

实验一

【实验名称】地图控件 MapView 的使用

【实验目的】了解 ArcGIS Runtime 开发桌面程序过程，熟悉.net 开发环境，初步掌握地图显示与图层加载的方法。

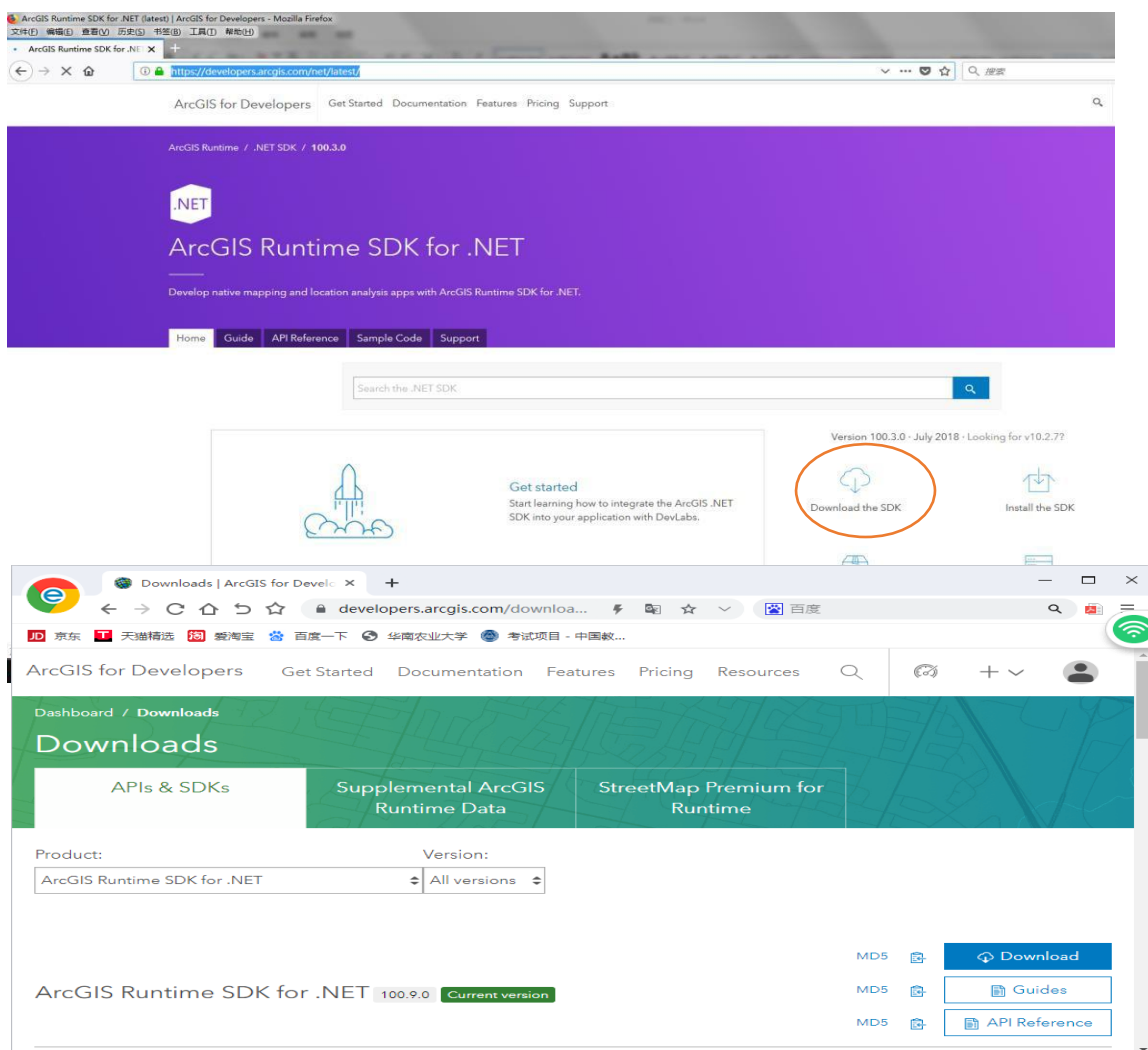
【实验内容】1. ArcGIS Runtime SDK 安装与配置；2. 创建模板项目；3. ArcGIS Runtime 应用程序开发。

【实验准备】1. 软件环境：Visual studio 2017（或更高版本） + ArcGIS Runtime 100.6（或更高版本）；2. 数据：shp 数据和 Geodatabase 数据。

【实验步骤】

一、ArcGIS Runtime SDK 安装与配置

1. 访问 <https://developers.arcgis.com/net/latest/> 登录网站，下载 ArcGIS Runtime SDK 安装文件。

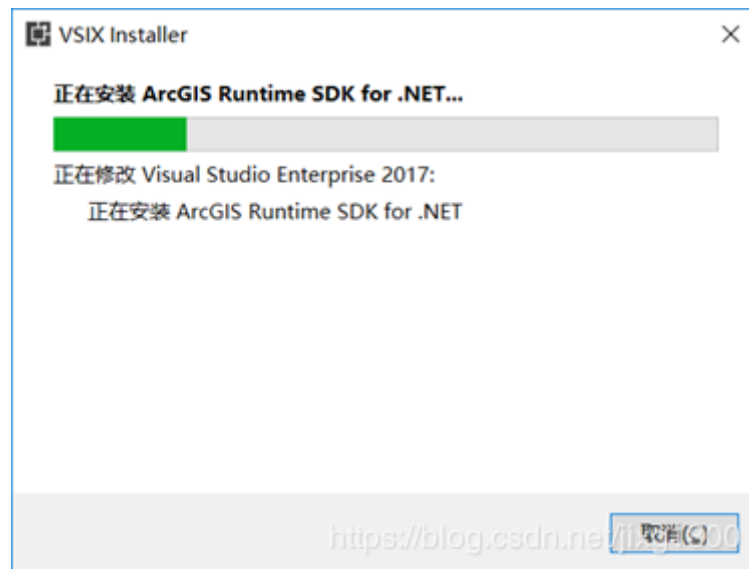


2. 安装 Visual studio 2017



选择图中红色方框即可，安装对应内容

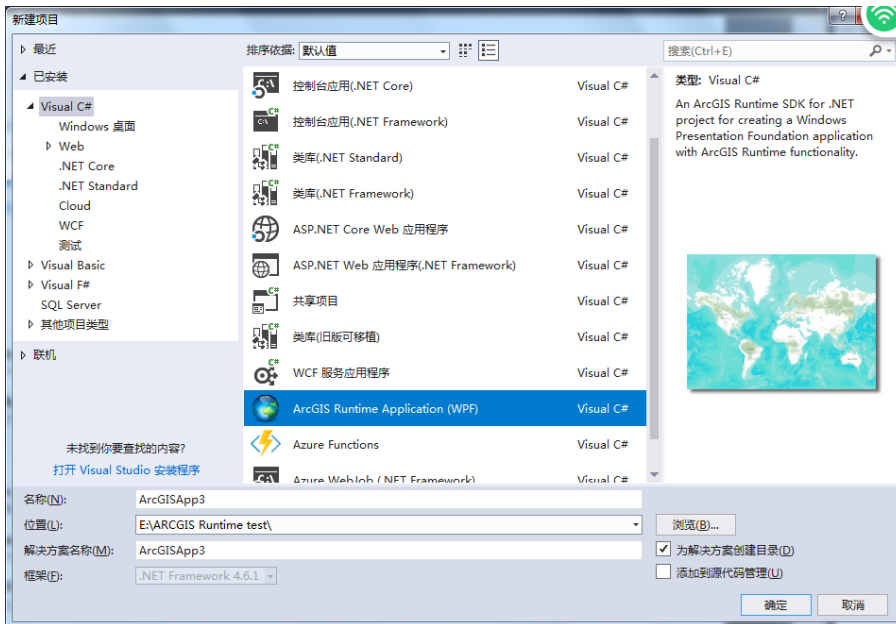
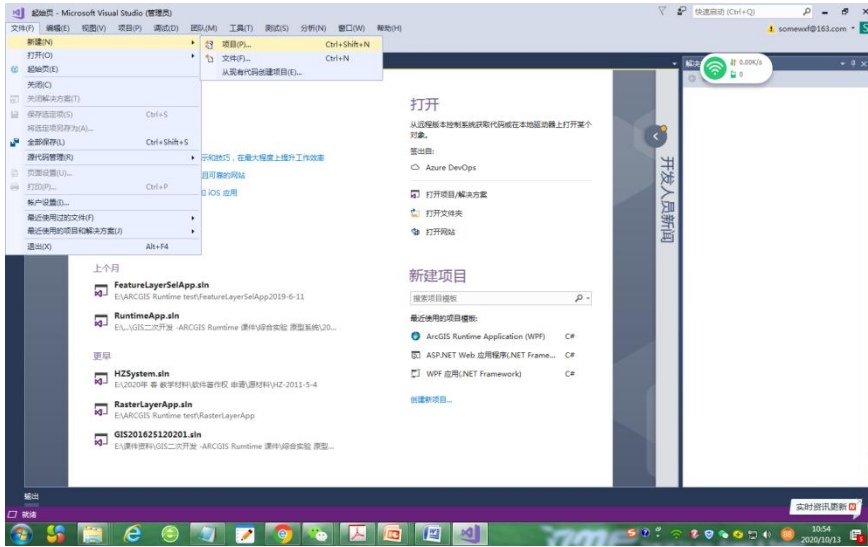
3. 安装 ArcGIS Runtime SDK .net



