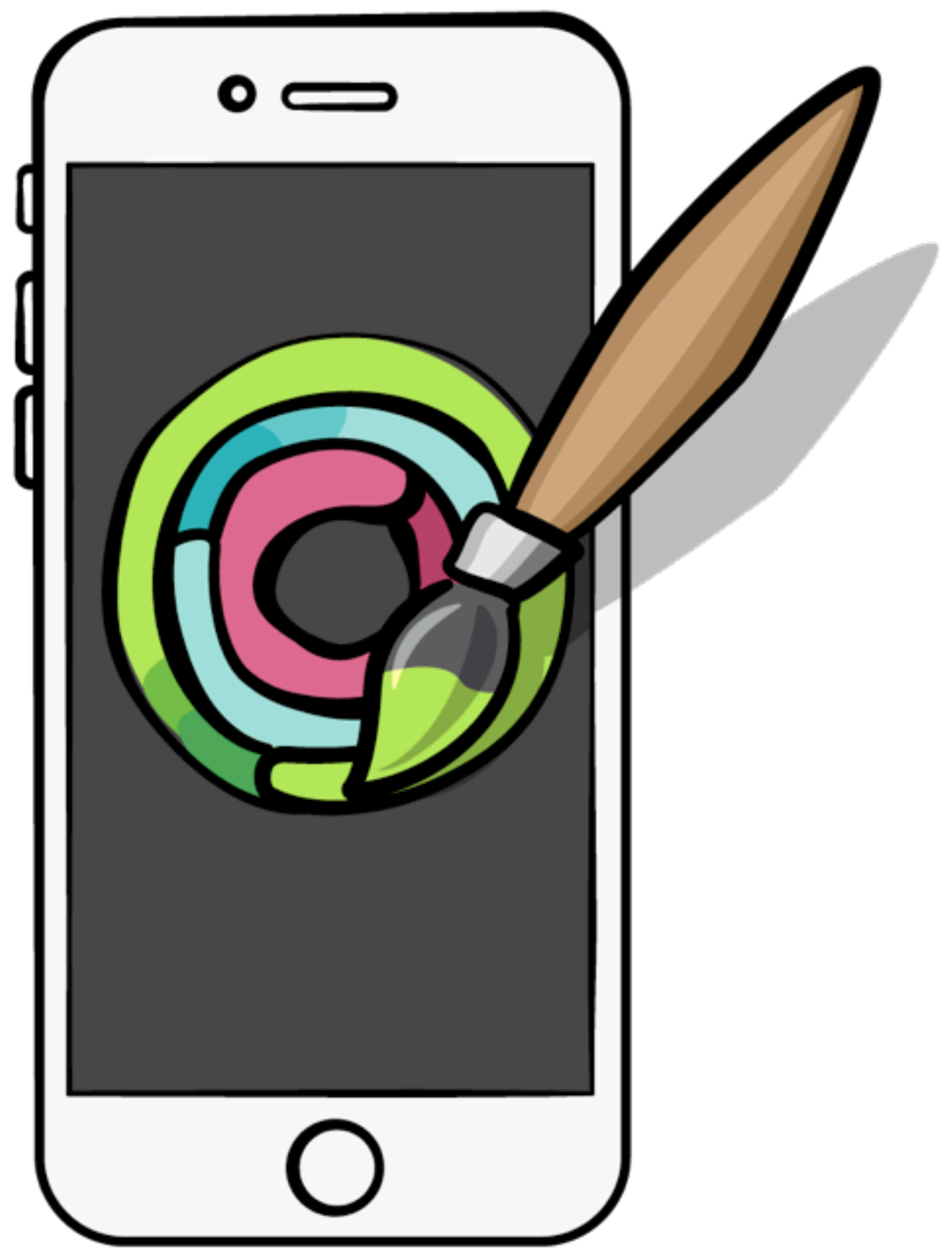
CUSTOM CONTROLS IN iOS



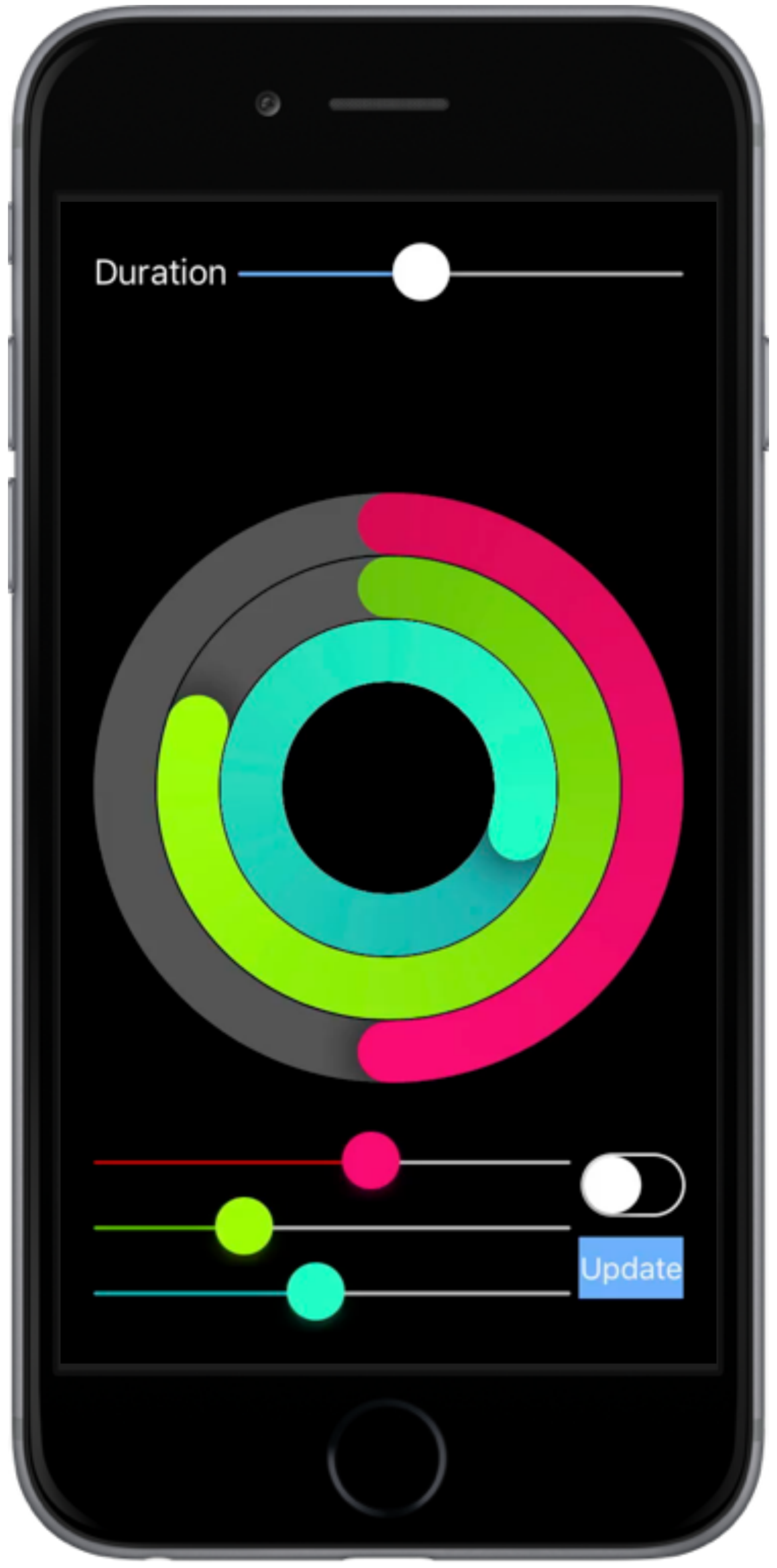
PART 12: CONCLUSION

Act I - Deluxe Button

9:41 AM
100%
ship it
DRINK ME!

