

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55



Quería un utilitario que me permitiera entre otras cosas hacer lo que se denomina "animación cuadro a cuadro". Esta es una técnica divertida (sobre todo si tienen niños de vacaciones, o como en mi caso, niñas) que consiste, en forma general, en "animar" objetos inanimados (en este caso, hablamos de muñecas barbie ...), a partir del proceso de sacarles una foto, moverlas ligeramente, sacar otra foto, moverlos un poquito mas ... sacar una nueva foto , y dada la suficiente cantidad de fotogramas, al poder verlos uno detrás del otro a velocidad, genera la ilusión de movimiento (autónomo).

Esta técnica es la base de infinidad de animaciones y personajes, tanto con muñecos como con masa / plastilina. No se si vieron alguna vez los videos de Wallace and Gromit ... (pueden buscar en youtube): Ahi tienen un buen ejemplo.

Entrando ahora si en el tema programación, la aplicación la hice en Silverlight (vb.net) por lo tanto puede ser ejecutada desde la web.

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

Primero vamos a pegarle un vistazo al archivo xaml. Para los que no han metido cuchara aún en Silverlight, el archivo xaml es un archivo XML cuyo objetivo fundamental es describir el diseño visual del interfase de nuestra aplicación con el usuario: Que controles tiene, con que efectos y en que posiciones deben mostrarse, e incluso como deben interactuar entre ellos o filtrar la información que despliegan. En nuestra aplicación, el archivo xaml tiene una única grid con los siguientes controles:

```
<Grid x:Name="LayoutRoot" Background="White">
```

```
    <Rectangle Height="480" HorizontalAlignment="Left" Margin="148,48,0,0"
    Name="webcamout" VerticalAlignment="Top" Width="640" />
```

```
    <sdk:Label Height="28" HorizontalAlignment="Left" Margin="272,8,0,0" Name="texto"
    VerticalAlignment="Top" Width="516" Content="Tu webcam como espejo o para animar!"
    FontSize="18" />
```

```
    <Button Visibility="Collapsed" Content="Activar Webcam" Height="23"
    HorizontalAlignment="Left" Margin="156,12,0,0" Name="encender" VerticalAlignment="Top"
    Width="110" />
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
<Button Visibility="Collapsed" Content="Sacar Foto" Height="23"
HorizontalAlignment="Left" Margin="21,12,0,0" Name="sacarfoto" VerticalAlignment="Top"
Width="110" />
```

```
<ListBox Height="412" ScrollViewer.HorizontalScrollBarVisibility="Disabled"
HorizontalAlignment="Left" Margin="12,48,0,0" Name="fotos" VerticalAlignment="Top"
Width="128">
```

```
<ListBox.ItemTemplate>
```

```
<DataTemplate>
```

```
<StackPanel Orientation="vertical">
```

```
<Image Source="{Binding Path=CapturedImage}" Stretch="UniformToFill"
Width="110" Height="80" />
```

```
<TextBlock Text="{Binding Path=nombre}"/>
```

```
</StackPanel>
```

```
</DataTemplate>
```

```
</ListBox.ItemTemplate>
```

```
</ListBox>
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
<Button Content="Borrar Fotos" IsEnabled="False" Height="23"
HorizontalAlignment="Left" Margin="21,467,0,0" Name="borrar" VerticalAlignment="Top"
Width="110" />
```

```
<Button Content="Grabar Foto" IsEnabled="false" Height="23" HorizontalAlignment="Left"
Margin="21,496,0,0" Name="grabar" VerticalAlignment="Top" Width="110" />
```

```
<Button Content="Animar Fotos" IsEnabled="false" Height="23"
HorizontalAlignment="Left" Margin="21,525,0,0" Name="anim" VerticalAlignment="Top"
Width="110" />
```

```
<Slider Height="23" IsEnabled="false" HorizontalAlignment="Left" Margin="300,534,0,0"
Name="velocidad" VerticalAlignment="Top" Width="199" Maximum="800" Minimum="50"
Value="150" />
```

```
<sdk:Label Height="23" HorizontalAlignment="Left" Margin="148,534,0,0" Name="Label1"
VerticalAlignment="Top" Width="146" Content="Velocidad de animación:" />
```

```
<sdk:Label Height="28" HorizontalAlignment="Left" Margin="513,534,0,0"
Name="milisegs" VerticalAlignment="Top" Width="120" Content="200 milisegundos" />
```

```
</Grid>
```

Una breve descripción de este xaml:

Primero tenemos el rectángulo "webcamout" de 640x480 pixeles, donde voy a mapear dentro,

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

en el código, el video de la webcam.

