

Windows Phone 7 - Tutoriel 6 : Imager

Visionneuse d'images simple pour Windows Phone 7 avec l'animation de projection

par Peter Bull ([Blog](#)) Deepin Prayag (Traduction) ([Home](#))

Date de publication : 23/03/2012

Dernière mise à jour : 23/03/2012

Ce tutoriel fait partie d'une série de niveau débutant-intermédiaire pour apprendre Windows Phone 7 par le biais d'exemples pas à pas.

Langage : [C#](#)

Public visé : niveau *Débutant*

Commentez ce tutoriel :

Traduction.....	3
Introduction.....	3
Visionneuse d'images simple pour Windows Phone 7 avec l'animation de projection.....	3
Étape 1.....	3
Étape 2.....	4
Étape 3.....	5
Étape 4.....	8
Étape 5.....	9
Étape 6.....	10
Étape 7.....	11
Étape 8.....	12
Étape 9.....	13
Conclusion.....	13
Liens.....	13
Remerciements.....	14

Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Peter Bull**, **Windows Phone 7 - Simple Image Viewer with Projection Animation for Windows Phone 7**.

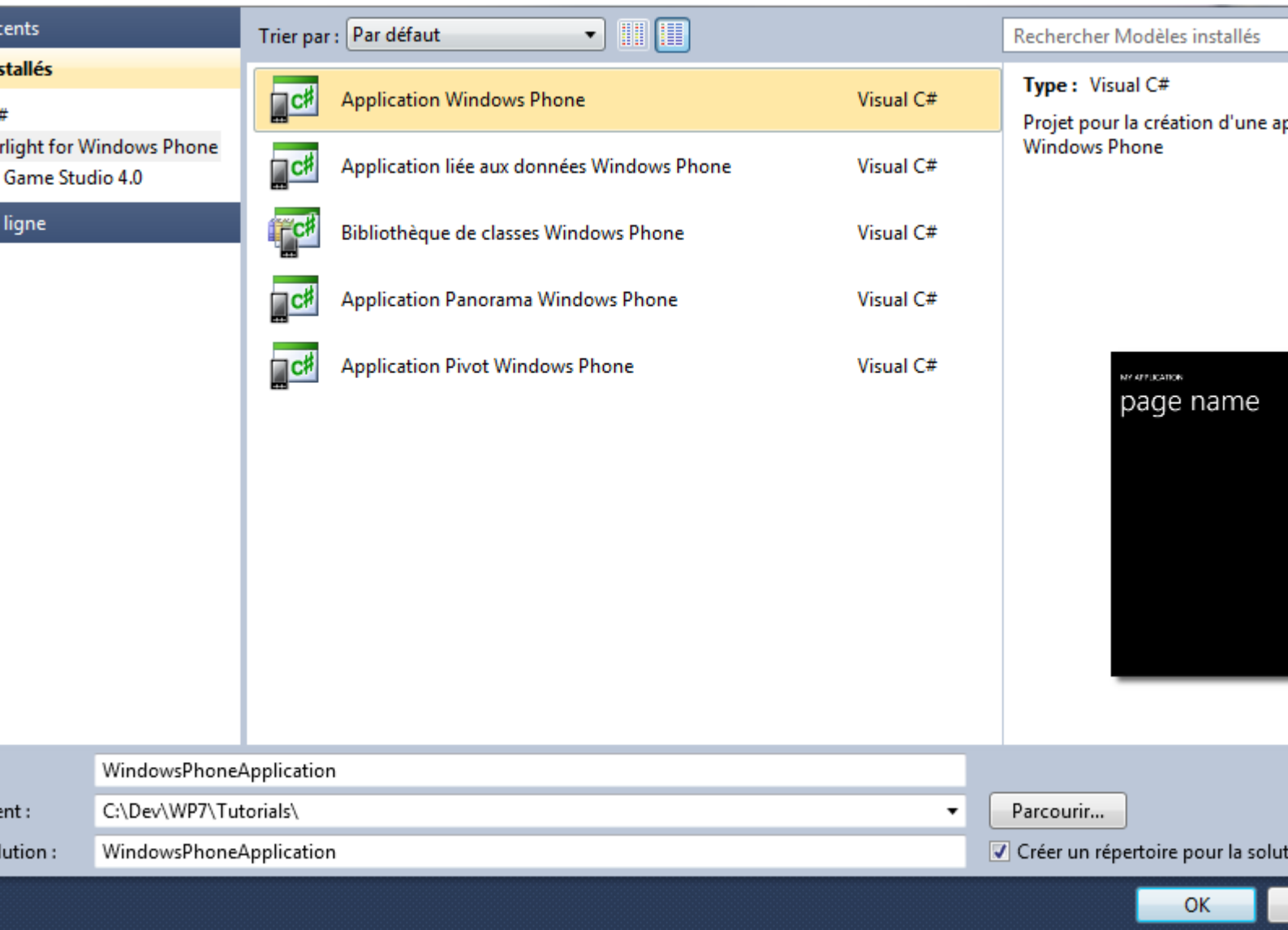
Introduction

Imager est une application simple de visualisation d'images, créée en utilisant Silverlight sur Windows Phone 7, avec des effets d'animations simples en utilisant l'animation de projection créée suivant une procédure.

Visionneuse d'images simple pour Windows Phone 7 avec l'animation de projection

Étape 1

Démarrez Microsoft Visual Web Developer 2010 Express pour Windows Phone, puis sélectionnez *Fichier* puis *Nouveau Projet...* Sélectionnez *Visual C#* ensuite *Silverlight for Windows Phone* puis *Application Windows Phone* dans les modèles installés, sélectionnez un emplacement si vous le souhaitez, puis entrez un nom pour le projet et appuyez sur OK :



Étape 2

Une page *Application Windows Phone* nommée *MainPage.xaml* devrait alors apparaître :



Étape 3

Dans le volet XAML pour *MainPage.xaml* entre les lignes `<Grid x:Name="ContentPanel" Grid.Row="1">` et `</Grid>` , tapez le code XAML suivant :

```

<Grid x:Name="ContentMain">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*"/>
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>
  <!-- Header -->
  <Grid Grid.Row="0" x:Name="Header">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="*"/>
      <ColumnDefinition Width="160"/>
    </Grid.ColumnDefinitions>
    <TextBox Grid.Column="0" Name="Location">
      <TextBox.InputScope>
        <InputScope>
          <InputScopeName NameValue="Url"/>
