

[](https://www.tutorialr.com/tutorials/)

Windows App SDK

Parallax View





# Parallax View

**Parallax View** shows how you can use **ParallaxView** which allows you to combine the scroll of a list

to a background element so as the list scrolls it animates the background element with **Parallax** in an

application using the **Windows App SDK**.

## Step 1

Follow **Setup and Start** on how to get **Setup** and **Install** what you need for **Visual Studio 2022** and **Windows App SDK**.

|  |  |
| --- | --- |
| In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it. | Text  Description automatically generated |
| Once **Visual Studio 2022** has started select **Create a new project**. | **Graphical user interface, text  Description automatically generated** |
| Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**. | **Graphical user interface, text  Description automatically generated** |
| After that in **Configure your new project** type in the **Project name** as *ParallaxView*, then select a Location and then select **Create** to start a new **Solution**. | **Graphical user interface, text, application, email  Description automatically generated** |

## Step 2

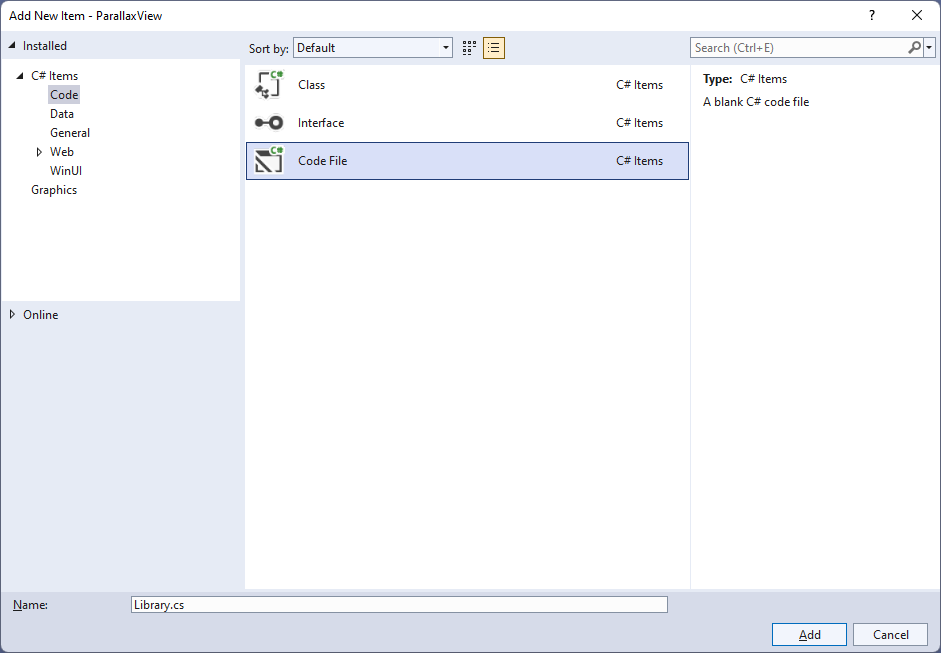
Then in **Visual Studio** within **Solution** **Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Add** then **New Item…**

Table

Description automatically generated with low confidence

## Step 3

Then in **Add New Item** from the **C# Items** list, select **Code** and then select **Code File** from the list next to this, then type in the name of *Library.cs* and then **Click** on **Add**.



## Step 4

You will now be in the **View** for the **Code** of *Library.cs*, within this type the following **Code**:

using Microsoft.UI.Xaml.Controls;

using System;

internal class Library

{

private class Item

{

public Guid Id { get; set; } = Guid.NewGuid();

public string Text { get; set; } = string.Empty;

}

public static void Add(ListView listView, string value)

{

listView.Items.Add(new Item

{

Text = value

});

}

public static void Remove(ListView listView, object sender)

{

Item item = (sender as AppBarButton).Tag as Item;

listView.Items.Remove(item);

}

}

The **Class** that has been defined in *Library.cs* has a **Class** within it of **Item** which is marked **private**, this is just for use with *Library.cs* which will represent what will be added or removed from the **ListView** which is performed in the **Methods** of **Add** and **Remove** which are declared as **static** so an **Instance** of the **Class** is not needed.

