

Name of Experiment: Camera Operation (Focus Setting)

Exp No: WP7

Background: Student should have basic knowledge of C#.

Summary: This experiment is again based on the camera operation, in this experiment user can set the Focus of the camera lens at his desired location on the mobile screen as well as he/she can set for Auto Focusing.

Learning Objective: Student now able to set the focus of the camera lens in three modes such as auto focusing, focusing at centre and focusing on the desired location on the mobile screen by simply tapping on the mobile screen.

Target Platforms: The application is tested on Windows Phone Emulator and Windows Phone (Lumia 800).

Procedure:

Step1. Repeat all steps [1-6] as in experiment no WP6.

Step2. Add two buttons, as shown in UI. Set the Content of button as FOCUS and AUTOFS and add their click event handlers.[Refer fig. no 1]

Step3. Now, initialise a Boolean variable `_isfocus= true` before the MainPage constructor.

Step4. Add one new event handler named `focus_Tapped` inside the `OnNavigationTo` method in order to get the coordinates of the tap location.[Refer Source Code MainPage.xaml.cs]

Step5. Now, perform the action on `focusButton_Click`, check whether the focusing is supported by the device by applying `IsFocusAtPoint.Supported`. After that set the Focus of lens at the centre using `FocusAtPoint(0.5,0.5)`.

Step6. Again perform the action against `AFButton_Click`, and call the `AutoFocusCompleted` event handler.

Step7. Now, define the body for `focus_Tapped` event to get the tap coordinates on the mobile screen and place the focus of lens at that place.

Step8. Now save all changes made to the experiment.

Step9. Press F5, to deploy the application on the Windows Emulator for the debugging purpose.[Refer fig. no 2,3,4]

Source Code	Comments
<pre> MainPage.xaml <!--LayoutRoot is the root grid where all page content is placed--> <Grid x:Name="LayoutRoot" Background="Transparent"> <Grid.ColumnDefinitions> <ColumnDefinition Width="640" /> <ColumnDefinition Width="160" /> </Grid.ColumnDefinitions> <Canvas x:Name="viewfinderCanvas" Width="640" Height="480" HorizontalAlignment="Left" > <!--Camera viewfinder --> <Canvas.Background> <VideoBrush x:Name="viewfinderBrush" /> </Canvas.Background> <TextBlock x:Name="focusBracket" FontSize="40" Text="[]" HorizontalAlignment="Center" VerticalAlignment="Center" FontWeight="ExtraBold" Visibility="Collapsed" Canvas.Left="296" Canvas.Top="192" /> </Canvas> <!--Button StackPanel to the right of viewfinder--> <StackPanel Grid.Column="1" > <Button x:Name="ShutterButton" Content="CAPTURE" Click="ShutterButton_Click" FontSize="26" FontWeight="ExtraBold" Height="75" /> <Button x:Name="FocusButton" Content="FOCUS" Click="FocusButton_Click" FontSize="26" FontWeight="ExtraBold" Height="75" /> <Button x:Name="AFButton" Content="AUTOFS" Click="AFButton_Click" FontSize="26" FontWeight="ExtraBold" Height="75" /> </StackPanel> <!--Used for debugging --> <TextBlock Height="40" HorizontalAlignment="Left" Margin="8,428,0,0" Name="txtDebug" VerticalAlignment="Top" Width="626" FontSize="24" FontWeight="ExtraBold" /> </Grid> MainPage.xaml.cs using System.Windows; using Microsoft.Phone.Controls; using Microsoft.Devices; using Microsoft.Xna.Framework.Media; using System; using System.Windows.Input; using System.Windows.Controls; namespace Cam_Test { public partial class MainPage : PhoneApplicationPage { PhotoCamera cam = null; int savedCounter = 0; bool _isfocus = true; MediaLibrary library = new MediaLibrary(); // Constructor public MainPage() </pre>	<pre> ← Button Content= "CAPTURE" Name= "ShutterButton" ←Content= "FOCUS" Name= "FocusButton" ←Content= "AUTOFS" Name= "AFButton" ←Microsoft.Xna.Framework. </pre>

```

{
    InitializeComponent();
}

protected override void
OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    //base.OnNavigatedTo(e);
    cam = new PhotoCamera();
    viewfinderBrush.SetSource(cam);
    cam.Initialized += new
EventHandler<CameraOperationCompletedEventArgs>(cam_Initialized);
    cam.CaptureImageAvailable += new
EventHandler<ContentReadyEventArgs>(cam_CaptureImageAvailable);
    cam.CaptureCompleted += new
EventHandler<CameraOperationCompletedEventArgs>(cam_CaptureCompleted);
    viewfinderCanvas.Tap += new
EventHandler<System.Windows.Input.GestureEventArgs>(focus_Tapped);
}
protected void
OnNavigatedFrom(System.Windows.Navigation.NavigatingCancelEventArgs e)
{
    //base.OnNavigatedFrom(e);
    if (cam != null) {
        cam.Dispose();
        cam.Initialized -= cam_Initialized;
        cam.CaptureCompleted -= cam_CaptureCompleted;
        cam.CaptureImageAvailable -= cam_CaptureImageAvailable;
        cam.AutoFocusCompleted -= cam_AutoFocusCompleted;
    }
}

public void cam_Initialized(object sender,
Microsoft.Devices.CameraOperationCompletedEventArgs e)
{
    if (e.Succeeded) {
        Deployment.Current.Dispatcher.BeginInvoke(delegate()
        {
            txtDebug.Text = "Camera is Initialised";
        });
    }
}

public void cam_CaptureCompleted(object
sender, CameraOperationCompletedEventArgs e) {
    savedCounter++;
    Deployment.Current.Dispatcher.BeginInvoke(delegate()
    {
        txtDebug.Text = "Picture is saved for you";
    });
}

public void cam_AutoFocusCompleted(object
sender, CameraOperationCompletedEventArgs e) {
    _isfocus = true;
    Deployment.Current.Dispatcher.BeginInvoke(delegate()
    {
        txtDebug.Text = "Auto Focusing Completed";
        focusBracket.Visibility = Visibility.Collapsed;
    });
}

public void cam_CaptureImageAvailable(object
sender, Microsoft.Devices.ContentReadyEventArgs e) {

