



孟祥月的博客

昨晚多几分钟的准备，今天少几小时的麻烦。



iOS9-by-Tutorials-学习笔记九：3D Touch

By 孟祥月

🕒 发表于 2016-02-23

声明：本文章版权归作者所有，转载请注明出处。

孟祥月的博客：<http://www.mengxiangyue.com>
<http://blog.csdn.net/mengxiangyue>

文章目录

1. Getting Started
2. UITouch force
3. Peeking and popping
 - 3.1. Preview actions
4. Home screen quick actions
 - 4.1. Adding a static shortcut
 - 4.2. Adding a dynamic shortcut

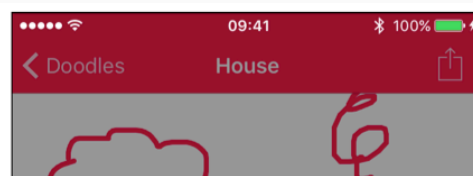
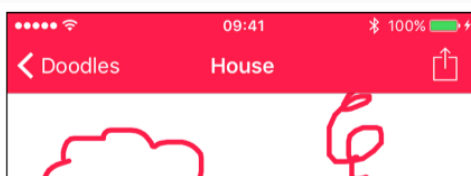
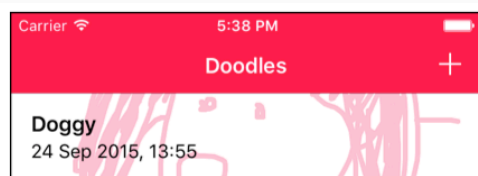
很久没有更新文章了，自己也感觉自己懒得要发霉了。为了让自己不至于发霉，今天开始继续更新文章。

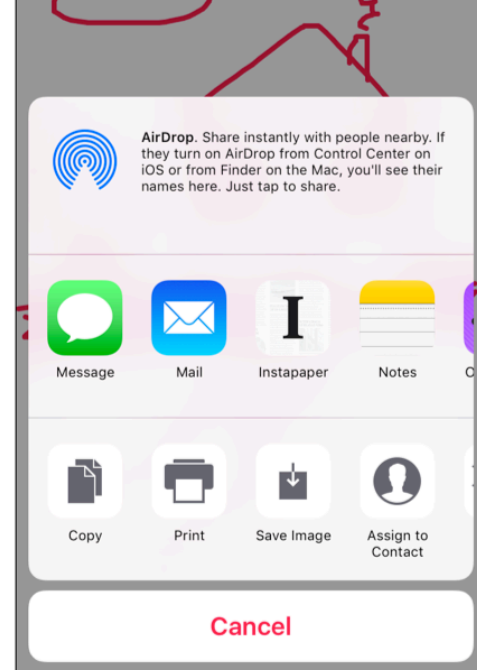
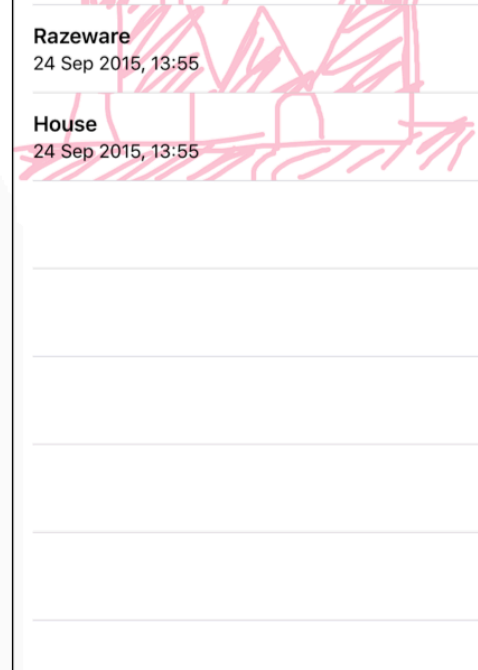
本篇文章介绍一下3D Touch。3D Touch 是iPhone 6s推出的一项新的技术，利用该技术能够更快的预览内容，更加平滑的进行多任务等。Apple提供一下与3D Touch 相关的API：

