Spatial Controls

The Spatial Controls

- The Spatial controls allow us to create spatial components such as boarders and decorators.

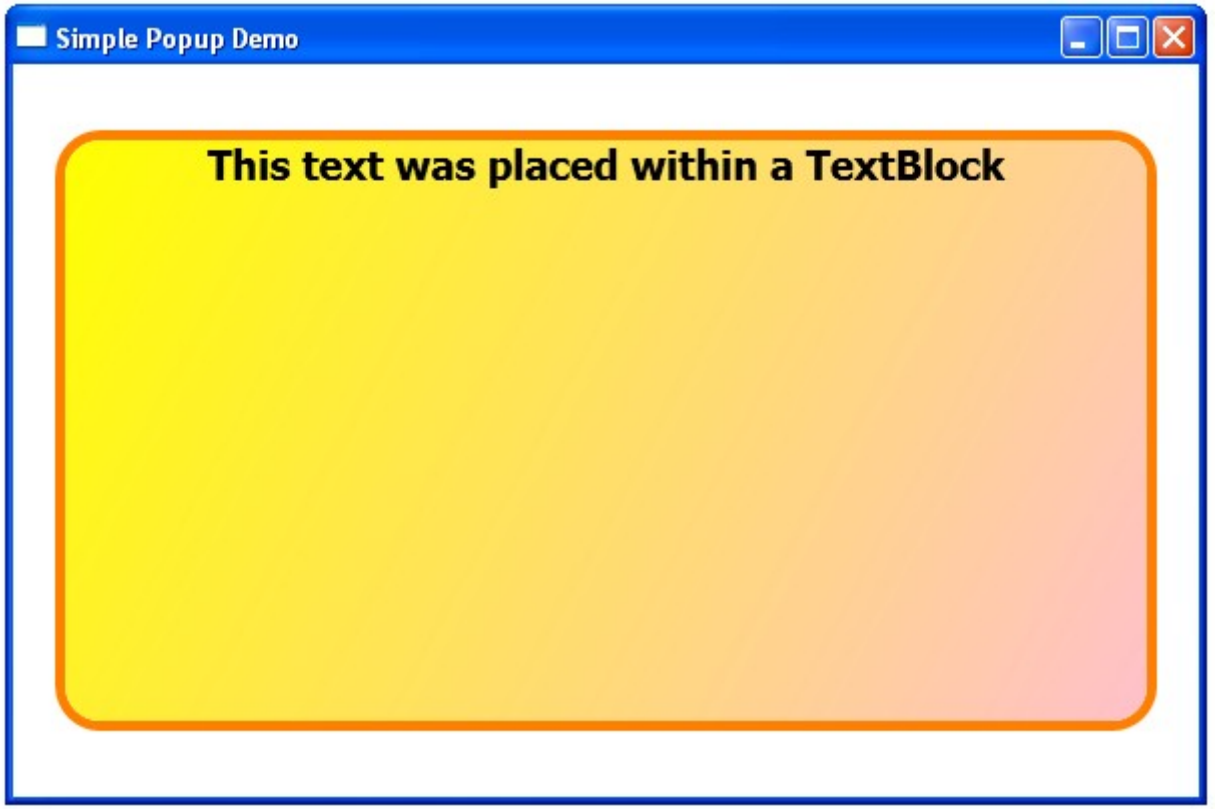
The `Border` Control

- The `Border` control creates a small border that surrounds other controls.
- We can use this control for creating backgrounds as well.

The `Border` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Title="Simple Popup Demo"
        Width="600" Height="400" FontSize="20" FontWeight="SemiBold">

    <Border     HorizontalAlignment="Center" VerticalAlignment="Center"
                Width="550" Height="300" CornerRadius="20"
                BorderBrush="#FFFF8000" BorderThickness="5">

        <Border.Background>
            <LinearGradientBrush>
                <GradientStop Color="Yellow" Offset="0"/>
                <GradientStop Color="Pink" Offset="1"/>
            </LinearGradientBrush>
        </Border.Background>

        <TextBlock  Width="400" Height="300" TextWrapping="Wrap"
                    TextAlignment="Center"
                    Text="This text was placed within a TextBlock"/>

