenuStripMoreSymbol.Show(this.btnMoreSymbols.Location);
    }
```

（3） 添加contextMenuStripMoreSymbol控件的ItemClicked事件。

当单击某一菜单项时响应ItemClicked事件，将选中的ServerStyle文件导入到SymbologyControl中并刷新。当用户单击“添加符号”菜单项时，弹出打开文件对话框，供用户选择其它的ServerStyle文件。代码如下：

```
        /// <summary>
        /// “更多符号”按钮弹出的菜单项单击事件
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void contextMenuStripMoreSymbol_ItemClicked(object sender, ToolStripItemC
lickedEventArgs e)
        {
            ToolStripMenuItem pToolStripMenuItem = (ToolStripMenuItem)e.ClickedItem;
            //如果单击的是“添加符号”
            if (pToolStripMenuItem.Name == "AddMoreSymbol")
            {
                //弹出打开文件对话框
                if (this.openFileDialog.ShowDialog() == DialogResult.OK)
                {
                    //导入style file到SymbologyControl
                    this.axSymbologyControl.LoadStyleFile(this.openFileDialog.FileName);
                    //刷新axSymbologyControl控件
                    this.axSymbologyControl.Refresh();
                }
            }
            else//如果是其它选项
            {
                if (pToolStripMenuItem.Checked == false)
                {
                    this.axSymbologyControl.LoadStyleFile(pToolStripMenuItem.Name);
                    this.axSymbologyControl.Refresh();
                }
                else
                {
                    this.axSymbologyControl.RemoveFile(pToolStripMenuItem.Name);
                    this.axSymbologyControl.Refresh();
                }
```


- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- （连载三）VC、OpenGL、ArcG
- （连载二）VC、OpenGL、ArcG
- （连载一）VC、OpenGL、ArcG

栏目热门

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- ArcGIS Engine中的数据访问
- （连载一）VC、OpenGL、ArcG
- ArcGIS Engine的license部署

```
            }
        }
```

2.8 编译运行

相信你已经盼这一步很久了吧，按照惯例，按下F5吧！大功造成。

以上代码在AE9.2+VS2005+XP中编译通过。

致谢：

在本讲的实现过程中，参考了GIS-Lin的Lin-GIS，在此特别感谢！

下一讲中，我将给大家带来的是属性数据表的查询显示，敬请关注3SDN.Net！

关于《ArcGIS Engine+C#实例开发教程》的几点说明：

本教程发表以来，得到了许多网友的支持，在此感谢他们，如果本教程能够给你带来一点一滴的帮助，我的目的也就达到了。有部分网友反映本教程是单纯的贴代码，而我自认为贴代码才是本教程的特色。本教程定位于入门教程。试想，对于初学者，如果一味的讲解AE原理，有几个能够独立写出实现代码？如果你能，那说明本教程并不适合你！以我多年的编程体会，我认为参考别人的代码，进行实例学习，是一种很好的学习编程方式。因此本教程详述了每一操作步骤的要点，提供了完全可运行的代码。而具体的原理方法，则以注释的方式进行解释。

如果，你只是单纯Ctrl+C And Ctrl+V，这是我不愿意看到的。我希望你在学习的过程中，能够配合注释和OMD理解其中的大部分代码，进而初步了解接着深入了解AE的体系架构与开发流程和方法。坚持，直到烂熟于心！有朝一日，你就是一看原理就能写出代码的高手了！

6 来顶一下 返回首页

发表评论 共有 3条评论

用户名： 密码：

验证码： 1552 匿名发表

提交留言

站内搜索： 文章 搜索 高级搜索

网站首页 | 关于我们 | 服务条款 | 广告服务 | 联系我们 | 网站地图 | 免责声明 | 返回顶部

Copyright ? 2008-2009 3sdn.net, All rights reserved.

3sdnnet@Gmail.com