        </InputScope>
      </TextBox.InputScope>
    </TextBox>
    <Button Grid.Column="1" Content="go" Click="Go_Click"/>
  </Grid>
  <ContentPanel Grid.Row="1" />
  <ContentPanel Grid.Row="2" />
</Grid>

```

```
</Grid>
<!-- Content -->
<Image Grid.Row="1" Margin="50" Stretch="Uniform" Name="Display">
  <Image.Projection>
    <PlaneProjection x:Name="Target"/>
  </Image.Projection>
</Image>
<!-- Footer -->
<Grid Grid.Row="2" x:Name="Buttons" >
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="160"/>
    <ColumnDefinition Width="160"/>
    <ColumnDefinition Width="160"/>
  </Grid.ColumnDefinitions>
  <!-- Buttons -->
  <Button Grid.Column="0" Content="pitch" Click="Pitch_Click"/>
  <Button Grid.Column="1" Content="roll" Click="Roll_Click"/>
  <Button Grid.Column="2" Content="yaw" Click="Yaw_Click"/>
</Grid>
</Grid>
```

```

<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <Grid x:Name="ContentMain">
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="*/>
      <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <!-- Header -->
    <Grid Grid.Row="0" x:Name="Header">
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*/>
        <ColumnDefinition Width="160"/>
      </Grid.ColumnDefinitions>
      <TextBox Grid.Column="0" Name="Location">
        <TextBox.InputScope>
          <InputScope>
            <InputScopeName NameValue="Url"/>
          </InputScope>
        </TextBox.InputScope>
      </TextBox>
      <Button Grid.Column="1" Content="go" Click="Go_Click"/>
    </Grid>
    <!-- Content -->
    <Image Grid.Row="1" Margin="50" Stretch="Uniform" Name="Display">
      <Image.Projection>
        <PlaneProjection x:Name="Target"/>
      </Image.Projection>
    </Image>
    <!-- Footer -->
    <Grid Grid.Row="2" x:Name="Buttons" >
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
        <ColumnDefinition Width="160"/>
      </Grid.ColumnDefinitions>
      <!-- Buttons -->
      <Button Grid.Column="0" Content="pitch" Click="Pitch_Click"/>
      <Button Grid.Column="1" Content="roll" Click="Roll_Click"/>
      <Button Grid.Column="2" Content="yaw" Click="Yaw_Click"/>
    </Grid>
  </Grid>
</Grid>

```

Design :



Étape 4

Faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Afficher le code*. Dans la *vue de code* au-dessus de `namespace Imager` tapez le code suivant :

```
using System.Windows.Media.Imaging;
```