    </Border>

</Window>
```

The `Border` Control



The `BulletDecorator` Control

- The `BulletDecorator` control displays an item together with a bullet within a bullet-ed list.

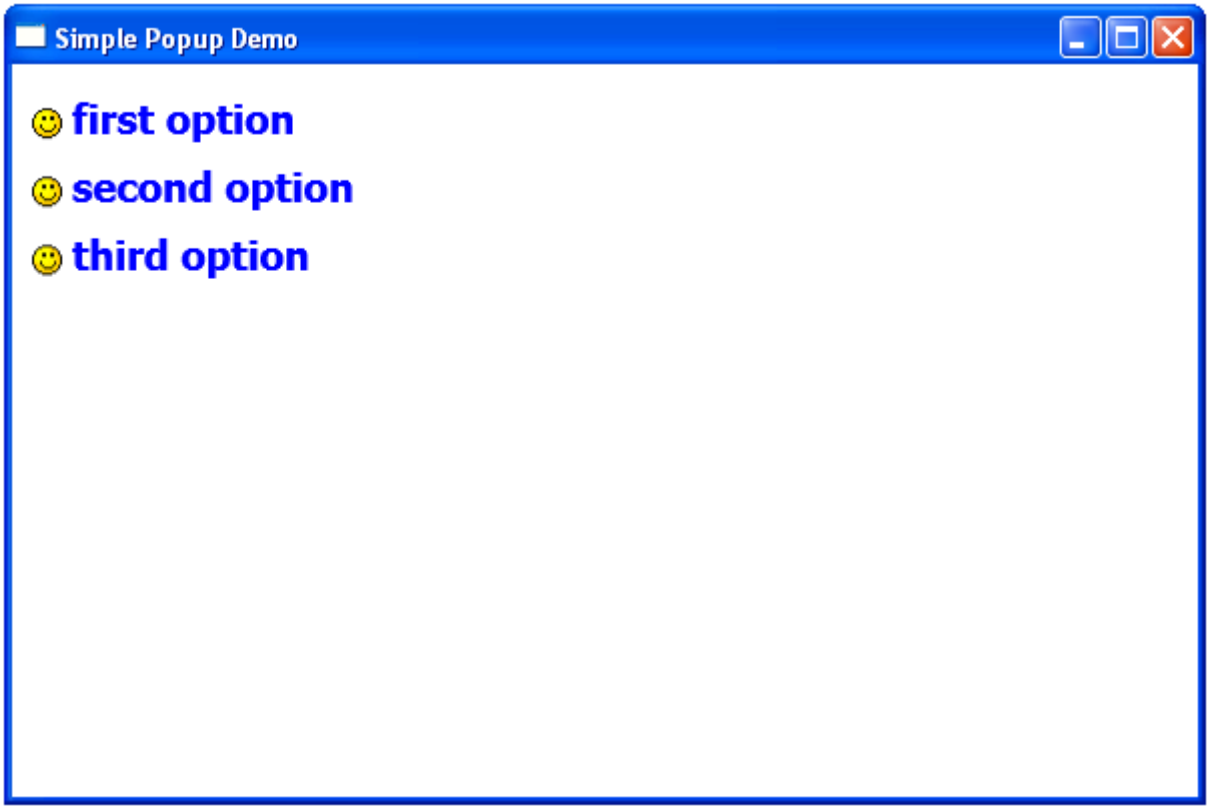
The `BulletDecorator` Control

```
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
x:Class="gaga.MainWindow"
Title="Simple Popup Demo"
Width="600" Height="400"
FontSize="20" FontWeight="SemiBold">

    <StackPanel Margin="10">
        <BulletDecorator>
            <BulletDecorator.Bullet>
                <Image  Width="20" Height="20"
                        Source="smile.gif" Stretch="None"
                        HorizontalAlignment="Left"
                        VerticalAlignment="Center" >
                </Image>
            </BulletDecorator.Bullet>
            <Label Content="first option" Foreground="Blue"/>
        </BulletDecorator>
```

The `BulletDecorator` Control

