

CHƯƠNG VI: MEDIA VÀ ANIMATION

1 Animation

Animation là một ảo giác được tạo ra bằng cách thay đổi một loạt các hình ảnh. Não nhận các hình ảnh như là một thay đổi cảnh. Trong phim ảo giác này được tạo bằng cách sử dụng các camera có ghi lại rất nhiều hình ảnh, khi hình ảnh được phát trở lại bằng một máy chiếu, các khán giả thấy một hình ảnh chuyển động. Silverlight có chứa rất nhiều các đối tượng có thể thực hiện được việc tạo ra ảo giác chuyển động.

1.1 Storyboard

Kiểm soát các animation với một timeline và cung cấp các đối tượng và tài nguyên nhằm mục đích cung cấp thông tin cho các animation con

```
<Canvas>
  <Canvas.Resources>
    <Storyboard x:Name="myStoryboard">

      <!-- Animate the center point of the ellipse. -->
      <PointAnimation Storyboard.TargetProperty="Center"
Storyboard.TargetName="MyAnimatedEllipseGeometry"
Duration="0:0:5"
From="20,200"
To="400,100"
RepeatBehavior="Forever" />
    </Storyboard>
  </Canvas.Resources>

  <Path Fill="Blue">
    <Path.Data>
      <!-- Describes an ellipse. -->
      <EllipseGeometry x:Name="MyAnimatedEllipseGeometry"
Center="20,20" RadiusX="15" RadiusY="15" />
    </Path.Data>
  </Path>

  <StackPanel Orientation="Horizontal" Canvas.Left="10" Canvas.Top="265">
    <!-- Button that begins animation. -->
    <Button Click="Animation_Begin"
Width="65" Height="30" Margin="2" Content="Begin" />

    <!-- Button that pauses Animation. -->
    <Button Click="Animation_Pause"
Width="65" Height="30" Margin="2" Content="Pause" />

    <!-- Button that resumes Animation. -->
    <Button Click="Animation_Resume"
Width="65" Height="30" Margin="2" Content="Resume" />
  </StackPanel>
</Canvas>
```

```
<!-- Button that stops Animation. Stopping the animation returns the
ellipse to its original location. -->
<Button Click="Animation_Stop"
Width="65" Height="30" Margin="2" Content="Stop" />
</StackPanel>

</Canvas>

private void Animation_Begin(object sender, RoutedEventArgs e)
{
    myStoryboard.Begin();
}
private void Animation_Pause(object sender, RoutedEventArgs e)
{
    myStoryboard.Pause();
}
private void Animation_Resume(object sender, RoutedEventArgs e)
{
    myStoryboard.Resume();
}
private void Animation_Stop(object sender, RoutedEventArgs e)
{
    myStoryboard.Stop();
}
```

1.2 Key-Frame Animations

Key-frame animations cho phép bạn tương tác vào animation nhiều hơn hai đối tượng và kiểm soát một animation bằng phương pháp nội suy.

Key-frame ko có các thuộc tính như From / To / bởi Animation. Bạn tạo ra key-frame, các đối tượng animation được key-frame kiểm soát, khi các hình ảnh động chạy nó sẽ chuyển giữa các frame mà bạn xác định tạo ra một chuỗi chuyển động liền mạch

```
<Canvas>
<Canvas.Resources>
    <Storyboard x:Name="myStoryboard">

        <!-- Animate the TranslateTransform's X property
        from 0 to 350, then 50, then 200 over 10 seconds. -->
        <DoubleAnimationUsingKeyFrames
Storyboard.TargetName="MyAnimatedTranslateTransform"
```

```

Storyboard.TargetProperty="X"
Duration="0:0:10">

    <!-- Using a LinearDoubleKeyFrame, the rectangle moves
    steadily from its starting position to 500 over
    the first 3 seconds. -->
    <LinearDoubleKeyFrame Value="500" KeyTime="0:0:3" />

    <!-- Using a DiscreteDoubleKeyFrame, the rectangle
suddenly
    appears at 400 after the fourth second of the animation. -->
    <DiscreteDoubleKeyFrame Value="400" KeyTime="0:0:4" />

    <!-- Using a SplineDoubleKeyFrame, the rectangle moves
at
    back to its starting point. The animation starts out slowly
    first and then speeds up. This KeyFrame ends after the 6th
    second. -->
    <SplineDoubleKeyFrame KeySpline="0.6,0.0 0.9,0.00"
Value="0" KeyTime="0:0:6" />

    </DoubleAnimationUsingKeyFrames>
</Storyboard>
</Canvas.Resources>

<Rectangle MouseLeftButtonDown="Mouse_Clicked" Fill="Blue"
Width="50" Height="50">
    <Rectangle.RenderTransform>
        <TranslateTransform x:Name="MyAnimatedTranslateTransform"
X="0" Y="0" />
    </Rectangle.RenderTransform>
</Rectangle>

</Canvas>

// When the user clicks the Rectangle, the animation
// begins.
private void Mouse_Clicked(object sender, MouseEventArgs e)
{
    myStoryboard.Begin();
}

