



Part 7

With Lloyd Humphreys

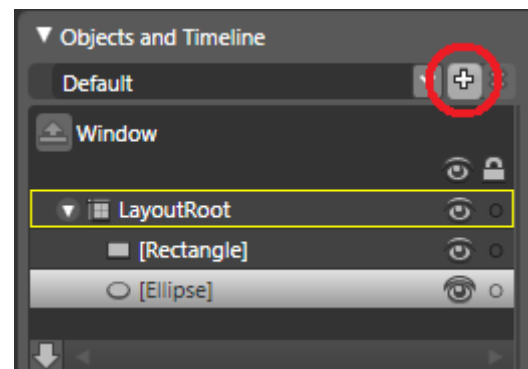
(<http://www.lloydhumphreys.com>)

Animation

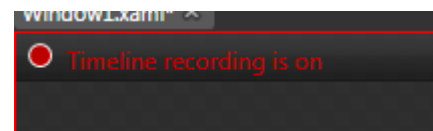
Animation in Expression Blend is extremely simple. In fact, monkeys could work it out, if they could operate computers. But they can't, so monkey-anime will never be, unfortunately.

To begin with, we're going to have to create a Timeline, but first I think it's important to set up the toolbars/panels so it's easier to animate. To do so, hit F7; or click Window > Animation Workspace. To change back, hit F6. After this, add anything you wish to animate (in my case, I'm going to animate a ball bouncing around in a box).

To create a timeline, click the small plus sign on the Objects and Timeline panel. This will pop open a dialog box; "Create Storyboard Resource". Name it what you like, in this instance I'll call it BouncyBall. And of course, to delete them, click the "X" next to the plus sign when illuminated.



So, once you've done this you'll notice the design area is now flanked by a red line, and a timeline has appeared at the bottom. You'll also probably notice a little icon in the top indicating that Timeline Recording is on. This means you don't need to manually create keyframes, and that Blend will do it for you.



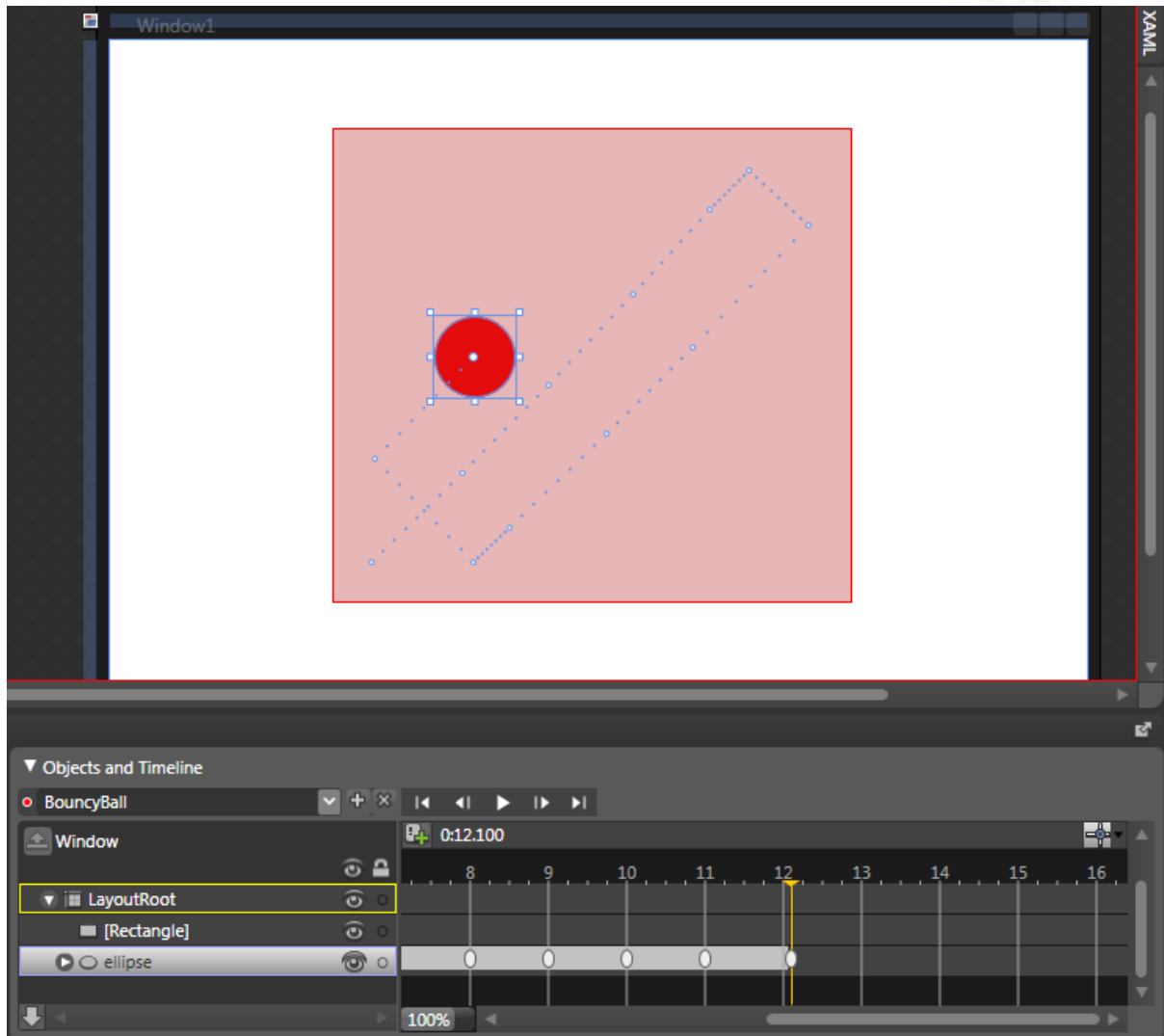
If you head over to the XAML, you should see something like this:

```
<Window.Resources>
  <Storyboard x:Key="BouncyBall">
```