```
        <BulletDecorator>
            <BulletDecorator.Bullet>
                <Image  Width="20" Height="20"
                        Source="smile.gif" Stretch="None"
                        HorizontalAlignment="Left"
                        VerticalAlignment="Center" >
                </Image>
            </BulletDecorator.Bullet>
            <Label Content="second option" Foreground="Blue"/>
        </BulletDecorator>
        <BulletDecorator>
            <BulletDecorator.Bullet>
                <Image  Width="20" Height="20"
                        Source="smile.gif" Stretch="None"
                        HorizontalAlignment="Left"
                        VerticalAlignment="Center" >
                </Image>
            </BulletDecorator.Bullet>
            <Label Content="third option" Foreground="Blue"/>
        </BulletDecorator>

    </StackPanel>

</Window>
```

The `BulletDecorator` Control



The `GroupBox` Control

- The `GroupBox` control is similar to `Border`. It creates a border around a single child.
- Using the `Foreground` property we can set the brush been used to draw the header text. Using the `BorderBrush` property we can set the brush been used to draw the border. Using the `Background` property we can set the fill.

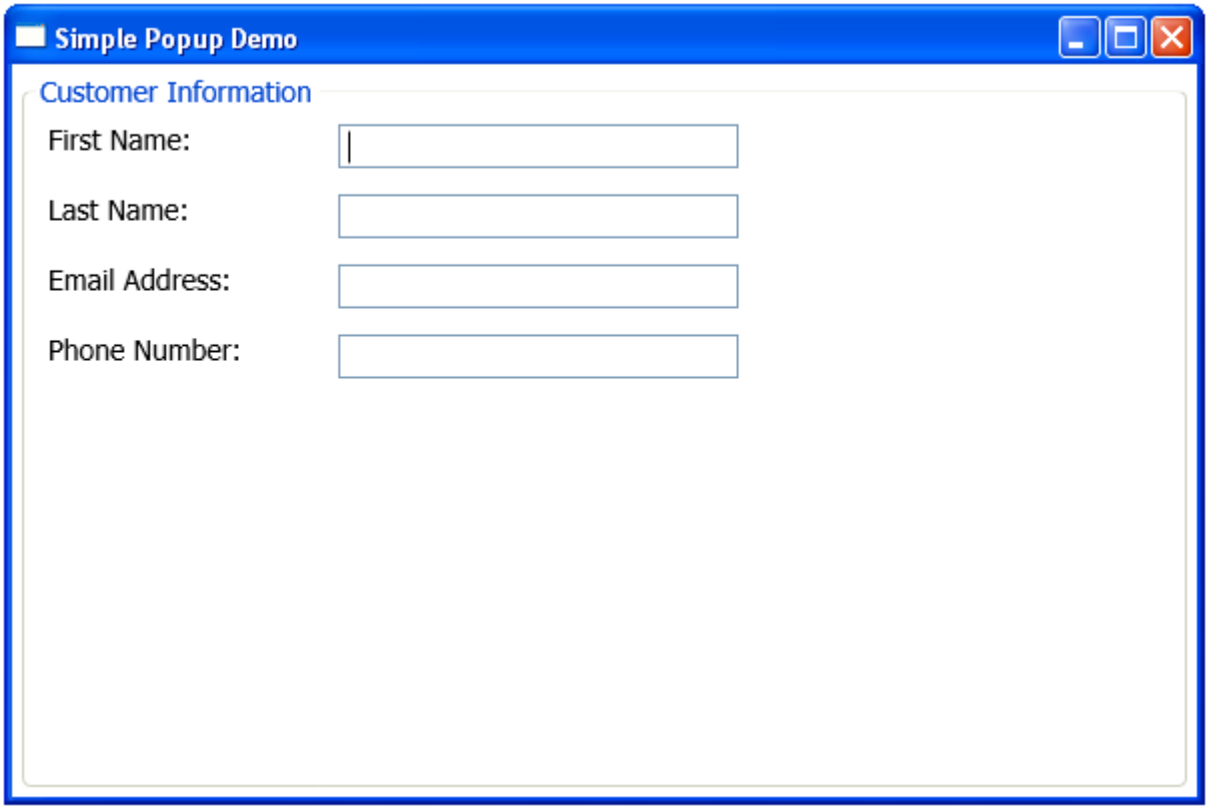
The `GroupBox` Control

