

# Style In Motion

Step Christopher  
Big Nerd Ranch

The image shows two side-by-side screenshots of an iPhone's Mail app displaying an email thread. The left screenshot shows the inbox screen with a selected message. The right screenshot shows the full thread view.

**Inbox Screen (Left):**

- Header: 9:58 AM, 90% battery
- Subject: cocoa conf and redeye...
- Message 2 of 5: From: Mark Dalrymple > To: schristopher >
- Subject: Re: cocoa conf and redeye flight (November 15, 2013, 10:48 PM)
- Text: badge, swagbag, posse t-shirt and stickers all snagged and in the room.
- Text: On Fri, Nov 15, 2013 at 8:31 AM, Mark Dalrymple <[markd@bignerdranch.com](mailto:markd@bignerdranch.com)> wrote:  
Yep. Actually, I'll snag your badge and swagbag today.  
++md

**Thread View (Right):**

- Header: AT&T LTE, 10:07 PM, 79% battery
- From: Mark Dalrymple >
- To: schristopher >
- Subject: Re: cocoa conf and redeye flight (November 15, 2013 at 7:48 PM)
- Text: badge, swagbag, posse t-shirt and stickers all snagged and in the room.
- Text: On Fri, Nov 15, 2013 at 8:31 AM, Mark Dalrymple <[markd@bignerdranch.com](mailto:markd@bignerdranch.com)> wrote:  
Yep. Actually, I'll snag your badge and swagbag today.  
++md

# Push Me

# 12 Principles of Cartoon Animation



<http://www.centolodigiani.com/117722/3078861/work/the-illusion-of-life>

# Map it

---

principles.map { \$0.code }

# Git the codes

---

<https://github.com/bignerdranch/iOS7Demos/Collections>

# 10 Steps for App Aliveness

---



# 10 Steps for App Aliveness

---

1. Animate.

2. Spring.

3. Anticipate.

4. Continuous.

5. Interactive

6. Timing.

7. Arc.

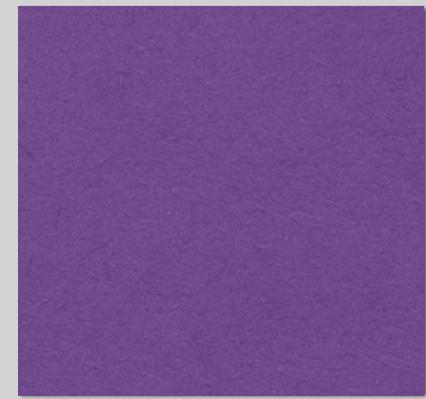
8. Solidarity.

9. Appeal.

10. Secondary Action

# Animate

---



[print version](#)

