Name of Experiment: Display Accelerometer Data	**Exp No:** SSN1

Background: Student should have a basic knowledge of C#.

Summary: Accelerometer is a type of sensor that detects the force of gravity along with any forces resulting from the movement of the Phone. The acceleration value is expressed as a 3D vector representing the acceleration components in the X,Y,Z axes in gravitational Units.

Learning Objective: Student can understand the concept of 3D rotation of the Mobile. This experiment explains the basic of 3D game development.

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

Procedure:

Step1. Repeat the steps [1-4] as in experiment no MC1(HelloWorld).[Refer Exp No MC1]

Step2. Add Reference for Microsoft.Devices.Sensors and Microsoft.Xna.Framework.

Step3. Initialise the Accelerometer before the MainPage constructor.[Refer Code Section]

Step4. Inside the startButton click handler, initialise the Accelerometer object, add CurrentValueChanged event, add TimeBetweenUpdates,and also add start method to start the accelerometer.

Step5. Define a UpdateUI method to display the updating data of Accelerometer.

Step6. At last call the stop_Button event handler to stop all the services of Accelerometer.

Step7. Reset the values of X,Y,Z coordinate axis.

Step8. Save all the changes that we have made by simply pressing the Alt+S or Save option from the above pallete.

Step9. Go to Debug option from the menu and click on the start debug.

Step10. An emulator will start after few minute and Accelerometer Page is shown to you.[Refer Snapshots.]

Step11. By this way, a Display Accelerometer Data program can be deployed on the WP7 emulator.

Step12. For deploying this experiment on the Window Phone device (Nokia Lumia 800).[Refer section Deployment of the WP7 application on the Target Device.]

Source Code	Comments

MainPage.xaml
 	
`<!--LayoutRoot is the root grid where all page content is placed-->`
`<Grid x:Name="LayoutRoot" Background="Transparent">`
`<Grid.RowDefinitions>`
`<RowDefinition Height="Auto"/>`
`<RowDefinition Height="*"/>`
`</Grid.RowDefinitions>`
 	
`<!--TitlePanel contains the name of the application and page title-->`
`<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">`
`<TextBlock x:Name="ExperimentTitle" Text="Exp No. SSN1" TextAlignment="Right" Style="{StaticResource PhoneTextNormalStyle}"/>`
`<TextBlock x:Name="ApplicationTitle" Text="Display Accelerometer Data" Style="{StaticResource PhoneTextNormalStyle}"/>`
`<TextBlock x:Name="PageTitle" Text="Accelerometer" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/>`
`</StackPanel>`
 	
`<!--ContentPanel - place additional content here-->`
`<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">`
`<Button Content="Start" Height="72" HorizontalAlignment="Left" Margin="20,10,0,0" Name="startButton" VerticalAlignment="Top" Width="160" Click="startButton_Click" />`
`<Button Content="Stop" Height="72" HorizontalAlignment="Right" Margin="0,10,20,0" Name="stopButton" VerticalAlignment="Top" Width="160" Click="stopButton_Click"/>`
`<TextBlock Height="30" HorizontalAlignment="Left" Margin="20,100,0,0" Name="xTextBlock" Text="X: 1.0" VerticalAlignment="Top" Foreground="Red" FontSize="28" FontWeight="Bold"/>`
`<TextBlock Height="30" HorizontalAlignment="Center" Margin="0,100,0,0" Name="yTextBlock" Text="Y: 1.0" VerticalAlignment="Top" Foreground="Green" FontSize="28" FontWeight="Bold"/>`
`<TextBlock Height="30" HorizontalAlignment="Right" Margin="0,100,20,0" Name="zTextBlock" Text="Z: 1.0" VerticalAlignment="Top" Foreground="Blue" FontSize="28" FontWeight="Bold"/>`
`<Line x:Name="xLine" X1="240" Y1="350" X2="340" Y2="350" Stroke="Red" StrokeThickness="4"></Line>`
`<Line x:Name="yLine" X1="240" Y1="350" X2="240" Y2="270" Stroke="Green" StrokeThickness="4"></Line>`
`<Line x:Name="zLine" X1="240" Y1="350" X2="190" Y2="400" Stroke="Blue" StrokeThickness="4"></Line>`
`<TextBlock Height="50" HorizontalAlignment="Center" Margin="0,524,12,0" Name="statusTextBlock" Text=" " VerticalAlignment="Top" Width="444" />`
`</Grid> </Grid>`
	← Name= “ExperimentTitle” Text= “Exp No.SSN1”
	