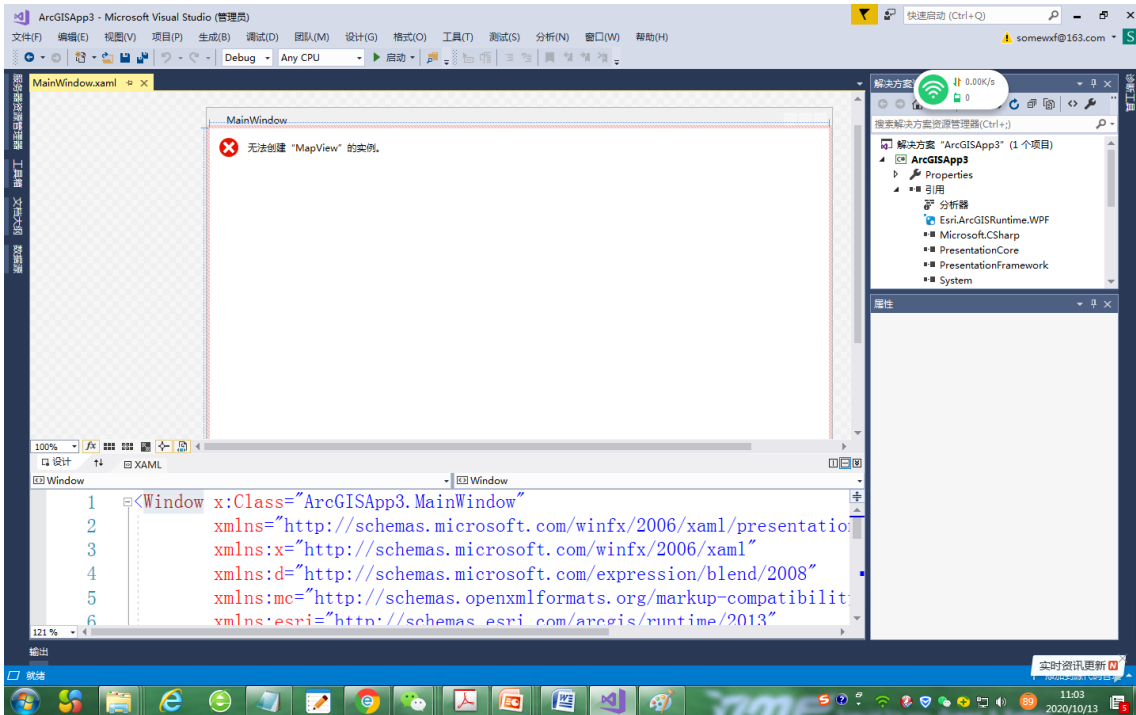
安装完成即可。

二、创建 ARCGIS Runtime Application WPF 模板项目

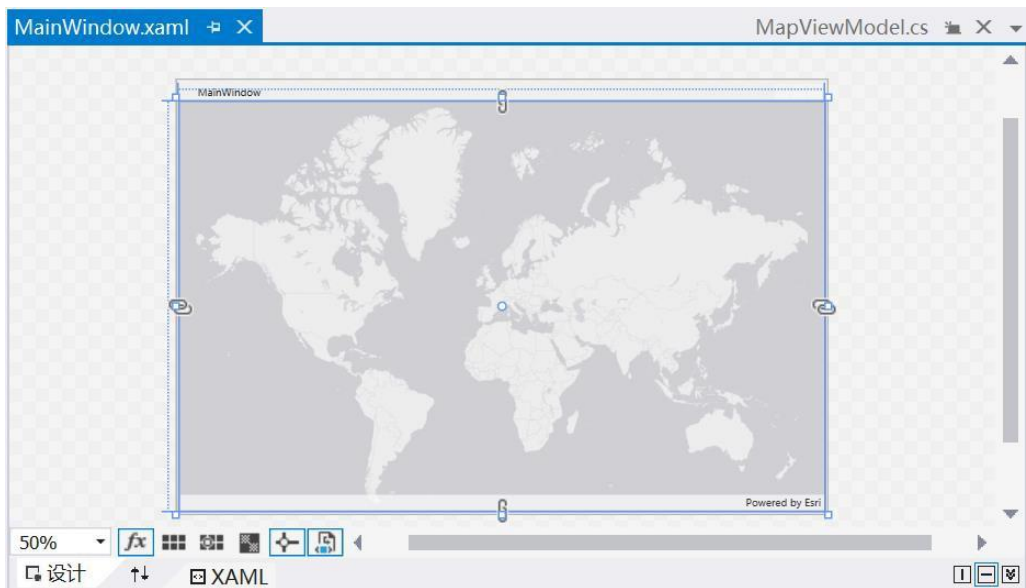
1. 启动 Visual studio 2017，新建一个项目，选择 WPF 模板



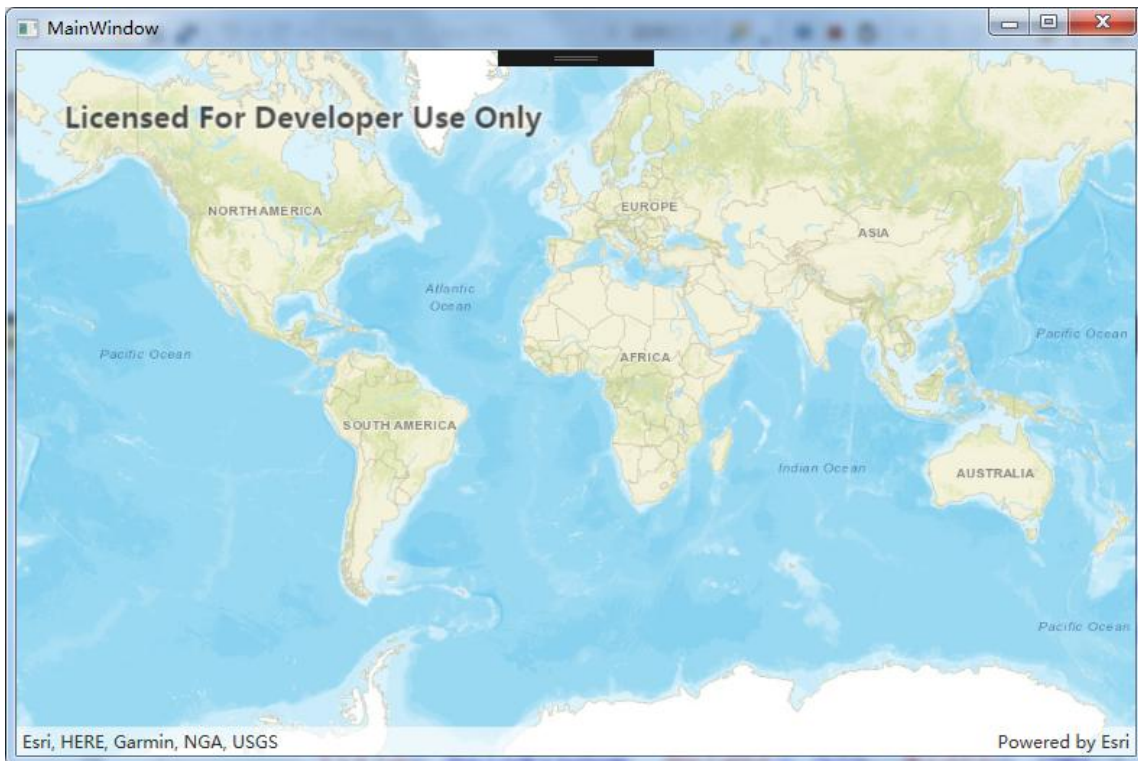
2 新生成的项目界面如下



2.1 选择“生成”->“生成***”命令，生成项目，可以看到地图控件的界面



2.2 点击“启动”运行程序，程序界面如下



(注意,如果依然无法正常显示 MapView 控件,关闭 Visual Studio 2017,以管理员身份重新启动 VS2017,打开之前创建的项目,可以正常显示 MapView 控件)

