Windows Phone アプリケーション開発

回る・回す機能の実装方法

日本マイクロソフト株式会社
デベロッパー＆プラットフォーム統括本部 監修

内容

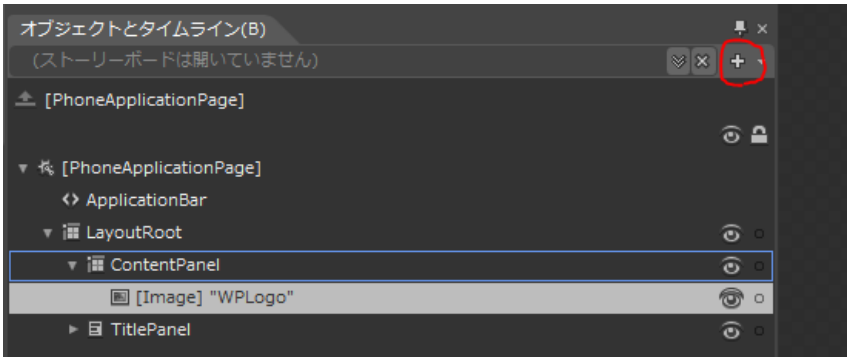
アニメーションでロゴを回す

ロゴの準備

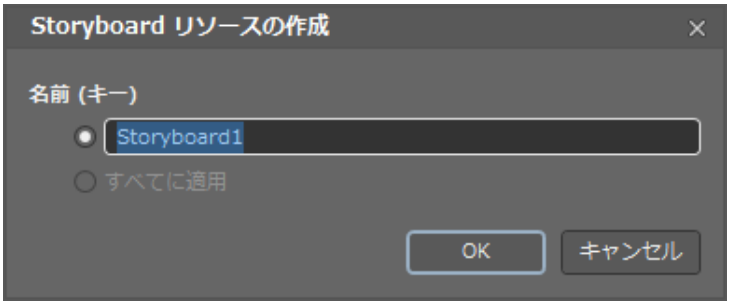
1. Expression Blend でプロジェクトを開きます。
2. ロゴ画像を準備して、ドラッグ&ドロップでロゴを配置します。