- UITouch有一个 **force** 属性用来检测用户按下的程度。
- UIViewController提供了一个新的API，允许你展示新的view Controller的预览，这个预览成为 **peek**。此时继续用力，将会 **pop** 该预览，进入对应的View Controller 界面。
- UIApplicationShortcutItem 是一个新的类，你能使用它给home屏幕上的icon添加快捷操作。

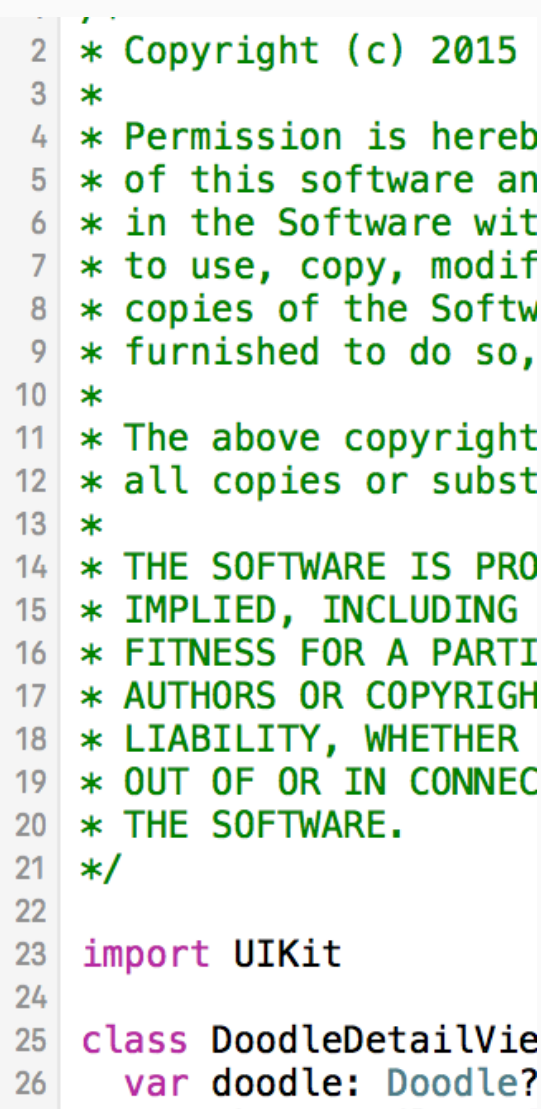
Getting Started

在这篇文章中将会使用一个叫做 **Doodles(涂鸦)** App, 这个APP 比较简单，就两个界面：列表+详情，截图如下：