```
<Window     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
            x:Class="gaga.MainWindow" Title="Simple GroupBox Demo"
            Width="600" Height="400" FontSize="14">

    <Grid>

        <GroupBox   HorizontalAlignment="Stretch"
                    VerticalAlignment="Stretch"
            Header="Customer Information"
                    Margin="5,5,5,5"
                    BorderThickness="1">
            <Grid Margin="2">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="150"/>
                    <ColumnDefinition Width="200"/>
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="35"/>
                    <RowDefinition Height="35"/>
                    <RowDefinition Height="35"/>
                    <RowDefinition Height="35"/>
                    <RowDefinition Height="35"/>
                </Grid.RowDefinitions>
```

The `GroupBox` Control

```
                <Label Grid.Row="0" Grid.Column="0" Content="First Name:"/>
                <TextBox Grid.Row="0" Grid.Column="1" Height="22"/>
                <Label Grid.Row="1" Grid.Column="0" Content="Last Name:"/>
                <TextBox Grid.Row="1" Grid.Column="1" Height="22"/>
                <Label Grid.Row="2" Grid.Column="0" Content="Email Address:"/>
                <TextBox Grid.Row="2" Grid.Column="1" Height="22"/>
                <Label Grid.Row="3" Grid.Column="0" Content="Phone Number:"/>
                <TextBox Grid.Row="3" Grid.Column="1" Height="22"/>
            </Grid>
        </GroupBox>

    </Grid>

</Window>
```

The `GroupBox` Control



The `ListView` Control

- The `ListView` control displays a set of data items. We can choose the preferred layout.

The `ListView` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Title="Simple Demo" Width="600"
        Height="400" FontSize="14">

    <Grid x:Name="LayoutRoot">

        <ListView Name="students"
                  HorizontalAlignment="Stretch"
                  HorizontalContentAlignment="Left"
                  Visibility="Visible"
                  ItemsSource="{Binding}"
                  >
            <ListView.View>
                <GridView>
                    <GridViewColumn Header="First Name" Width="140"
                          DisplayMemberBinding="{Binding Path=FirstName}"/>
```

The `ListView` Control

```
                    <GridViewColumn Header="Last Name" Width="140"
                          DisplayMemberBinding="{Binding Path=LastName}"/>
                    <GridViewColumn Header="Average" Width="100"
                          DisplayMemberBinding="{Binding Path=Average}"/>
                </GridView>
            </ListView.View>
        </ListView>

    </Grid>

</Window>
```

