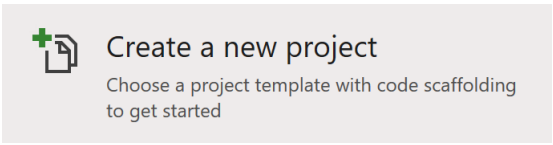
# Universal Windows Platform – Image Rotate

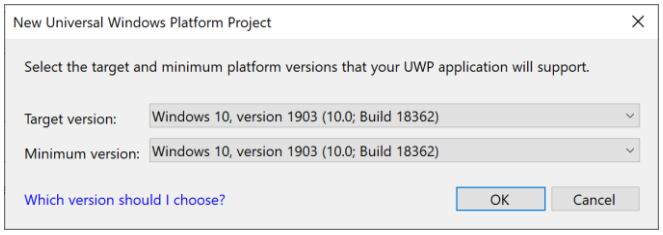
**Image Rotate** shows how to use a **Storyboard** to create a simple **DoubleAnimation** to rotate an image in the **X**, **Y** and **X** axis

## 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**

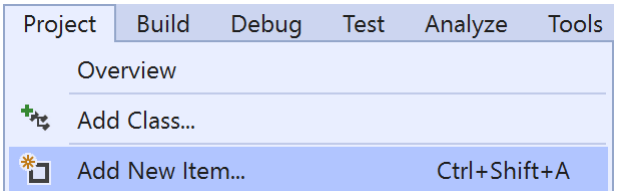

Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **ImageRotate** and select **Create**




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




Choose **Project** then **Add New Item...** from the **Menu** in **Visual Studio 2019**

## Step 3


Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add**

# Universal Windows Platform – Image Rotate

## Step 4

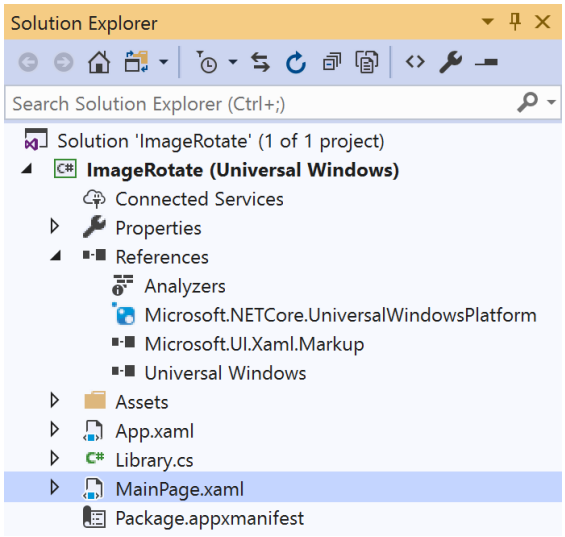
In the **Code** View of **Library.cs** will be displayed and in this the following should be entered:

```
using System;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Media.Animation;

public class Library
{
    private bool _rotating = false;
    private Storyboard _rotation = new Storyboard();

    public void Rotate(string axis, ref Image target)
    {
        if (_rotating)
        {
            _rotation.Stop();
            _rotating = false;
        }
        else
        {
            DoubleAnimation animation = new DoubleAnimation
            {
                From = 0.0,
                To = 360.0,
                BeginTime = TimeSpan.FromSeconds(1),
                RepeatBehavior = RepeatBehavior.Forever
            };
            Storyboard.SetTarget(animation, target);
            Storyboard.SetTargetProperty(animation,
            $"(UIElement.Projection).(PlaneProjection.Rotation{axis})");
            _rotation.Children.Clear();
            _rotation.Children.Add(animation);
            _rotation.Begin();
            _rotating = true;
        }
    }
}
```

There is a `using` statement to include functionality needed for the application. `Storyboard` is used as part of the later `DoubleAnimation` which will animate between 0 and 360 and this will repeat Forever after 1 second. The `PlaneProjection.Rotation` value for each Axis is set on the `UIElement` which in this case is the `Image` named `target` and uses String Interpolation Syntax or `$`

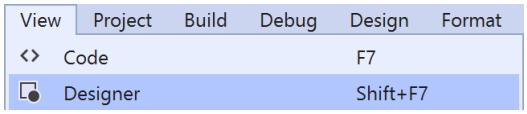
# Universal Windows Platform – Image Rotate

## Step 5




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

## Step 6




Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

## Step 7

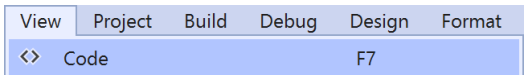
In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **Grid** and **/Grid** elements enter the following **XAML**:

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>
    <TextBox Grid.Row="0" Name="Value" Margin="20"
    InputScope="Url" KeyDown="Go_KeyDown"/>
    <Image Grid.Row="1" Margin="100" Stretch="Uniform" Name="Display">
        <Image.Projection>
            <PlaneProjection/>
        </Image.Projection>
    </Image>
</Grid>
<CommandBar VerticalAlignment="Bottom">
    <AppBarButton Icon="RepeatAll" Label="Pitch" Click="Pitch_Click"/>
    <AppBarButton Icon="Rotate" Label="Roll" Click="Roll_Click"/>
    <AppBarButton Icon="Refresh" Label="Yaw" Click="Yaw_Click"/>
</CommandBar>
```

The first block of XAML the main user interface features a TextBox. The second block of XAML is is the CommandBar which contains Pitch – to rotate the X Axis, Roll – to rotate the Y Axis and Yaw – to rotate the Z Axis

# Universal Windows Platform – Image Rotate

## Step 8




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

## Step 9

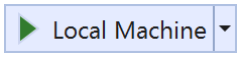
Once in the **Code** View, below the end of **`public MainPage() { ... }`** the following Code should be entered:

```
Library library = new Library();

private void Go_KeyDown(object sender, KeyRoutedEventArgs e)
{
    if (e.Key == Windows.System.VirtualKey.Enter)
    {
        Display.Source = new Windows.UI.Xaml.Media.Imaging
        .BitmapImage(new Uri(Value.Text));
    }
}

private void Pitch_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("X", ref Display);
}

private void Roll_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("Y", ref Display);
}

private void Yaw_Click(object sender, RoutedEventArgs e)
{
    library.Rotate("Z", ref Display);
}
```

Below the MainPage(...) method an instance of the `Library` Class is created. In the `Go_KeyDown(...)` Event handler the Image has the `Source` property set to the contents any URL entered in the TextBox, the `Pitch_Click(...)`, `Roll_Click(...)` and `Yaw_Click(...)` event handler will use the `Rotate` method to set which Axis the Image should be rotated by

# Universal Windows Platform – Image Rotate

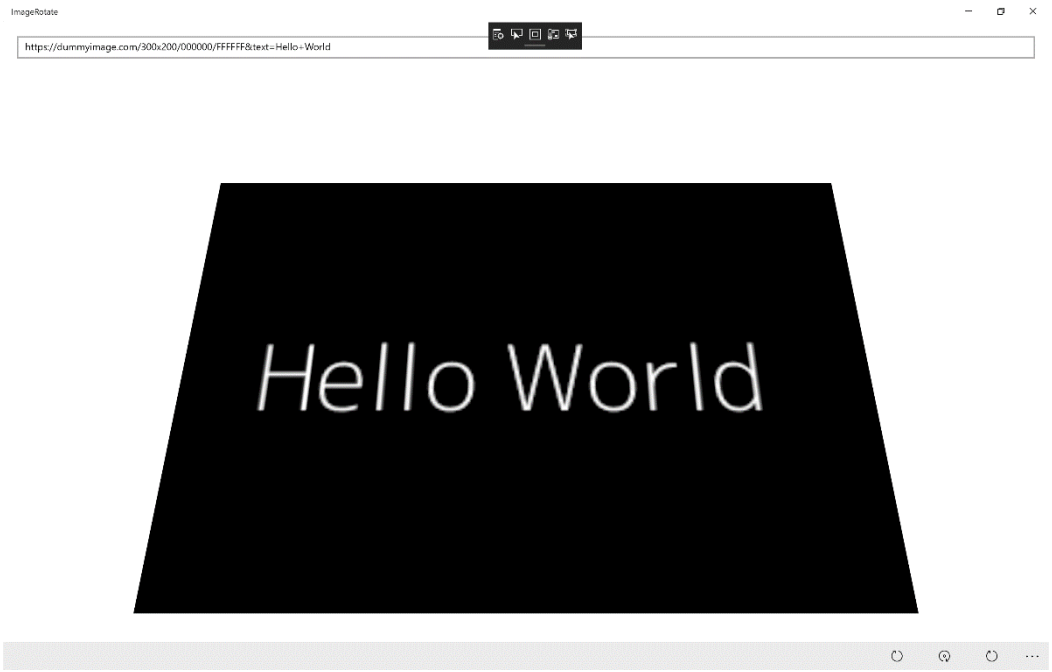
## Step 10




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 type in the URL of any image e.g. https://dummyimage.com/300x200/000000/FFFFFF&text=Hello+World then press or tap Enter to load it, then use the **Pitch**, **Roll** or **Yaw** buttons to rotate the **Image**




## Step 12

To Exit the Application, select the **Close** button in the top right of the Application