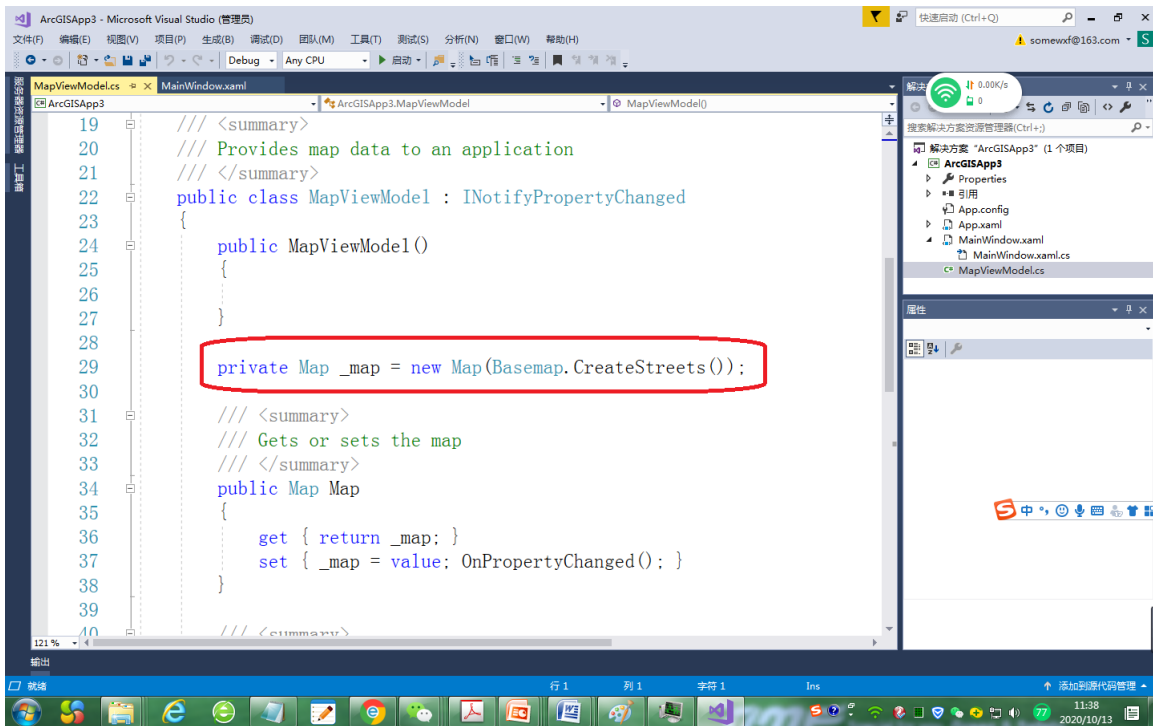
3、查看 MainWindow.xaml 文件代码,理解各标签含义。

```

<Window x:Class="RuntimeExp2.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:esri="http://schemas.esri.com/arcgis/runtime/2013"
    xmlns:local="clr-namespace:RuntimeExp2"
    mc:Ignorable="d"
    Title="MainWindow" Height="525" Width="790">
    <Window.Resources>
        <local:MapViewModel x:Key="MapViewModel" />
    </Window.Resources>
    <Grid>
        <esri:MapView Map="{Binding Map, Source={StaticResource MapViewModel}}" />
    </Grid>
</Window>

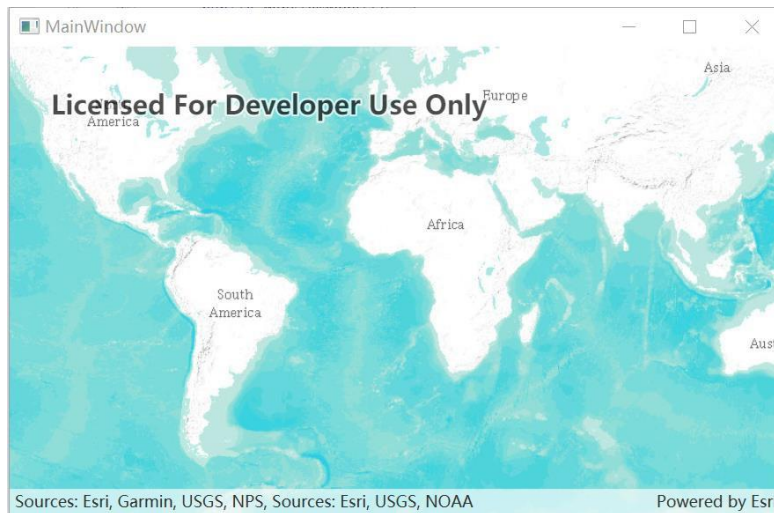
```

4、查看 MapViewModel.cs 文件代码，了解其作用。



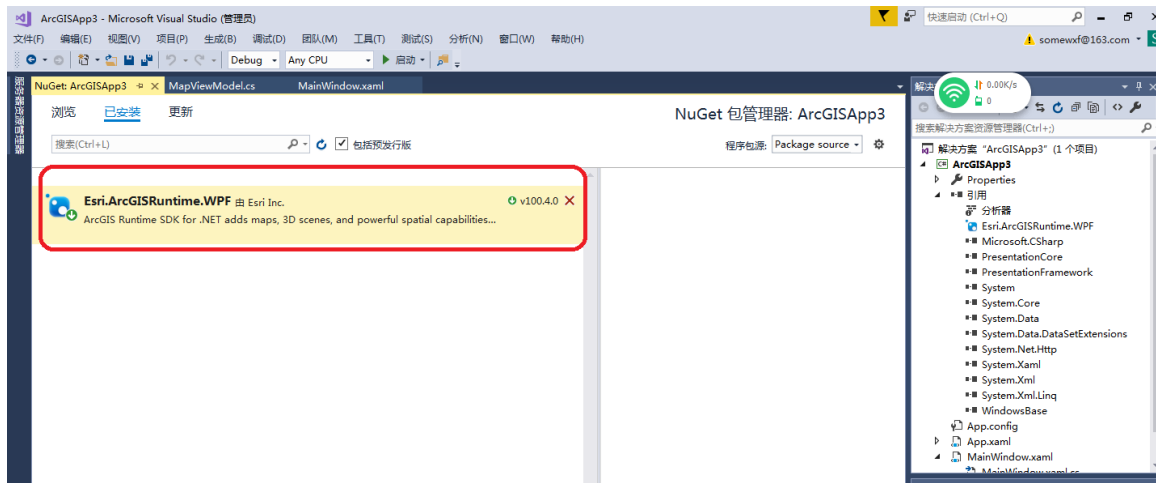
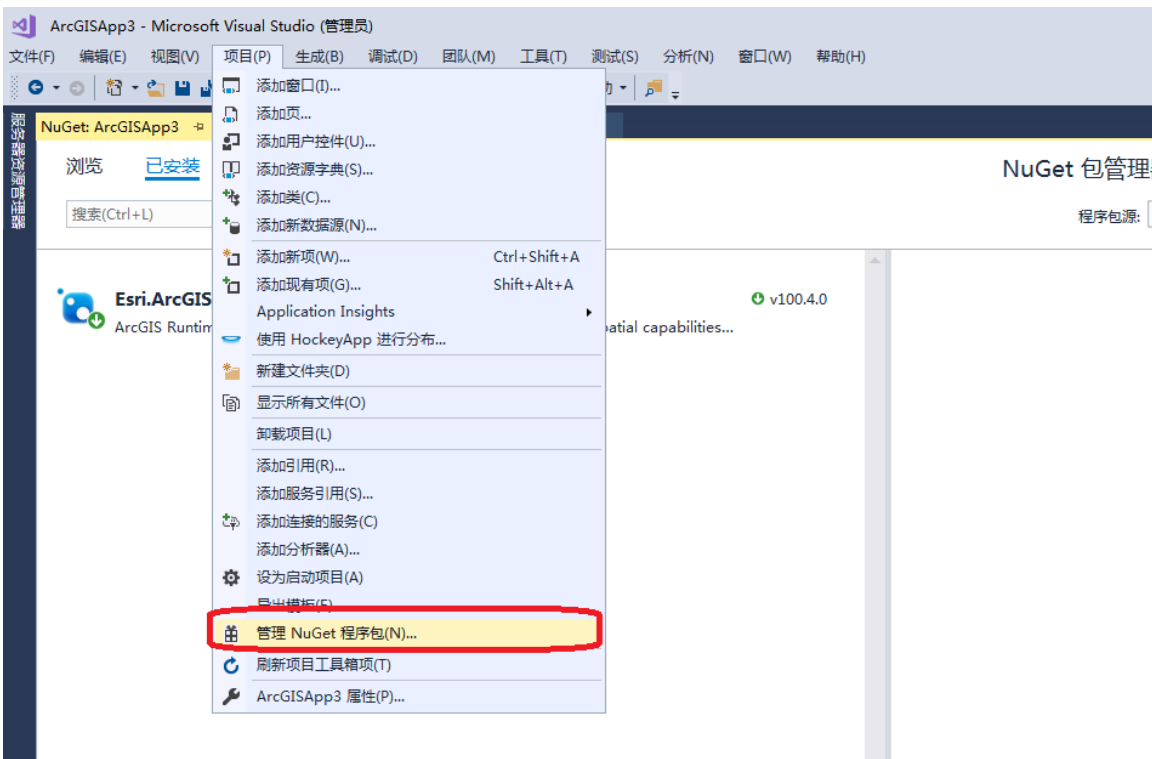
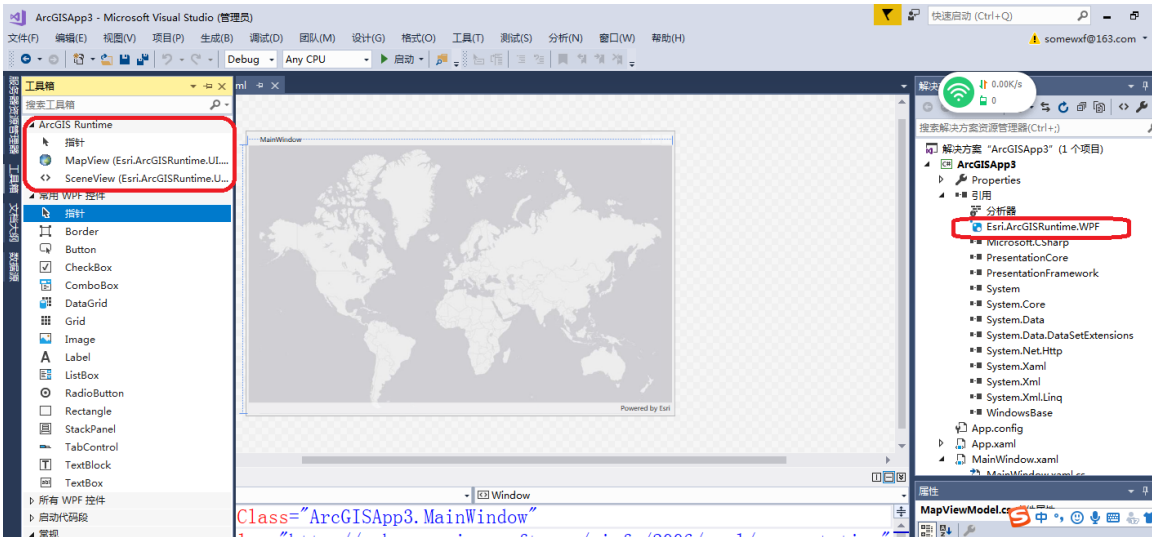
可更改以上代码，如改为 CreateTopographic()。

