

Disabling Actions during Animation

Christian Wenz

Overview

The Animation control in the ASP.NET AJAX Control Toolkit is not just a control but a whole framework to add animations to a control. It also supports actions, like mouse clicks. However when a mouse click starts an animation, it is desirable to disable mouse clicks during the animation.

Steps

First of all, include the **ScriptManager** in the page; then, the ASP.NET AJAX library is loaded, making it possible to use the Control Toolkit:

```
<asp:ScriptManager ID="asm" runat="server" />
```

The animation will be applied to an HTML button like this:

```
<input type="button" ID="Button1" runat="server" Value="Launch Animation" />
```

Note that an HTML Control is used instead of a Web Control since we do not want the button to create a postback; it shall just launch the client-side animation for us.

Then, add the **AnimationExtender** to the page, providing an **ID**, the **TargetControlID** attribute and the obligatory **runat="server"**:

```
<ajaxToolkit:AnimationExtender ID="ae" runat="server" TargetControlID="Button1">
```

Within the **<Animations>** node, **<OnClick>** is the right element to handle the mouse click. However, the button could be clicked during the animation, as well. The **<EnableAction>** element can take care of that. Setting **Enabled="false"** disables the button as part of the animation. Since we are using several individual animations (disabling the button and the actual animations), the **<Parallel>** element is required to glue the single animations together into one. Here is the complete markup for **AnimationExtender**:

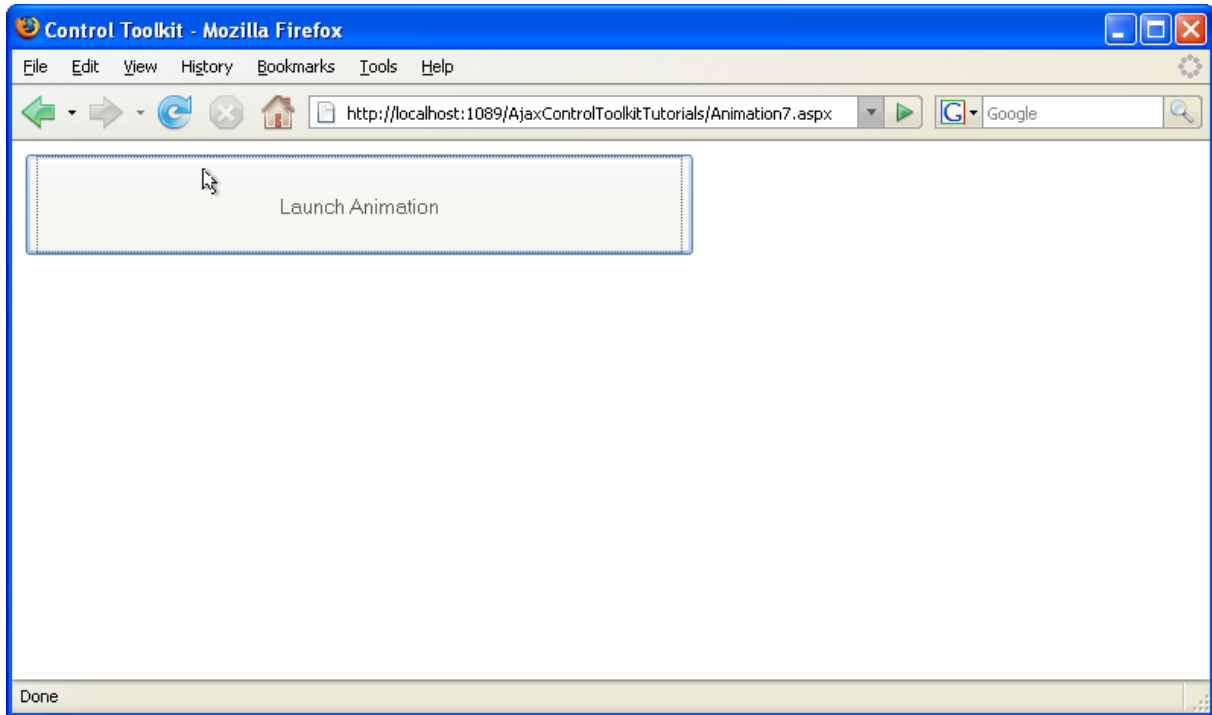
```
<ajaxToolkit:AnimationExtender ID="ae" runat="server" TargetControlID="Button1">
  <Animations>
    <OnClick>
      <Parallel>
        <EnableAction Enabled="false" />
        <FadeOut Duration="1.5" Fps="24" />
        <Resize Width="1000" Height="150" Unit="px" />
      </Parallel>
    </OnClick>
  </Animations>
</ajaxToolkit:AnimationExtender>
```

```
</Animations>  
</ajaxToolkit:AnimationExtender>
```

It would also be possible to re-enable the button after the animation, using the following XML element at the end of the list:

```
<EnableAction Enabled="true" />
```

However in the given scenario this would be useless since the button fades out and is not visible at the end of the animation.



The button is disabled as soon as the animation runs