# Animate

---

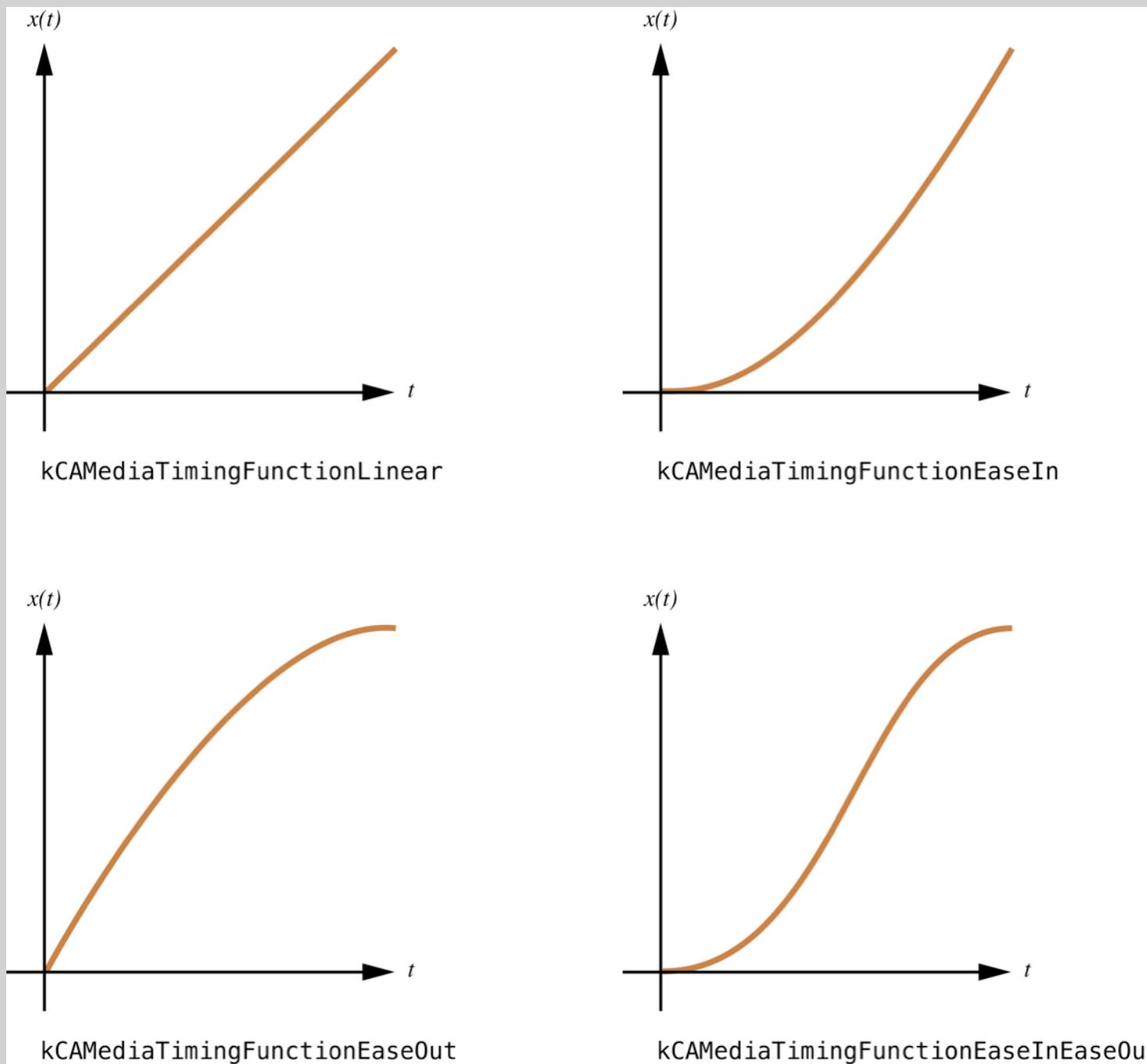
- Position
  - Frame
  - Center
  - Color
  - Opacity
- Constraints
  - Height
  - Offset
  - More!