5、试着改变地图底图并运行。



6、ArcGIS runtime 在 Visual studio 中的体现

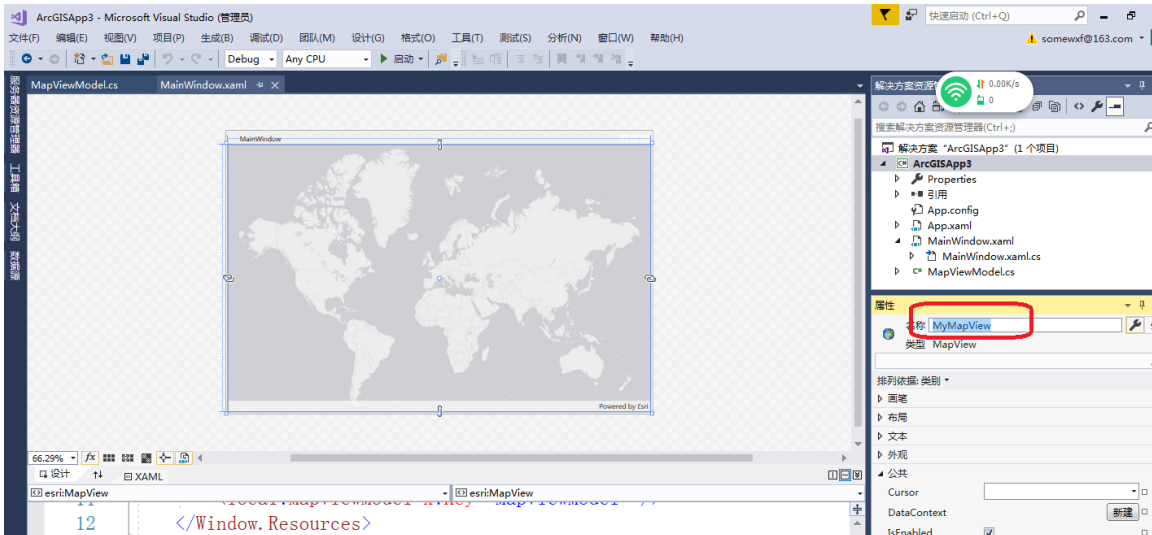
如果 ArcGIS runtime 安装正常，在以下红色标记处可正常看到相关内容，若无法正常显示，在 Nuget 管理中卸载 ArcGIS runtime，再次加载安装 ArcGIS runtime。



三、ArcGIS Runtime 应用程序开发

3.1 通用环境设置

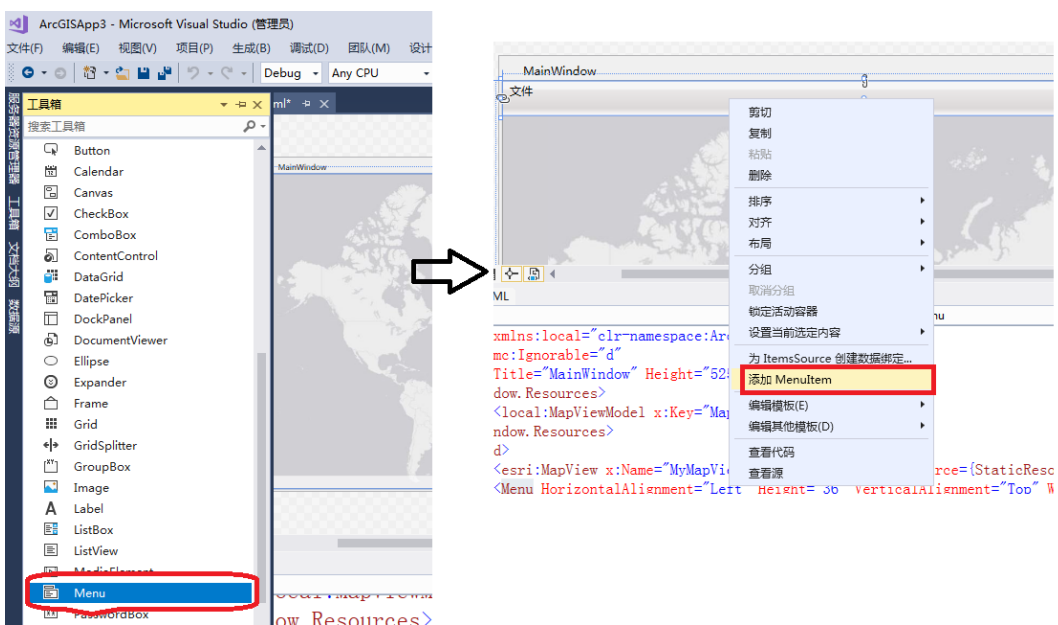
按照以上步骤创建 ARCGIS Runtime Application WPF 模板项目，设置 MapView 控件的名称为 **MyMapView**，以便通过该名称 MyMapView，在编写代码时调用该控件的属性和方法。