ストーリーボードの作成

3. オブジェクトとタイムラインの右端の[＋]を押してストーリーボード（アニメーション）を作成します。

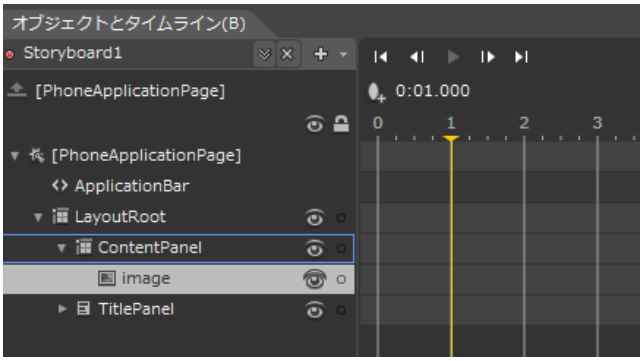


4. 名前を付けて OK を押します。



回転の設定

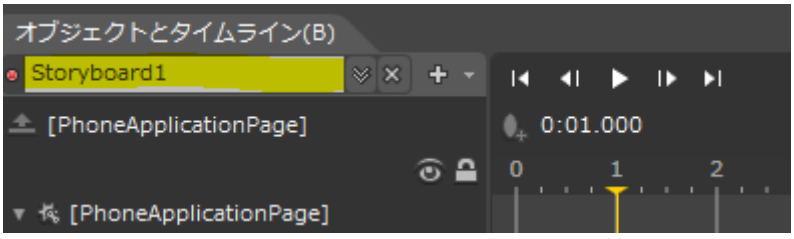
5. 黄色のタイムラインを 1 秒進めます。



6. 変換パネルの Projection（下）の回転タブ(一番左)の Y を 90 に設定します。

繰り返しの設定

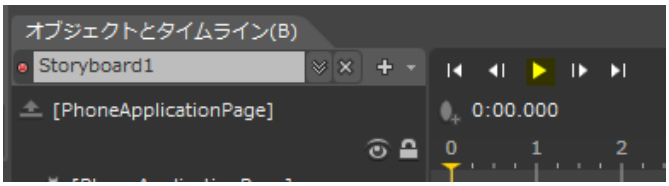
7. オブジェクトとタイムラインのストーリーボード名を選択します。



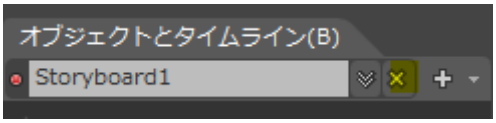
8. 共通プロパティの AutoReverse にチェックを入れ、RepeatoBehavior は Forever を選択します。

動作確認

9. オブジェクトとタイムラインパネルの、再生ボタンを押して動作を確認します。（往復分 1 回のみ再生）



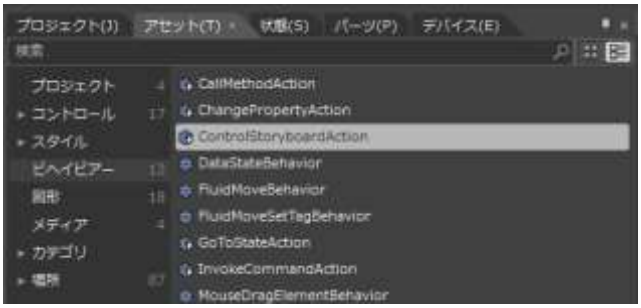
10. オブジェクトとタイムラインの[×]ボタンを押して、ストーリーボードの編集モードを終了します。



ユーザーコントロール化と実行制御の実装

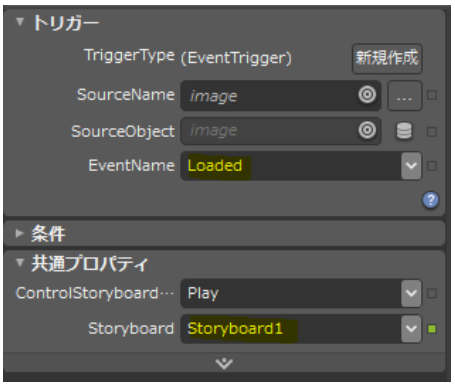
11. ロゴを選択して、「ツール」メニューから「ユーザーコントロールの作成」をおすか F8 キーを押します。このコントロールの名前を付けて OK を押します。（ユーザーコントロールにしない場合はこの処理を行いません）

