

# A new realm of interaction



Gabriela Zagarova

[gabriela.zagarova@mentormate.com](mailto:gabriela.zagarova@mentormate.com)

[LinkedIn](#)

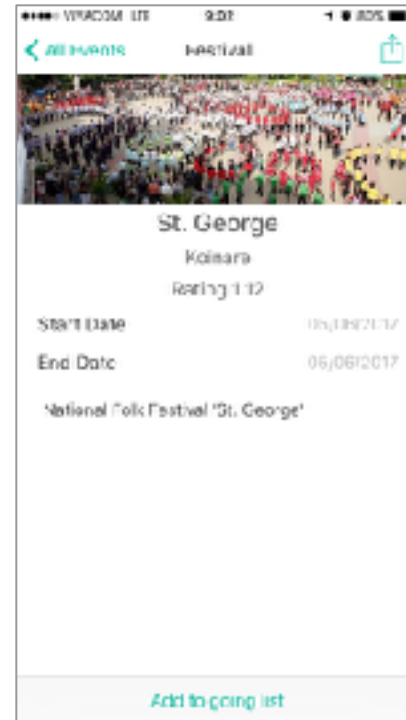
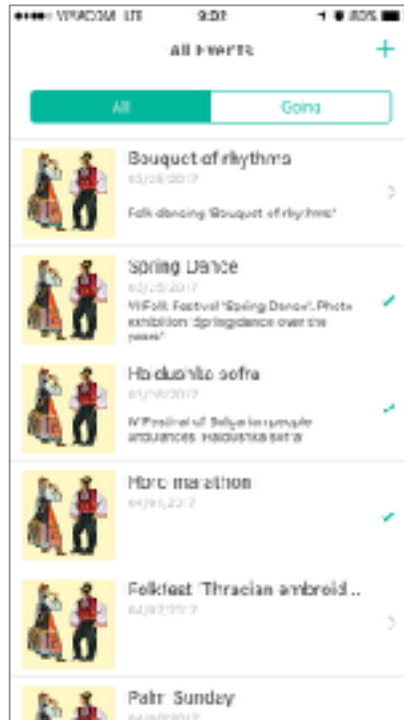
Siyana Slavova

[siyana.slavova@mentormate.com](mailto:siyana.slavova@mentormate.com)

[LinkedIn](#)



# Our app idea - Festivals 💡



<https://github.com/MentorMate/bgfestivals-copy>

# What will we learn today?

What is 3D touch

How to adopt Home Screen Quick Actions and Peek and Pop

Suggestions by Apple

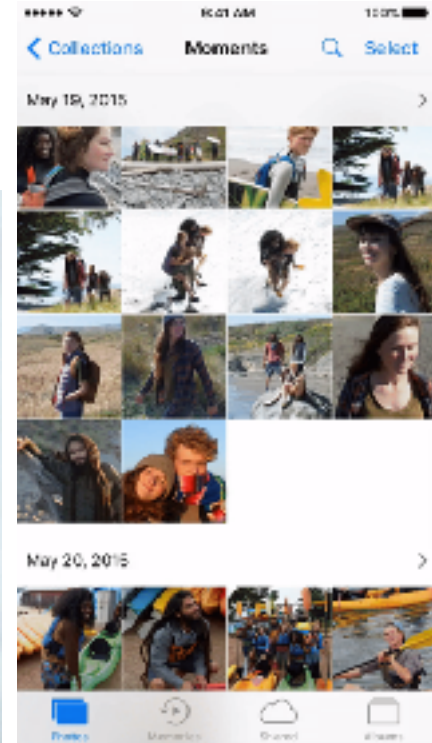
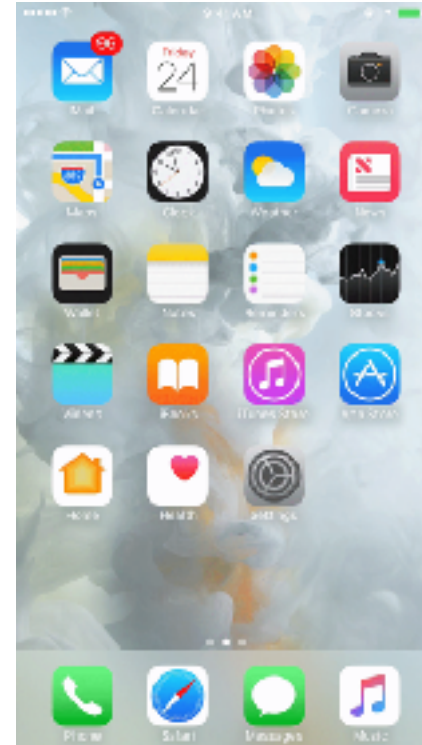
# Overview