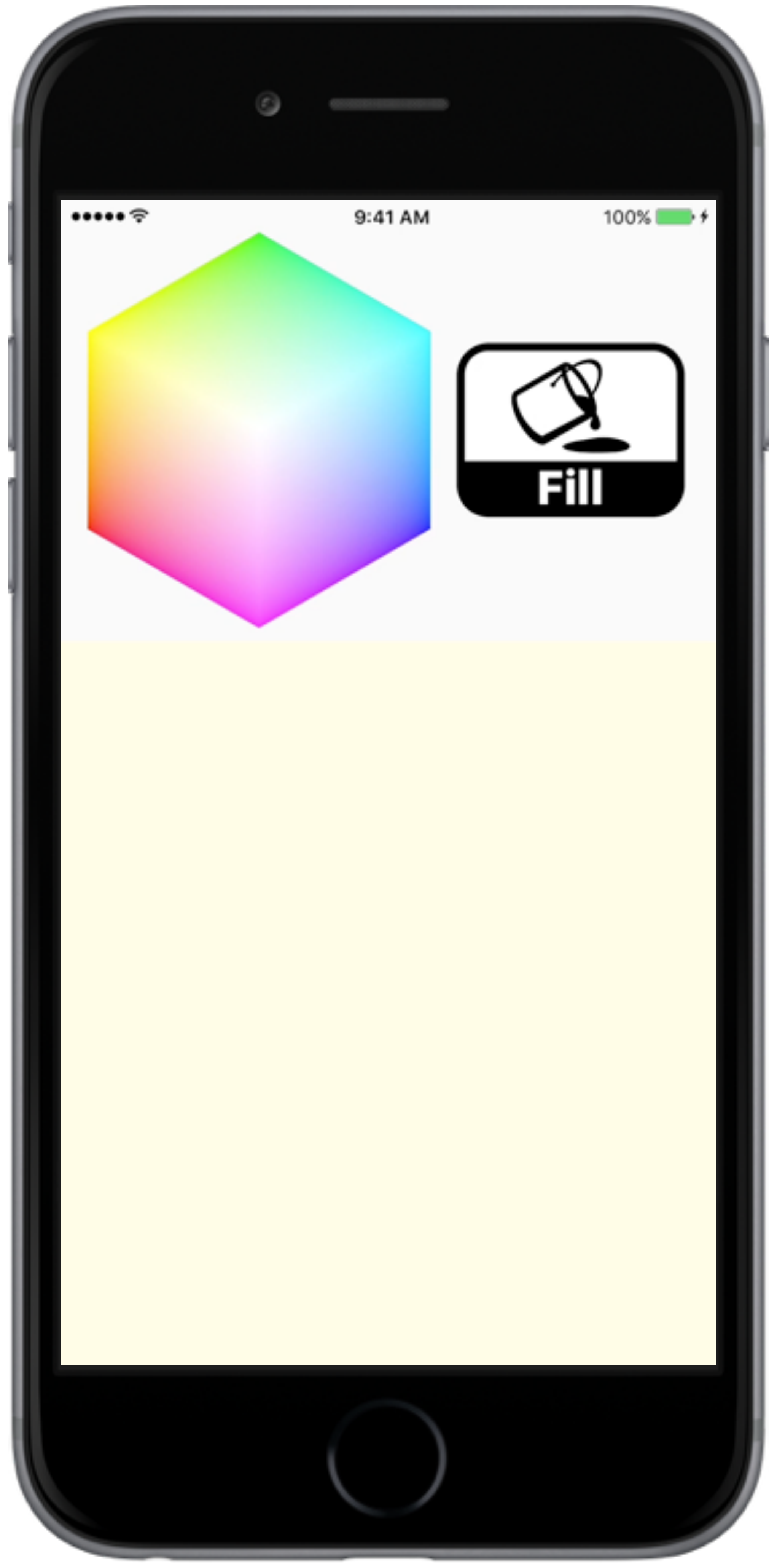
- Creating a custom control via composition
- Playground Driven Development
- Integration with Interface Builder
- UIControl to implement interaction

Act II - Three Ring Control



- Construction of the appearance using Core Animation layers
- Adding animation to a custom control

Act III - Sketchpad



- Making controls reusable across projects
- Using Core Graphics and Core Image to draw the interface
- Creating custom gesture recognizers to enhance interaction

Which Approach?