Toujours dans la *vue de code* au-dessus de `public MainPage()` tapez les déclarations suivantes :

```
private bool Rotating = false;
private Storyboard Rotation = new Storyboard();
```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.Windows.Media.Imaging;

namespace Imager
{
    public partial class MainPage : PhoneApplicationPage
    {
        private bool Rotating = false;
        private Storyboard Rotation = new Storyboard();

        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }
    }
}

```

Étape 5

Puisque nous en sommes à la *vue de code* pour *MainPage.xaml.cs* au-dessus de la méthode `public MainPage()` tapez la *méthode* suivante :

```

private void Rotate(object Parameter)
{
    if (Rotating)
    {
        Rotation.Stop();
        Rotating = false;
    }
    else
    {
        DoubleAnimation _animation = new DoubleAnimation();
        _animation.From = 0.0;
        _animation.To = 360.0;
        _animation.Duration = new Duration(TimeSpan.FromSeconds(10));
        _animation.RepeatBehavior = RepeatBehavior.Forever;
        Storyboard.SetTarget(_animation, Target);
        Storyboard.SetTargetProperty(_animation, new PropertyPath(Parameter));
        Rotation.Children.Clear();
        Rotation.Children.Add(_animation);
        Rotation.Begin();
        Rotating = true;
    }
}

```

```

private void Rotate(object Parameter)
{
    if (Rotating)
    {
        Rotation.Stop();
        Rotating = false;
    }
    else
    {
        DoubleAnimation _animation = new DoubleAnimation();
        _animation.From = 0.0;
        _animation.To = 360.0;
        _animation.Duration = new Duration(TimeSpan.FromSeconds(10));
        _animation.RepeatBehavior = RepeatBehavior.Forever;
        Storyboard.SetTarget(_animation, Target);
        Storyboard.SetTargetProperty(_animation, new PropertyPath(Parameter));
        Rotation.Children.Clear();
        Rotation.Children.Add(_animation);
        Rotation.Begin();
        Rotating = true;
    }
}
    
```

Étape 6

Puisque nous en sommes toujours à la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `public MainPage()` tapez les *gestionnaires d'évènements* :

```

private void Go_Click(object sender, RoutedEventArgs e)
{
    Display.Source = new BitmapImage(new Uri(Location.Text));
}

private void Pitch_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationXProperty);
}

private void Roll_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationZProperty);
}

private void Yaw_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationYProperty);
}
    
```

```

private void Go_Click(object sender, RoutedEventArgs e)
{
    Display.Source = new BitmapImage(new Uri(Location.Text));
}

private void Pitch_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationXProperty);
}

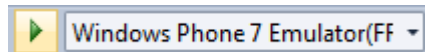
private void Roll_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationZProperty);
}

private void Yaw_Click(object sender, RoutedEventArgs e)
{
    Rotate(PlaneProjection.RotationYProperty);
}

```

Étape 7

Enregistrez le *Projet* maintenant que vous avez terminé l'application *Windows Phone Silverlight*. Sélectionnez l'option *Windows Phone Emulator*, ensuite sélectionnez *Debug* puis *Démarrer le débogage* ou cliquez sur *Démarrer le débogage* :



Une fois que vous l'aurez fait, ce qui suit apparaîtra dans *l'émulateur Windows Phone* après qu'il aura été chargé :



Étape 8

Sélectionnez le *TextBox* et entrez l'adresse web d'une image telle que [celle-ci](#). Ensuite cliquez sur le bouton « go », et une fois lancée l'image apparaîtra dans le contrôle Image. Vous pouvez l'animer avec les boutons *Pitch*, *Roll* and *Yaw* :



Étape 9

Vous pouvez ensuite *Arrêter l'application* en sélectionnant la fenêtre d'application *Visual Studio 2010* et en cliquant sur le bouton *Arrêter le débogage* :



Conclusion

Ceci est une application très simple sur l'animation d'une image en utilisant le code. Il est également possible d'avoir plusieurs animations - si vous déclarez un objet Animation pour chaque type, vous pouvez combiner des animations - voyez ce que vous pouvez ajouter et modifier pour la personnaliser !

Liens

Télécharger le code source du tutoriel.

Remerciements

Je tiens ici à remercier **Peter Bull** de m'avoir autorisé à traduire son tutoriel.
Je remercie **xxx** pour sa relecture technique et ses propositions.
Je remercie également **yyy** pour sa relecture orthographique et ses propositions.