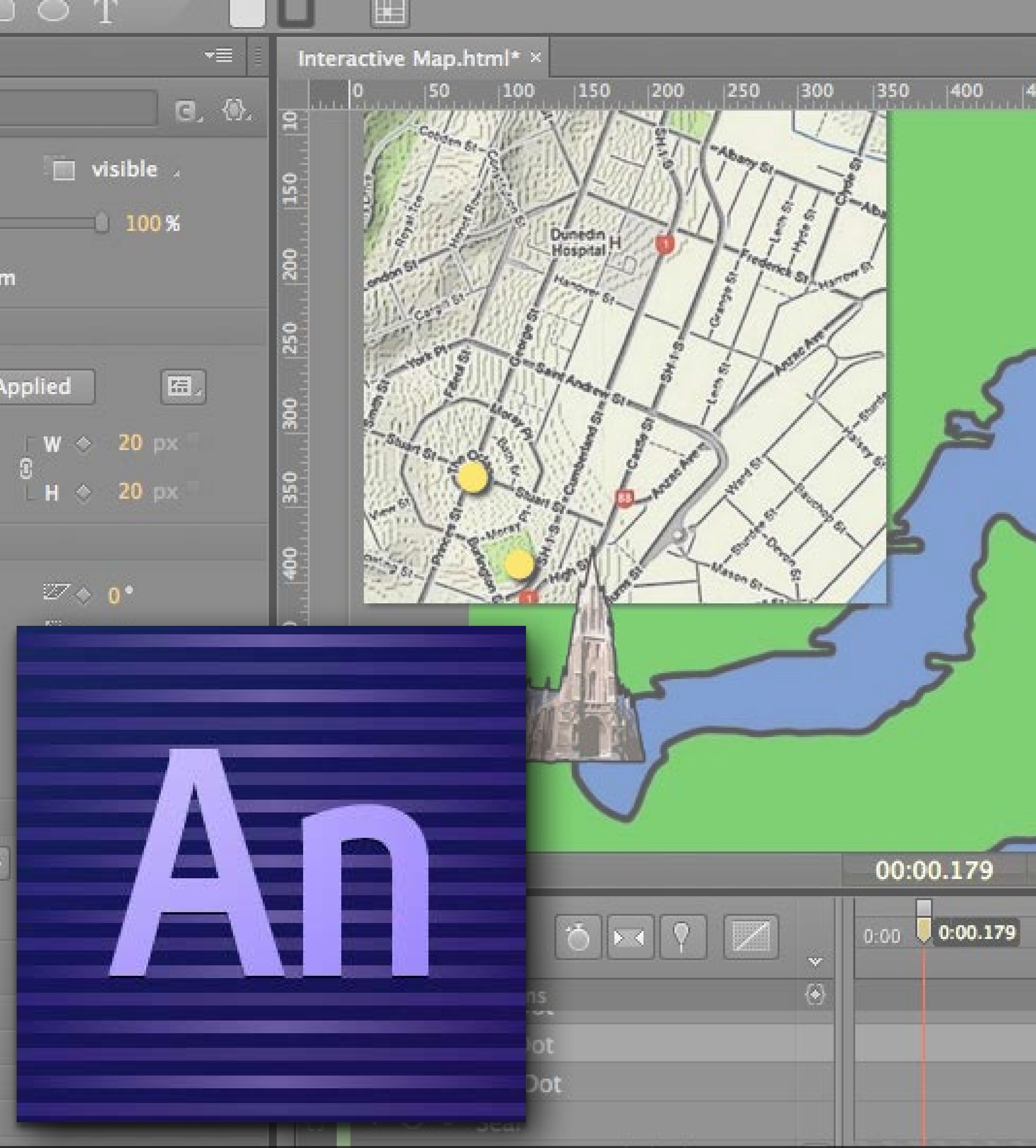