Since iPhone 6s & 6s Plus, iOS 9 +

Senses how deeply user press the display

Haptic feedback

Quick and simple actions



# A new realm of interaction in iOS world

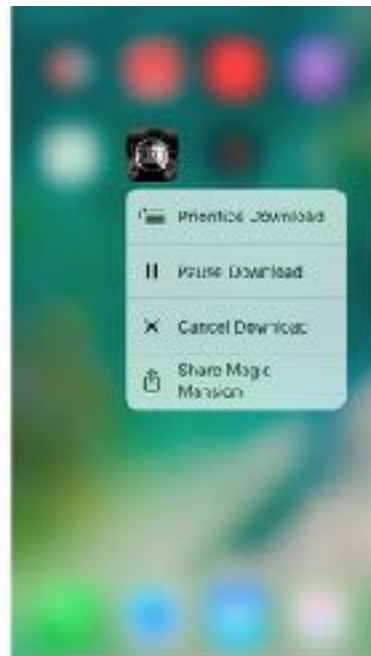
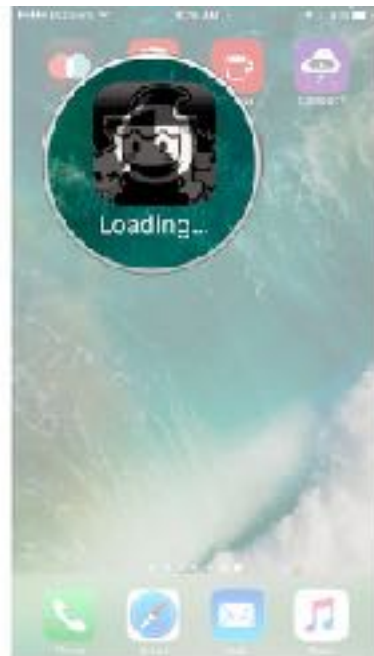