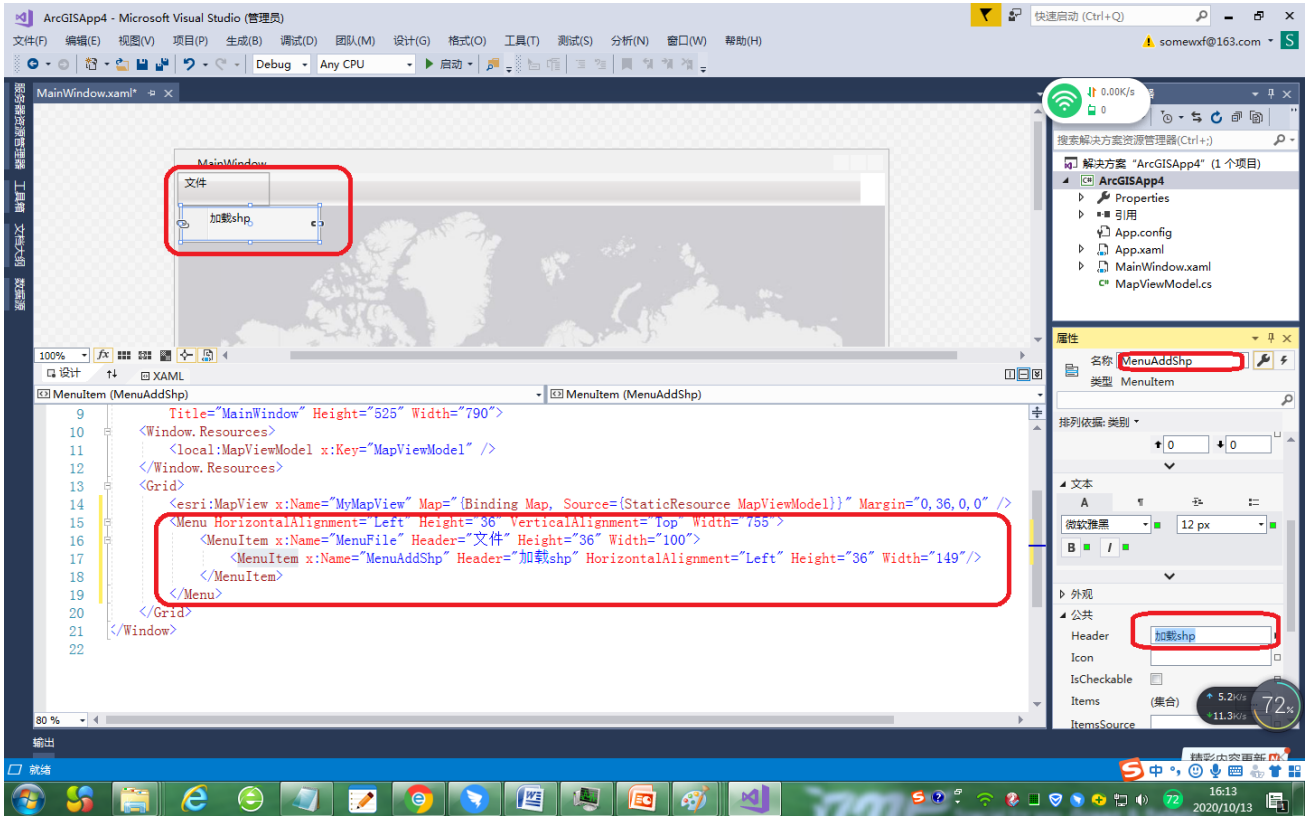


3.2 加载 shp 数据

3.2.1 界面设计

添加“文件”主菜单、“加载 shp”子菜单

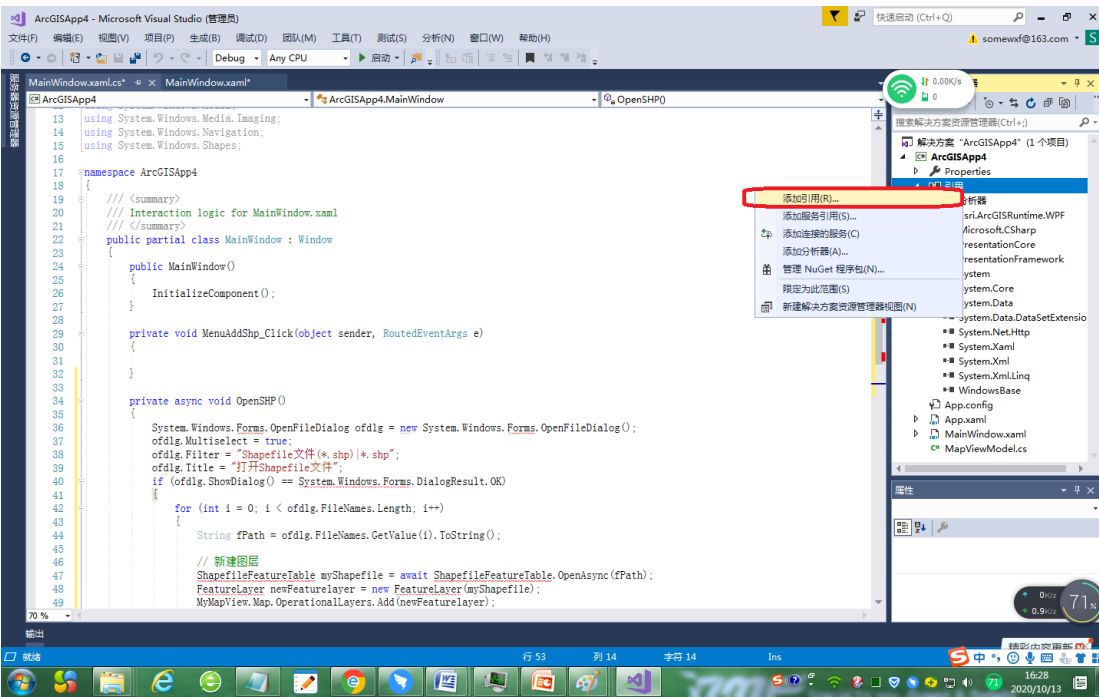


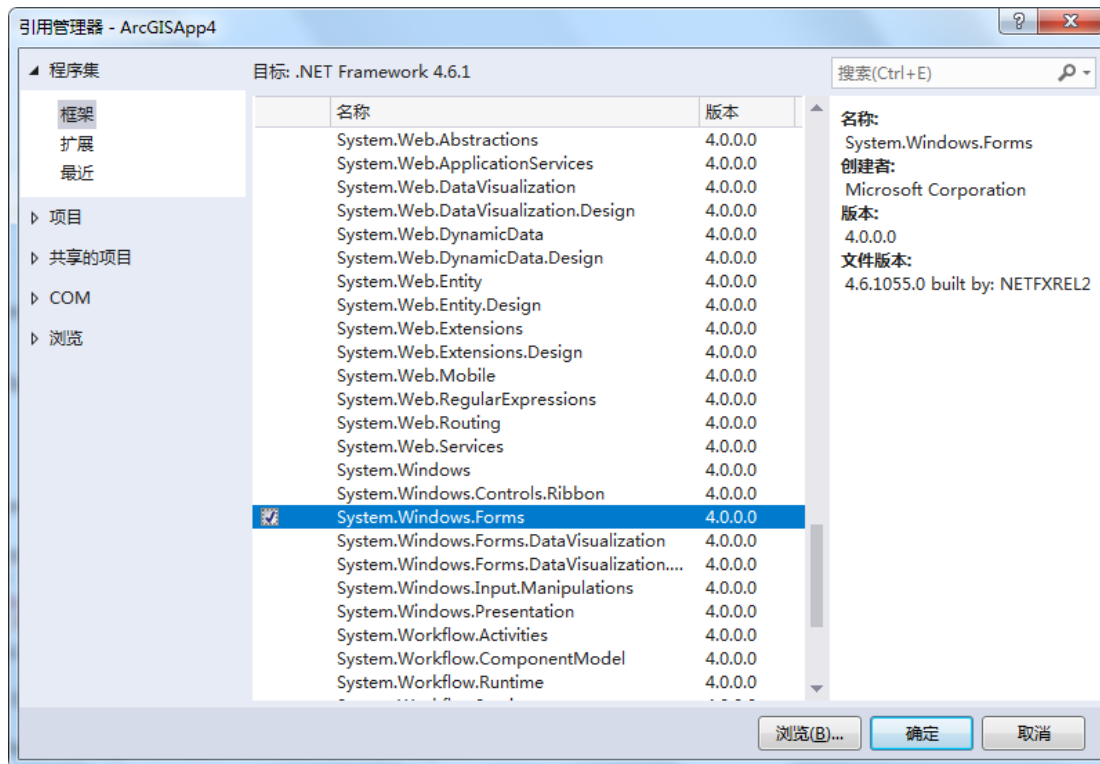


双击添加“加载 shp”的 Click 事件。

3.2.2 添加引用

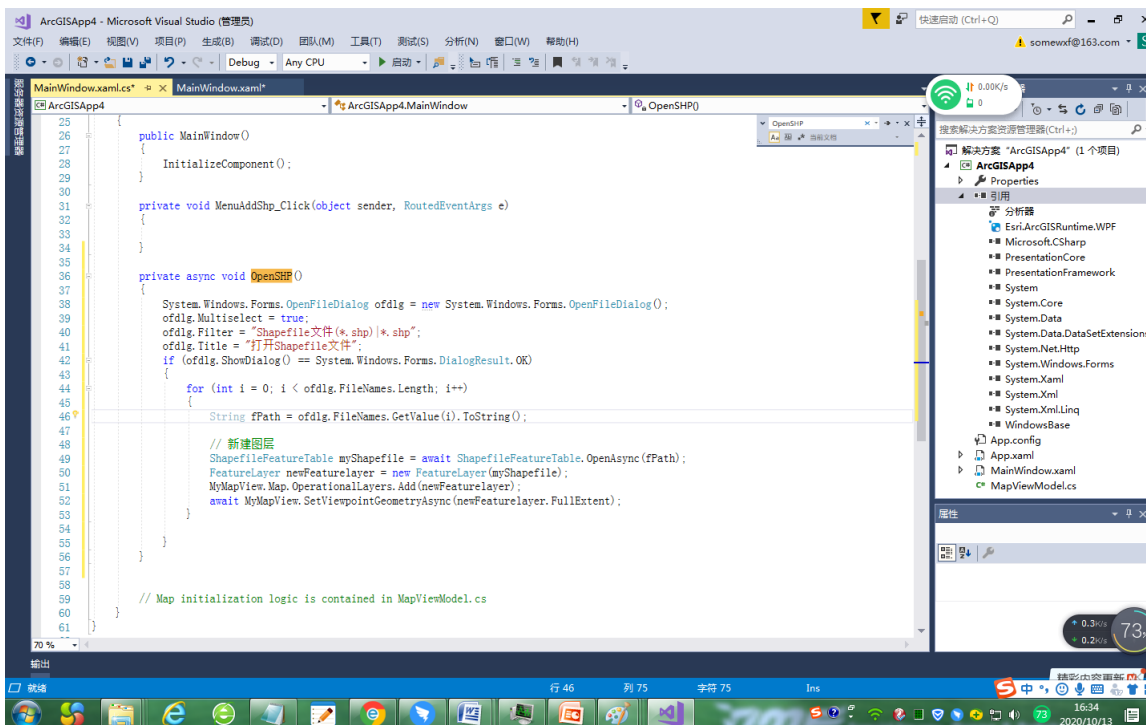
添加该引用主要是为了使用 System.Windows.Forms 中的 OpenFileDialog





3.2.3 代码编写

1、在 MainWindow.xaml.cs 文件中，添加一个自定义函数 `OpenSHP()`，函数如下，



```

private async void OpenSHP()
{
    System.Windows.Forms.OpenFileDialog ofdlg = new
System.Windows.Forms.OpenFileDialog();
    ofdlg.Multiselect = true;
    ofdlg.Filter = "Shapefile 文件(*.shp)|*.shp";
    ofdlg.Title = "打开 Shapefile 文件";
    if (ofdgl.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        for (int i = 0;i < ofdlg.FileNames.Length;i++)
        {
            String fPath = ofdlg.FileNames.GetValue(i).ToString();

// 新建图层
            ShapefileFeatureTable myShapefile = await
ShapefileFeatureTable.OpenAsync(fPath);
            FeatureLayer newFeaturelayer = new FeatureLayer(myShapefile);
            MyMapView.Map.OperationalLayers.Add(newFeaturelayer);
            await MyMapView.SetViewpointGeometryAsync(newFeaturelayer.FullExtent);
        }
    }
}