12. アセットパネルから「ビヘイビアー」の「ControlStoryBoardAction」を選択します

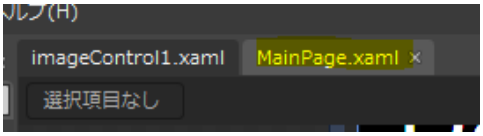


13. そのまま、ロゴの Image コントロールにドラッグ&ドロップします。

14. 右のトリガーパネルの EventName を Loaded に設定し、共通プロパティで作成したストーリーボードを選択します。



15. MainPage.xaml を開いて、いったん「プロジェクト」メニューの「プロジェクトのビルド」を選択します。（ユーザーコントロールにしなかった場合はこの処理を行いません）



実行確認

16. F5 キーを押して実行します。ロゴがずっと回転視し続けていることがわかります。User コントロールにした場合は、ほかのページにコピペして簡単に利用することができます。

作成された ストーリーボードの XAML コードはこれだけです。

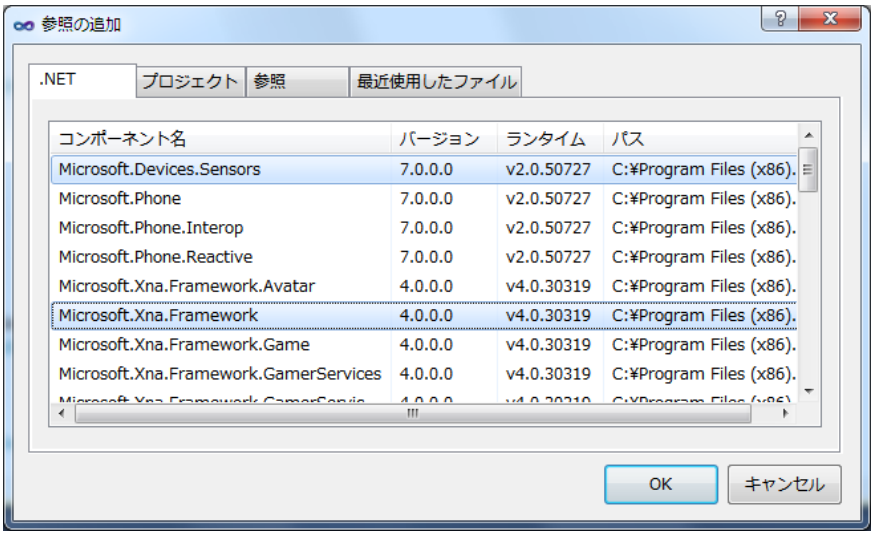
```
<Storyboard x:Name="Storyboard1" AutoReverse="True" RepeatBehavior="Forever">
            <DoubleAnimation Duration="0:0:1" To="90"
            Storyboard.TargetProperty="(UIElement.Projection).(PlaneProjection.RotationY)"
            Storyboard.TargetName="image" d:IsOptimized="True"/>
</Storyboard>
```

デバイスの回転検知

デバイスを表から裏に回したことを検出します。

プロジェクトの作成

1. アプリケーションのプロジェクトを作成して、MainPage.xaml.cs を開きます。
2. 「プロジェクト」メニューの「参照の追加」を選んで、Microsoft。Devices.Sensors と、Microsoft.XNA.Framework を選んで OK します。



3. Using 句を 2 行追加します。

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

センサー検知機能の追加

4. 以下のようにコードを追加します。

```
public partial class MainPage : PhoneApplicationPage
{
    Accelerometer acc = new Accelerometer();
    Boolean bflat, bside;
    int count = 0;

    // コンストラクター
    public MainPage()
    {
        InitializeComponent();
        acc.TimeBetweenUpdates = TimeSpan.FromSeconds(0.1);
        acc.CurrentValueChanged += new
EventHandler<SensorReadingEventArgs<AccelerometerReading>>(acc  CurrentValueChanged);
        acc.Start();
    }
```

```
    void acc_CurrentValueChanged(object sender, SensorReadingEventArgs<AccelerometerReading>
e)
    {
        Vector3 v = e.SensorReading.Acceleration;

        if (v.Z < -0.9)
            bflat = true;
        else if (Math.Abs(v.X) > 0.9)
            bside = true;
        else if (v.Z > 0.9 && bflat && bside)
        {
            Dispatcher.BeginInvoke(() => {
                this.PageTitle.Text = (++count).ToString();
            });
            bside = false;
            bflat = false;
        }

    }
}
```