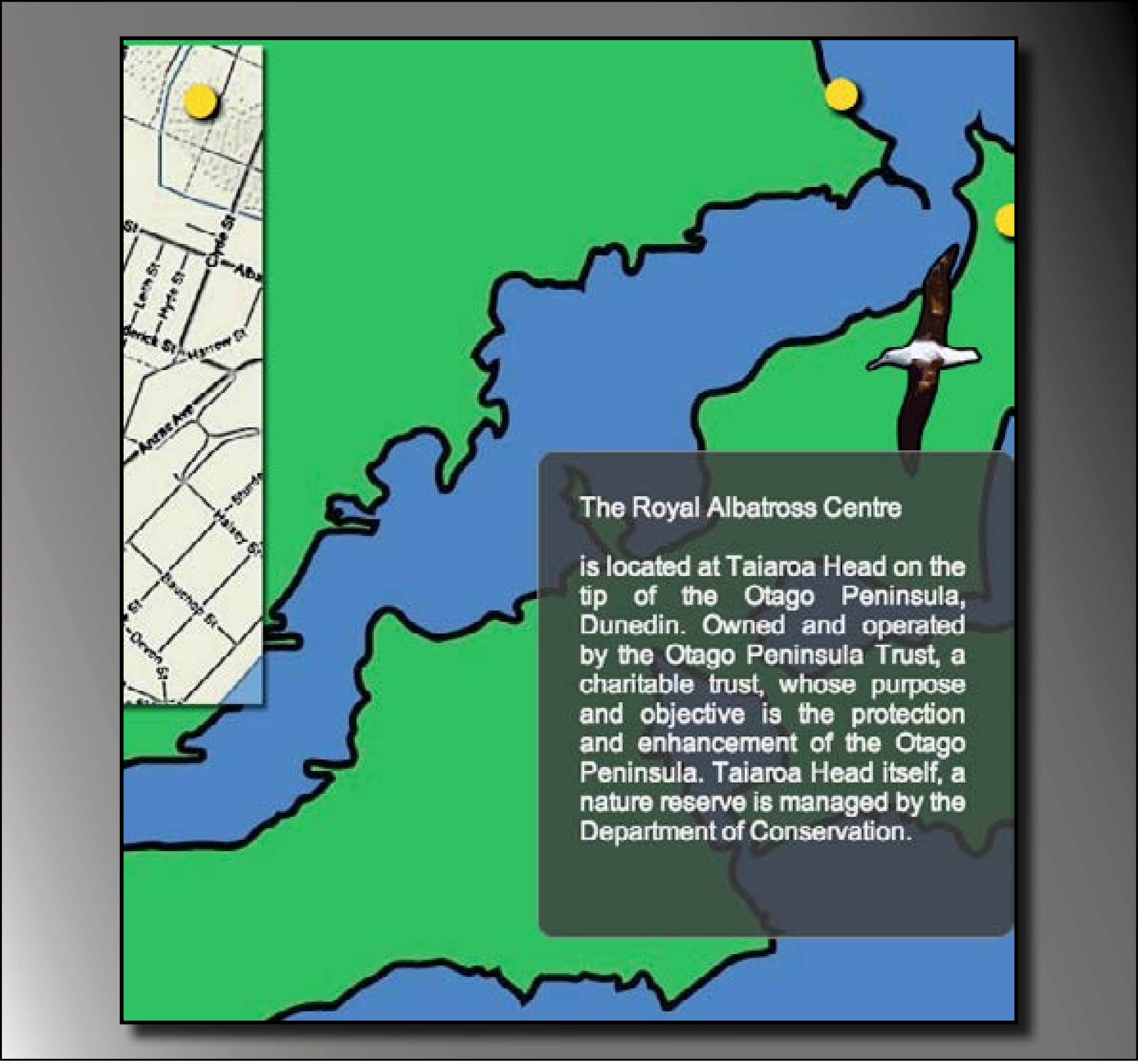