# Spring

---



# Timing Curve



# Spring

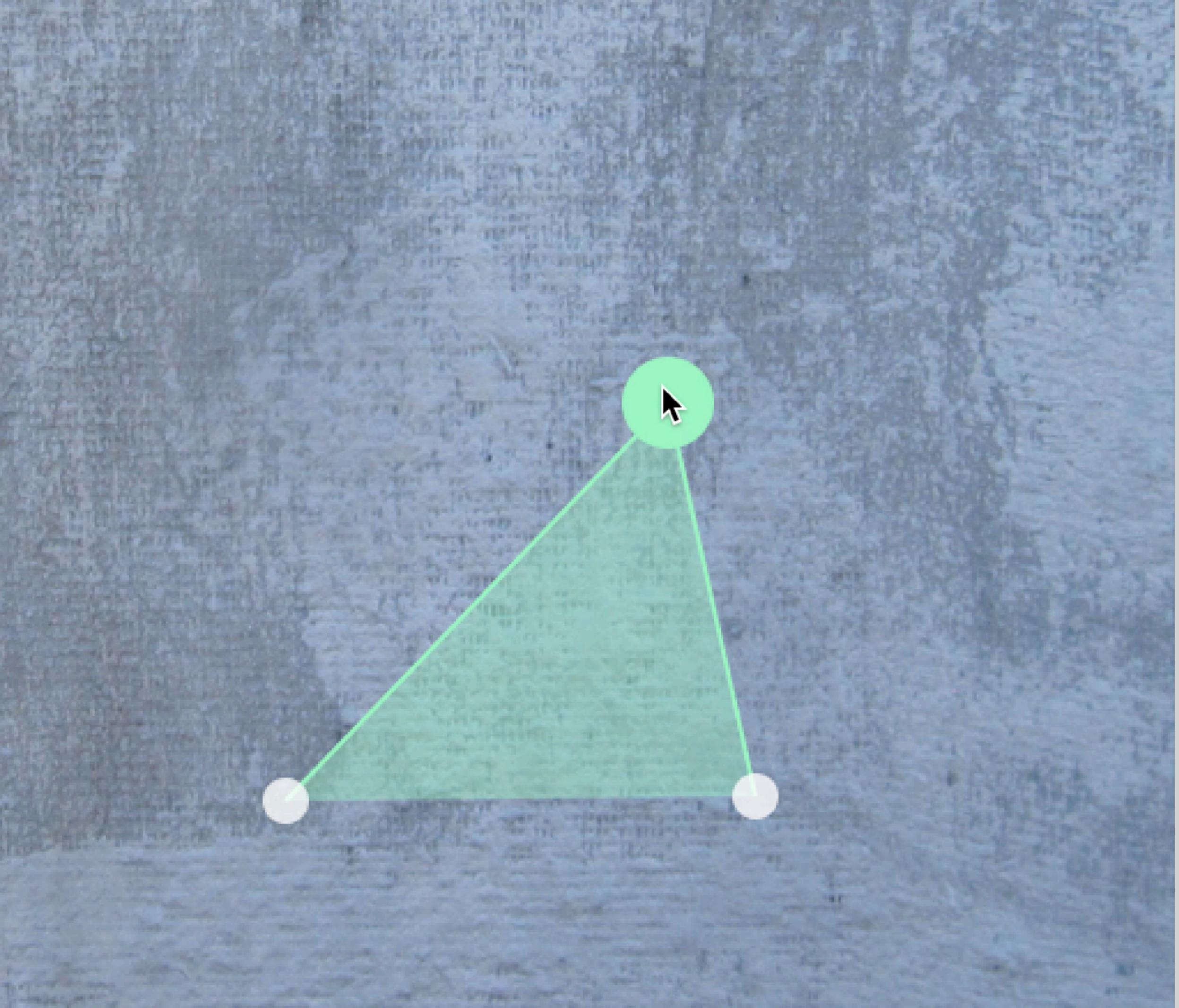
---

Go from 0.0 to 1.1 and back to 1.0

```
UIView animateWithDuration:  
    delay:  
    usingSpringWithDamping:  
    initialSpringVelocity:  
    options:
```

