iOS Animation with Swift

Hands-On Challenges

iOS Animation with Swift Hands-On Challenges

Challenge E: Keyframe animations

This challenge will get you to excercise the skills to create a basic keyframe animation.

Add a new method to the view controller class to define the keyframe animation for this challenge and position the status banner out of the screen bounds:

```
func animateStatusBannerWithKeyframes() {
  statusBanner.center.x -= view.bounds.size.width
  statusBanner.center.y -= 50.0

}
```

This code will move the button to the left of the screen and a bit upwards. Next add the keyframe animation to that same method:

```
UIView.animateKeyframesWithDuration(1.0, delay: 0.0, options: nil,
animations: {

}, completion: nil)
```

This configures the total duration and the only thing left to do is to add some keyframes from within the animations block.

First add the initial phase, in which the banner flies in the screen sideways (**NB:** make sure the code is inside the animations block):

```
UIView.addKeyframeWithRelativeStartTime(0.0, relativeDuration:
0.33, animations: {
  self.statusBanner.center.x += self.view.bounds.size.width
})
```

Then add the second keyframe to the animations block to move the button down to its final position:

```
UIView.addKeyframeWithRelativeStartTime(0.33, relativeDuration:
0.66, animations: {
  self.statusBanner.center.y += 50.0
})
```

That’s it for this challenge – to give the app a test you only need to call this new method at the bottom of `changeFlightDataAnimatedTo(data:)`:

```
animateStatusBannerWithKeyframes()
```

If you want to experiment a little bit you can add more keyframes to the animation and make the button make more zig-zags until it reaches its final position on screen.