代码结构如下：



UITouch force

iOS9后，UITouch 添加了一个force属性，这个属性表示的按压屏幕的力量，取值范围为0.0到1.0。这个值不是1，我们可以

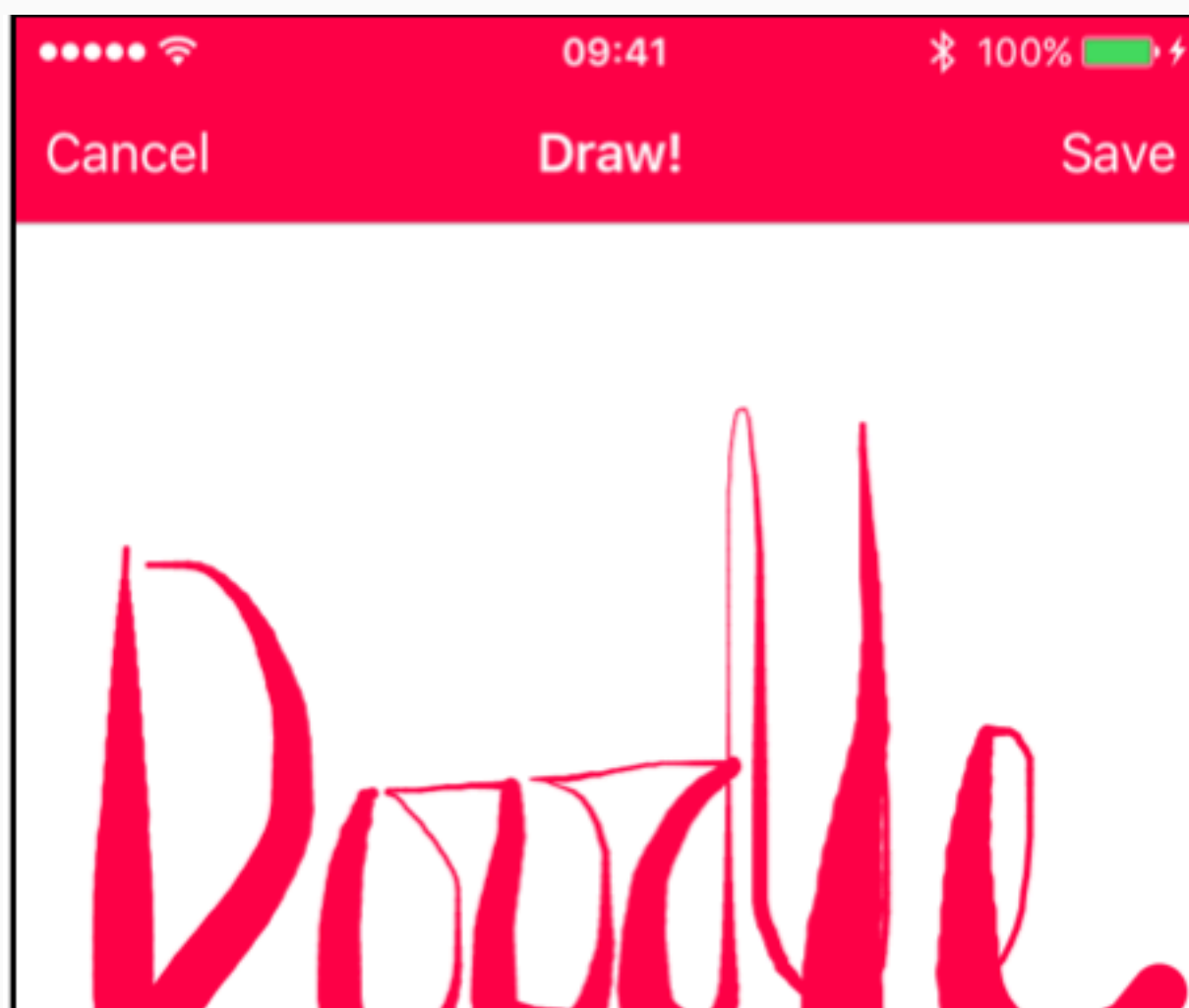
为0~maximumPossibleForce(touch.maximumPossibleForce,这个值不是1)。我们可以利用force属性去改造我们的APP，让线条的粗细随着按压屏幕力量的变化而变化。修改Canvas.swift中方法如下（下面标记update的为修改的）：

