

Intermediate iOS Animation

Hands-on Challenges

Intermediate iOS Animation Hands-On Challenges

Copyright © 2016 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.



Challenge A: Custom animation easing

Unlike in UIKit and view animations, with Core Animation and layer based animations you can actually set your own custom easing.

In this challenge you will get to try some custom curves for your animations just to get a feeling of what's possible.

Open **ViewController.swift** and find the line where you set the easing on your `flyRight` animation:

```
flyRight.timingFunction = CAMediaTimingFunction(name:
    kCAMediaTimingFunctionEaseOut)
```

You are using one of the pre-defined easing functions by providing a name to the function initializer.

There is a different `init` on `CAMediaTimingFunction` that allows you to use a cubic-bezier curve to define your own custom easing. Let's try a couple of things – replace the line above with the following:

```
flyRight.timingFunction = CAMediaTimingFunction(controlPoints:
    0.11, 1, 0.83, 0.67)
```

Run the app and check the text fields animation – the fields slow down way too early and almost stall for a bit before they pick up again towards the finishing line. You can try also another curve:

```
flyRight.timingFunction = CAMediaTimingFunction(controlPoints:
    0.99, 0, 0, 0.99)
```

This curve makes the fields stall for way too long in the beginning of the animation and almost instantly fly through the rest of the way :]

To start designing your own bezier curves I recommend: <http://cubic-bezier.com/>

This site allows you to visually design and try different easings and when you're done just copy over the four float numbers as the control points of `CAMediaTimingFunction`.

Have fun!