Luego tenemos un listbox denominado "fotos" donde iremos guardando visualmente las sucesivas "fotos" (fotogramas) que capturemos desde el video de la webcam, de tal manera de que se vayan almacenando una debajo de la otra, pero que el listbox muestre siempre la última de abajo (el fotograma mas reciente!).

También tenemos una serie de botones, cuyo funcionamiento puede ser inferido leyendo el texto del contenido de los mismos: Activar la webcam, Sacar Foto, Borrar Fotos, Grabar Foto y Animar Fotos.

Es importante resaltar que muchos de estos botones comienzan con su propiedad `isEnabled="false"` ya que sus funcionalidades dependen de determinados elementos: No se puede "Sacar Foto" si no está la Webcam activada, o no se puede "Animar Fotos" si no hay al menos 3 o 4 fotos sacadas, No se puede Grabar Foto si no has elegido de la tira de fotos sacadas, una foto en particular, para grabar ... O no se puede Borrar Fotos si para empezar, no hay fotos sacadas!. O sea los botones se "activan" (`isEnabled="true"`) solamente si corresponde su función en el contexto de la aplicación.

Por último tenemos un control deslizante, que tiene un rango de 50 a 800, cuyo valor luego va a ser inyectado al timer que regula la velocidad de la animación.

Ahora vamos a repasar el código en vb.net que hace de cohesión entre estos controles y las distintas clases del Silverlight, para lograr el funcionamiento de esta aplicación; Fraccionado en las distintas funciones y subs: Primero, defino una serie de objetos y variables y luego ante el evento "onloaded" de mi programita, comienzo a trabajar::

```
Imports System.Windows.Media.Imaging
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
Imports System.Collections.ObjectModel Imports System.IO Imports ImageTools Imports ImageTools.IO.Png
```

```
Dim WithEvents captura As New CaptureSource 'El withevents es necesario porque luego uso el evento de "oncapturecompleted"
```

```
Private capturedImages As New ObservableCollection(Of StillImage) 'Esto será el array de "still images" Dim cant As Integer = 0 Dim chorro As New VideoBrush Dim WithEvents eltimer As New System.Windows.Threading.DispatcherTimer
```

```
Dim i As Integer = 0
```

```
Dim fotograma As New ImageBrush
```

```
'Cargó mi aplicacion:
```

```
'-----
```

```
Private Sub MainPage_Loaded(ByVal sender As Object, ByVal e As System.Windows.RoutedEventArgs) Handles Me.Loaded captura.VideoCaptureDevice = CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice
```

```
If Not captura Is Nothing Then
```

```
If CaptureDeviceConfiguration.AllowedDeviceAccess Then
```

```
iniciarespejo()
```

```
Else
```

```
'No tiene permiso para el video! Le pongo el botón
```

```
encender.Visibility = Windows.Visibility.Visible
```

```
End If
```

```
Else
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

'No se puede crear el objeto de captura:

```
texto.Content = "Error: No encuentro tu webcam :("
```

```
webcamout.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
End Sub
```

Arrancamos agregando o declarando las clases que queremos usar en nuestro código. Las que dicen "ImageTools" si no saben de donde salieron, no se preocupen: Sobre el final de este texto se explica de donde sacarlas. Son de "terceros", gratuitas, y sirven para manejar formatos de imagenes y asi poder grabar por ejemplo las fotos capturadas de la webcam en formato PNG o JPG...

En cuanto a las variables que defino, resalto el array o collection de "stillimage". Stillimage es una clase muy sencilla que simplemente encapsula o define una imagen como writeablebitmap y un string como titulo único de esa imagen. También defino un timer, que luego será el "reloj" que haga funcionar la secuencia de imagenes capturadas como animación.

En cuanto al código, verifico que haya realmente una webcam disponible en el equipo del usuario. Si la hay, intento iniciar el stream de video si tengo permiso, o de pedir al usuario que la active él. Veamos con detalle: En ese proceso de inicio ya verificamos que sea TRUE el `CaptureDeviceConfiguration.AllowedDeviceAccess` lo que implica que previamente (en otro momento) este usuario YA permitió a esta aplicación el uso de su webcam, y por lo tanto directamente llamamos a la función que inicia la webcam. Si aún no dió permiso este usuario para activar la webcam, entonces le hacemos visible el botón de "Activar Webcam".

```
Private Sub encender_Click(ByVal sender As System.Object, ByVal e As System.Windows.RoutedEventArgs) Handles encender.Click
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
If Not captura Is Nothing Then If CaptureDeviceConfiguration.RequestDeviceAccess Then iniciarespejo()
```

```
Else
```

```
texto.Content = "Error: No tengo permiso para usar tu webcam :("
```

```
End If
```

```
encender.Visibility = Windows.Visibility.Collapsed
```

```
Else
```

```
'No se puede crear el objeto de captura: Abrá desconectado la webcam entre que cargó la pagina y apretó el botón?
```

```
texto.Content = "Error: No encuentro tu webcam :("
```

```
webcamout.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
End Sub
```

El evento onclick del botón "Activar Webcam" primero se fija si sigue habiendo una webcam conectada a la computadora del usuario. Si no está mas conectada, le avisa.

Si sigue disponible, entonces se fija si el usuario nos da permiso para utilizar la webcam, en cuyo caso ejecuta el iniciarspejo (y deja de mostrar el botón de "Activar Webcam") , y si el usuario no nos dá permiso para activar la webcam, lo avisa poniendo el texto correspondiente en el título de la aplicación (y deja de mostrar el rectángulo principal donde iba a ir el video).

```
Public Sub iniciarspejo() Try chorro.SetSource(captura) webcamout.Fill = chorro webcamout.UpdateLayout()
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
Dim value As New SolidColorBrush(Colors.Black)
```

```
webcamout.Stroke = value
```

```
webcamout.StrokeThickness = 1
```

```
Dim efecto As New Effects.DropShadowEffect
```

```
efecto.BlurRadius = 15
```

```
webcamout.Effect = efecto
```

```
fotos.Visibility = Windows.Visibility.Visible
```

```
fotos.Effect = efecto
```

```
captura.Start()
```

```
'Prendo el boton de sacar fotos:
```

```
sacarfoto.Visibility = Windows.Visibility.Visible
```

```
fotos.ItemsSource = capturedImages 'Bindeo la colección de "still images" al itemslist
```

