**Shade Effect** shows how to create a Shadow Effect on an element – in this case the **Visual Studio** logo, triggered with **Accept** and cleared with **Clear**

## Step 1

|  |  |
| --- | --- |
|  | Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project** |
| A screenshot of a cell phone  Description automatically generated | Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **ShadeEffect** and select **Create** |
| A screenshot of a social media post  Description automatically generated | Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK** |

**Target Version** will control the most recent features of **Windows 10** your application can use. To make sure you always have the most recent version, check for any **Notifications** or **Updates** in **Visual Studio 2019**

## Step 2

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | Choose **Project** then **Add New Item...** from the **Menu** in **Visual Studio 2019** |

## Step 3

|  |  |
| --- | --- |
| A close up of a logo  Description automatically generated | Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add** |

## Step 4

In the **Code** View of **Library.cs** will be displayed and in this the following should be entered:

|  |
| --- |
| using System.Numerics;  using Windows.UI;  using Windows.UI.Composition;  using Windows.UI.Xaml.Controls;  using Windows.UI.Xaml.Hosting;  using Windows.UI.Xaml.Shapes;  public class Library  {  private SpriteVisual \_visual;  public void Accept(ref Path path, ref Border border)  {  Compositor compositor = ElementCompositionPreview  .GetElementVisual(path).Compositor;  \_visual = compositor.CreateSpriteVisual();  \_visual.Size = new Vector2((float)path.ActualWidth,  (float)path.ActualHeight);  DropShadow shadow = compositor.CreateDropShadow();  shadow.Offset = new Vector3(10, 10, 0);  shadow.Mask = path.GetAlphaMask();  shadow.Color = Colors.Black;  \_visual.Shadow = shadow;  ElementCompositionPreview.SetElementChildVisual(border, \_visual);  }  public void Clear()  {  \_visual.Shadow = null;  }  } |

There is a SpriteVisual **member** and the Accept **method** uses CreateSpriteVisual combined with DropShadow to set up a Drop Shadow that will form the Shade Effect which will be applied to the Shadow **property** of the SpriteVisual and then will use the SetElementChildVisual Method to set the Border to be associated with the SpriteVisual. Clear **method** will set the SpriteVisual **property** of Shadow to null

## Step 5

|  |  |
| --- | --- |
|  | In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml** |

## Step 6

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019** |

## Step 7

In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **Grid** and **/Grid** elements enter the following **XAML**:

|  |
| --- |
| <Viewbox Margin="100">  <Grid Height="400" Width="400">  <Border x:Name="ShadowElement"/>  <Path Name="Logo" Fill="#FF5C2D91" Stretch="Uniform"  Data="M27.021,0l8.897,3.592v28.815L26.938,36L12.653,  21.796l-9.061,7.021L0,27.021V8.979l3.592-1.714l9.061,  7.102 L27.021,0z M3.592,12.653v10.939l5.388-5.551L3.592,  12.653z M17.633,18.041l9.306,7.348V10.693L17.633,18.041z"/>  </Grid>  </Viewbox>  <CommandBar VerticalAlignment="Bottom">  <AppBarButton Icon="Accept" Label="Accept" Click="Accept\_Click"/>  <AppBarButton Icon="Clear" Label="Clear" Click="Clear\_Click"/>  </CommandBar> |

The first block of **XAML** features a **Viewbox** which contains a **Grid** with a **Border** which will represent the Shade Effect and **Path** within which represents the Logo. The second block of **XAML** is the **CommandBar** which contains **Accept** – to apply the Shade Effect to the Logo and **Clear** – to remove the Shade Effect

## Step 8

|  |  |
| --- | --- |
|  | Choose **View** then **Code** from the **Menu** in **Visual Studio 2019** |

## Step 9

Once in the **Code** View, below the end of **public MainPage() { ... }** the following Code should be entered:

|  |
| --- |
| Library library = new Library();  private void Accept\_Click(object sender, RoutedEventArgs e)  {  library.Accept(ref Logo, ref ShadowElement);  }  private void Clear\_Click(object sender, RoutedEventArgs e)  {  library.Clear();  } |

Below the **MainPage(...)** method an instance of the Library **Class** is created. In the Accept\_Click(...) **Event** handler the Accept **method** is called and in the Clear\_Click(...) **event** handler the Clear **method** of the Library **class** is called

## Step 10

|  |  |
| --- | --- |
|  | That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application |

## Step 11

Once the Application is running you can then select **Accept** to start the Light Effect and use **Clear** to remove the effect



## Step 12

|  |  |
| --- | --- |
| A picture containing object  Description automatically generated | To Exit the Application, select the **Close** button in the top right of the Application |