This is basically telling us that in the resources, there's a new Storyboard named BouncyBall. We'll explore XAML in more depth later on; by making things repeat and reverse, etc.

Smooth Your Animations

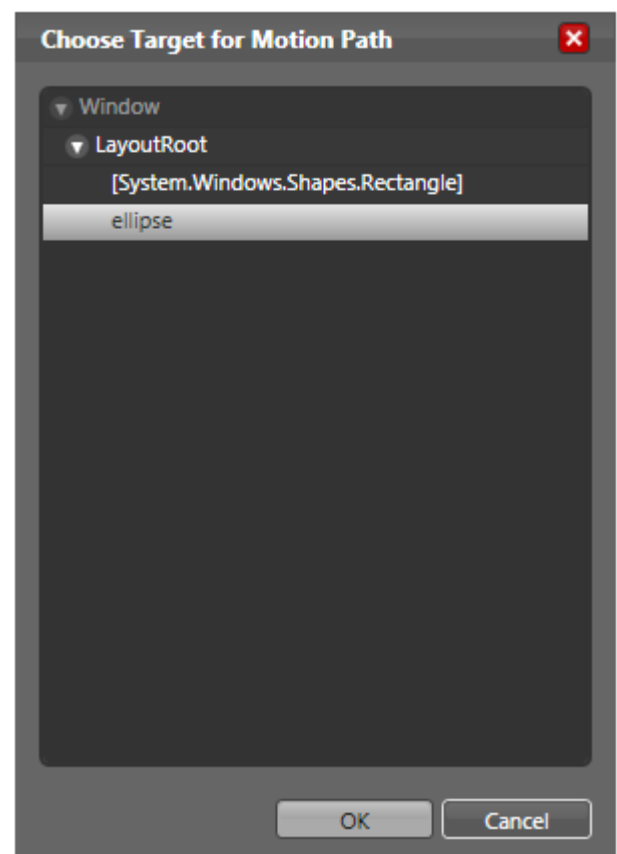
By right clicking on each Keyframe marker (the little white dot made when you add a Keyframe), you can set Ease In and Ease Out properties. This makes the animation smoother as it comes towards each marker, and as it leaves. The "Hold Out" property means that it will not smooth at all, and will simply jump without any smooth effects at all.