<http://giphy.com/search/download>

# Cursor control on keyboard



# Downloads



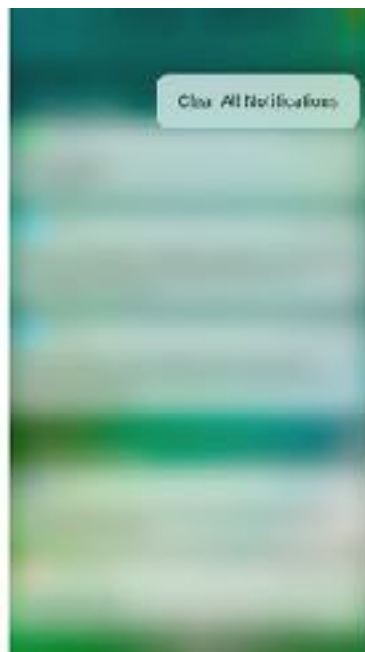
<http://www.imore.com/3d-touch>

# App switcher



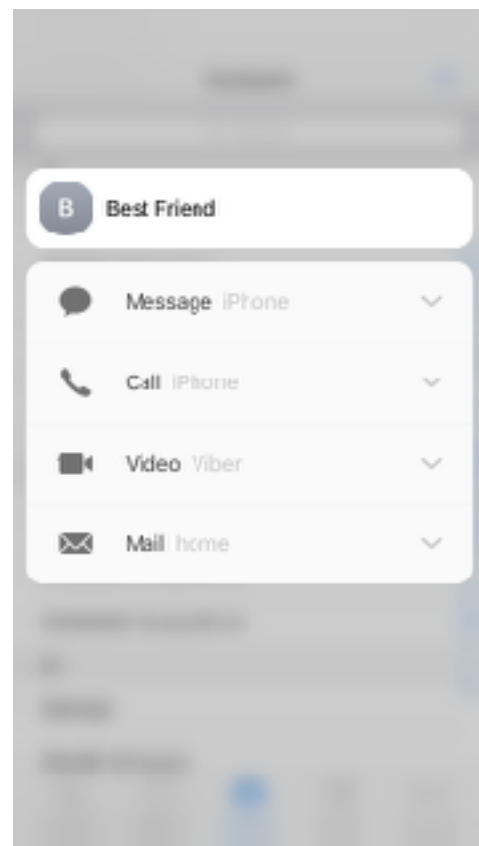
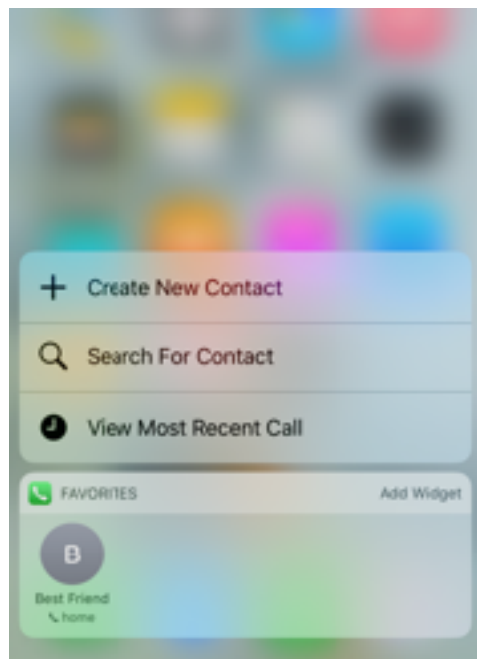


# Notification Center



<http://www.imore.com/3d-touch>

# Phone app

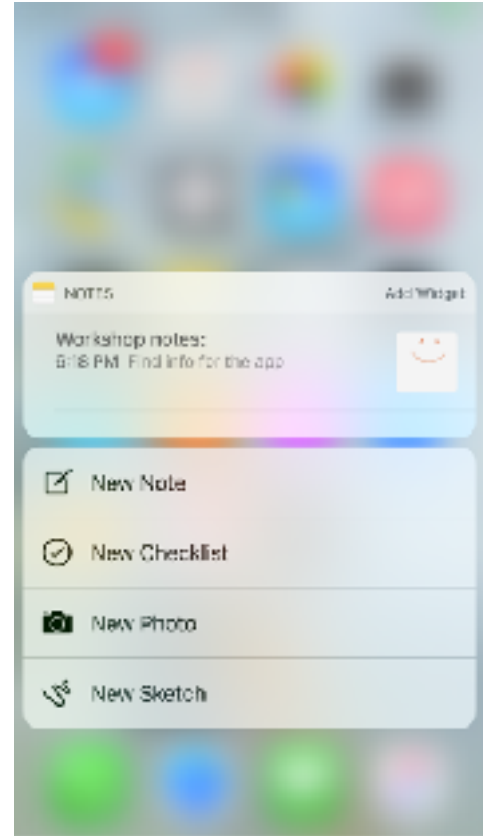


# Home Screen Quick Actions

Widget

Static and Dynamic - 4 slots

“Share <app name>” button



# Home Screen Quick Actions

Static

Specified at build time

Defined in Info.plist

Available after app install

▼ Information Property List	Dictionary	(16 items)
▼ UIApplicationShortcutItems	Array	(1 item)
▼ Item 0	Dictionary	(3 items)
UIApplicationShortcutItemIconType	String	UIApplicationShortcutIconTypeAdd
UIApplicationShortcutItemType	String	com.bgfestivals.createevent
UIApplicationShortcutItemTitle	String	Create Event