UIKit

Core Animation

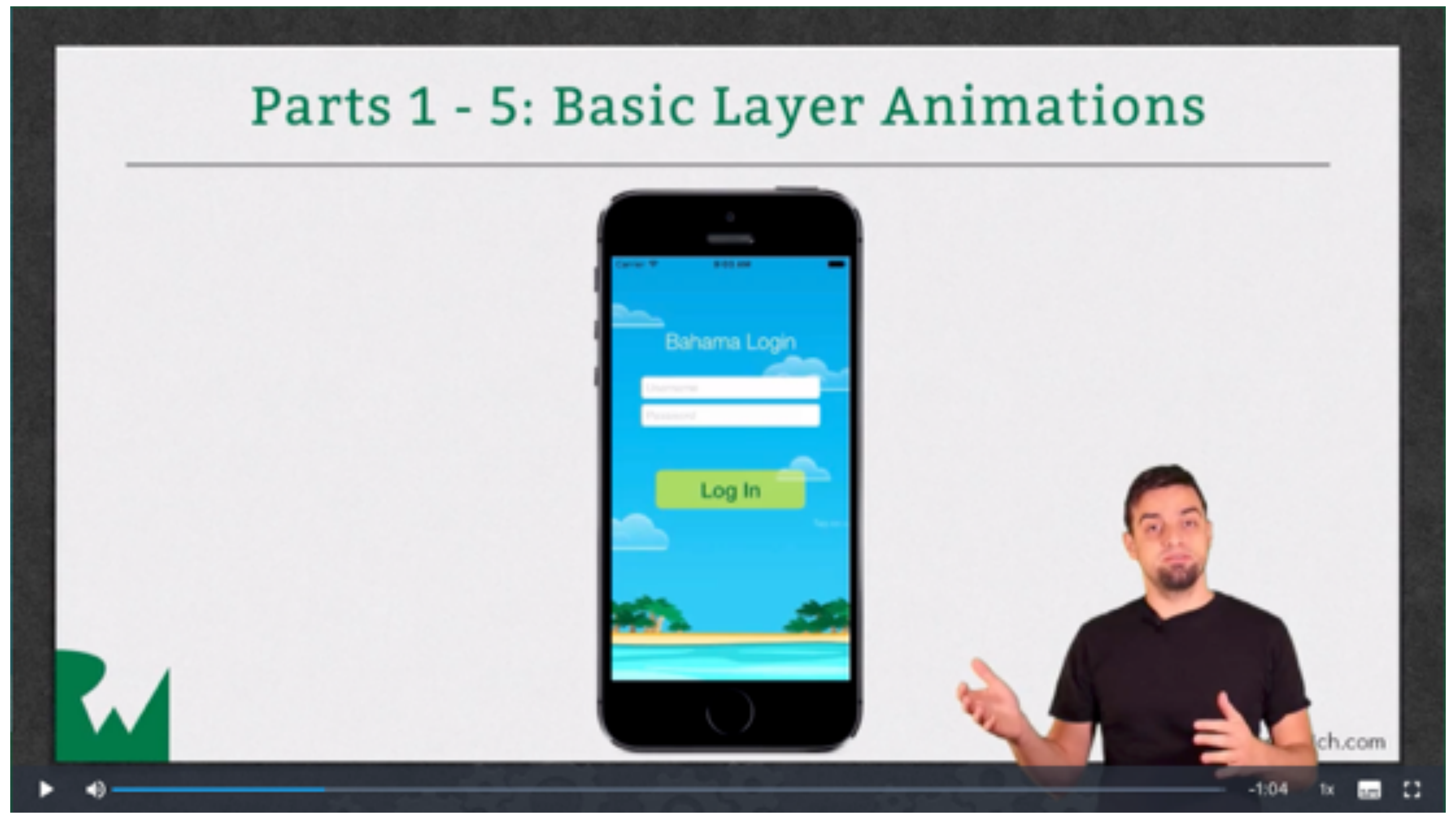
Core Graphics

Core Image

Where to go from here?

What Are CALayers?