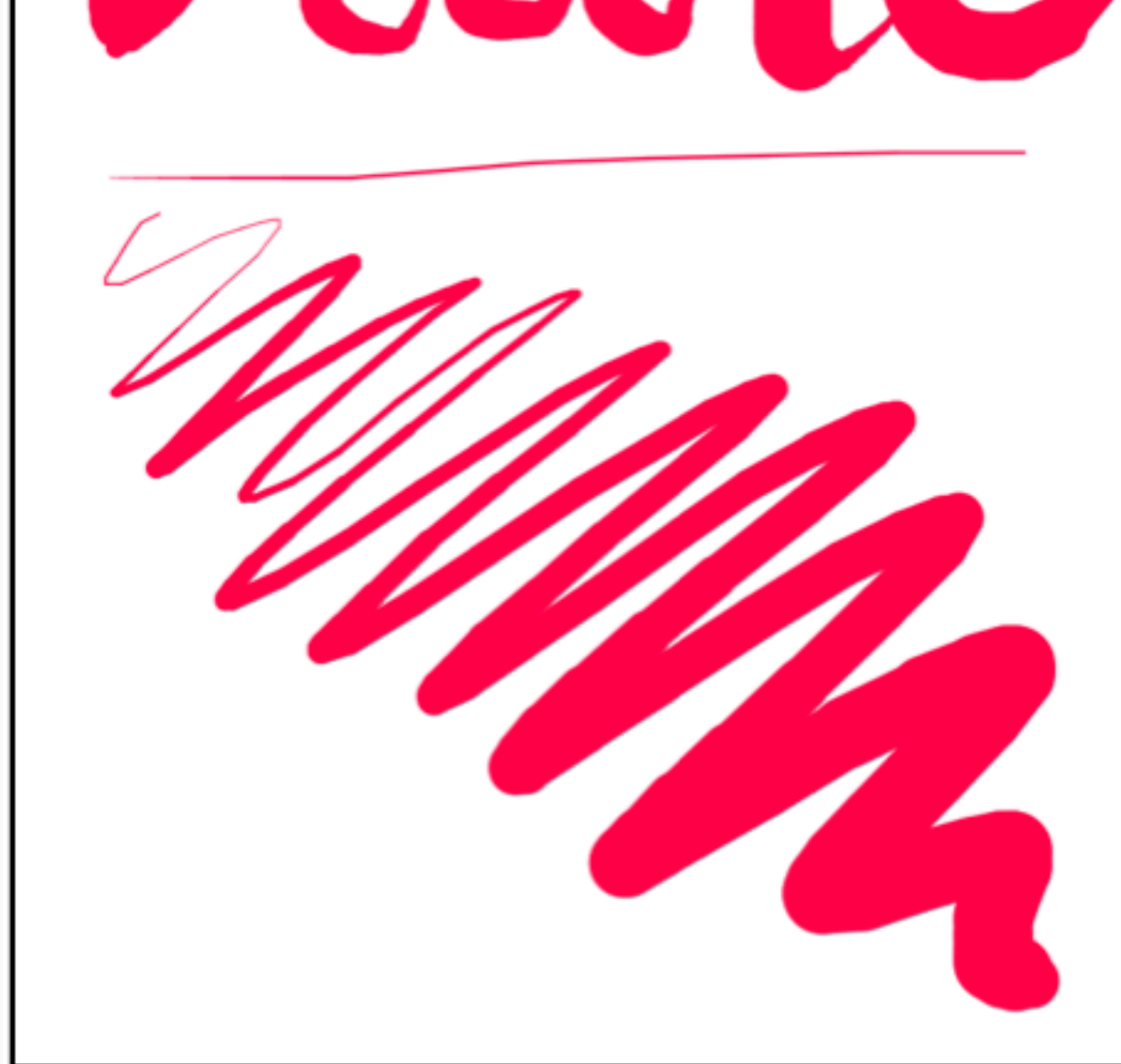
```
1 private func addLineFromPoint(from: CGPoint, toPoint: CGPoint, withForce force: CGFloat = 1) { // update
2     UIGraphicsBeginImageContextWithOptions(bounds.size, false, 0.0)
3     print("force:\(force)")
4     drawing?.drawInRect(bounds)
5
6     let cxt = UIGraphicsGetCurrentContext()
7     CGContextMoveToPoint(cxt, from.x, from.y)
8     CGContextAddLineToPoint(cxt, toPoint.x, toPoint.y)
9
10    CGContextSetLineCap(cxt, .Round)
11
12    CGContextSetLineWidth(cxt, strokeWidth * force) // update
13
14    strokeColor.setStroke()
15
16    CGContextStrokePath(cxt)
17
18    drawing = UIGraphicsGetImageFromCurrentImageContext()
19
20    layer.contents = drawing?.CGImage
21
22    UIGraphicsEndImageContext()
23 }
```

继续修改如下方法：

```
1 public override func touchesMoved(touches: Set<UITouch>, withEvent event: UIEvent?) {
2     if let touch = touches.first {
3         // 先去验证是否支持3D Touch
4         if traitCollection.forceTouchCapability == .Available {
5             addLineFromPoint(touch.previousLocationInView(self), toPoint: touch.locationInView(self), with
6         } else {
7             addLineFromPoint(touch.previousLocationInView(self), toPoint: touch.locationInView(self))
8         }
9     }
10 }
```