# Home Screen Quick Actions

## Static

Required keys:

```
UIApplicationShortcutItemType  
UIApplicationShortcutItemTitle
```

Optional keys:

```
UIApplicationShortcutItemSubtitle  
UIApplicationShortcutItemIconType  
UIApplicationShortcutItemIconFile  
UIApplicationShortcutItemUserInfo
```



# Home Screen Quick Actions

Dynamic

Created at runtime

Available after the first launch

Shown after static items

Can include icon



# Dynamic Home Screen Quick Actions

```
// MARK: Create
```

```
UIApplicationShortcutItem(type: String,  
                           localizedTitle: String)
```

```
UIApplicationShortcutItem(type: String,  
                           localizedTitle: String,  
                           localizedSubtitle: String?,  
                           icon: UIApplicationShortcutIcon?,  
                           userInfo: [AnyHashable : Any]? = nil))
```

# Dynamic Home Screen Quick Actions

```
// MARK: Register
```

```
application(_ application: UIApplication,  
             didFinishLaunchingWithOptions  
             launchOptions: [UIApplicationLaunchOptionsKey : Any]? = nil) -> Bool
```

```
application.shortcutItems = [item1,item2]
```



# Home Screen Quick Actions

*// MARK: Handling the case when the app becomes active*

```
func application(_ application: UIApplication,
    performActionFor shortcutItem: UIApplicationShortcutItem,
    completionHandler: @escaping (Bool) -> Void) {

    var didHandle = false
    if (shortcutItem.type == "YourType") {
        //handle here
        didHandle = true
    }
    completionHandler(didHandle)
}
```

# Home Screen Quick Actions

*// MARK: Handling the case when the app is launched*

```
func application(_ application: UIApplication,
                 didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey:
                 Any]?) -> Bool {

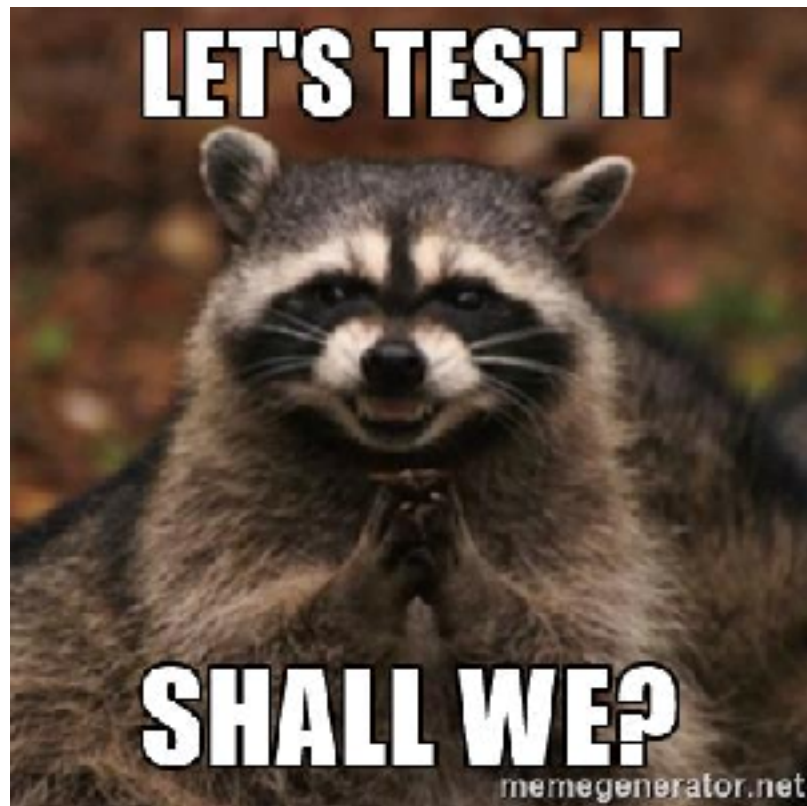
    var performAdditionalHandling = true

    if let shortcutItem = launchOptions?[UIApplicationLaunchOptionsKey.shortcutItem]
    as? UIApplicationShortcutItem {
        //handle here
        performAdditionalHandling = false
    }

    return performAdditionalHandling
}
```

