

Name of Experiment: Use of dedicated button for Camera Operation.	Exp No: WP10
Background: Student should have basic knowledge of C#.	
Summary: This experiment is based on camera operation using dedicated button. We can see that how this single dedicated button can be used for firing three different events.	
Learning Objective: Student should now understand the camera operation using the dedicated button.	
Target Platforms: This experiment is tested on Windows Phone Emulator and Windows Phone (Lumia - 800).	
<p>Procedure:</p> <p>Step1. Repeat all steps [1-4] as in experiment WP6.[Refer the fig. no 1]</p> <p>Step2. Initialise the Photo camera object, before the Main Page constructor.</p> <p>Step3. Inside the OnNavigatedTo method we will initialise the three different events associated with the hardware button as <code>CameraButtons.ShutterKeyHalfPressed+=new System.EventHandler(CameraButtons.ShutterKeyHalfPressed);</code> <code>CameraButtons.ShutterKeyPressed+=new System.EventHandler(CameraButtons.ShutterKeyPressed);</code> <code>CameraButtons.ShutterKeyReleased+=new System.EventHandler(CameraButtons.ShutterKeyReleased);</code></p> <p>Step4. Now initialise the mycamera.Initialized method in order to initialise the camera hardware, we also set the videoBrush.SetSource(mycamera).</p> <p>Step5. Define the body for ShutterKeyHalfPressed, set the camera for auto focus.</p> <p>Step6. Define the body for ShutterKeyPressed, set the camera capture event.</p> <p>Step7. Define the body for ShutterKeyReleased , cancel the camera focus.</p> <p>Step8. Save all the changes made to the experiment.</p> <p>Step9. Press F5, to debug the experiment and deploy it on the Windows Phone Emulator.[Refer fig. no 2]</p>	
Source Code	Comments
<p>MainPage.xaml</p> <pre><phone: PhoneApplicationPage x: Class="WP11.MainPage" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:phone="clr- namespace: Microsoft.Phone.Controls; assembly=Microsoft.Phone" xmlns:shell="clr- namespace: Microsoft.Phone.Shell; assembly=Microsoft.Phone" xmlns:d="http://schemas.microsoft.com/expression/bl end/2008" xmlns:mc="http://schemas.openxmlformats.org/markup-</pre>	

```

compatibility/2006"
    mc:Ignorable="d" FontFamily="{StaticResource
PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Landscape" Orientation="LandscapeLeft"
    shell:SystemTray.IsVisible="False" d:DesignHeight="480"
d:DesignWidth="800">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="640"/>
            <ColumnDefinition Width="150"/>
        </Grid.ColumnDefinitions>
        <Canvas x:Name="viewfinderCanvas" >
            <Canvas.Background>
                <VideoBrush x:Name="videoBrush"/>
            </Canvas.Background>
            <TextBlock Canvas.Left="0" Canvas.Top="414" Height="60"
Name="txtDebug" FontSize="22" FontWeight="ExtraBold" Width="513" />
        </Canvas>
        <!--TitlePanel contains the name of the application and
page title-->

        <!--ContentPanel - place additional content here-->
    </Grid>

</phone:PhoneApplicationPage>

```

MainPage.xaml.cs

```

using Microsoft.Phone.Controls;
using Microsoft.Devices;
using Microsoft.Xna.Framework;
using System;

namespace WP11
{
    public partial class MainPage : PhoneApplicationPage
    {
        PhotoCamera mycamera;
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }
        protected override void
OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            //base.OnNavigatedTo(e); if
(PhotoCamera.IsCameraTypeSupported(CameraType.Primary) == true)
            {
                mycamera = new Microsoft.Devices.PhotoCamera(CameraType.Primary);
                CameraButtons.ShutterKeyHalfPressed+=new
System.EventHandler(CameraButtons_ShutterKeyHalfPressed);
                CameraButtons.ShutterKeyPressed+=new
System.EventHandler(CameraButtons_ShutterKeyPressed);
                CameraButtons.ShutterKeyReleased+=new

```

← event on Half
Button pressed
← event on Full
Pressed Button

```

System. EventHandler(CameraButtons_ShutterKeyReleased);
mycamera.Initialized += new
System. EventHandler<CameraOperationCompletedEventArgs>(mycamera_Initiali
lized);
        videoBrush.SetSource(mycamera);
    }
    else {
        this.Dispatcher.BeginInvoke(delegate()
        {
            this.txtDebug.Text = "camera is not supported in
the device"; });
        }
    }
    protected override void
OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e)
    { //base.OnNavigatedFrom(e);
        if (mycamera != null) {
            mycamera.Dispose();
            mycamera.Initialized -= mycamera_Initialized;
            CameraButtons.ShutterKeyHalfPressed -=
CameraButtons_ShutterKeyHalfPressed;
            CameraButtons.ShutterKeyPressed -=
CameraButtons_ShutterKeyPressed;
            CameraButtons.ShutterKeyReleased -=
CameraButtons_ShutterKeyReleased;
        }
    }
    public void mycamera_Initialized(object sender,
CameraOperationCompletedEventArgs e) {
        if (mycamera != null) {
            this.Dispatcher.BeginInvoke(delegate()
            {
                this.txtDebug.Text = "camera is Initialised";
            });
        }
    }
    public void CameraButtons_ShutterKeyHalfPressed(object sender,
EventArgs e) {
        if (mycamera != null) {
            try
            {
                this.Dispatcher.BeginInvoke(delegate()
                {
                    this.txtDebug.Text = "Half Key Pressed: Focus";
                });
                mycamera.Focus();
            }
            catch (Exception focuserror) {
                this.Dispatcher.BeginInvoke(delegate() {
                    this.txtDebug.Text = focuserror.Message; });
            }
        }
    }
    public void CameraButtons_ShutterKeyPressed(object sender,
EventArgs e) {
        if (mycamera != null) {
            mycamera.CaptureImage();
        }
    }
    public void CameraButtons_ShutterKeyReleased(object
sender, EventArgs e) {
        if (mycamera != null) {

```

← event on Button
Released

```
mycamera.CancelFocus();  
    }  
    }  
}
```

Screenshots



Fig. no 1 UI design for experiment WP10



Fig. no 2 Snapshot taken from Windows Phone Emulator

Observations:

It is observed that using the single dedicated button, we can call the three basic camera operation events.