修改完后运行一下，用不同压力绘画，效果如下：





Peeking and popping

对于 **Peek** 和 **Pop** 等写完下面的例子自然就明白了。

修改DoodlesViewController.swift对应方法如下：

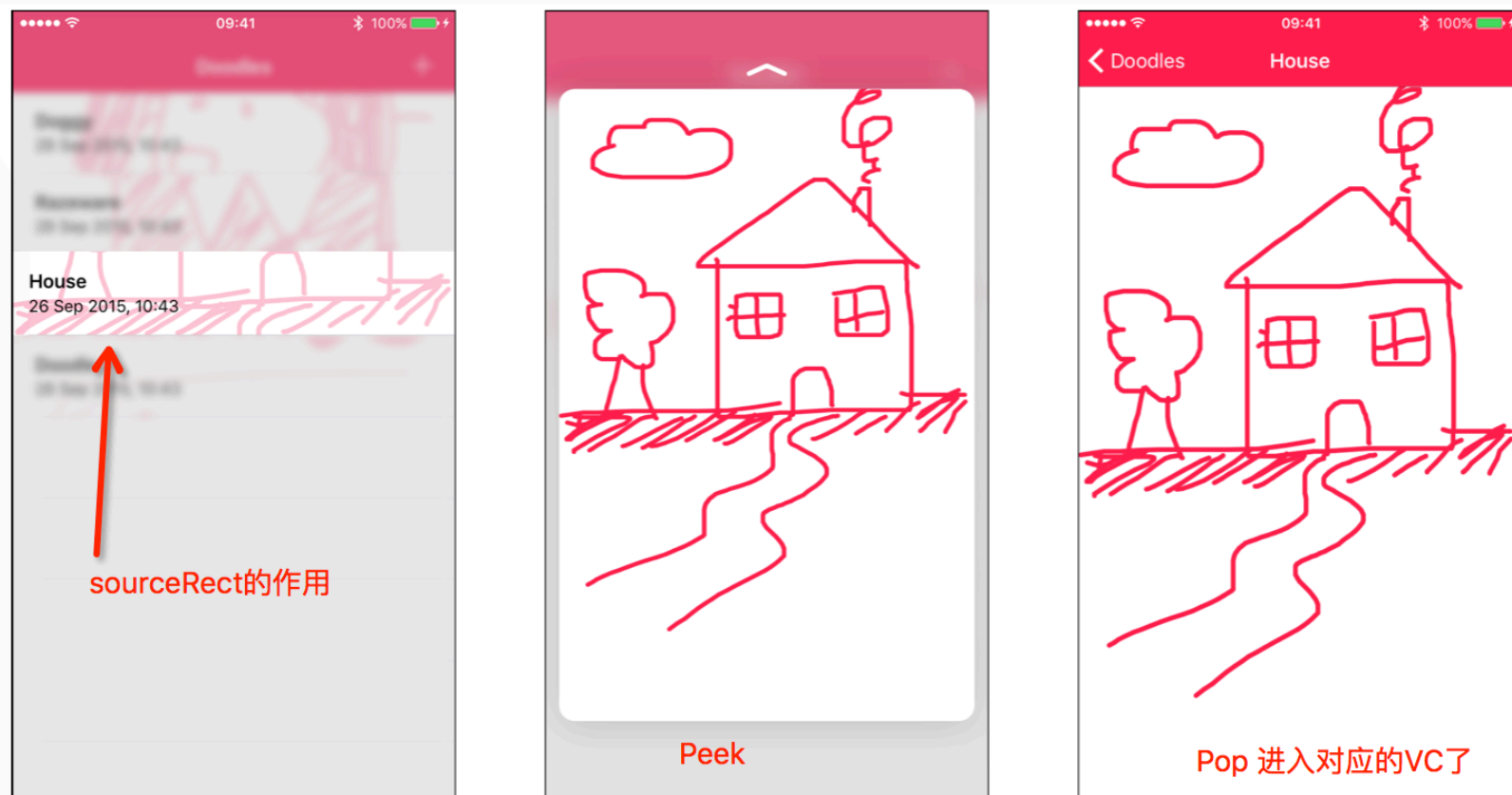
```
1  override func viewDidLoad() {
2      super.viewDidLoad()
3      // update
4      if traitCollection.forceTouchCapability == .Available {
5          // 添加对于3D Touch的代理
6          registerForPreviewingWithDelegate(self, sourceView: view)
7      }
8  }
9
10 .....
11
12 // 扩展该类实现对应的协议 有两个方法用于处理peek pop
13 //MARK: - 3D Touch
14 extension DoodlesViewController: UIViewControllerPreviewingDelegate {
15     // peek 回调 返回一个包含对应内容的UIViewController
16     func previewingContext(previewingContext: UIViewControllerPreviewing, viewControllerForLocation location: CGPoint) {
17         guard let indexPath = tableView.indexPathForRowAtPoint(location), cell = tableView.cellForRowAtIndexPath(indexPath) else {
18             return nil
19         }
20         let identifier = "DoodleDetailViewController"
21         guard let detailVC = storyboard?.instantiateViewControllerWithIdentifier(identifier) as? DoodleDetailViewController else {
22             return nil
23         }
24         detailVC.doodle = cell.doodle // 设置数据
25
26         // 显示那部分区域响应的3D Touch 在peek之前, 该frame之外的内容将会被模糊
27         previewingContext.sourceRect = cell.frame //CGRect(x: 0, y: 100, width: 300, height: 40)// cell.frame
28         return detailVC
29     }
30
31     // pop commitViewController是上面方法返回的
32     func previewingContext(previewingContext: UIViewControllerPreviewing, commitViewController viewControllerToCommit: UIViewController) {
```

```

33 // peek之后继续用力 将会调用该方法 进入对应的view Controller
34 showViewController(viewControllerToCommit, sender: self)
35 }
36 }