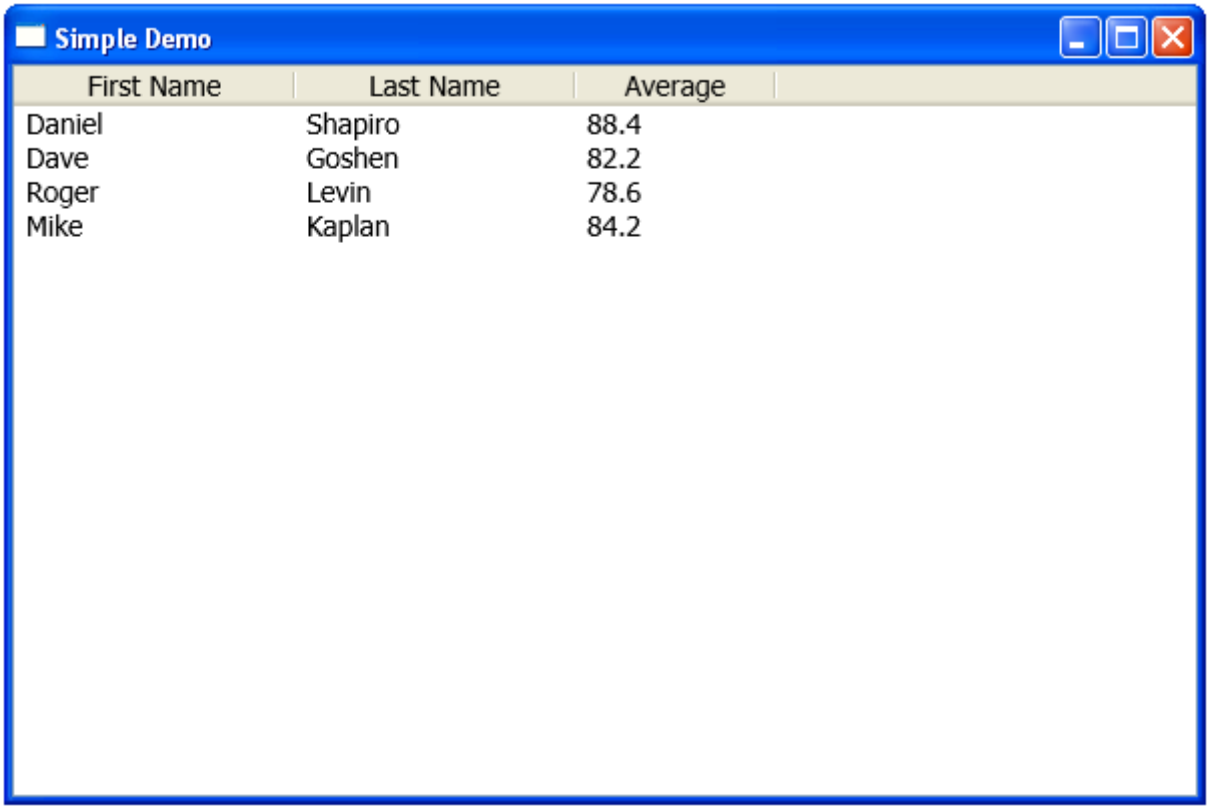
The `ListView` Control

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        this.InitializeComponent();
        ObservableCollection<Student> stds =
            new ObservableCollection<Student>();
        stds.Add(new Student("Daniel", "Shapiro", 88.4));
        stds.Add(new Student("Dave", "Goshen", 82.2));
        stds.Add(new Student("Roger", "Levin", 78.6));
        stds.Add(new Student("Mike", "Kaplan", 84.2));
        students.DataContext = stds;
    }
}
```

The `ListView` Control

```
public class Student
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public double Average { get; set; }
    public Student(string firstName, string lastName, double average)
    {
        FirstName = firstName;
        LastName = lastName;
        Average = average;
    }

}
```

The `ListView` Control



The `ProgressBar` Control

- The `ProgressBar` control displays a progress bar. We can use it to update the user about the progress of work performed in the background.

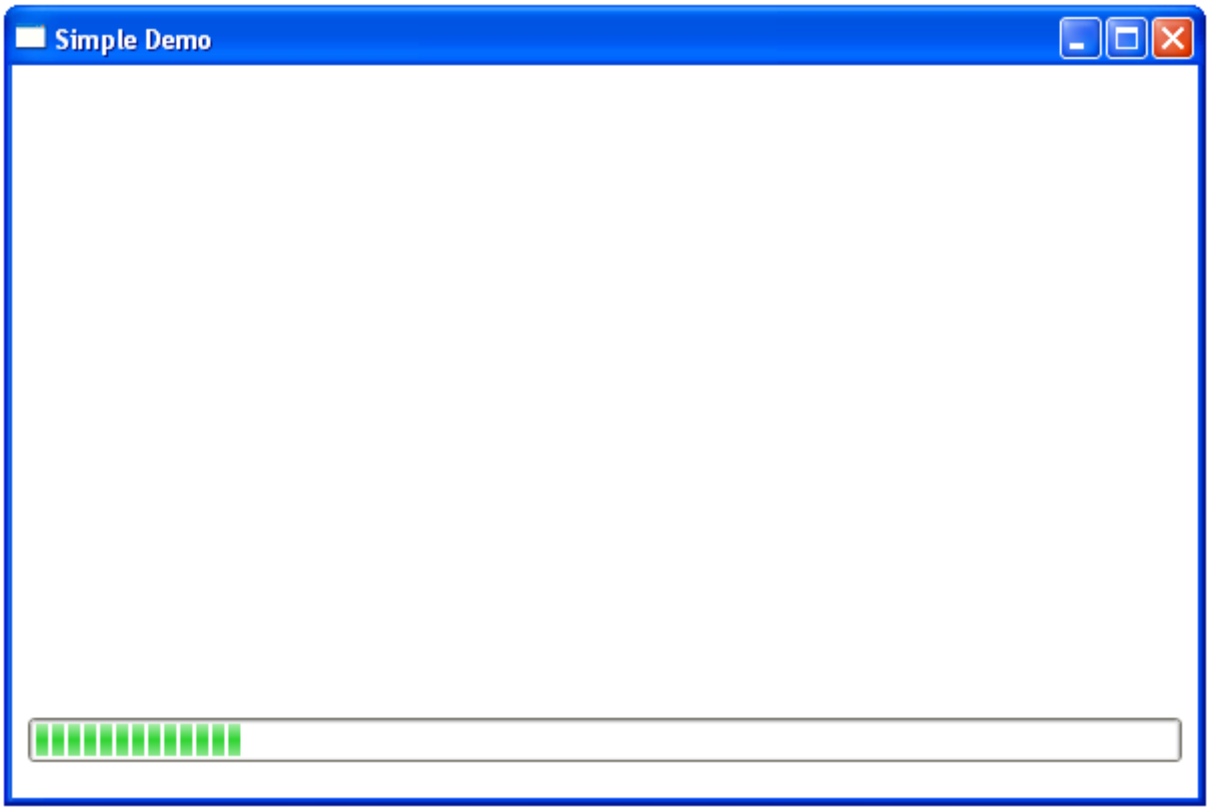
The `ProgressBar` Control

