

Custom iOS 7 UIViewController Transitions

Somewhat explained by
Joachim Bondo (@osteslag)
CocoaHeads, Copenhagen, 10 April 2014

Joachim... Who?

- of Deep Green, Quibbler fame.
- Working at Black Pixel.

What Up?

- Not yet another tutorial.
- First a little bit basics, though, 5-7 minutes (presentation).
- The a little bit something you *won't* see on the Internet (code).

Custom Transition Animations, What Are They?

A framework-supported way of customizing the animated transition from one view controller to the other

- Keyword is *framework-supported*, could be done before, but basically only with the `animated:YES` parameter.
- We now have a wealth of new API, mainly protocols.

Which Transitions Can Be Customized, Then?

- Navigation Controller Pushes and Pops
- Tab Bar Controller Selection Changes (a first!)
- Modal Presentations and Dismissals
(`.modalPresentationStyle ==`
`UIModalPresentationCustom` or
`UIModalPresentationFullScreen`)
- Interactive vs. non-interactive.

Sounds Easy, Let's Go...

All it takes, really, is... *Animation Controllers, Interaction Controllers, Transitioning Delegates, Transitioning Contexts, Transition Coordinators.*

...glued together by lovely, LIX-heavy protocols.

Animation Controllers

Animation Controllers perform the animation.

They conform to the
`UIViewControllerAnimatedTransitioning`
protocol.

Interaction Controllers

Interaction Controllers control the animation by conforming to the `UIViewControllerInteractiveTransitioning` protocol.

Anything can drive the interaction, typically gestures though.

Transitioning Delegates

Transitioning Delegates vend the correct animation and interaction controllers depending on the kind of transition to be performed. Depending on the parent view controller class they conform to `UIViewControllerTransitioningDelegate`, `UINavigationControllerDelegate` or `UITabBarControllerDelegate`.

Transitioning Contexts

Transitioning Contexts define properties of the view controller views participating in the transition – *and are provided by the system.*

They are so incredibly kind to conform to the `UIViewControllerContextTransitioning` protocol and are passed on to the animation controllers and interaction controllers.

Transition Coordinators

Transition Coordinators run code blocks when transitions finish or get cancelled so that view controllers can clean up and restore state. They conform to the `UIViewControllerTransitionCoordinator` protocol.

FINAL NOTE: Custom *Transitions*, or Custom *Containment*?

- This all pertains to view controller parent/child hierarchies (often *containment*).
- But does *not* support your own containment view controllers!

Demo

The Code.