```

http://samples.msdn.microsoft.com/Silverlight/silverlight_next/Animations/doubleanimationusingkeyframes/ClientBin/TestPage.html

1.3 Double Animation

Double Animation biến đổi giá trị giữa hai biến của đối tượng Animation bằng cách sử dụng nội suy tuyến tính trên một khoảng thời gian xác định

```

<StackPanel>
    <StackPanel.Resources>
        <Storyboard x:Name="myStoryboard">
            <DoubleAnimation

```

```
Storyboard.TargetName="MyAnimatedRectangle"
Storyboard.TargetProperty="Opacity"
From="1.0" To="0.0" Duration="0:0:5"
AutoReverse="True" RepeatBehavior="Forever" />
</Storyboard>
</StackPanel.Resources>

<Rectangle Loaded="Start_Animation" x:Name="MyAnimatedRectangle"
Width="100" Height="100" Fill="Blue" />

</StackPanel>
private void Start_Animation(object sender, EventArgs e)
{
    myStoryboard.Begin();
}
http://samples.msdn.microsoft.com/Silverlight/silverlight\_next/Animations/duobleanimation/ClientBin/TestPage.html
```

1.4 Color Animation

Color Animation biến đổi màu giữa hai biến của đối tượng Animation bằng cách sử dụng nội suy tuyến tính trên một khoảng thời gian xác định

```
<StackPanel x:Name="myStackPanel" Background="Red"
Loaded="Start_Animation">
    <StackPanel.Resources>
        <Storyboard x:Name="colorStoryboard">

            <!-- Animate the background color of the canvas from red to
green
over 4 seconds. -->
            <ColorAnimation BeginTime="00:00:00"
Storyboard.TargetName="myStackPanel"

Storyboard.TargetProperty="(Panel.Background).(SolidColorBrush.Color)"
From="Red" To="Green" Duration="0:0:4" />

        </Storyboard>
    </StackPanel.Resources>
</StackPanel>
private void Start_Animation(object sender, EventArgs e)
{
    colorStoryboard.Begin();
}
http://samples.msdn.microsoft.com/Silverlight/silverlight\_next/Animations/coloranimation/ClientBin/TestPage.html
```

1.5 Point Animation

Point Animation biến đổi giá trị tọa độ của một điểm giữa hai biến của đối tượng Animation bằng cách sử dụng nội suy tuyến tính trên một khoảng thời gian xác định

```

<Canvas Width="400" Height="300">
  <Canvas.Resources>
    <Storyboard x:Name="myStoryboard">

      <!-- Animate the center point of the ellipse from 100 X,
300 Y
      to 400 X, 100 Y over 5 seconds. -->
      <PointAnimation
        Storyboard.TargetProperty="Center"
        Storyboard.TargetName="MyAnimatedEllipseGeometry"
        Duration="0:0:5"
        From="100,300"
        To="400,100"
        RepeatBehavior="Forever" />

    </Storyboard>
  </Canvas.Resources>
  <Path Fill="Blue" Loaded="Start_Animation">
    <Path.Data>

      <!-- Describes an ellipse. -->
      <EllipseGeometry x:Name="MyAnimatedEllipseGeometry"
        Center="200,100" RadiusX="15" RadiusY="15" />
    </Path.Data>
  </Path>
</Canvas>
// Start the animation when the object loads
private void Start_Animation(object sender, EventArgs e)
{
    myStoryboard.Begin();
}

```

http://samples.msdn.microsoft.com/Silverlight/silverlight_next/Animations/pointanimation/ClientBin/TestPage.html

2 Media

Các tính năng đa phương tiện của Silverlight và mô tả cách tích hợp audio và video vào trang Silverlight

2.1 MediaElement Object

Để thêm media(video, audio) vào trang web của bạn ta sử dụng MediaElement .
MediaElement sẽ cung cấp cho bạn một Form để hiển thị media , bạn chỉ cần cung cấp các

nguồn tài nguyên media cho URI, như là một file video chẳng hạn. Các MediaElement bắt đầu play khi trang web của bạn tải về.

```
<MediaElement x:Name="media" Source="xbox.wmv"
    Width="300" Height="300" />
http://samples.msdn.microsoft.com/Silverlight/Silverlight_1_0/media/media_ov
erview_snip/js/media_simple.html
```