```

←System.Windows.Input

```

string filename = savedCounter + ".jpg";
library.SavePictureToCameraRoll(filename, e.ImageStream);
}

private void ShutterButton_Click(object sender, RoutedEventArgs e)
{
if (cam != null && _isfocus)
{
cam.CaptureImage();
}
else
{
Deployment.Current.Dispatcher.BeginInvoke(delegate()
{
txtDebug.Text = "Camera device is not available";
});
}
}

private void FocusButton_Click(object sender, RoutedEventArgs e)
{
if (cam.IsFocusAtPointSupported) {
focusBracket.Visibility = Visibility.Visible;
cam.FocusAtPoint(0.5,0.5);

}
Deployment.Current.Dispatcher.BeginInvoke(delegate()
{
txtDebug.Text = "Focussing at center";
});
}

private void AFButton_Click(object sender, RoutedEventArgs e)
{
cam.AutoFocusCompleted += new
EventHandler<CameraOperationCompletedEventArgs>(cam_AutoFocusCompleted);
if (cam.IsFocusSupported) {
_isfocus = false;
focusBracket.Visibility = Visibility.Visible;
cam.Focus();

}
}

public void focus_Tapped(object sender, GestureEventArgs e)
{
if (cam != null)
{
if (cam.IsFocusAtPointSupported == true)
{
try
{ // Determine the location of the tap.
Point tapLocation = e.GetPosition(viewfinderCanvas);

// Position the focus brackets with the estimated offsets.
focusBracket.SetValue(Canvas.LeftProperty, tapLocation.X - 30);
focusBracket.SetValue(Canvas.TopProperty, tapLocation.Y - 28);

// Determine the focus point.
double focusXPercentage = tapLocation.X / viewfinderCanvas.Width;
double focusYPercentage = tapLocation.Y / viewfinderCanvas.Height;

// Show the focus brackets and focus at point.
focusBracket.Visibility = Visibility.Visible;