# Anticipate

---





11:35

Thursday, September 18

> Deslizar para desbloquear



# Continuous

---



# Continuous

---



# Drawers

---



OR



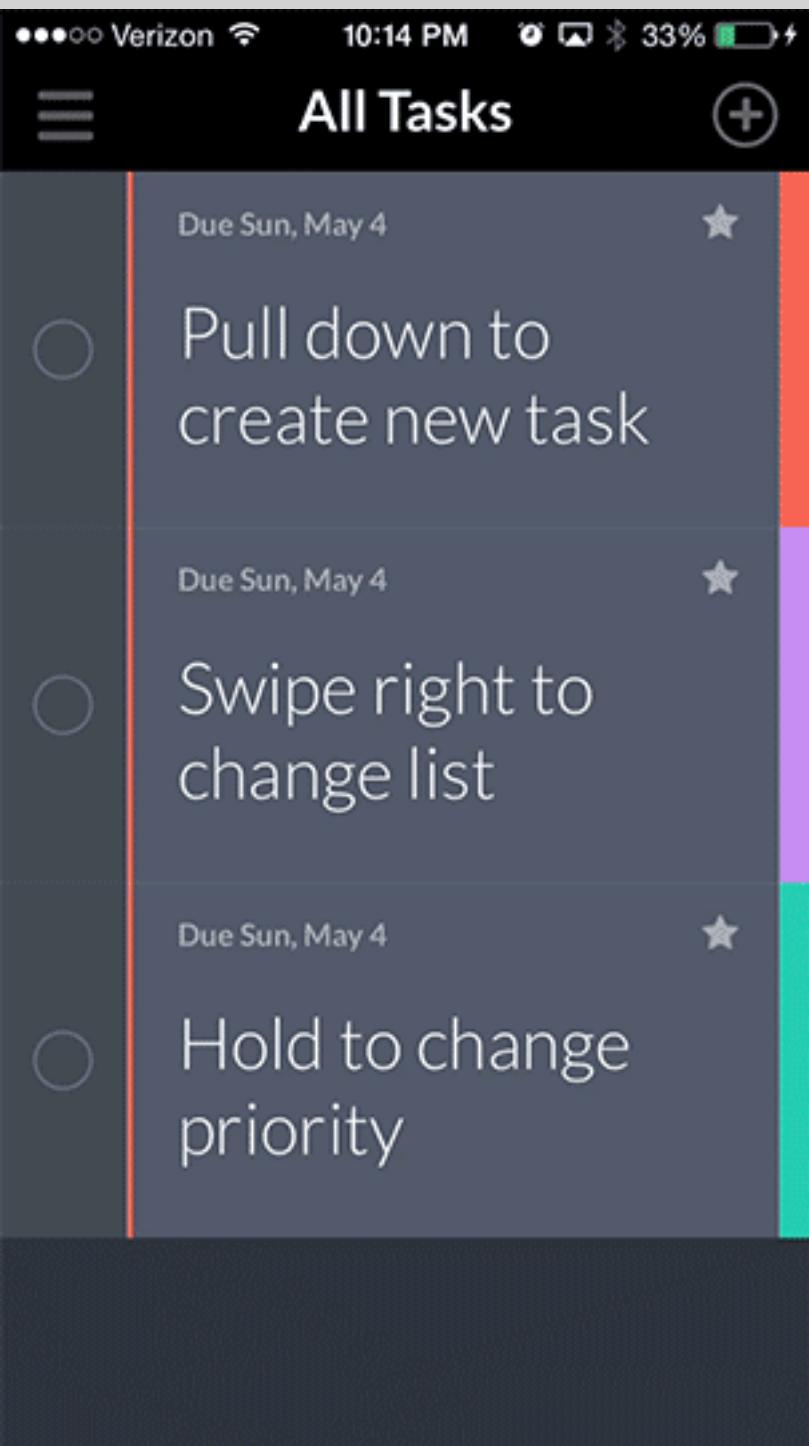
# Drawers

---



[print version](#)

# Drawers



# Interactive & Reversible

---



# 99.69 mi

 34  
RUNS

 12'19"/mi  
AVG PACE

 999 +  
AVG NIKEFUEL

AUGUST



Create a Run Reminder  
Get an alert when it's time to get out  
there.

# Interlude

---

We need to talk about view controller  
transitions

# Springboard Effect

---

- Modal Presentation from Collection View Cell
- Modal Transition Delegate: UIViewController property