5. デバイスをつないで、実行します。裏に返すごとにカウントが 1 つ追加されます。
6. デバイスをもって手で回すことを検出する場合はこのようになります。

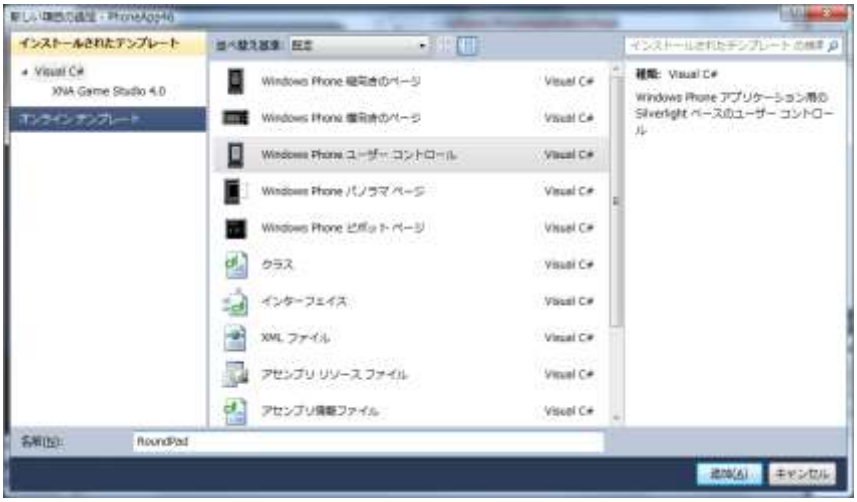
```
void acc_CurrentValueChanged(object sender, SensorReadingEventArgs<AccelerometerReading> e)
{
    Vector3 v = e.SensorReading.Acceleration;
    //Debug.WriteLine("{0} {1} {2}", v.X, v.Y, v.Z);

    if (v.X < -0.9)
        bleft = true;
    else if (v.X > 0.9)
        bright = true;
    else if (v.Y < -0.9)
        bdown = true;
    else if (v.Y > 0.9)
        bup = true;

    if( bup && bdown && bleft && bright)
    {
        Dispatcher.BeginInvoke(() => {
            this.PageTitle.Text = (++count).ToString();
        });
        bup = false;
        bdown = false;
        bleft = false;
        bright = false;
    }
}
```

くるくるパッドの作り方

画面上でのタッチ回転を検出するパッドを作ります。

プロジェクトの作成

1. プロジェクトを作成します。
2. 「プロジェクト」メニューの「新しい項目の追加」で、Windows Phone ユーザーコントロールを選んで RounchPad と名付けます。



RoundPad の作成

3. LayoutRoot グリッドの中に、以下のコードを追加します。

```
<Grid x:Name="LayoutRoot">
    <Grid.RowDefinitions>
        <RowDefinition Height="*" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

    <Rectangle Name="rectangle1" Fill="#01000000"
               Grid.Column="0" Grid.Row="0"
               MouseLeave="Rectangle_MouseLeave"
               MouseEnter="rectangle_MouseEnter" />

    <Rectangle Name="rectangle2" Fill="#01000000"
               Grid.Column="1" Grid.Row="0"
               MouseLeave="Rectangle_MouseLeave"
               MouseEnter="rectangle_MouseEnter" />

    <Rectangle Name="rectangle3" Fill="#01000000"
               Grid.Column="0" Grid.Row="1"
               MouseLeave="Rectangle_MouseLeave"
               MouseEnter="rectangle_MouseEnter" />

    <Rectangle Name="rectangle4" Fill="#01000000"
               Grid.Column="1" Grid.Row="1"
               MouseLeave="Rectangle_MouseLeave"
               MouseEnter="rectangle_MouseEnter" />

    <Ellipse Name="ellipse1"
               Grid.ColumnSpan="2" Grid.RowSpan="2" Margin="30"
               HorizontalAlignment="Stretch" VerticalAlignment="Stretch"
               StrokeThickness="3" />
</Grid>
```