	←Name= “ApplicationTitle” Text= “Display Accelerometer Data”
	
	← Name= “PageTitle” Text= “Accelerometer”

MainPage.xaml.cs

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


namespace SensorsAcc
{
    public partial class Main2 : PhoneApplicationPage
    {
        Accelerometer myAccelerometer;
        public MainPage()
        {
    InitializeComponent();
  if (!Accelerometer.IsSupported) {
 statusTextBlock.Text = "Accelerometer is not supported by
the device";
       startButton.IsEnabled = false;
       stopButton.IsEnabled = false;

       }
     }

private void startButton_Click(object sender,
System.Windows.RoutedEventArgs e)
        {
            if (myAccelerometer == null)
            {
      myAccelerometer = new Accelerometer();
      myAccelerometer.TimeBetweenUpdates =
      TimeSpan.FromMilliseconds(100);
   myAccelerometer.CurrentValueChanged += new
EventHandler<SensorReadingEventArgs<AccelerometerReading>>
(myAccelerometer_CurrentValueChanged);
            }
            try
            {
   statusTextBlock.Text = "Starting Accelerometer";
   myAccelerometer.Start();
            }
      catch (InvalidOperationException ex)
            {
  statusTextBlock.Text = "Unable to start Accelerometer";
          }

    }

private void myAccelerometer_CurrentValueChanged(object
sender, SensorReadingEventArgs<AccelerometerReading> e) {
     Dispatcher.BeginInvoke(() =>
UpdateUI(e.SensorReading));
        }
        private void UpdateUI(AccelerometerReading
reading) {
    statusTextBlock.Text = "Getting data";
  Vector3 acceleration = reading.Acceleration;
 xTextBlock.Text = "X: " +
acceleration.X.ToString("0.00");
 yTextBlock.Text = "Y: " +
```

← Referencing Microsoft.Devices.Sensors; Microsoft.Xna.Framework;

← Accelerometer object declaration

← Start Button Click handler

← Calling Start()

←Calling thread for Updating the Accelerometer readings

← Updating the UI and Displaying the Values in X,Y,Z Coordinates

```
acceleration.Y.ToString("0.00");
            zTextBlock.Text = "Z: " +
acceleration.Z.ToString("0.00");