## Step 5

|  |  |
| --- | --- |
| Then from **Solution** **Explorer** for the **Solution** double-click on **MainWindow.xaml** to see the **XAML** for the **Main Window**. |  |

## Step 6

In the **XAML** for **MainWindow.xaml** there be some **XAML** for a **StackPanel**, this should be **Removed** by removing the following:

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center" VerticalAlignment="Center">

<Button x:Name="myButton" Click="myButton\_Click">Click Me</Button>

</StackPanel>

## Step 7

While still in the **XAML** for **MainWindow.xaml** above **</Window>**, type in the following **XAML**:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<AutoSuggestBox Grid.Row="0" Margin="25" Name="Value" QueryIcon="Add"

QuerySubmitted="Value\_QuerySubmitted"/>

<Grid Grid.Row="1">

<ParallaxView Source="{x:Bind Display}" VerticalShift="100">

<StackPanel Spacing="5" Orientation="Vertical"

HorizontalAlignment="Center">

<Rectangle Margin="10" Width="75" Height="75" Fill="Black"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Gray"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Red"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Orange"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Yellow"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Green"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Cyan"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Blue"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Magenta"/>

<Rectangle Margin="10" Width="75" Height="75" Fill="Purple"/>

</StackPanel>

</ParallaxView>

<ListView x:Name="Display">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="Auto"/>

</Grid.ColumnDefinitions>

<TextBlock Grid.Column="0" Text="{Binding Text}"

VerticalAlignment="Center"/>

<AppBarButton Grid.Column="1" Icon="Remove" Label="Remove"

Tag="{Binding}" Click="Remove\_Click"/>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

<ListView.ItemContainerStyle>

<Style TargetType="ListViewItem">

<Setter Property="HorizontalContentAlignment" Value="Stretch" />

</Style>

</ListView.ItemContainerStyle>

</ListView>

</Grid>

</Grid>

This **XAML** features a **Grid** with two **Rows**, the first **Row** is for a **AutoSuggestBox** to add an **Item**, then the second **Row** is the **ParallaxView** containing **Rectangle** elements which has the **Source** set to the **ListView** which uses a **DataTemplate** which controls how each **Item** in the **ListView** will look like and contains the **AppBarButton** which will be used to remove an **Item**.

## Step 8

|  |  |
| --- | --- |
| Then, within **Solution** **Explorer** for the **Solution** select the arrow next to **MainWindow.xaml** then double-click on **MainWindow.xaml.cs** to see the **Code** for the **Main Window**. |  |

## Step 9

In the **Code** for **MainWindow.xaml.cs** there be a **Method** of **myButton\_Click(...)** this should be **Removed** by removing the following:

private void myButton\_Click(object sender, RoutedEventArgs e)

{

myButton.Content = "Clicked";

}

## Step 10

Once **myButton\_Click(...)** has been removed, type in the following **Code** below the end of the **Constructor** of **public MainWindow() { ... }**:

private void Value\_QuerySubmitted(AutoSuggestBox sender,

AutoSuggestBoxQuerySubmittedEventArgs args)

{

Library.Add(Display, Value.Text);

}

private void Remove\_Click(object sender, RoutedEventArgs e)

{

Library.Remove(Display, sender);