```

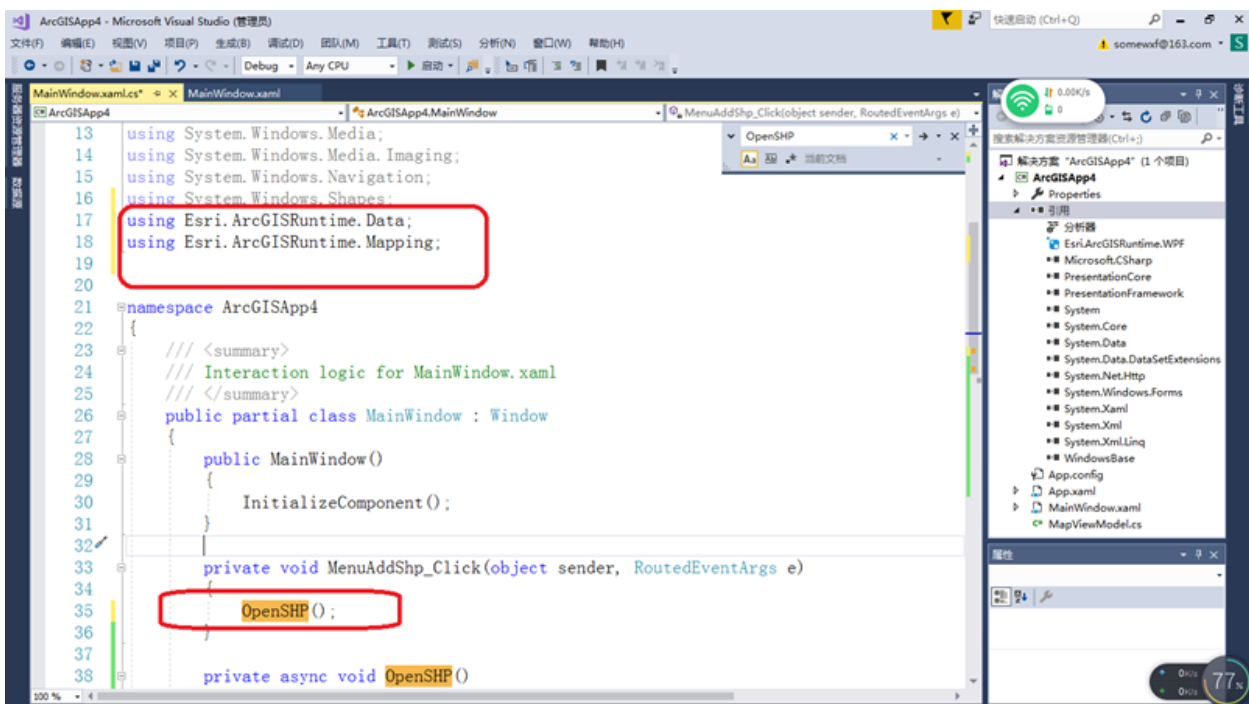
2、在MainWindow ()类的头文件中添加ArcGISRuntime命名空间，消除错误提示。

```

using Esri.ArcGISRuntime.Data;
using Esri.ArcGISRuntime.Mapping;

```

3、在“加载 shp”的 Click 事件 MenuAddShp_Click 中调用 OpenSHP()函数



4、启动程序，运行结果



3.3 加载 Geodatabase 数据

3.3.1 界面设计

参考“3.2 加载 shp 数据”的界面设计，在“文件”主菜单下添加“加载 Geodatabase”子菜单（注意“加载 Geodatabase”子菜单的名称和 Header 需要自行设置，不可以和之前的一样），加载其 Click 事件。

3.3.2 代码编写

1、MainWindow()类中设置全局变量

```
private FeatureLayer _featureLayer;
```

2、在 MainWindow()类中编写函数 OpenGeoDatebase ()

```
private async void OpenGeoDatebase()  
{  
    string _slocalGeodatabasePath="";  
    Geodatabase _localGeodatabase;  
    //打开对话框，选择 geodatabase 文件，获取路径  
    //需在引用中添加 System.Windows.Forms  
    System.Windows.Forms.OpenFileDialog ofdlg = new
```

```

System.Windows.Forms.OpenFileDialog();
    ofdlg.Filter = "Geodatabase 文件(*.geodatabase)|*.geodatabase";
    ofdlg.Title = "打开 Geodatabase 文件";
    if (ofdgl.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        _slocalGeodatabasePath = ofdlg.FileName;
    }
    if (System.IO.File.Exists(_slocalGeodatabasePath))
    {
        _localGeodatabase = await Geodatabase.OpenAsync(_slocalGeodatabasePath);

        Map pMap = new Map();

        int beginCount = _localGeodatabase.GeodatabaseFeatureTables.Count - 1;
        for (int i = beginCount; i >= 0; i--)
        {
            // 加载数据
            GeodatabaseFeatureTable table =
            _localGeodatabase.GeodatabaseFeatureTables[i];
            await table.LoadAsync();
            // 创建图层，增加到地图中
            FeatureLayer layer = new FeatureLayer(table);
            layer.Name = table.TableName;
            pMap.OperationalLayers.Add(layer);
            //将图层赋值给全局变量。
            _featureLayer = layer;
        }
        Viewpoint pVP = new Viewpoint(_featureLayer.FullExtent);
        pMap.InitialViewpoint = pVP;
        MyMapView.Map = pMap;
    }
}

```

3、“加载 Geodatabase” 菜单或按钮的点击事件中，调用 **OpenGeoDatebase ()** 函数。

4、编译运行。

3.4 加载在线数据（一）

3.4.1 界面设计

参考“3.2 加载 shp 数据”的界面设计，在“文件”主菜单下添加“打开在线数据”子菜单，加载其 Click 事件。

3.4.2 代码编写

1、MainWindow()类中设置全局变量

```
private string _statesUrl =  
“https://services.arcgis.com/jIL9msH90I208GCb/arcgis/rest/services/USA\_Daytime  
Population\_2016/FeatureServer/0” ;  
private ServiceFeatureTable _featureTable;  
private FeatureLayer _featureLayer; //若之前已有该变量，不用重复添加这一行
```

2、在 MainWindow ()类的头文件中添加命名空间

```
//using System.Windows.Media; //注意：把这一个命名空间注释掉  
using System.Drawing;  
using Esri.ArcGISRuntime.Geometry;  
using Esri.ArcGISRuntime.Symbology;
```

3、在 MainWindow ()类中编写函数 OpenOnlineData ()

```
private async void OpenOnlineData()  
{  
    //创建地图  
    Map pMap = new Map();  
    pMap.Basemap = Basemap.CreateTopographic();  
    _featureTable = new ServiceFeatureTable(new Uri(_statesUrl));  
    _featureLayer = new FeatureLayer(_featureTable);  
    _featureLayer.MaxScale = 1.0;  
    await _featureLayer.LoadAsync();  
    Viewpoint pVP = new Viewpoint(_featureLayer.FullExtent);  
    pMap.InitialViewpoint = pVP;  
    //设置图层的符号样式  
    SimpleMarkerSymbol pMarkerSymbol = new  
SimpleMarkerSymbol(SimpleMarkerSymbolStyle.Circle, Color.Red, 5);  
    SimpleLineSymbol pLineSymbol = new SimpleLineSymbol(SimpleLineSymbolStyle.Solid,  
Color.Blue, 1);  
    SimpleFillSymbol pFillSymbol = new SimpleFillSymbol(SimpleFillSymbolStyle.Solid,  
Color.Pink, pLineSymbol);  
    SimpleRenderer pRenderere = new SimpleRenderer();  
    if (_featureLayer.FeatureTable.GeometryType == GeometryType.Polygon)  
    {  
        pRenderere.Symbol = pFillSymbol;  
    }  
    else if (_featureLayer.FeatureTable.GeometryType == GeometryType.Polyline)  
    {  
        14
```

```

        pRenderere.Symbol = pLineStyleSymbol;
    }
    else if (_featureLayer.FeatureTable.GeometryType == GeometryType.Point)
        pRenderere.Symbol = pMarkerSymbol;
    else
        pRenderere.Symbol = pFillSymbol;
    _featureLayer.Renderer = pRenderere;
    pMap.OperationalLayers.Add(_featureLayer);
    MyMapView.Map = pMap;
    MyMapView.SelectionProperties.Color = Color.Cyan;
}

```

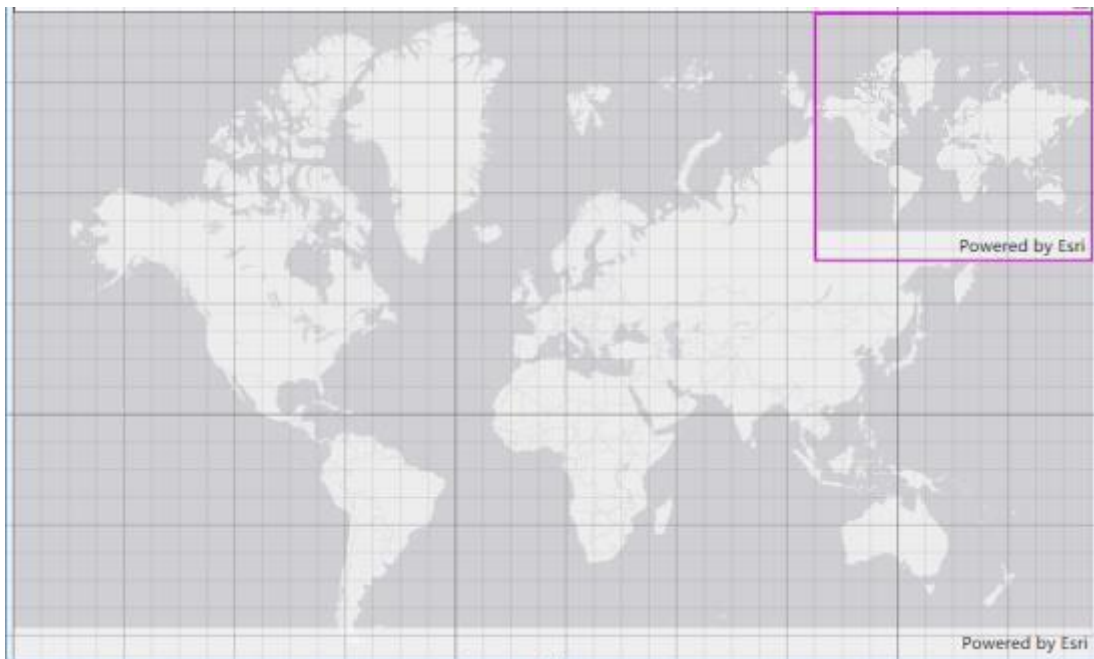
4、“打开在线数据”菜单或按钮的点击事件中，调用 OpenOnlineData（）函数。