# Demo & Questions

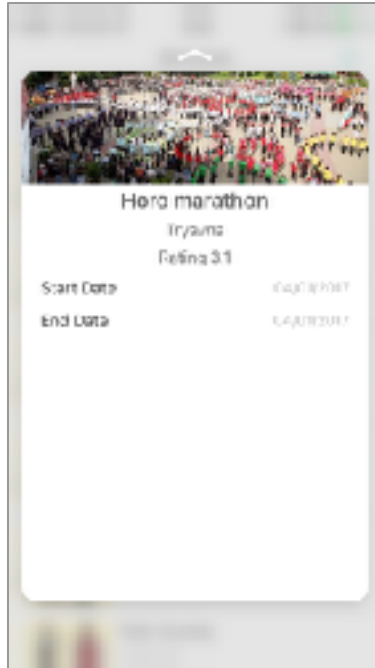




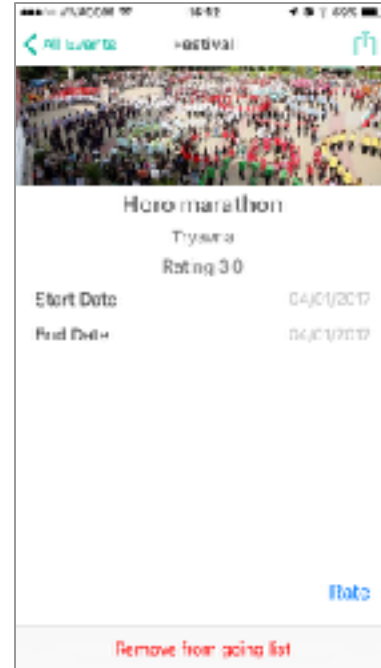
<https://memegenerator.net>

# Peek and Pop

## States



Preview

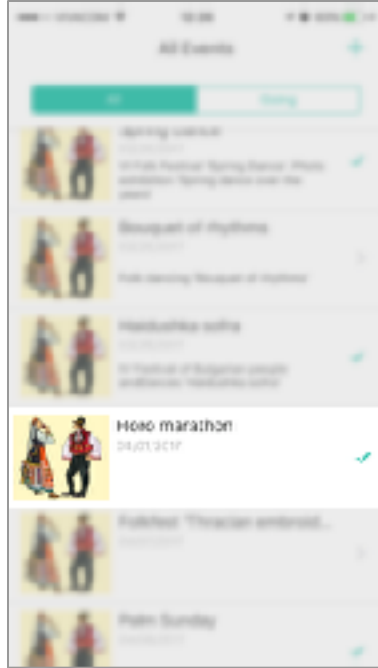


Commit

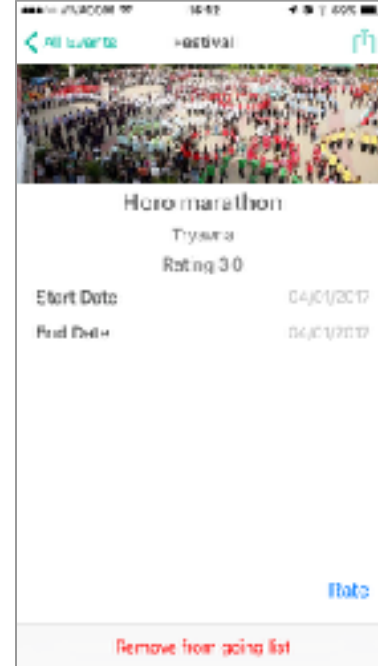
# Peek and Pop

Components

Source view →



← Source rect



Registered View Controller

Previewed View Controller

# Peek and Pop

*// MARK: Register*

**UIViewControllerPreviewingDelegate**

```
registerForPreviewing(with delegate: UIViewControllerPreviewingDelegate,  
                        sourceView: UIView) -> UIViewControllerPreviewing
```