```
Catch ex As Exception
```

```
texto.Content = "Error: No pude iniciar la Webcam: Ya está en uso?"
```

```
End Try
```

```
End Sub
```

Aqui tenemos la pequeña sub que activa la webcam enganchando el video al rectángulo en la pantalla destinado para ello. Lo hace a través de un videobrush llamado "chorro" que fue definido al inicio del código, en forma general. Como extra, generamos un efecto de sombreado y se lo aplicamos al rectángulo, para que quede mejor, con mas peso visual, dado que tiene video ahora, adentro. También aprovechamos ya para activar el uso del botón de Sacar Fotos.

Por las dudas, este código está encapsulado en un Try --- end Try. Si este código llegara a fallar, todo indica que (tal como se lo avisamos al usuario) la webcam ya estaba previamente en uso por otra aplicación.

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
Private Sub sacarfoto_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles sacarfoto.Click 'Mando sacar la foto y mientras
tanto el botón de sacar foto queda inhabilitado
```

```
captura.CaptureImageAsync()
```

```
sacarfoto.IsEnabled = False
```

```
End Sub
```

```
Private Sub captura_captureimagecompleted(ByVal sender As Object, ByVal e As
System.Windows.Media.CaptureImageCompletedEventArgs) Handles
captura.CaptureImageCompleted
```

```
Dim imagen As New StillImage
```

```
imagen.CapturedImage = e.Result
```

```
imagen.nombre = "Foto " + (cant + 1).ToString
```

```
capturedImages.Add(imagen)
```

```
fotos.UpdateLayout()
```

```
fotos.ScrollIntoView(fotos.Items.Item(cant)) 'Y muestro la ultima foto
```

```
sacarfoto.IsEnabled = True
```

```
borrar.IsEnabled = True
```

```
If cant = 3 Then anim.IsEnabled = True
```

```
cant += 1
```

```
End Sub
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

Por un lado aquí tenemos el evento de click del botón de "Sacar Foto": El mismo simplemente ejecuta la funcionalidad de capturar imagen del propio objeto CaptureSource, que se ejecuta asincrónicamente, y cuando esté pronto el fotograma capturado, Silverlight llamará automáticamente al sub que maneje el evento "CaptureImageCompleted" (que está inmediatamente abajo).

Es importante notar que desactivamos el botón de "Sacar Foto" mientras este proceso ocurre.

El sub que recibe y procesa el fotograma capturado, crea un nuevo "stillimage" y le asigna la imagen y un nombre, que será el texto "FOTO" seguido del número de fotograma. Luego agrega ese "Stillimage" y lo agrega a la colección o array donde estamos almacenando las "stillimage", que a su vez está atado (binded) al Listbox "fotos" por lo que allí aparecerá automáticamente esta nueva foto.

Para que el Listbox nos muestre siempre la última foto, lo forzamos a que automáticamente se desplace hasta esa foto que acabamos de agregar. Si esta foto es ya la tercera que el usuario saca, le activamos el botón de "Animar Fotos".

```
Private Sub borrar_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs) Handles borrar.Click    capturedImages.Clear()    cant = 0    borrar.IsEnabled = False

anim.IsEnabled = False

grabar.IsEnabled = False

End Sub
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
Private Sub fotos_SelectionChanged(ByVal sender As Object, ByVal e As  
System.Windows.Controls.SelectionChangedEventArgs) Handles fotos.SelectionChanged
```

```
'Si eligo algo en la tira de fotos, le activo el boton de guardar foto
```

```
grabar.IsEnabled = True
```

```
End Sub
```

Dos subs autoexplicativas: La primera maneja el evento de click del botón de "Borrar Fotos", haciendo un clear del collection o array "capturedimages", reseteando el contador "cant" que va contando las imagenes capturadas, y desactivando los botones de Borrar, Animar y Grabar fotos. La segunda maneja el evento que ocurre cuando se agrega una foto al Listbox "fotos": si el usuario ha elegido una foto, entonces le activo el botón de "Grabar Foto".

```
Private Sub anim_Click(ByVal sender As System.Object, ByVal e As  
System.Windows.RoutedEventArgs) Handles anim.Click If captura.State =  
Windows.Media.CaptureState.Stopped Then
```

```
'Asumo que estaba haciendo animacion
```

```
eltimer.Stop()
```

```
anim.Content = "Animar Fotos"
```

```
iniciarespejo() 'Reinicio funcion de espejo
```

```
velocidad.IsEnabled = False
```

```
borrar.IsEnabled = True 'Le permito borrar los fotogramas porque no los estoy usando
```

```
Else
```

```
'No está haciendo animacion, se lo enciendo:
```

```
captura.Stop()
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
borrar.IsEnabled = False 'No le permito borrar los fotogramas porque está usandolos en la animación.
```

```
sacarfoto.Visibility = Windows.Visibility.Collapsed
```

```
eltimer.Interval = TimeSpan.FromMilliseconds(CInt(velocidad.Value))
```

```
eltimer.Start()
```

```
anim.Content = "PARAR"
```

```
velocidad.IsEnabled = True
```

```
End If
```

```
End Sub
```

Aquí manejamos el evento de click del botón de "Animar Fotos": Si estaba animando, paramos la animación y volvemos a mostrar la webcam. Si estaba mostrando la webcam, paramos y comenzamos el timer que va a secuenciar las fotos capturadas. En ese caso el botón de "Animar Fotos" pasa a tener el contenido de "PARAR" ya que es este mismo botón y evento el que usaremos para controlar ambos estados.