CALayers


Parts 1 - 5: Basic Layer Animations

iOS
Animations

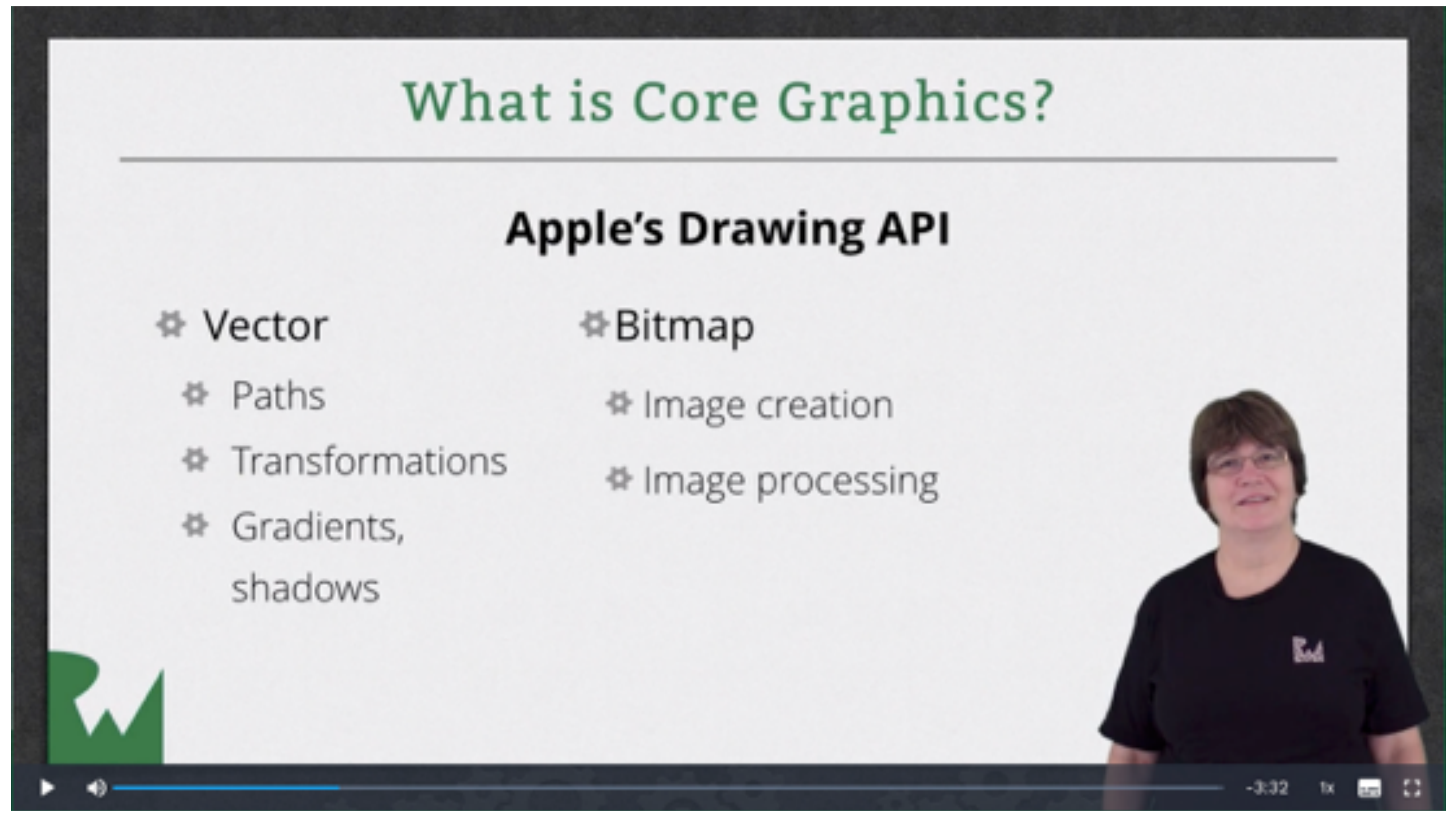
Up to date for iOS 10, Xcode 8 & Swift 3