# View Snapshots

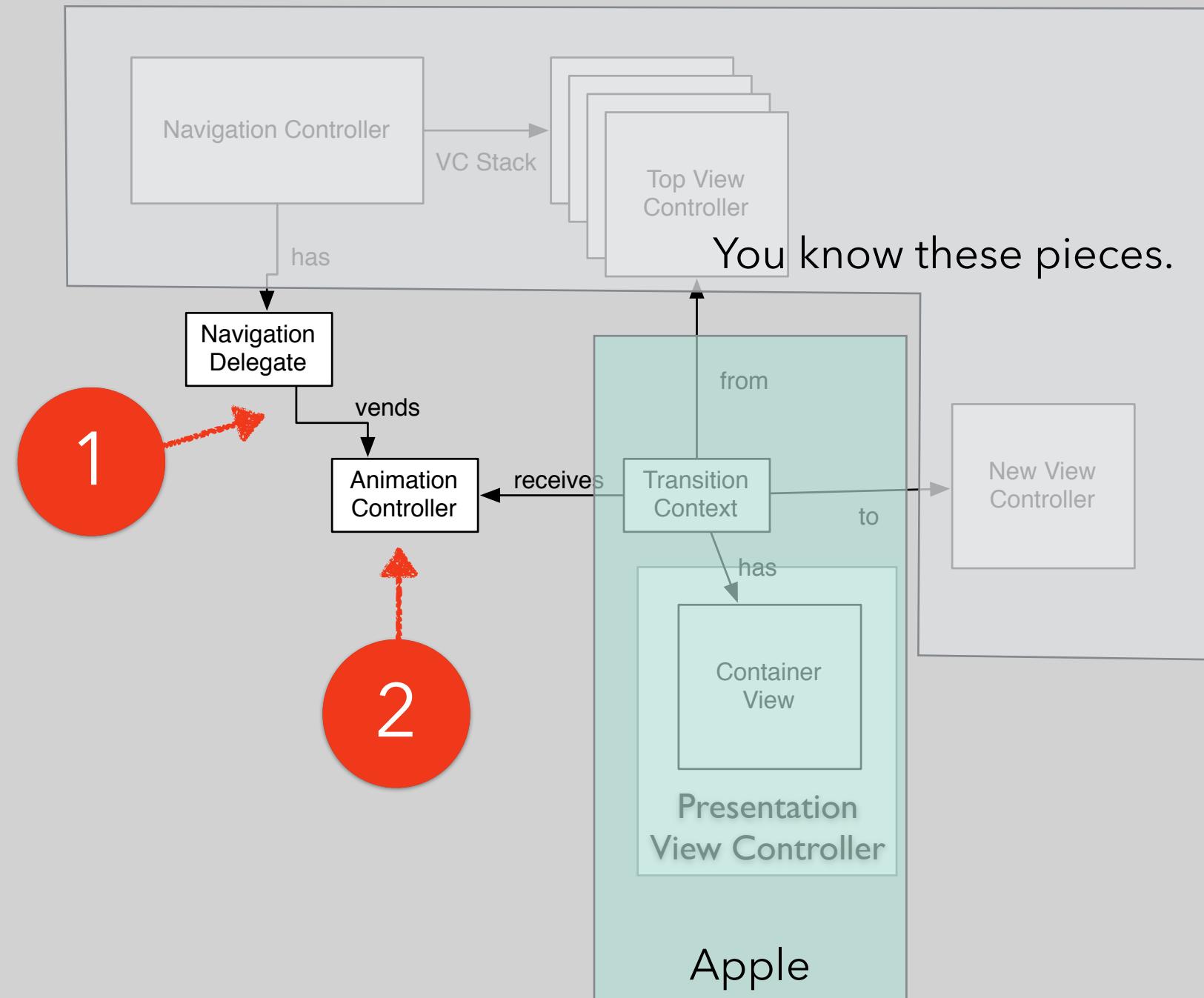
---

- `(UIView *)snapshotViewAfterScreenUpdates:(BOOL)afterUpdates;`
- `(UIView *)resizableSnapshotViewFromRect:(CGRect)rect afterScreenUpdates:(BOOL)afterUpdates withCapInsets:(UIEdgeInsets)capInsets;`

# Using View Snapshots

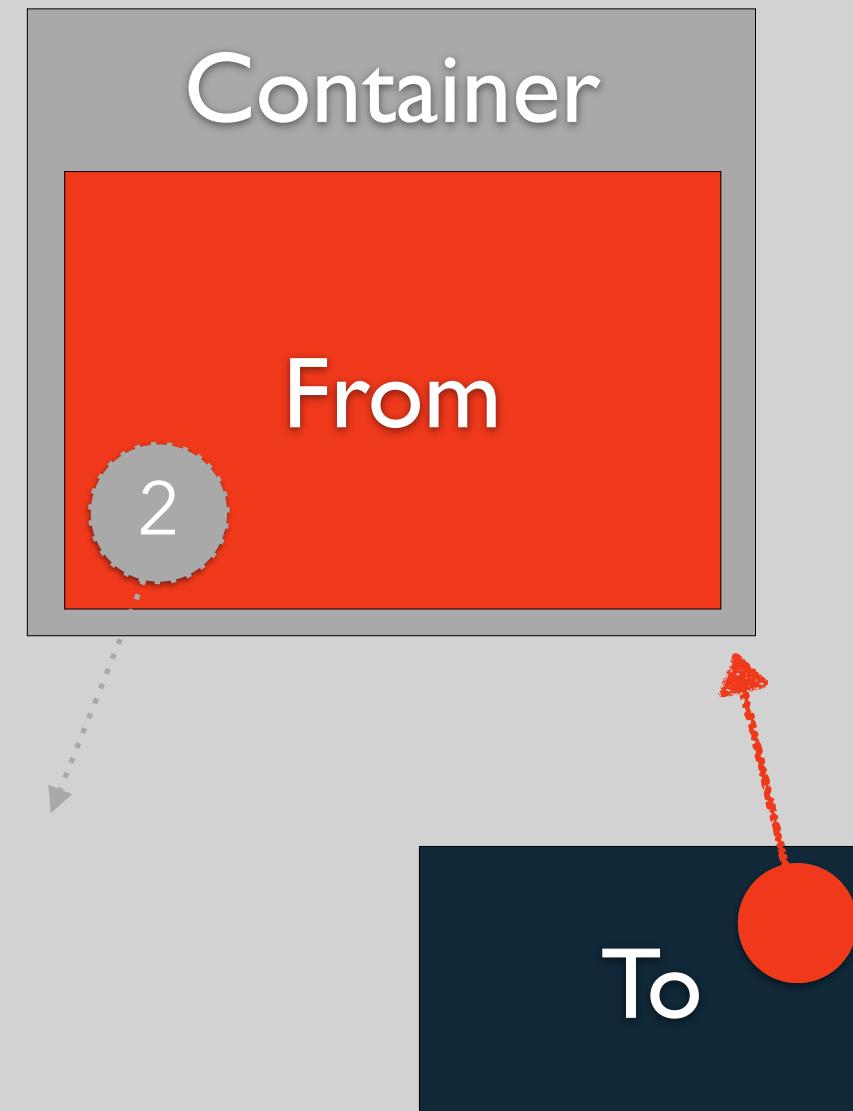
---

```
UIViewController *fromVC = [transitionContext  
viewControllerForKey:UITransitionContextFromViewControllerKey];  
  
UIView *fromSnapshotView = [fromVC.view snapshotViewAfterScreenUpdates:YES];  
  
[transitionContext.containerView addSubview:fromSnapshotView];
```