5、编译运行。

3.5 鹰眼视图

3.5.1 界面设计

在窗体上添加一个 border 控件，然后在其上再添加一个地图视图控件，并设置相关属性。



其 xaml 代码为：

```

<Border x:Name="EagleBorder" BorderBrush="Magenta" BorderThickness="2" HorizontalAlignment="Right"
Height="180" Margin="0,30,0,0" VerticalAlignment="Top" Width="202" Background="#FF5CFD72"
Grid.ColumnSpan="2" >
<esri:MapView x:Name="eagleMapView" IsEnabled="False" />

```

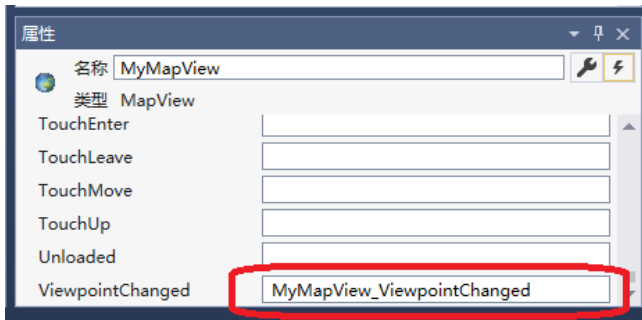
3.5.2 代码编写

1、设置鹰眼图 eagleMapView 控件的底图数据

在 MainWindow() 构造函数中添加以下代码

```
public MainWindow()  
{  
    InitializeComponent();  
    //鹰眼图中加载和主视图一样的数据  
    //eagleMapView.Map = MyMapView.Map.Clone();  
    eagleMapView.Map = new Map(Basemap.CreateStreets());  
}
```

2. 为窗体中的 MyMapView 地图控件添加 ViewpointChanged 事件处理代码



```
private void MyMapView_ViewpointChanged(object sender, EventArgs e)  
{  
    //将鹰眼图上的图形元素清空  
    eagleMapView.GraphicsOverlays.Clear();  
    //获取主视图上地图的显示范围  
    Esri.ArcGISRuntime.Geometry.Polygon vExtent = MyMapView.VisibleArea;  
    Envelope eagleViewEnv = vExtent.Extent;  
    //构建线符号  
    Color lineColor = Color.FromArgb(255, 0, 0);  
    SimpleLineSymbol lineSymbol = new  
    Esri.ArcGISRuntime.Symbology.SimpleLineSymbol(SimpleLineSymbolStyle.Dash, lineColor, 2.0);  
    //构建面符号  
    Color fillColor = Color.FromArgb(0, 255, 255, 255);  
    SimpleFillSymbol polySymbol = new  
    Esri.ArcGISRuntime.Symbology.SimpleFillSymbol(SimpleFillSymbolStyle.Solid, fillColor,  
    lineSymbol);  
    //构建图形元素图层  
    var graphicOverlay = new Esri.ArcGISRuntime.UI.GraphicsOverlay();  
    //构建图形元素，将元素放入图形元素图层  
    var envGraphic = new Esri.ArcGISRuntime.UI.Graphic(eagleViewEnv, polySymbol);  
}
```

```
graphicOverlay.Graphics.Add(envGraphic);  
//图形元素图层放入到鹰眼图中  
eagleMapView.GraphicsOverlays.Add(graphicOverlay);  
}
```

3. 编译运行

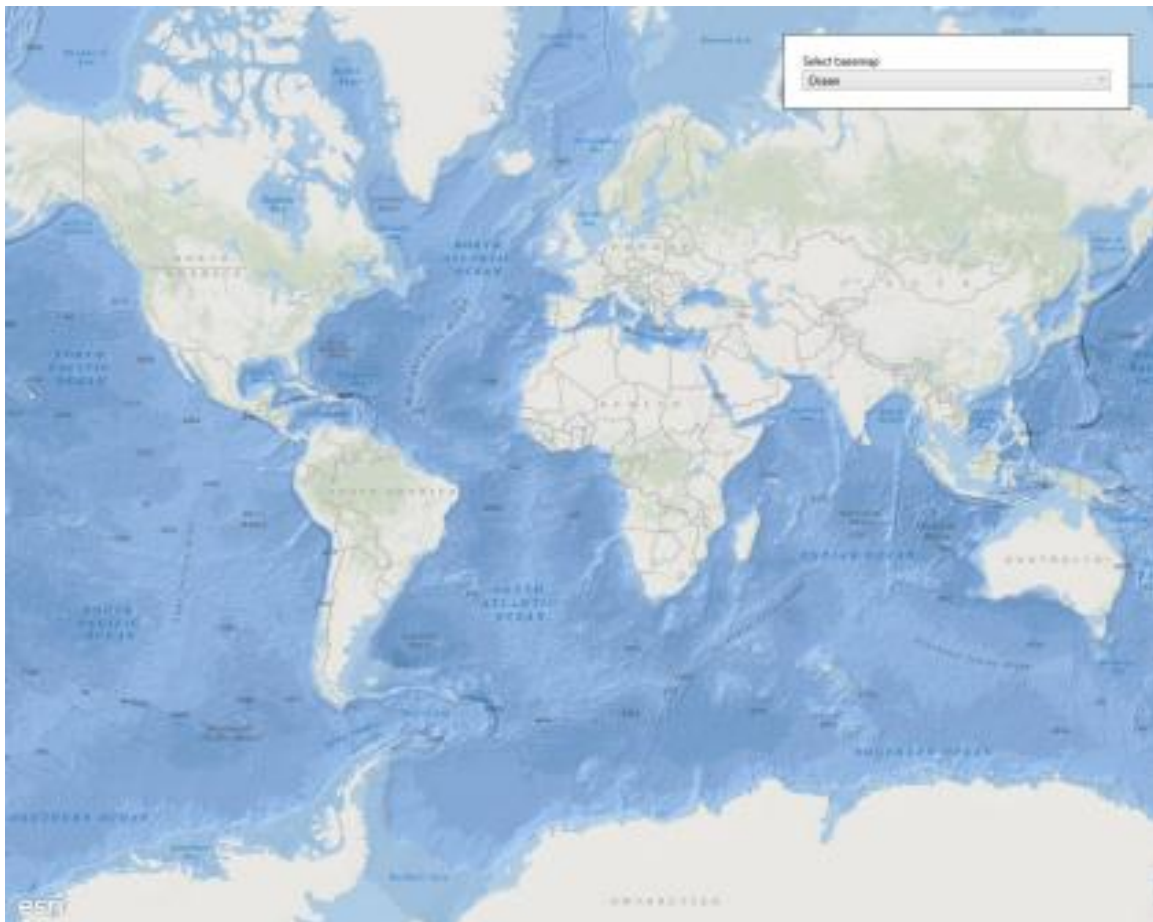
3.6 修改底图

3.6.1 界面设计

1、添加 Border

2、添加 ComboBox ， 将其命名为 BasemapChooser

```
<Border BorderBrush="Black" BorderThickness="1" HorizontalAlignment="Left"  
Height="57" Margin="580, 210, 0, 0" VerticalAlignment="Top" Width="202"  
Background="{DynamicResource {x:Static SystemColors.ControlBrushKey}}">  
    <ComboBox x:Name="BasemapChooser" HorizontalAlignment="Left"  
Height="57" Margin="9, -1, 0, -1" VerticalAlignment="Top" Width="182"  
SelectionChanged="BasemapChooser_SelectionChanged"/>  
</Border>
```



3.6.2 代码编写

1、在 MainWindow 主窗体中添加以下全局变量

```
private readonly Dictionary<string, Basemap> _basemapOptions = new Dictionary<string, Basemap>()
{
    {"Streets (Raster)", Basemap.CreateStreets()},
    {"Streets (Vector)", Basemap.CreateStreetsVector()},
    {"Streets - Night (Vector)", Basemap.CreateStreetsNightVector()},
    {"Imagery (Raster)", Basemap.CreateImagery()},
    {"Imagery with Labels (Raster)", Basemap.CreateImageryWithLabels()},
    {"Imagery with Labels (Vector)",
        Basemap.CreateImageryWithLabelsVector()},
    {"Dark Gray Canvas (Vector)", Basemap.CreateDarkGrayCanvasVector()},
    {"Light Gray Canvas (Raster)", Basemap.CreateLightGrayCanvas()},
    {"Light Gray Canvas (Vector)", Basemap.CreateLightGrayCanvasVector()},
    {"Navigation (Vector)", Basemap.CreateNavigationVector()},
    {"OpenStreetMap (Raster)", Basemap.CreateOpenStreetMap()}
};
```

2、MainWindow 类中添加以下 Initialize () 函数，以初始化数据

```
private void Initialize() {
    // Assign a new map to the MapView
    MyMapView.Map = new Map(_basemapOptions.Values.First());
    // Set basemap titles as a items source
    BasemapChooser.ItemsSource = _basemapOptions.Keys;
    // Show the first basemap in the list
    BasemapChooser.SelectedIndex = 0;
}
```

3、MainWindow 类的构造函数调用 Initialize () 函数

```
public MainWindow()
{
    InitializeComponent();
    //...初始化数据
    Initializ();
}
```