iOS Animations by Tutorials

THIRD EDITION

iOS 10 and Swift 3 edition

By Marin Todorov

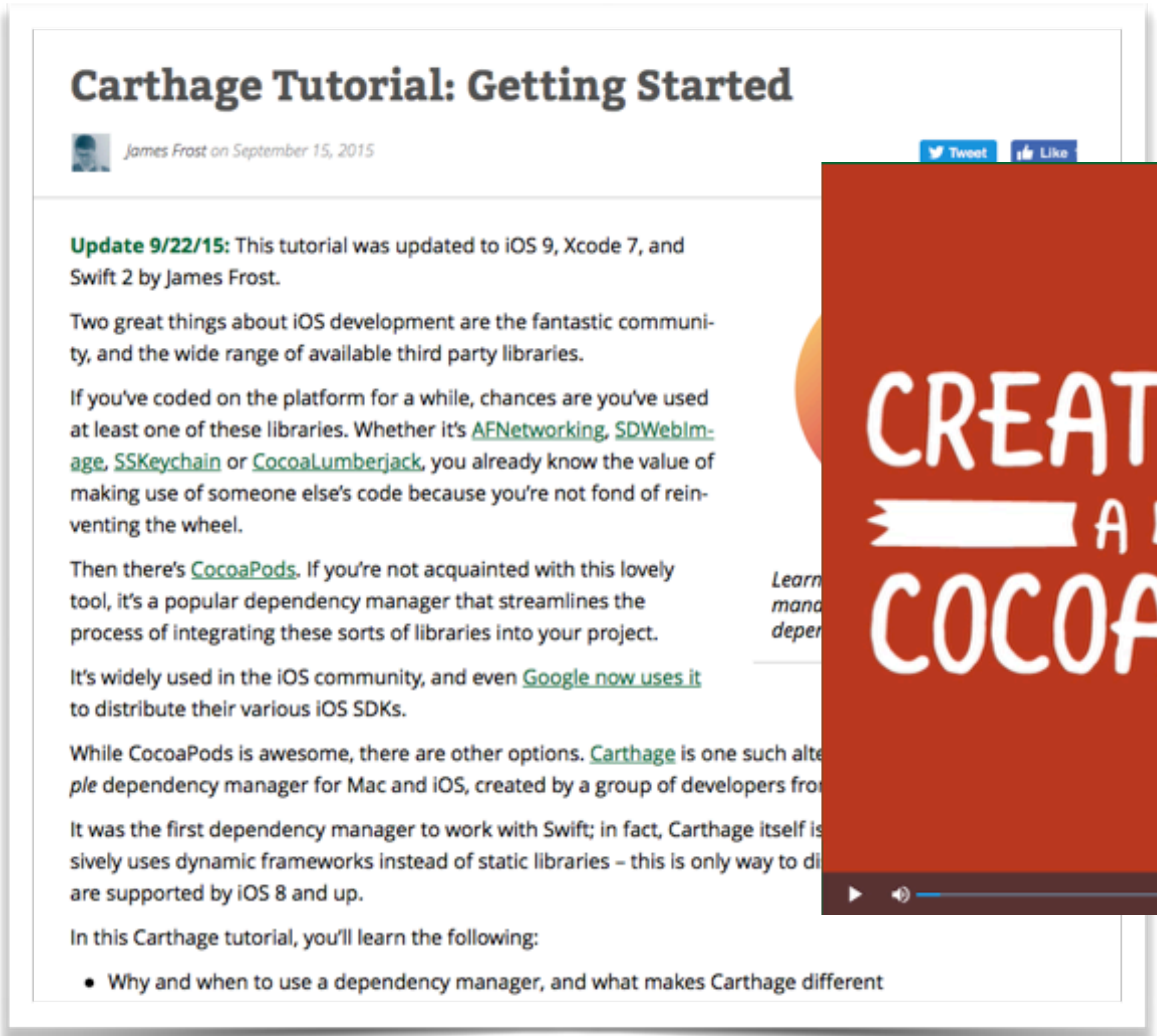

What is Core Graphics?

Apple's Drawing API

- Vector
  - Paths
  - Transformations
  - Gradients, shadows
- Bitmap
  - Image creation
  - Image processing

Beginning
Core Graphics

Where to go from here?



Carthage Tutorial: Getting Started

James Frost on September 15, 2015

Update 9/22/15: This tutorial was updated to iOS 9, Xcode 7, and
Swift 2 by James Frost.

Two great things about iOS development are the fantastic communi-
ty, and the wide range of available third party libraries.

If you've coded on the platform for a while, chances are you've used
at least one of these libraries. Whether it's AFNetworking, SDWebIm-
age, SSKeychain or CocoaLumberjack, you already know the value of
making use of someone else's code because you're not fond of rein-
venting the wheel.

Then there's CocoaPods. If you're not acquainted with this lovely
tool, it's a popular dependency manager that streamlines the
process of integrating these sorts of libraries into your project.

It's widely used in the iOS community, and even Google now uses it
to distribute their various iOS SDKs.