In my example, I'm simply having a ball bounce around a little, and I'll include it with this tutorial. So, I click on the 1 second marker, and drag the ball a little. Since Timeline Recording is on, an ellipse will appear on the 1 second line, telling us that it has recorded a movement. Then I click on the 2 second, and move it again – another ellipse. Then hit F5 and watch the ball move around!

Path Animation

Animating along a path is simple too. Say for example I wanted the circle to travel around the square, instead of bounce inside of it. I select the square, head to Object, Path and "Convert to Motion Path". Of course, something will have to be tied to the motion path, and in this case, the "ellipse", as I've selected here. Note that it's been named for you. Then simply hit play, and the circle will begin following the lines – but only once. In order to get it

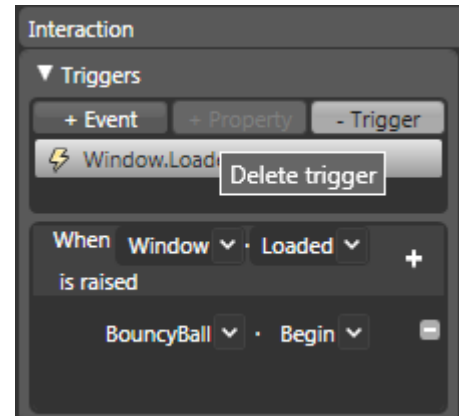


to repeat, you'll have to go into some XAML – which I'll cover later on.

Triggers

So far, should you have tested the application you'll have noticed that everything begins to happen instantly. But what if you want some control? This is where triggers are used.

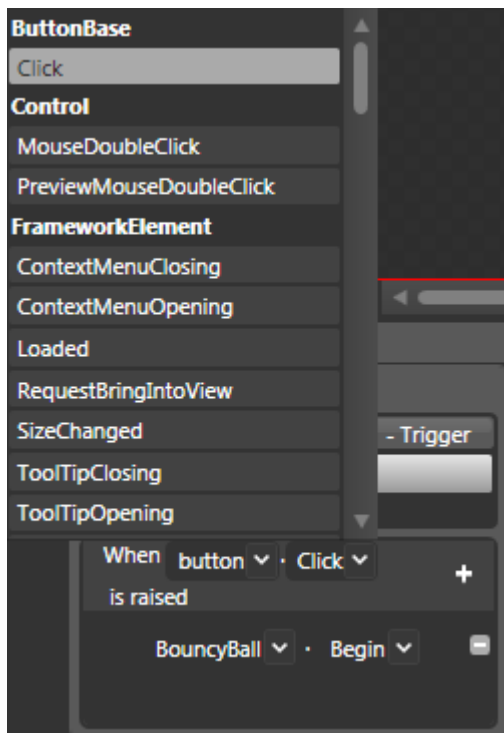
If you expand the triggers panel on the left, you'll see a "Window.Loaded" trigger. This means that when the window loads, "trigger" the beginning of the animation. To create our own trigger, we must first delete the old one. Select "Window.Loaded" and click "-Trigger".



To create a new Trigger, we must change the

When [Window] : [Loaded]

Into :



This tells Expression that when we click the button, begin the BouncyBall timeline.

Of course, we can also stop it, pause it, etc. It's all there in the dropdown. There are so many different triggers in the actions element (the dropdown where it says "click") that I could write a whole tutorial detailing what each one does. For further information on these, you'll just have to check out the awesome documentation for Blend.

XAML Editing

You'll have to edit some XAML while animating, even for the most basic things. You can even create new triggers using XAML, and some people actually prefer to do that.

There are many things you can do in the XAML, including repeating the animation. To do this, you'll have to edit part of the storyboard tag and add the following:

```
<Storyboard x:Key="BouncyBall" RepeatBehavior="Forever">
```

As you may have guessed, this repeats the animation indefinitely.

You can also automatically reverse each animation. So, when it hits the end of the animation, it will just reverse, and this can sometimes work rather well.

To do this, simply add

```
<Storyboard x:Key="BouncyBall" RepeatBehavior="Forever" AutoReverse="True">
```