```

运行效果如下：



Preview actions

在 **Peek** 状态下，我们可以添加一些快捷操作，通过上划就能显示出来这些快捷操作。这些快捷操作是需要添加到 **Peek** 显示的View Controller中的。

在DoodleDetailViewController.swift 末尾添加如下代码：

```

1  weak var doodlesViewController: DoodlesViewController?
2  override func previewActionItems() -> [UIPreviewActionItem] {
3      // UIPreviewAction 专门为3D Touch使用
4      let shareAction = UIPreviewAction(title: "分享", style: .Default) { (previewAction, viewController) ->
5          if let doodlesVC = self.doodlesViewController, activityViewController = self.activityViewController {
6              doodlesVC.presentViewController(activityViewController, animated: true, completion: nil)
7          }
8      }
9
10     let deleteAction = UIPreviewAction(title: "删除", style: .Default) { (previewAction, viewController) ->
11         guard let doodle = self.doodle else {
12             return
13         }
14         Doodle.deleteDoodle(doodle)
15         if let doodlesViewController = self.doodlesViewController {
16             doodlesViewController.tableView.reloadData()
17         }
18     }
19     return [shareAction, deleteAction]
20 }

```

上面的代码添加了两个快捷键，一个是分享一个是删除。

在DoodlesViewController.swift中也需要修改一下代码：

```

1 // 注释new add 是新添加的

```

```

2 func previewingContext(previewingContext: UIViewControllerPreviewing, viewControllerForLocation location:
3     guard let indexPath = tableView.indexPathForRowAtPoint(location), cell = tableView.cellForRowAtIndexPa
4         return nil
5     }
6     let identifier = "DoodleDetailViewController"
7     guard let detailVC = storyboard?.instantiateViewControllerWithIdentifier(identifier) as? DoodleDetailV
8         return nil
9     }
10    detailVC.doodle = cell.doodle // 设置数据
11
12    detailVC.doodlesViewController = self // new add
13
14    // 显示那部分区域响应的3D Touch 在peek之前，该frame之外的内容将会被模糊
15    previewingContext.sourceRect = cell.frame //CGRect(x: 0, y: 100, width: 300, height: 40)// cell.frame
16    return detailVC
17 }