# VC/V Hierarchy

- Putting To VC in container starts transition (usually)
- Animation Controller must add To view
- May want to remove From view
- At end, From controller / view removed



[print version](#)

# Interactive

---

- Navigation Delegate also returns an Interaction Controller

# Interaction Controller

---

```
- (id<UIViewControllerInteractiveTransitioning>)
navigationController:
    (UINavigationController *)navigationController
interactionControllerForAnimationController:
    (id<UIViewControllerAnimatedTransitioning>)animationController
```

# Interaction Controller

---

- Object conforming to  
UIViewControllerInteractiveTransitioning
- - (void)startInteractiveTransition
- Completion speed, curve

## UIPercentDrivenInteractiveTransition

---

- Concrete class implementing  
UIViewControllerInteractiveTransitioning
- Create, call
  - `(void)updateInteractiveTransition:(CGFloat)percentComplete`

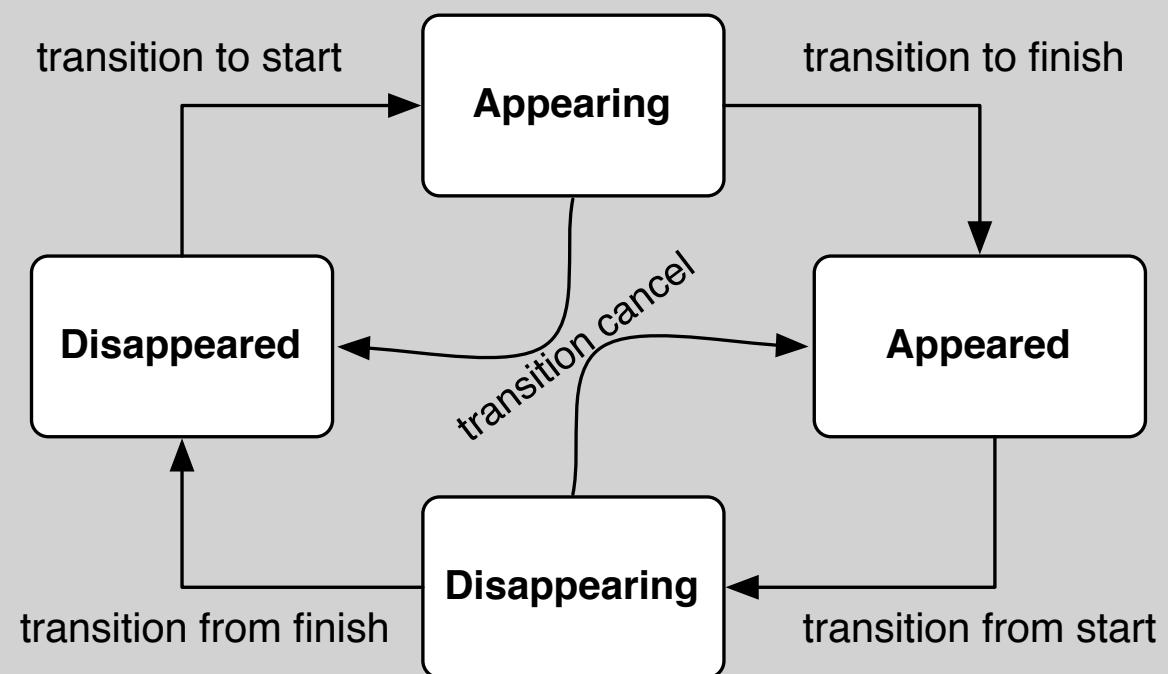
# viewMightAppear?