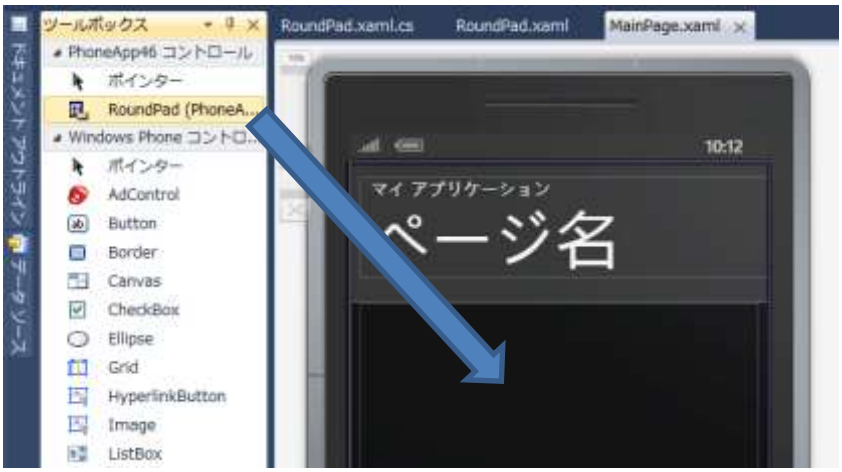
4. 続けて RoundPad.xaml.cs を開いて以下のコードを追加します。

```
public partial class RoundPad : UserControl
{
    //4 象限のタッチ数。4 で 1 回転と判断
    private static int count = 0;

    //オリジナルイベント
    public delegate void CountUpEventHandler(object sender, EventArgs e);
    public event CountUpEventHandler CountUp;

    //初期化処理
    private void Init()
    {
        count = 0;
        rectangle1.Tag = "";
        rectangle2.Tag = "";
        rectangle3.Tag = "";
        rectangle4.Tag = "";
    }

    public RoundPad()
    {
        InitializeComponent();

        this.Loaded += (e, s) => { ellipse1.Stroke = this.BorderBrush; };
        Init();
    }

    private void Rectangle_MouseLeave(object sender, MouseEventArgs e)
    {
        //一度タッチしたところは Tag に記録
        (sender as Rectangle).Tag = "Tap";

        //4 か所タッチし終えたら 1 回転としてイベントを発生させる
        if (++count == 4)
            CountUp(this, EventArgs.Empty);
    }

    private void rectangle_MouseEnter(object sender, MouseEventArgs e)
    {
        //すでにタッチ済みのパネルを触ったらもう一度初めから
        if ((String)((sender as Rectangle).Tag) != "")
            Init();
    }
}
```

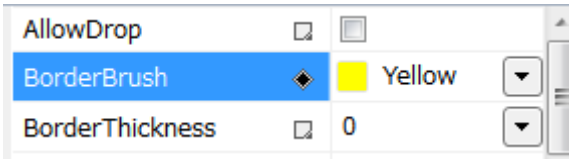
5. いったん、F6 キーを押してビルドします。

RoundPad の利用

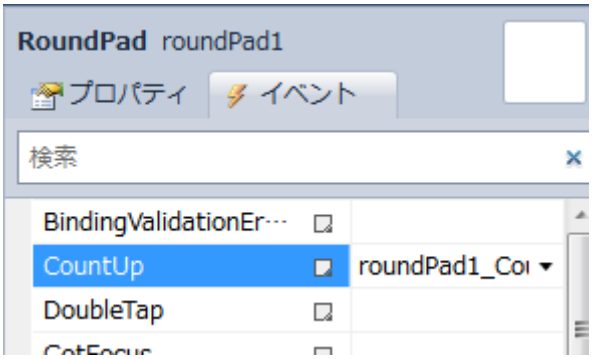
6. そのまま MainPage.xaml を開き、追加された RoundPad コントロールをドラッグ&ドロップし、操作しやすい大きさに拡大します。



7. プロパティの BorderBrush を Yellow に設定します。



8. イベントタブを開き CountUp の右欄をダブルタップします。



9. 以下のようにコードを追加します。

```
public int count = 0;

private void roundPad1  CountUp(object sender, EventArgs e)
{
    this.PageTitle.Text = (++count).ToString();
}
```

テストと実行

10. F5 キーを押してテストします。黄色い円をなぞるようにタッチするとカウントがアップします。