```

运行程序，在 **Peek** 状态上划效果如下：



分享

删除

Home screen quick actions

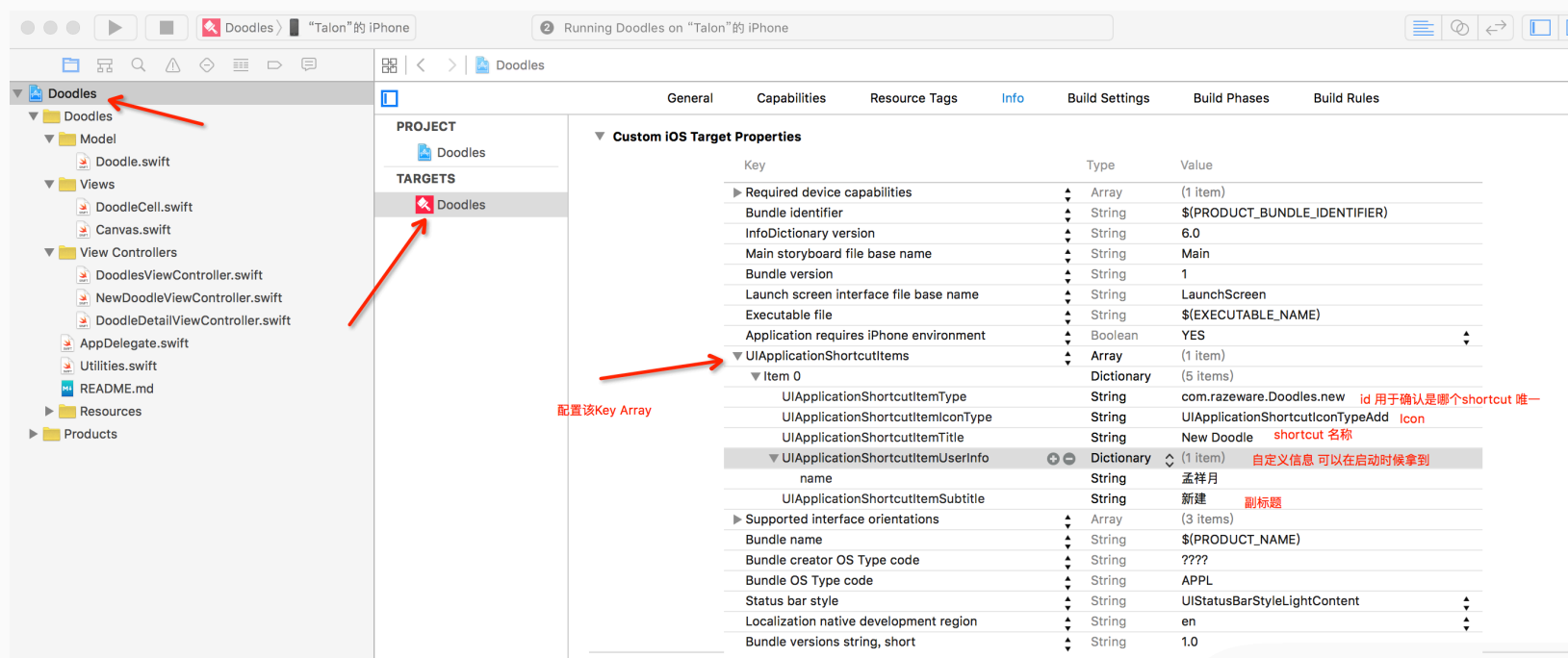
3D Touch 另一个作用就是能够在Home界面的图标上，通过按压在图标的周围显示一个快捷操作（shortcuts）的菜单，通过这个菜单，我们能够更加快速直接的进入某项功能。现在主流的App都已经支持这项功能，可以在图标上按按试试。

这种快捷操作（shortcuts）分两种：

- **Static shortcuts** 静态的，在Info.plist里面配置，在程序安装后就会存在。
- **Dynamic shortcuts** 动态的，在运行时创建的，直到程序第一次运行后才会存在。

Adding a static shortcut

添加一个静态的shortcut，首先需要在Info.plist里面配置，按照下图进行相应的配置：



配置好了之后运行程序，在Home页面App Icon上按压能够看到我们配置的shortcut。

点击后发现应用启动，但是不是我们想要的结果。我们需要在AppDelegate.swift中添加如下的代码：

```
1
2 // 响应shortcut的点击
3 func application(application: UIApplication, performActionForShortcutItem shortcutItem: UIApplicationShort
4     handleShortcutItem(shortcutItem)
5
6 // 如果你自己处理了shortcut 这里参数为true 否则为false
7 completionHandler(true)
8 }
9
10 func handleShortcutItem(shortcutItem: UIApplicationShortcutItem) {
11     print(shortcutItem.userInfo)
12     switch shortcutItem.type {
13     case "com.razeware.Doodles.new":
14         presentNewDoodleViewController()
15     default:
16         break
17     }
18 }
19
20 func presentNewDoodleViewController() {
21     let identifier = "NewDoodleNavigationController"
22     let doodleViewController = UIStoryboard.mainStoryboard .instantiateViewControllerWithIdentifier(identi
23
24     window?.rootViewController?.presentViewController(doodleViewController, animated: true, completion: n
25 }
```