# Peek and Pop

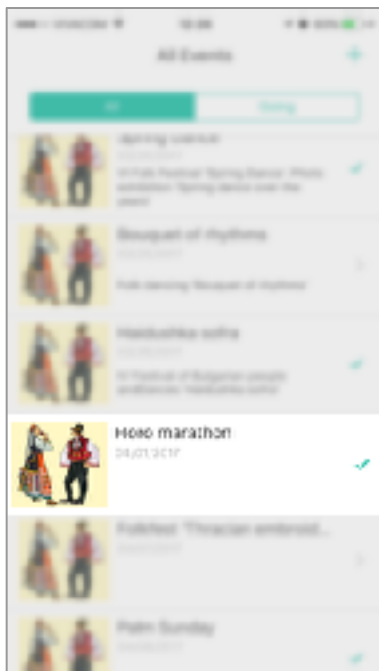
```
// MARK: UIViewControllerPreviewingDelegate  
methods
```

```
previewingContext(_ previewingContext: UIViewControllerPreviewing,  
                   viewControllerForLocation location: CGPoint) ->  
UIViewController? {  
    var controller = nil  
  
    if ableToPreview {  
        controllerToPreview.item = itemToPreview  
        controller = controllerToPreview  
        previewingContext.sourceRect = sourceRect(fromLocation: location)  
    }  
  
    return controller  
}
```

```
previewingContext(_ previewingContext: UIViewControllerPreviewing,  
                   commit viewControllerToCommit: UIViewController)
```