4、添加 BasemapChooser 的 SelectionChanged 事件，编写以下代码

```
private void OnBasemapChooserSelectionChanged(object sender, SelectionChangedEventArgs e)
{
    // Get the title of the selected basemap
```

```
var selectedBasemapTtile = e.AddedItems[0].ToString();  
// Retrieve the basemap from the dictionary  
MyMapView.Map.Basemap = _basemapOptions[selectedBasemapTtile];  
}
```

5、编译运行

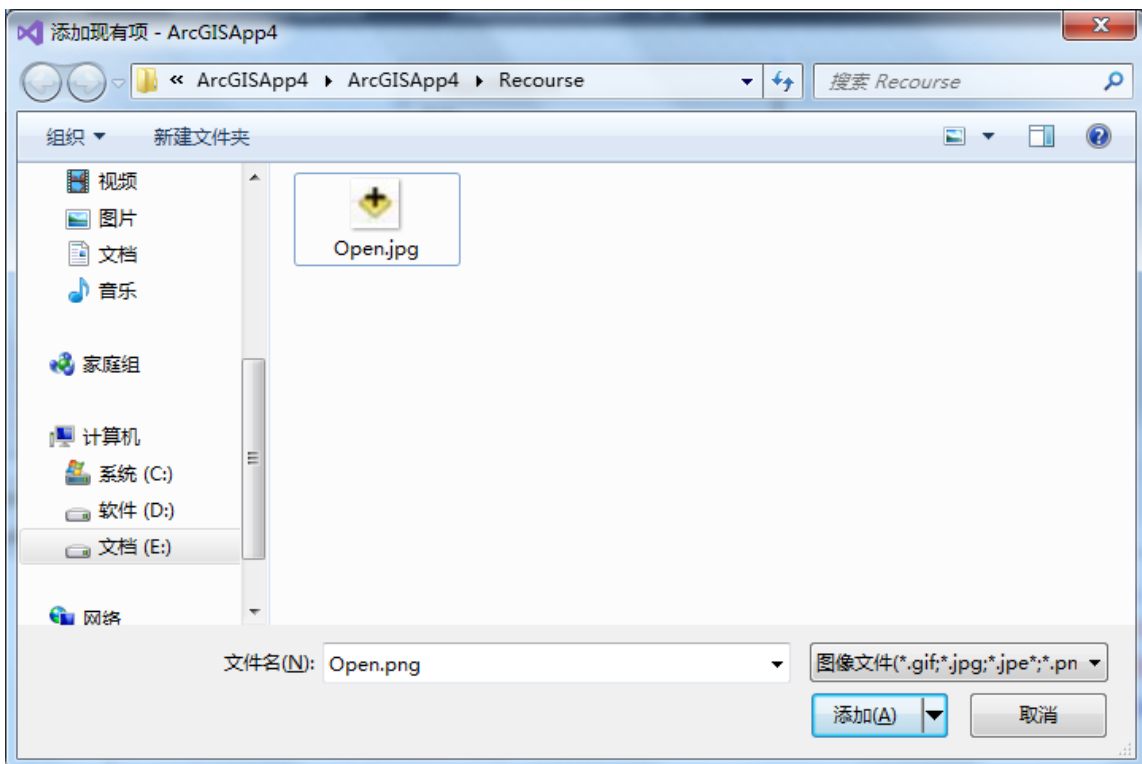
3.7 地图放大缩小等

添加“视图”主菜单，在其下添加“放大”“缩小”“漫游”“全图”等，可参考课件代码，点击“放大”、“缩小”等菜单即实现图形的缩放，也可以考虑在点击“放大”、“缩小”等菜单后，需在 MapView 上进行点击后，以该点为中心点，进行放大和缩小，需自行修改代码。

(步骤略)

3.8 添加工具条的图标按钮工具，实现加载 shp 数据

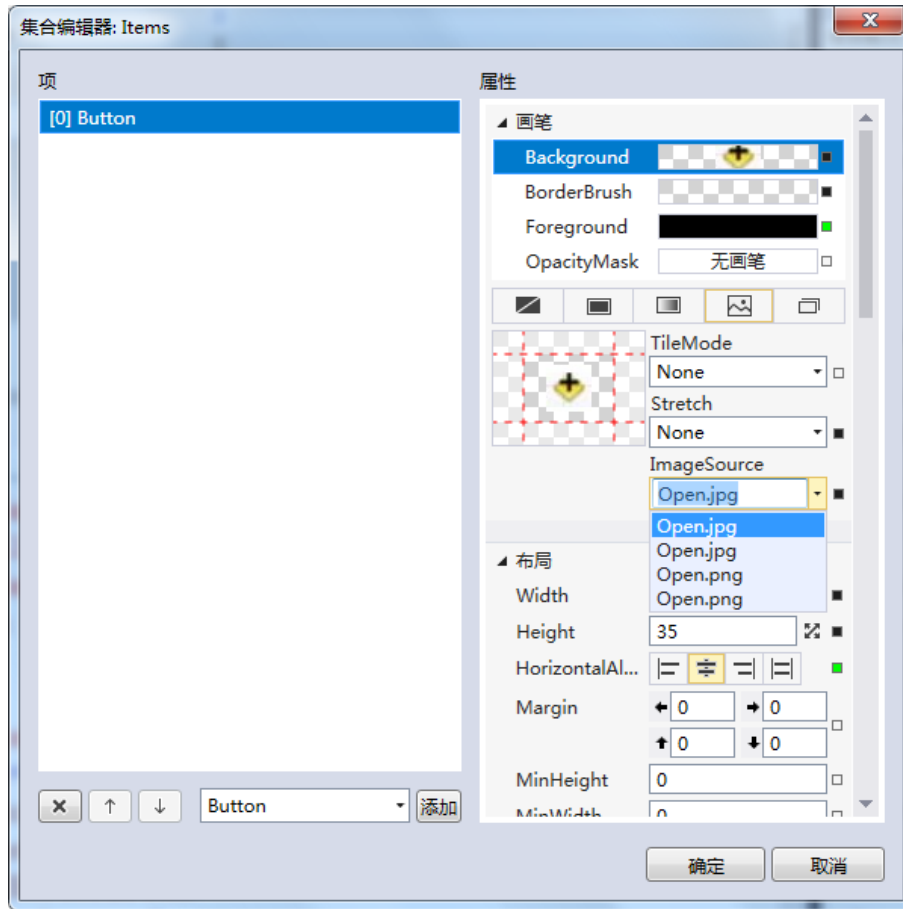
1. 工程项目中添加图片资源，已方便后续工具条上使用该图片。



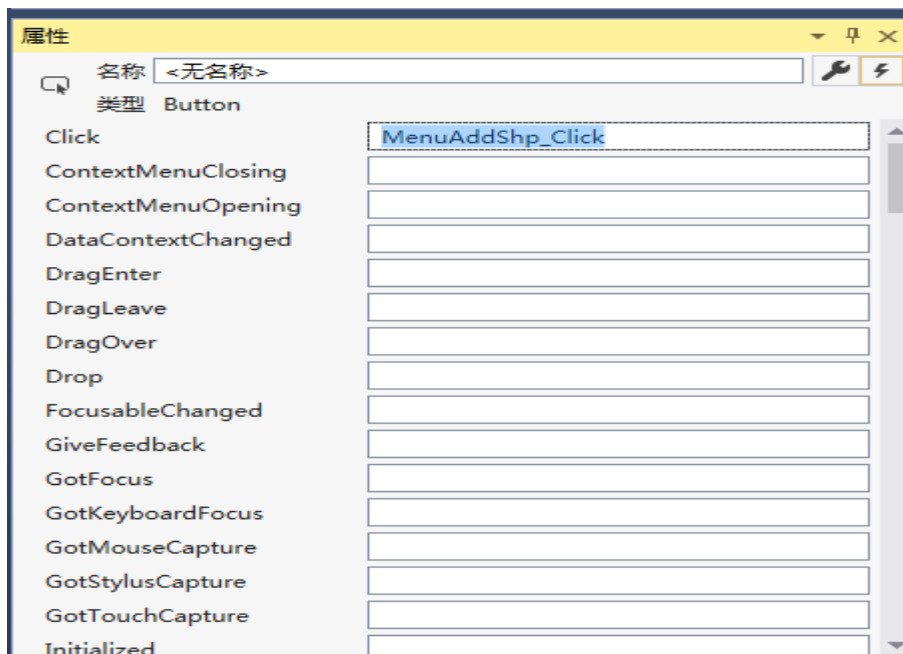
2. 在主窗体添加一个 toolbar 控件，在其上点击 toolbar 的 item 属性为其添加工具



3、设置工具按钮的图片



4、添加其 Click 事件名称，设置其事件名称 MenuAddShp_Click，即为“加载 shp”子菜单的 Click 事件名称，不用再编写代码，点击该工具即可实现 shp 文件加载。



可参考以上方式，添加各种工具按钮，实现各子菜单所对应的按钮工具功能。

3.9 退出程序

在“文件”主菜单下添加“退出程序”子菜单，加载其 Click 事件，在 Click 事件中添加如下代码。

```
if(System.Windows.MessageBox.Show("确定退出? ", "退出程序", MessageBoxButton.YesNo)
== MessageBoxResult.Yes)
{
    System.Windows.Application.Current.Shutdown();
}
```

【作业与练习】：

1. 创建自己的 ArcGIS Online 地图，并加载到自己的应用中；
参考网址 <https://doc.arcgis.com/zh-cn/arcgis-online/get-started/get-started-with-maps.htm>
2. 参考帮助文档，实现 MapView 中 [ShowCalloutAt](#) 代码实例。
3. 参考帮助文档，实现 MapView 中 [SetViewpointRotationAsync](#) 代码实例。