运行程序，点击shortcut会进入新建View Controller。

Adding a dynamic shortcut

当然我们也能动态的通过程序添加shortcut，这个有一个限制就是程序必须运行一次才会生效。

修改Doodle.swift对应的方法如下（注释add地方为新加的）：

```
1 static func addDoodle(doodle: Doodle) {
2     allDoodles.append(doodle)
3     Doodle.configureDynamicShortcuts() // add
4 }
5
6 static func deleteDoodle(doodle: Doodle) {
7     if let index = allDoodles.indexOf({ $0.name == doodle.name }) {
8         allDoodles.removeAtIndex(index)
9         Doodle.configureDynamicShortcuts() // add
10    }
11 }
12
13 // 这个方法添加
14 static func configureDynamicShortcuts() {
15     if let mostRecentDoodle = Doodle.sortedDoodles.first {
16         let shortcutType = "com.razeware.Doodles.share"
17         let shortcutItem = UIApplicationShortcutItem(type: shortcutType, localizedTitle: "Share latest Doo
18         UIApplication.sharedApplication().shortcutItems = [shortcutItem]
19     } else {
20         UIApplication.sharedApplication().shortcutItems = []
21     }
22 }
```

修改AppDelegate.swift对应的方法：

```
1 func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyOb
2     configureAppAppearance()
3     Doodle.configureDynamicShortcuts() // add
4     return true
```



```

5  }
6
7  .....
8
9  func handleShortcutItem(shortcutItem: UIApplicationShortcutItem) {
10     print(shortcutItem.userInfo)
11     switch shortcutItem.type {
12     case "com.razeware.Doodles.new":
13         presentNewDoodleViewController()
14     case "com.razeware.Doodles.share": // 处理share
15         shareMostRecentDoodle()
16     default:
17         break
18     }
19 }
20
21 func shareMostRecentDoodle() {
22     guard let mostRecentDoodle = Doodle.sortedDoodles.first, navigationController = window?.rootViewContro
23         return
24     }
25
26     let identifier = "DoodleDetailViewController"
27     let doodleViewController = UIStoryboard.mainStoryboard.instantiateViewControllerWithIdentifier(identif
28
29     doodleViewController.doodle = mostRecentDoodle
30     doodleViewController.shareDoodle = true
31     navigationController.pushViewController(doodleViewController, animated: true)
32 }
33

```

运行程序，然后退出到Home界面，按压图标，点击share，能够进入程序对应的详情页面，调用起来分享。





3D Touch相关的内容到这里就已经更新完了，初步计划以后录制视频关于动画相关的，希望有兴趣的朋友关注一下。

 [iOS 9 by Tutoials](#)  [iOS 9](#)




[上一篇：](#)

[◀ 一场站着听完的iOS技术分享会](#)

[下一篇：](#)

[▶ iOS9 by Tutorials 学习笔记八:Storyboard新特性](#)

还没有评论，沙发等你来抢

社交帐号登录:  微信  微博  QQ  人人 [更多»](#)



Q 写点什么吧...



发布

孟祥月的博客正在使用多说

0条评论

40 条新浪微博

最新 最早 最热

还没有评论，沙发等你来抢

社交帐号登录:  微信  微博  QQ  人人 [更多»](#)



Q 写点什么吧...



发布

孟祥月的博客正在使用多说

分类

React Native ¹

iOS 9 by Tutoials ⁹

iOS Tips ²

工具 ¹

标签

iOS 9 ¹⁰

MShare ¹

React Native ¹

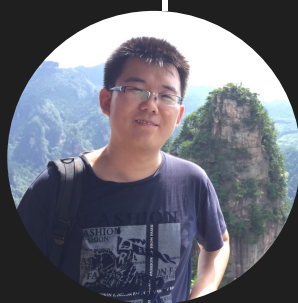
Github ¹

Hexo ¹

友情链接

My CSDN Blog

 RSS 订阅



iOS小菜鸟一枚，在努力的路上前进着

喜欢交朋友、聊与技术有关无关的，可以加我QQ、邮件联系我。QQ&mail :
598660766@qq.com

