Universal Windows Platform – Image Rotate

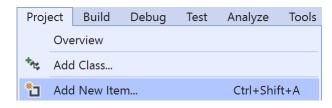
Image Rotate shows how to use a **Storyboard** to create a simple **DoubleAnimation** to rotate an image in the **X**, **Y** and **X** axis

Step 1

Create a new project Choose a project template with code scaffolding to get started	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
Blank App (Universal Windows)	Windows) and select Next and then in
A project for a single-page Universal Windows Platform (UWP) app that has no predefined controls or layout.	Configure your new project enter the
C# Windows Xbox UWP Desktop	Project name as ImageRotate and select
	Create
New Universal Windows Platform Project	× Finally, in New Universal Windows Platform
Select the target and minimum platform versions that your UWP application will support.	Project pick the Target version and
Target version: Windows 10, version 1903 (10.0; Build 18362)	Minimum version to be at least Windows
Minimum version: Windows 10, version 1903 (10.0; Build 18362)	10, version 1903 (10.0; Build 18362) and
Which version should I choose? OK Cancel	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

Code File

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



Visual C#



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:

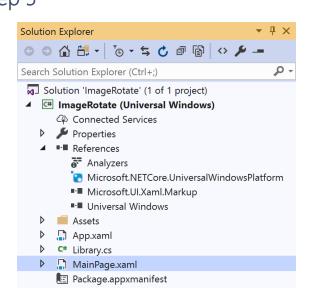
```
using System;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Media.Animation;
public class Library
{
    private bool _rotating = false;
    private Storyboard _rotation = new Storyboard();
    public void Rotate(string axis, ref Image target)
    {
        if (_rotating)
        {
            _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

View	Project	Build	Debug	Design	Format
<> c	ode			F7	
Designer		Shift+F7			

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> <rowdefinition height="*"></rowdefinition> </grid.rowdefinitions> <textbox <br="" grid.row="0" margin="20" name="Value">InputScope="Url" KeyDown="Go_KeyDown"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image.projection> </image.projection> </textbox></grid> <commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar>		
<pre><rowdefinition height="Auto"></rowdefinition></pre>	<grid></grid>	
<pre><rowdefinition height="*"></rowdefinition> <textbox grid.row="0" inputscope="Url" keydown="Go_KeyDown" margin="20" name="Value"></textbox> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image.projection> </image.projection> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton> </pre>	<grid.< th=""><th>.RowDefinitions></th></grid.<>	.RowDefinitions>
 <textbox <br="" grid.row="0" margin="20" name="Value">InputScope="Url" KeyDown="Go_KeyDown"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image.projection> <planeprojection> </planeprojection></image.projection> <commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar></textbox>	< R	RowDefinition Height="Auto"/>
<textbox <br="" grid.row="0" margin="20" name="Value">InputScope="Url" KeyDown="Go_KeyDown"/> <image grid.row="1" margin="100" name="Display" stretch="Uniform"/> <image.projection> <planeprojection> </planeprojection></image.projection> <commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar></textbox>	< R	RowDefinition Height="*"/>
<pre>InputScope="Url" KeyDown="Go_KeyDown"/></pre>	<th>d.RowDefinitions></th>	d.RowDefinitions>
<pre><image grid.row="1" margin="100" name="Display" stretch="Uniform"/></pre>	<textb< th=""><th>Box Grid.Row="0" Name="Value" Margin="20"</th></textb<>	Box Grid.Row="0" Name="Value" Margin="20"
<pre><image.projection></image.projection></pre>	InputS	<pre>Scope="Url" KeyDown="Go_KeyDown"/></pre>
<pre></pre>	< Image	e Grid.Row="1" Margin="100" Stretch="Uniform" Name="Display">
 <commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar>	<1	Image.Projection>
 <commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar>		<planeprojection></planeprojection>
<commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar>	</th <th>/Image.Projection></th>	/Image.Projection>
<commandbar verticalalignment="Bottom"> <appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></commandbar>	<th>ge></th>	ge>
<pre><appbarbutton click="Pitch_Click" icon="RepeatAll" label="Pitch"></appbarbutton></pre>		
	<commandba< th=""><th>ar VerticalAlignment="Bottom"></th></commandba<>	ar VerticalAlignment="Bottom">
	< AppBa	arButton Icon="RepeatAll" Label="Pitch" Click="Pitch_Click"/>
<pre><appbarbutton click="Roll_Click" icon="Rotate" label="Roll"></appbarbutton></pre>	< AppBa	arButton Icon="Rotate" Label="Roll" Click="Roll_Click"/>
<pre><appbarbutton click="Yaw_Click" icon="Refresh" label="Yaw"></appbarbutton></pre>	< AppBa	arButton Icon="Refresh" Label="Yaw" Click="Yaw_Click"/>
	<th>Bar></th>	Bar>

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

 View
 Project
 Build
 Debug
 Design
 Format

 <>
 Code
 F7

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

▶ Local Machine ▼

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/FFFFF&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