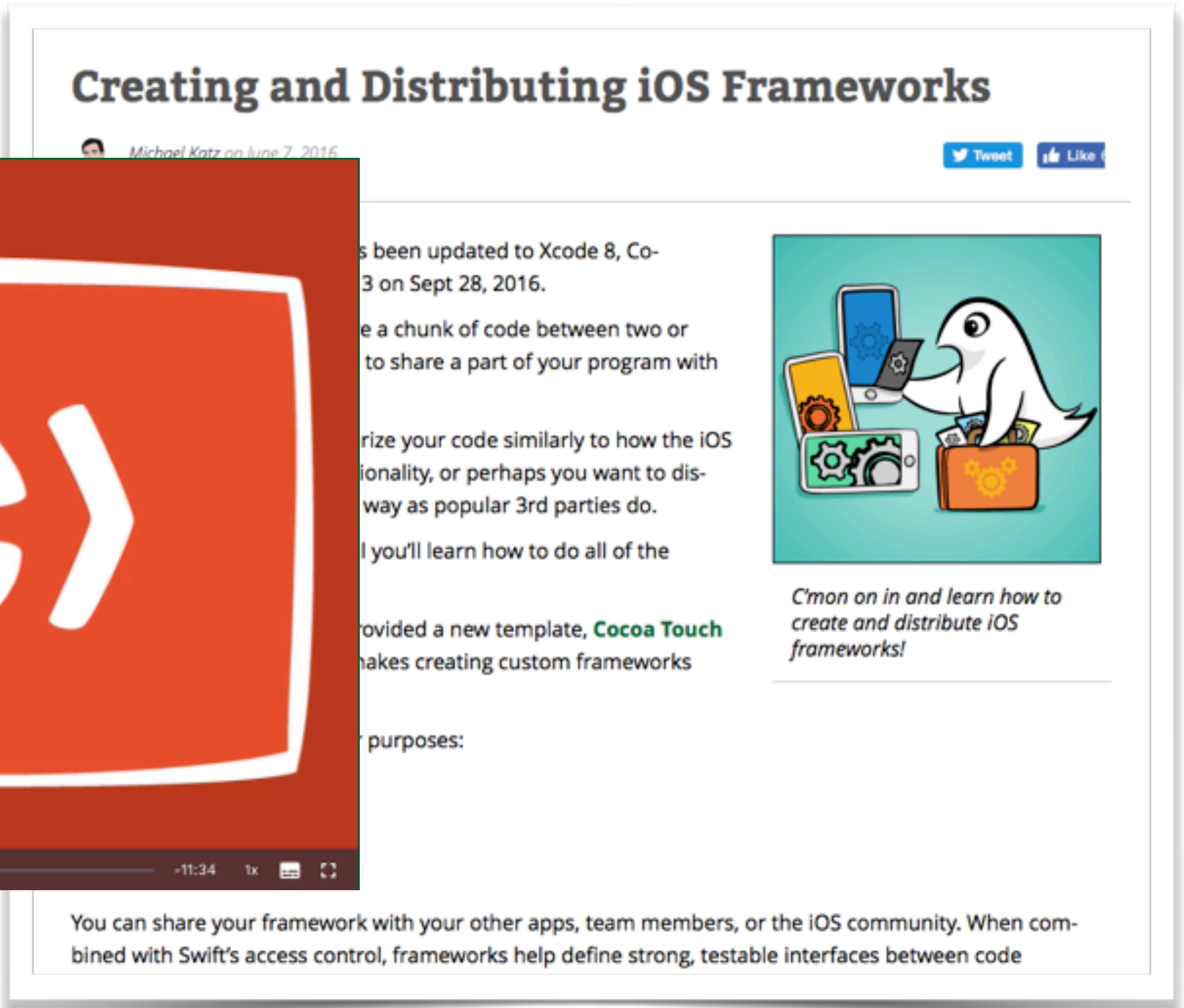
While CocoaPods is awesome, there are other options. Carthage is one such alte
ple dependency manager for Mac and iOS, created by a group of developers fro

It was the first dependency manager to work with Swift; in fact, Carthage itself is
sively uses dynamic frameworks instead of static libraries – this is only way to di
are supported by iOS 8 and up.

In this Carthage tutorial, you'll learn the following:

- Why and when to use a dependency manager, and what makes Carthage different

Learn
mana
deper



Creating and Distributing iOS Frameworks

Michael Katz on June 7, 2016

s been updated to Xcode 8, Co-
3 on Sept 28, 2016.

e a chunk of code between two or
to share a part of your program with

rize your code similarly to how the iOS
ionality, or perhaps you want to dis-
way as popular 3rd parties do.

l you'll learn how to do all of the

rovided a new template, **Cocoa Touch**
nakes creating custom frameworks

purposes:

C'mon on in and learn how to create and distribute iOS frameworks!

You can share your framework with your other apps, team members, or the iOS community. When com-
bined with Swift's access control, frameworks help define strong, testable interfaces between code