```

←System.Windows.Controls

```

cam.FocusAtPoint(focusXPercentage, focusYPercentage);

// Write a message to the UI.
this.Dispatcher.BeginInvoke(delegate()
{
txtDebug.Text = String.Format("Camera focusing at point: {0:N2} , {1:N2}",
focusXPercentage, focusYPercentage);
});
}
catch (Exception focusError)
{
// Cannot focus when a capture is in progress.
this.Dispatcher.BeginInvoke(delegate()
{
// Write a message to the UI.
txtDebug.Text = focusError.Message;
// Hide focus brackets.
focusBracket.Visibility = Visibility.Collapsed;});
}
}
else {
// Write a message to the UI.
this.Dispatcher.BeginInvoke(delegate()
{ txtDebug.Text = "Camera does not support FocusAtPoint.";});
}
}
}
}
}

```

← System

Screenshots



Fig no1 UI design for the experiment no WP6



Fig. no 2 Output Screen running on Windows Emulator

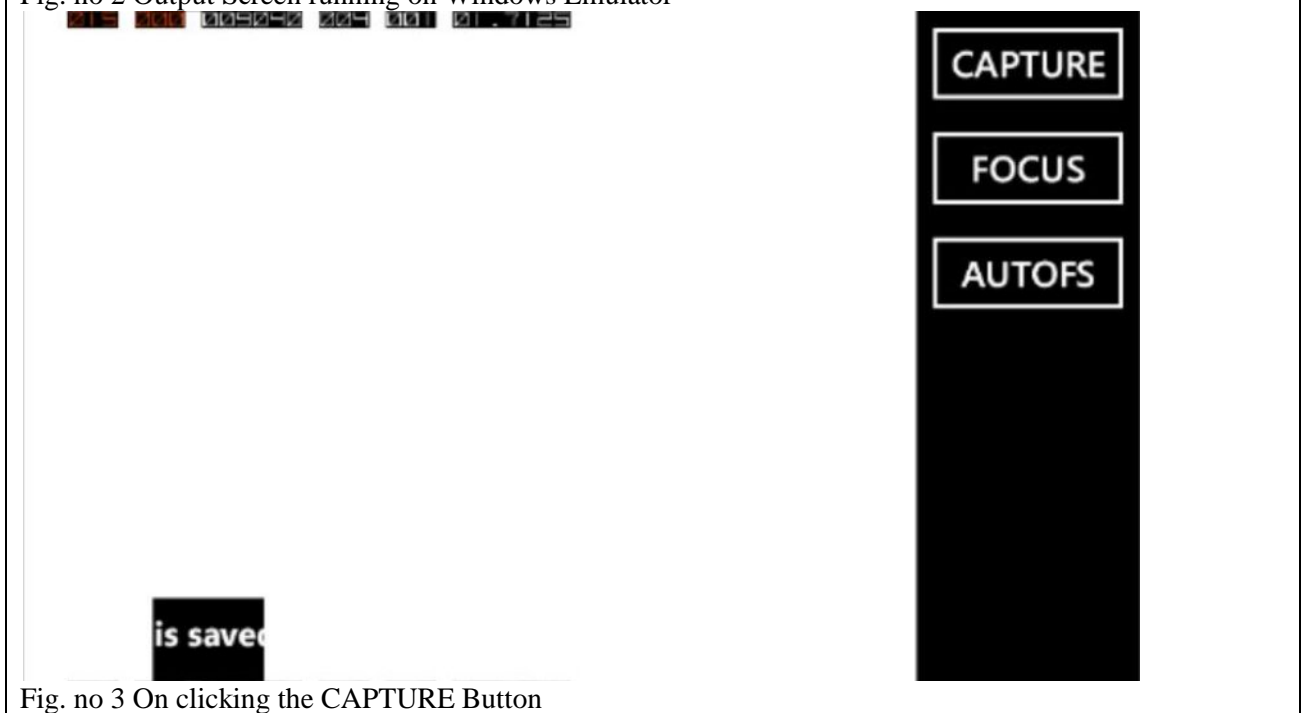


Fig. no 3 On clicking the CAPTURE Button



Fig. no 4 After clicking on AUTOFS Button

Observations: It is observed that focusing is also an important feature of the camera device and must be enabled in order to capture the desired object.