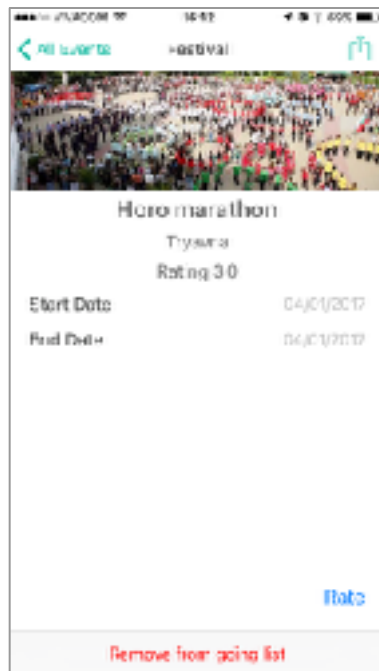


# Interface Builder Support for Peek and Pop

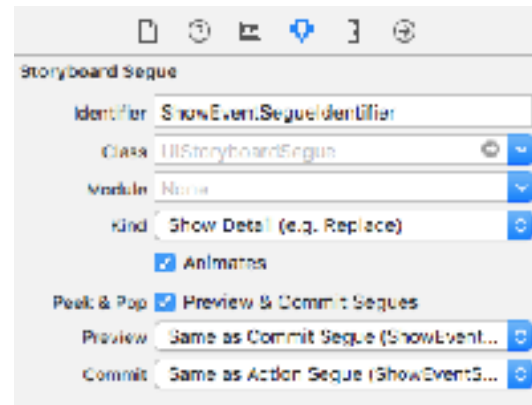


Registered

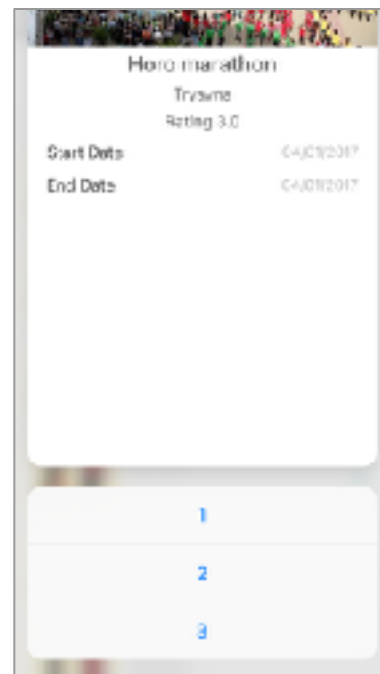
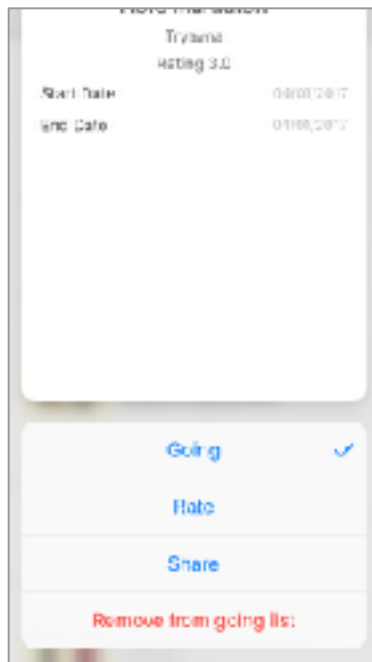
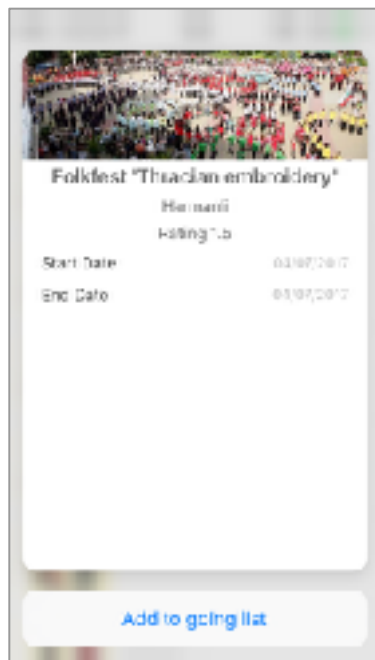
Segue →



Previewed



# Preview Quick Actions



# Preview Quick Actions

*// MARK: The quick actions displayed. Implement in previewed controller*

```
UIPreviewAction(title: String,  
                style: UIPreviewActionStyle,  
                handler: (UIPreviewAction, UIViewController) -> Void
```

*// MARK: Groups*

```
UIPreviewActionGroup(title: String,  
                     style: UIPreviewActionStyle,  
                     actions: [UIPreviewAction])
```

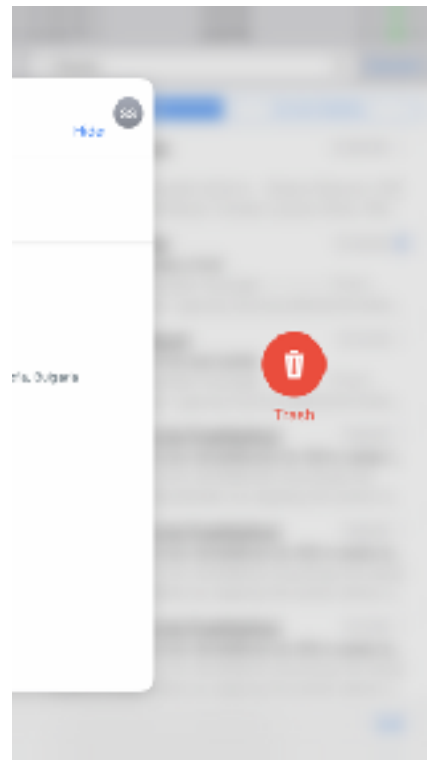
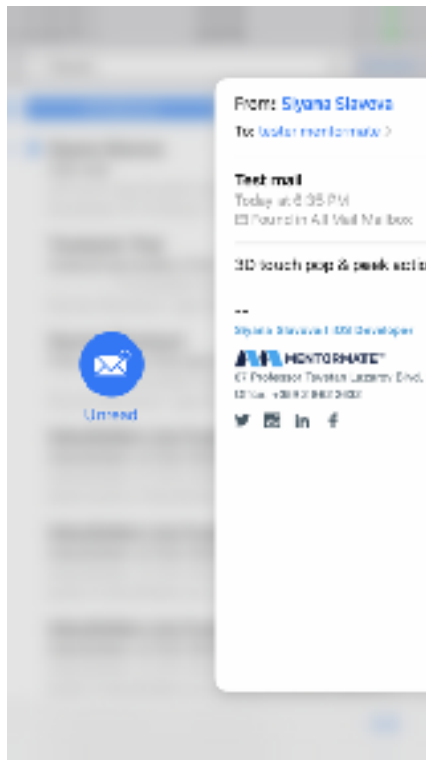
# Preview Quick Actions