2.2 Controlling Media Playback Interactively

Bạn có thể tương tác điều khiển media bằng cách sử dụng Play, Pause, Stop, và các thuộc tính của một đối tượng MediaElement. Ví dụ như định nghĩa một đối tượng MediaElement với một số nút điều khiển cho các file (video, audio) phát.

```
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <MediaElement x:Name="media" Source="xbox.wmv" Width="300"
Height="300"
                Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3" />

    <!-- Stops media playback.-->
    <Button Click="StopMedia"
        Grid.Column="0" Grid.Row="1" Content="Stop" />

    <!-- Pauses media playback. -->
    <Button Click="PauseMedia"
        Grid.Column="1" Grid.Row="1" Content="Pause" />

    <!-- Begins media playback. -->
    <Button Click="PlayMedia"
        Grid.Column="2" Grid.Row="1" Content="Play" />

</Grid>

private void StopMedia(object sender, RoutedEventArgs e)
{
    media.Stop();
}
private void PauseMedia(object sender, RoutedEventArgs e)
{
    media.Pause();
}
private void PlayMedia(object sender, RoutedEventArgs e)
{
    media.Play();
}
```

```
}  
http://samples.msdn.microsoft.com/Silverlight/silverlight\_next/Media/media\_ovw\_controlling\_media/ClientBin/TestPage.html
```

2.3 Timeline Markers

Một timeline marker là siêu dữ liệu liên kết với một điểm trong một tập tin media. các đánh dấu thường được tạo trước và được lưu trữ trong các tập tin media riêng. chúng thường được sử dụng để đặt tên khác nhau trong một cảnh quay video, cho phép người sử dụng tìm đến một vị trí hoặc cung cấp các phân cảnh.

Khi MediaElement đạt đến một thời gian đánh dấu trong quá trình phát, nó sẽ tăng MarkerReached, sự kiện này có thể sử dụng để kích hoạt hành động khác.

Một MediaElement của đối tượng Marker đều cho phép bạn truy cập vào tiêu đề (nhúng, đánh dấu lưu trữ trong tập tin media).

Một tập tin có thể chứa cả hai bao gồm tiêu đề nhúng tập lệnh và tập lệnh riêng biệt

```
<StackPanel Margin="40">  
    <StackPanel Orientation="Horizontal">  
        <TextBlock FontSize="12"  
Foreground="DarkGray">Time:</TextBlock>  
        <TextBlock x:Name="timeTextBlock" FontSize="12" />  
    </StackPanel>  
    <StackPanel Orientation="Horizontal">  
        <TextBlock FontSize="12"  
Foreground="DarkGray">Type:</TextBlock>  
        <TextBlock x:Name="typeTextBlock" FontSize="12" />  
    </StackPanel>  
    <StackPanel Orientation="Horizontal">  
        <TextBlock FontSize="12"  
Foreground="DarkGray">Value:</TextBlock>  
        <TextBlock x:Name="valueTextBlock" FontSize="12" />  
    </StackPanel>  
  
    <!-- The MediaElement has the MarkerReached event attached. -->  
    <MediaElement MarkerReached="OnMarkerReached"  
HorizontalAlignment="Left"  
Source="thebutterflyandthebear.wmv" Width="300" Height="200" />  
  
</StackPanel>  
  
private void OnMarkerReached(object sender,  
TimelineMarkerRoutedEventArgs e)  
{  
    timeTextBlock.Text = e.Marker.Time.Seconds.ToString();  
    typeTextBlock.Text = e.Marker.Type.ToString();  
    valueTextBlock.Text = e.Marker.Text.ToString();  
}
```

2.4 Server-Side Playlist

Một server-side playlist là một dãy các tài nguyên media cho phép quản trị viên trên máy chủ kiểm soát các tài nguyên media của người sử dụng.

một danh sách media phía máy chủ được sử dụng để phục vụ việc phát media, nó có thể ko không cho phép download từ client.

Silverlight sử dụng tập tin wsx để cấu hình xác định một server-side playlist(SSPL) có thể được phục vụ cho client thông qua các đối tượng MediaElement.

Một số lợi ích của việc sử dụng (SSPL) như sau

- Giảm băng thông do giảm số lần phía client kết nối tới server để lấy nội dung.
- Bạn có thể tùy chọn danh sách media khi mà bạn đang phát một danh sách media từ trước.
- Bạn có thể kiểm soát, bổ sung media. Ví dụ bạn có thể chơi chỉ một phần của một tập tin media.

```
<?wsx version="1.0"?>
<smil>
  <seq id="sq1">
    <media id="video1" src="clip1.wmv" />
    <media id="video2" src="clip2.wmv" />
    <media id="video3" src="clip3.wmv" />
  </seq>
</smil>
```