**Light Effect** shows how to create a **PointLight** on an element – in this case the **Visual Studio** Logo and uses an animation to demonstrate the effect passing over the logo, triggered with **Play** and cleared with **Stop**

## 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 **LightEffect** 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;  using System.Numerics;  using Windows.UI;  using Windows.UI.Composition;  using Windows.UI.Xaml.Hosting;  using Windows.UI.Xaml.Shapes;  public class Library  {  private PointLight \_light;  public void Play(ref Path path)  {  Compositor compositor = ElementCompositionPreview  .GetElementVisual(path).Compositor;  Visual visual =  ElementCompositionPreview.GetElementVisual(path);  \_light = compositor.CreatePointLight();  \_light.Offset = new Vector3(-(float)path.ActualWidth \* 2,  (float)path.ActualHeight / 2, (float)path.ActualHeight);  \_light.CoordinateSpace = visual;  \_light.Color = Colors.White;  \_light.Targets.Add(visual);  ScalarKeyFrameAnimation animation =  compositor.CreateScalarKeyFrameAnimation();  animation.IterationBehavior = AnimationIterationBehavior.Forever;  animation.InsertKeyFrame(1, 2 \* (float)path.ActualWidth);  animation.Duration = TimeSpan.FromSeconds(5.0f);  \_light.StartAnimation("Offset.X", animation);  }  public void Stop()  {  if (\_light != null)  {  \_light.Targets.RemoveAll();  }  }  } |

There is a PointLight **member**, the Play **method** uses GetElementVisual to get the Path to apply the effect to then the PointLight is set up and a ScalarKeyFrameAnimation animation is set up which will “move” the PointLight across the Path. Stop **method** uses RemoveAll to clear the effect and has null check to prevent a NullReferenceException

## 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">  <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="Play" Label="Play" Click="Play\_Click"/>  <AppBarButton Icon="Stop" Label="Stop" Click="Stop\_Click"/>  </CommandBar> |

The first block of **XAML** is a **Viewbox** which contains a **Grid** with a **Path** within which represents the Logo. The second block of **XAML** is the **CommandBar** which contains **Play** – to apply the Light Effect to the Logo and **Stop** – to remove the Light Effect from the Logo

## 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 Play\_Click(object sender, RoutedEventArgs e)  {  library.Play(ref Logo);  }  private void Stop\_Click(object sender, RoutedEventArgs e)  {  library.Stop();  } |

Below the **MainPage(...)** method an instance of the Library **Class** is created. In the Play\_Click(...) **Event** handler the Play **method** is called, and in the Stop\_Click(...) **event** handler the Stop **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 **Play** to start the Light Effect and use **Stop** 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 |