*// MARK: Add actions to the Previewed View Controller*

```
override var previewActionItems: [UIPreviewActionItem] {  
    return [item1,item2]  
}
```

# Preview Quick Actions

Private API 🙄



# Demo & Questions



VIA 9GAG.COM

Chuck Norris doesn't ask questions.  
He knows all the answers.

# Preview Interaction

```
@available(iOS 10.0, *)
```

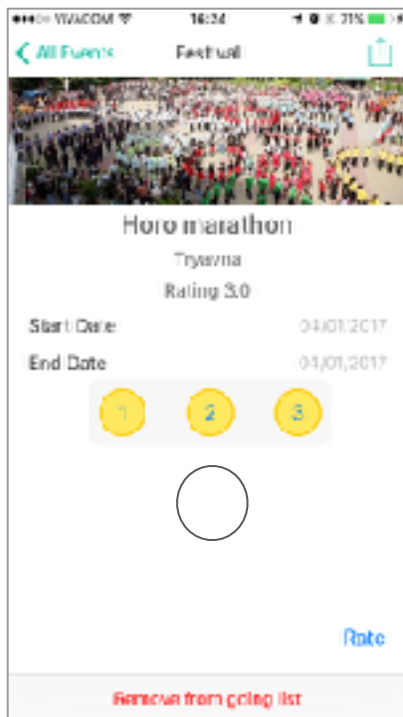
Same Peek and Pop  
Force Processing

Automatic Haptic  
Feedback

Your User Interface

# Preview Interaction

@available(iOS 10.0, \*)





# Preview Interaction

*// MARK: Create preview interaction*

**UIPreviewInteractionDelegate**

```
previewInteraction = UIPreviewInteraction(view: view)  
previewInteraction.delegate = self
```



# Preview Interaction

Interact

```
previewInteractionShouldBegin(_ previewInteraction: UIPreviewInteraction) ->  
Bool
```

Preview

```
previewInteraction(_ previewInteraction: UIPreviewInteraction,  
    didUpdatePreviewTransition transitionProgress: CGFloat,  
    ended: Bool)
```

Commit

```
previewInteraction(_ previewInteraction: UIPreviewInteraction,  
    didUpdateCommitTransition transitionProgress: CGFloat,  
    ended: Bool)
```

```
previewInteractionDidCancel(_ previewInteraction: UIPreviewInteraction)
```

# Demo & Questions



# Apple Suggestions

## Home Screen Quick Actions

Provide quick access to high-value tasks

Make quick actions predictable

Handle dynamic quick actions from a previous version of your app

Don't add functionality that is only accessible using quick actions

Provide an icon when possible

# Apple Suggestions

Peek and Pop

Tappable content should support Peek and Pop

Be consistent

Don't block the main queue

Set the previewing context **sourceRect**

Provide action buttons when appropriate

Each actions should be accessible without 3D touch

**“There is beauty when  
something works and it  
works intuitively.”**

Jonathan Ive  
Apple Chief Design Officer

# Resources

1. <http://www.macworld.co.uk/how-to/iphone/15-great-3d-touch-tips-tricks-for-iphone-6s-6s-plus-ios-10-3626954/>
2. <http://www.imore.com/3d-touch>
3. <https://krakendev.io/uipreviewinteraction/>
4. <https://developer.apple.com/>
5. <https://developer.apple.com/videos/play/wwdc2016/228/?time=1616>
6. <https://developer.apple.com/library/prerelease/content/documentation/UserExperience/Conceptual/Adopting3DTouchOniPhone/index.html>
7. <https://developer.apple.com/ios/human-interface-guidelines/extensions/home-screen-actions/>
8. <https://developer.apple.com/ios/3d-touch/>