Adobe Edge Animate Tutorial
Mouse Events

Interactive Tourist Map



The Royal Albatross Centre

is located at Taiaroa Head on the tip of the Otago Peninsula, Dunedin. Owned and operated by the Otago Peninsula Trust, a charitable trust, whose purpose and objective is the protection and enhancement of the Otago Peninsula. Taiaroa Head itself, a nature reserve is managed by the Department of Conservation.

Lesson 3

Animate a symbol

Now that you've got a map with landmarks which can be clicked to open another web page, it's time to add some more fun to your interactive map.

This lesson aims to teach you how to:

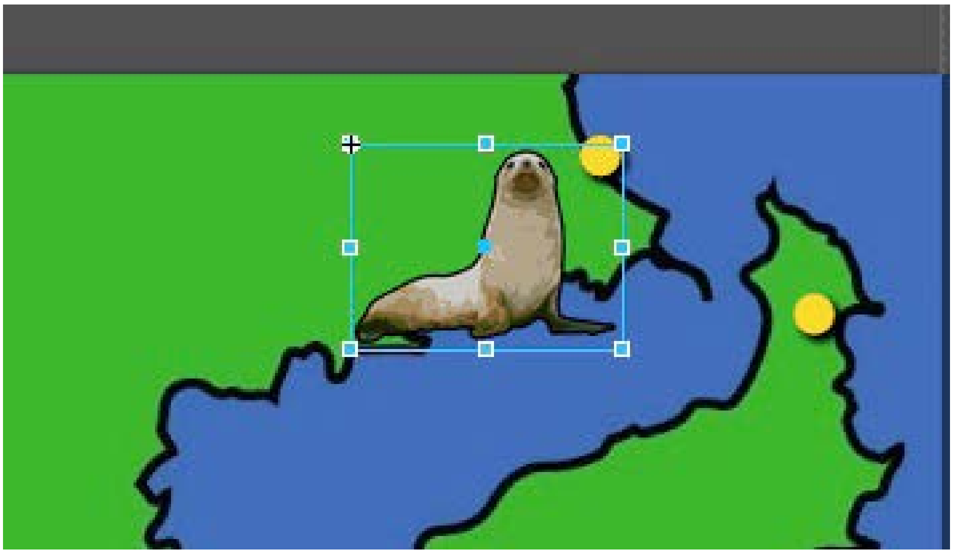
- Turn images into symbols
- Use a symbol's timeline
- Create animation transitions
- Employ more mouse events

1. Setting up for symbols

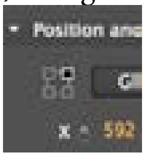
Remember the images you imported in the first lesson? In this lesson you'll put to use all the images you haven't already placed on the stage.

Drag the image called **seal** onto the stage. It will probably be quite big, so you'll need to drag a corner in to get it the right size. Make sure you hold down the shift key while you drag so that your image doesn't change shape.

Align the image so that one corner is near the dot you created in earlier lessons for where seals can be found. It should look a bit like the picture below.



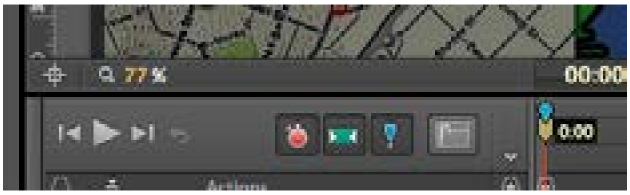
Ensure that your picture is at a lower level than the dot so that, even when the picture can't be seen, the dot is still hover- and click-able. Do this by dragging the name of the image below the name of the dot in the **Elements** panel. Also, right under where it says Position and Size in the Properties panel, change the relative corner to be the one closest to the dot. In this case, top right.



2. Creating the animation

Now that you've got an image on the stage, right click on it and select **Convert to symbol...** just like you did with the dot in an earlier lesson. Double click on you nice new (and sensibly named, of course) symbol to enter it's timeline. The rest of the stage will darken.

You'll notice that now the **Timeline** panel at the bottom only contains the seal. You're going to use this to create a transition. Make sure all the buttons above the timeline controls are lit to tell you they're on.



Auto-keyframe mode

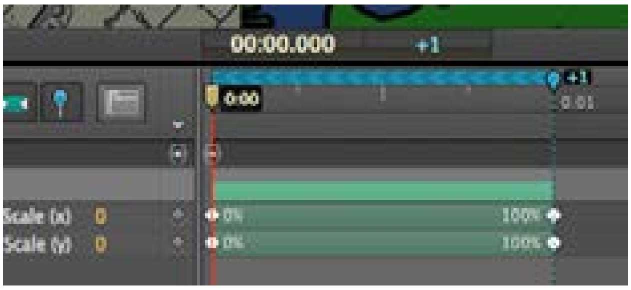
Auto-transition mode

Toggle pin

Click the **Easing** selector next to the blue toggle pin button and, since you are going to make this a fun animated bouncing transition, choose **Ease out: Elastic**.

Now drag the blue pin to about 0:01 on the timeline (that's one second) and you'll see a blue section form behind it. The arrows in that blue section indicate that whatever property you change will apply to the start of the transition, and the current state of the object will be the end of the transition.