```
namespace gaga
{
    public partial class MainWindow : Window
    {
        private BackgroundWorker background;
        public MainWindow()
        {
            this.InitializeComponent();
            background = new BackgroundWorker();
            background.WorkerReportsProgress = true;
            background.DoWork += BackgroundWorker_DoWork;
            background.ProgressChanged += BackgroundWorker_ProgressChanged;
            background.RunWorkerCompleted += BackgroundWorker_RunWorkerCompleted;
        }

        private void bt_Click(object sender, RoutedEventArgs e)
        {
            progress.Value = 0;
            progress.Visibility = Visibility.Visible;
            progress.Visibility = Visibility.Visible;
            bt_handle.Visibility = Visibility.Hidden;
            background.RunWorkerAsync();
        }
```

The `ProgressBar` Control

```
        private void BackgroundWorker_RunWorkerCompleted(Object sender,
             System.ComponentModel.RunWorkerCompletedEventArgs e)
        {
            progress.Visibility = Visibility.Hidden;
            working_lbl.Visibility = Visibility.Hidden;
            bt_handle.Visibility = Visibility.Visible;
        }

        private void BackgroundWorker_ProgressChanged(Object sender,
             ProgressChangedEventArgs e)
        {
            progress.Value = e.ProgressPercentage;
        }