      xLine.X2 = xLine.X1 + acceleration.X * 200;
      yLine.Y2 = yLine.Y1 - acceleration.Y * 200;
      zLine.X2 = zLine.X1 - acceleration.Z * 100;
      zLine.Y2 = zLine.Y1 + acceleration.Z * 100;
    }
private void stopButton_Click(object sender,
System.Windows.RoutedEventArgs e)
        {
            if (myAccelerometer != null) { }
            myAccelerometer.Stop();
  statusTextBlock.Text = "Accelerometer is Stopped";
            Reset();
    }
      void Reset()
        {
     xTextBlock.Text = "1.0";
     yTextBlock.Text = "1.0";
     zTextBlock.Text = "1.0";
 xLine.X1 = 240; xLine.Y1 = 350; xLine.X2 = 340;
 xLine.Y2 = 350;
 yLine.X1 = 240; yLine.Y1 = 350; yLine.X2 = 240;
 yLine.Y2 = 270;
 zLine.X1 = 240; zLine.Y1 = 350; zLine.X2 = 190;
 zLine.Y2 = 400;
        }
    }
}
```

← Stop Button Click handler

Screenshots:

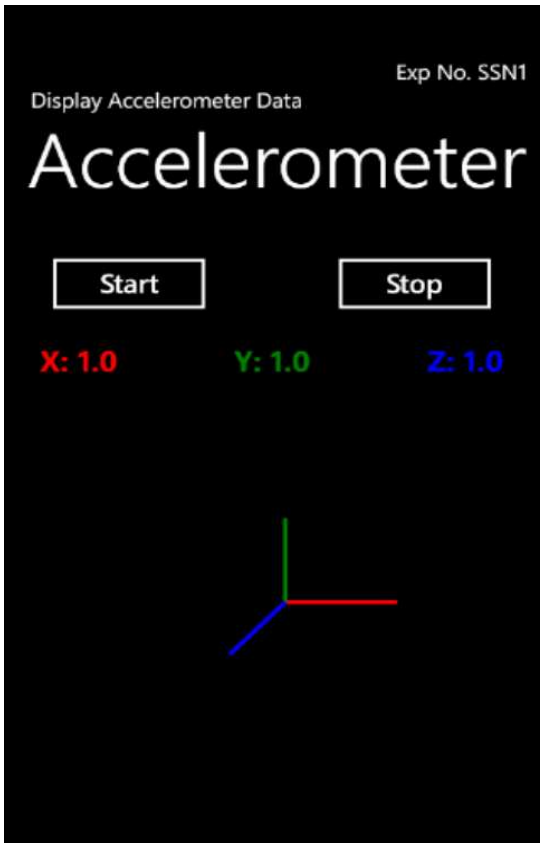

Fig. No1 Home Screen

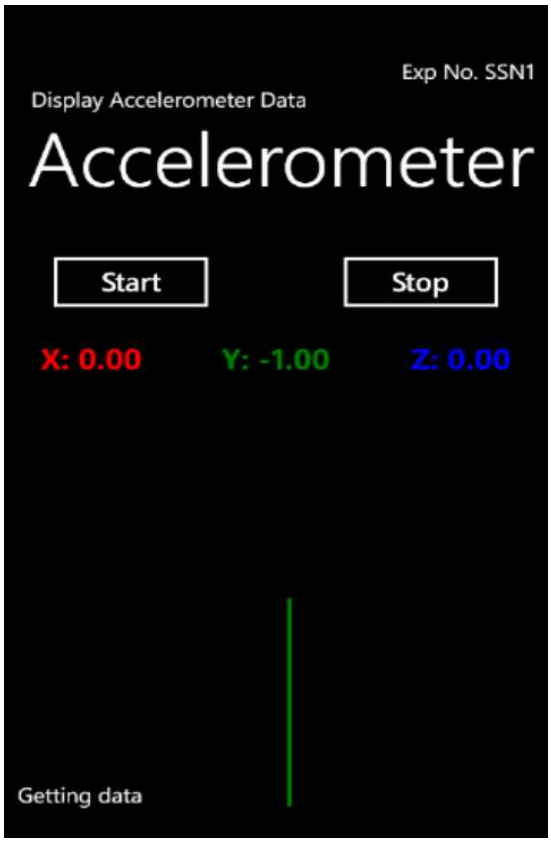

Fig.No2 After Click on Start Button

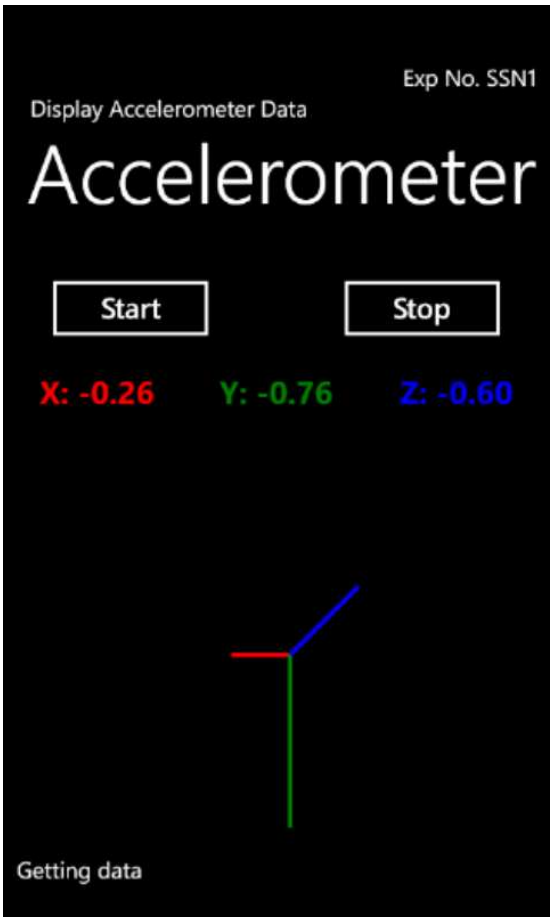

Fig. No3 Reading the Aceelerometer

Fig.No4 Keeping the Device in Z-axis only

Fig.No 5 Keeping the Device in X-axis only

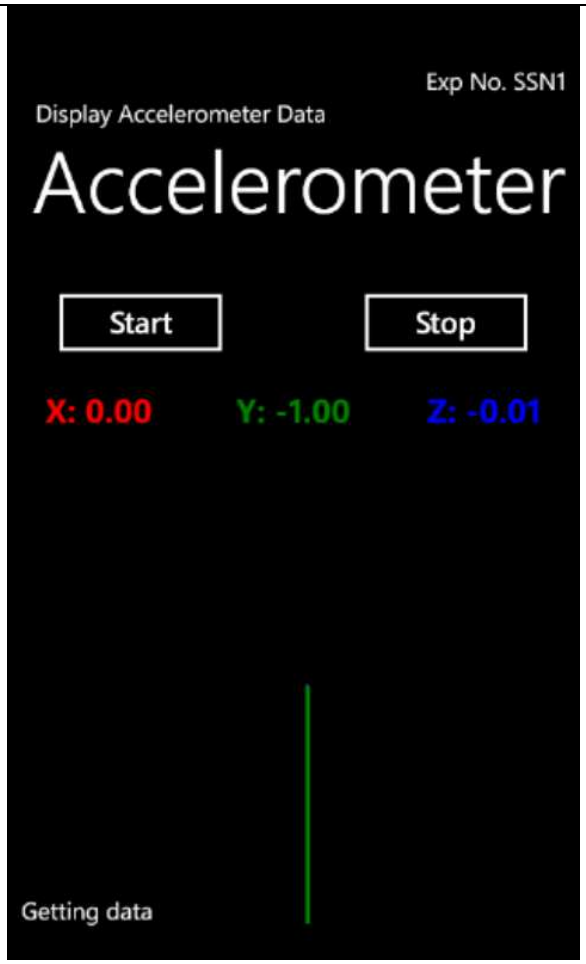

Fig. No6 Keeping the Device in Y-axis only

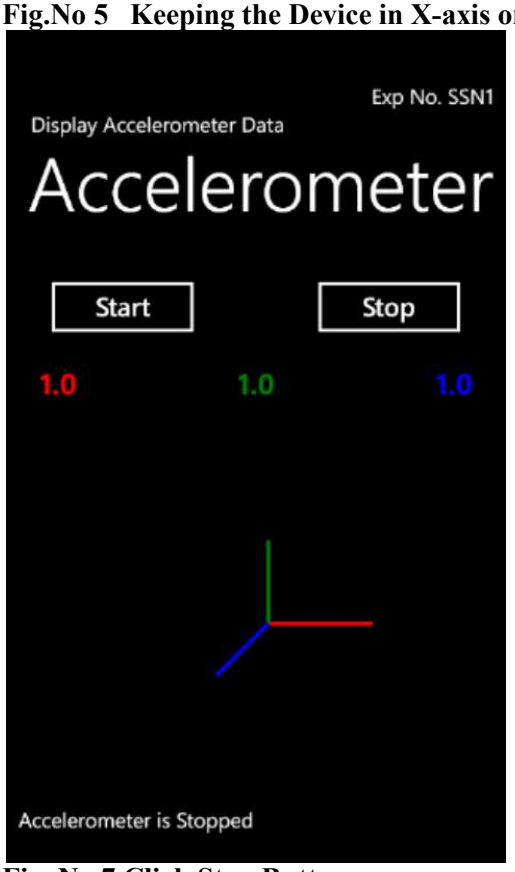

Fig. No 7 Click Stop Button

Observations:

This experiment is used in various applications which require the movement of device as an input parameter for it. Also, used in 3D game development.