```
'PIDE NUEVO FRAME EN LA ANIMACIÓN: '----- Private Sub  
timerwork() Handles eltimer.Tick
```

```
fotograma.ImageSource = capturedImages(i).CapturedImage
```

```
webcamout.Fill = fotograma
```

```
webcamout.UpdateLayout()
```

```
i += 1
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
If i = capturedImages.Count Then i = 0
```

```
End Sub
```

```
'CAMBIA VELOCIDAD DE ANIMACIÓN:
```

```
'-----
```

```
Private Sub velocidad_ValueChanged(ByVal sender As System.Object, ByVal e As  
System.Windows.RoutedPropertyChangedEventArgs(Of System.Double)) Handles  
velocidad.ValueChanged
```

```
eltimer.Interval = TimeSpan.FromMilliseconds(CInt(velocidad.Value))
```

```
milisegs.Content = CInt(velocidad.Value).ToString + " milisegundos"
```

```
End Sub
```

La primer Sub es la que llama el timer periódicamente cuando está activo. El código simplemente muestra un fotograma, avanzando de a uno en uno, cada vez que esta rutina es llamada, hasta llegar al último, en cuyo caso vuelve a empezar por el primero. La segunda Sub es llamada en el evento de que cambió el valor del control deslizante, inyectando ese nuevo valor al tiempo de demora del timer, y modificando así efectivamente la velocidad entre imagen e imagen.

```
Private Sub grabar_Click(ByVal sender As System.Object, ByVal e As  
System.Windows.RoutedEventArgs) Handles grabar.Click 'Quiere grabar la foto! Dim grabo  
As New SaveFileDialog()
```

```
grabo.Filter = "PNG Files (*.png)|*.png|All Files (*.*)|*.*"
```

```
grabo.DefaultExt = ".png"
```

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

```
grabo.FilterIndex = 1
```

```
'Grabo la imagen en cuestión:
```

```
If grabo.ShowDialog Then
```

```
Dim elstream As Stream = grabo.OpenFile
```

```
Dim rawimage As ImageTools.ExtendedImage
```

```
rawimage = capturedImages(fotos.SelectedIndex).CapturedImage.ToImage
```

```
Dim encoder As New PngEncoder
```

```
encoder.Encode(rawimage, elstream)
```

```
elstream.Close()
```

```
End If
```

```
End Sub
```

Este código se ejecuta cuando el usuario hace click sobre Grabar Foto. Es importante comentar que Silverlight (al menos en su versión 4) no incluye clases para codificar imagenes en ningún formato, por lo que fue necesaria la funcionalidad de unas dll de terceros, gratuitas, que agregan esa capacidad al Silverlight (e inflan el código final compilado, ya que tienen que agregarse al código que el usuario debe ejecutar en su navegador!). Estas Dll son las "imagetools" y pueden ser descargadas [haciendo click aqui](#) .

Al descargar esas dll, que vienen empaquetadas en un archivo .rar, deben tirar para la carpeta BIN de su proyecto las siguientes DLL:

ImageTools.dll

ImageTools.IO.Png.dll

Silverlight y la Webcam en vb.net: Diversión!

Escrito por Enrique

Lunes, 03 de Enero de 2011 13:50 - Actualizado Lunes, 26 de Septiembre de 2011 15:55

Y hacer la referencia a las mismas en vuestro proyecto.

```
Public Class StillImage Public Property CapturedImage As WriteableBitmap Public Property nombre As String
```

```
End Class
```

Por último, abajo del todo, creamos la clase "stillimage" que simplemente es un marco para guardar cada unidad de "imagen y su nombre".

Cualquier pregunta, no duden en plantearla!!