        private void BackgroundWorker_DoWork(Object sender, DoWorkEventArgs e)
        {
            int value = 0;
            Random rand = new Random();
            while (value < 100)
            {
                Thread.Sleep(200);
                value += rand.Next(5, 10);
                background.ReportProgress(value);
            }
        }
    }
}
```

The `ProgressBar` Control



The `Seperator` Control

- The `Separator` control draws a line that separates between two items in a menu or two items in a toolbar.

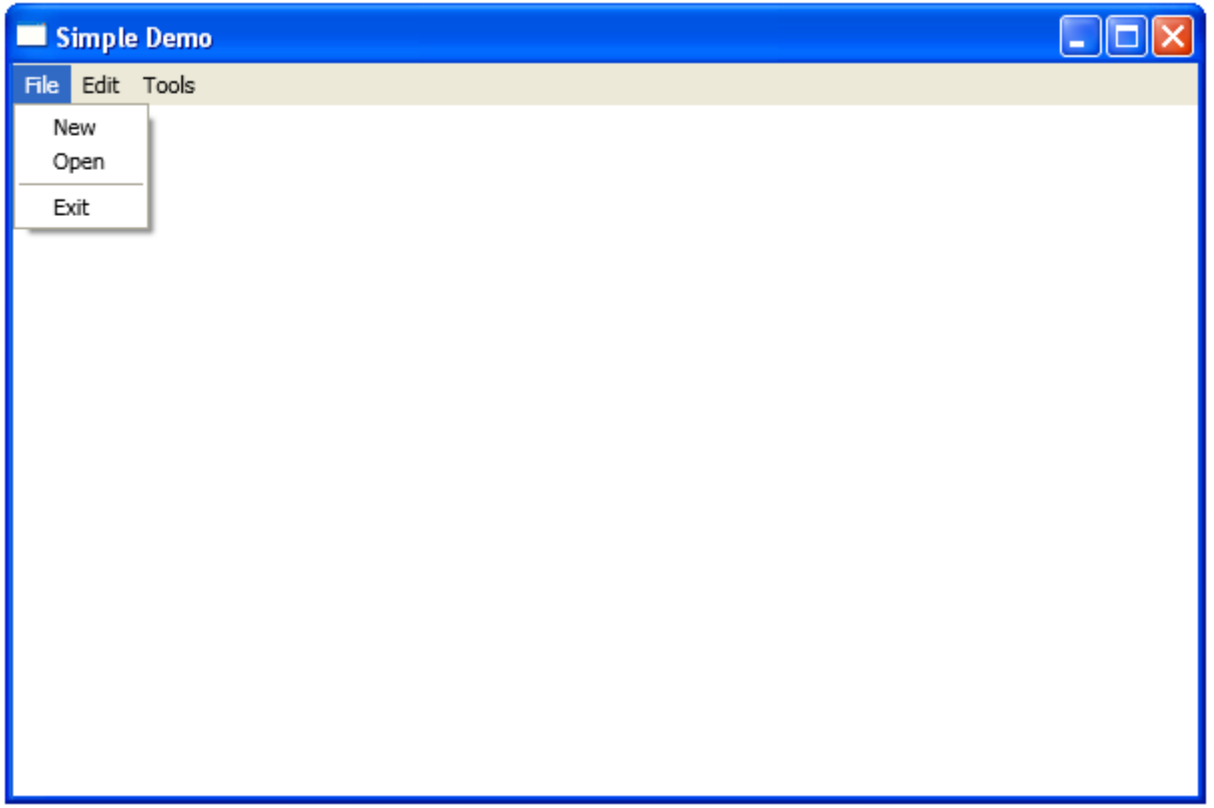
The `Seperator` Control

```
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
x:Class="gaga.MainWindow"
Title="Simple Demo"
Width="600" Height="400"
FontSize="14">

    <Menu HorizontalAlignment="Stretch"
            DockPanel.Dock="Top" VerticalAlignment="Top">
        <MenuItem Header="File">
            <MenuItem Header="New">
             </MenuItem>
            <MenuItem Header="Open">
            </MenuItem>
            <Separator/>
            <MenuItem Header="Exit"/>
        </MenuItem>
        <MenuItem Header="Edit">
            <MenuItem Header="Copy">
            </MenuItem>
            <MenuItem Header="Cut">
            </MenuItem>
            <MenuItem Header="Paste"/>
        </MenuItem>
```

The `Seperator` Control

```
        <MenuItem Header="Tools">
            <MenuItem Header="Setting">
            </MenuItem>
            <MenuItem Header="Options">
            </MenuItem>
        </MenuItem>
    </Menu>
</Window>
```

The `Seperator` Control



The `TreeView` Control

- The `TreeView` control displays hierarchical data. It contains `TreeViewItem` controls.

The `TreeView` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Title="Simple Demo"
        Width="600" Height="400" FontSize="14">

    <TreeView Margin="5">
        <TreeViewItem Header="My Folders" IsExpanded="True">
            <TreeViewItem Header="C:" IsExpanded="True">
                <TreeViewItem Header="Temp" IsExpanded="True">
                    <TreeViewItem Header="Letters"/>
                    <TreeViewItem Header="letter_to_bank.doc"/>
                    <TreeViewItem Header="letter_to_garden.doc"/>
                </TreeViewItem>
                <TreeViewItem Header="Documents" IsExpanded="True">
                    <TreeViewItem Header="summar_of_courses.doc"/>
                    <TreeViewItem Header="list_of_students.doc"/>
                    <TreeViewItem Header="list_of_teachers.xsl"/>
                </TreeViewItem>
            </TreeViewItem>
        </TreeViewItem>
    </TreeView>

</Window>
```

The `TreeView` Control