}

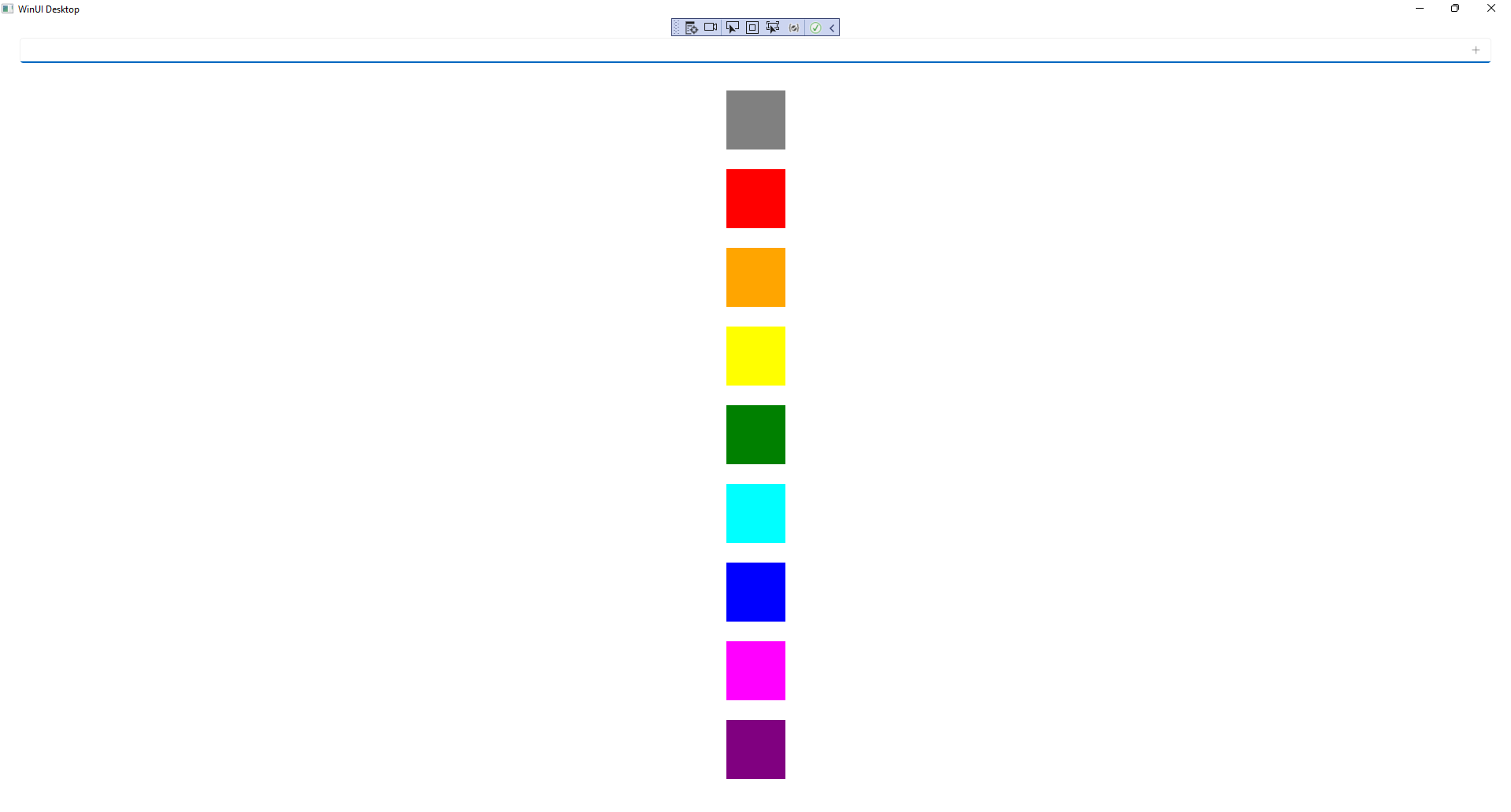
The **Method** of **Value\_QuerySubmitted** will call the **Method** within *Library.cs* of **Add** from **Library** passing in the **ListView** and also passes in the **Property** for **Text**. The **Method** of **Remove\_Click** will call the **Method** for **Remove** with the **ListView** and the **object** for **sender**.

## Step 11

|  |  |
| --- | --- |
| That completes the **Windows App SDK** application. In **Visual Studio 2022** from the **Toolbar** select **ParallaxView (Package)** to **Start** the application. |  |

## Step 12

Once running you should see a **AutoSuggestBox** and some **Rectangle** elements.

****

## Step 13

You can type into the **AutoSuggestBox** any values and then press **Enter** or **Click** on the **+** option to add them, then when you add a few, you can try scrolling the **ListView** and you should see the **Parallax View** in action.



## Step 14

|  |  |
| --- | --- |
| To **Exit** the **Windows App SDK** application, select the **Close** button from the top right of the application as that concludes this **Tutorial** for **Windows App SDK** from [tutorialr.com](https://tutorialr.com)! |  |