Aside:
There are lots of ways to make a transition. You can do the whole thing manually by dragging the whole playhead and using the properties panel or you can drag the pin or the yellow part of the playhead to highlight the transition scope.



In the **Properties** panel, make sure the **Scale** controls are linked and then change one to *0*. They should both change and you should see a transition appear on the timeline.

You also only want this to happen when you tell it to, not as soon as you visit the map. To do this, click once off the seal picture to deselect it (but stay in symbol edit mode) and you'll see the **Properties** panel change. Find the tick box for **Autoplay** and click it to take away the tick.

Press the space bar or the triangular play button to make sure your transition behaves in the right way, then head back to the **Stage**. You shouldn't be able to see your picture there at all, but if you click where you know it is you'll get that blue box around the invisible symbol.

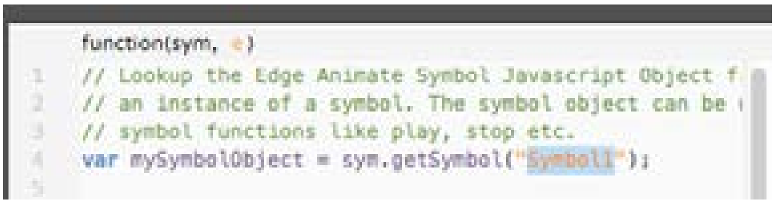
3. Inserting code

Just like you did when you make click events in lesson 2, you've going to make another mouse event for the dot that causes the picture of the seal to pop up when the mouse hovers over the dot. This is called, oddly enough, a **mouseover**.

Open the code window for the dot by the seal by selecting it and clicking the code dot by its name. This time, click the plus symbol in the top left corner of the code window and choose **mouseover** from the list.

With the code window open and your click event ready to be filled with code, you're going to again make use of the handy code snippet shortcuts to the right of the code panel.

Click on the Get Symbol code snippet (you may have to scroll down the list to find it) and you'll get code that looks a bit like this:



```
function(sym, e)
// Lookup the Edge Animate Symbol Javascript Object f
// an instance of a symbol. The symbol object can be
// symbol functions like play, stop etc.
var mySymbolObject = sym.getSymbol("Symbol1");
```

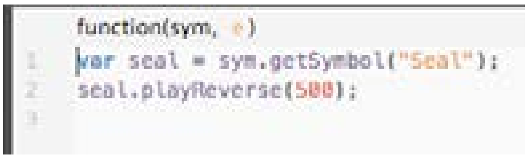
You're going to need to change some things to make this code apply to your new symbol. You can also dlete the comments in green if you want to tidy up a bit.

Where it says `mySymbolObject`, change it to say something like *seal*. Where it says `“Symbol1”`, change it to the exact name you gave your seal symbol, with the quotation marks still around it. Like *“Seal”*. Now your mouseover event knows you want to refer to that seal symbol for what to do next.

TIP
When you write code, you have to be as accurate as possible. One little typo could mean nothing will happen at all. That's why code snippets are so handy - you just change a few things to suit your project and there's far less opportunity for typos.

You haven't quite finished putting in code yet! You now want that symbol to play its nice bouncing animation. To do this, click the button to add the code snippet called **Play from**. It will add another line of code to your event (and another comment too). Where it says `sym.play`, change it to say *seal.play* (where ‘seal’ is the exact same word you used above to replace mySymbolObject). Then in the brackets, where it says `1000`, change it to just read *0*. This will make your symbol play from the beginning when the mouse hovers over the dot.

But you're still not done yet! You need the picture to bounce back in when the mouse comes off the dot. To do this, add another mouse event called **mouseout**. Use the same exact **Get Symbol** code as you did in the **mouseover**, but then instead of a **Play from** you want a **Play reverse** (to make it bounce back in) and probably start it at around *500*. It should then look like this:



```
function(sym, e)
var seal = sym.getSymbol("Seal");
seal.playReverse(500);
```

Now when you test your project using **Ctrl+Enter** or **Cmd+Return** you should be able to see your seal picture pop up out of the dot when you hover the mouse over it, and pop bak in when you move off.

Well done!

You've successfully completed the thirs lesson and learned how to animate your Adobe Edge Animate symbol.

- You've placed and image on the stage and made a symbol,
- the symbol has an animated transition,
- and you've made mouse events with code which plays your animation.

For more practice at the new skills you have learnt, visit **edgemap.weebly.com/exercise.html**