---

- Interactive transitions could cause `viewWillAppear` to be called but not `viewDidAppear`

# Transition Sequence

## View Lifecycle

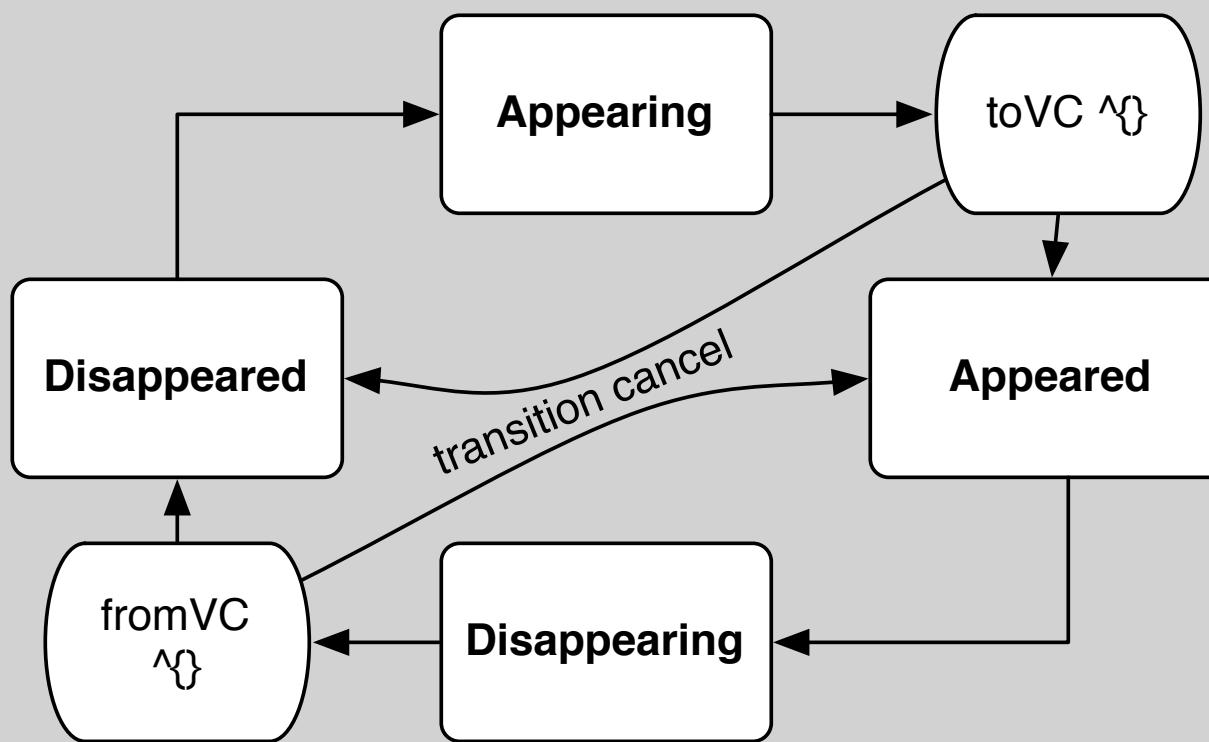


# Transition Coordinator

---

- `viewWillAppear` should now clean up cancelled transition using `notifyWhenInteractionEndsUsingBlock:`
- `viewWillDisappear`: called after this block  
`viewWillAppear`: called on start

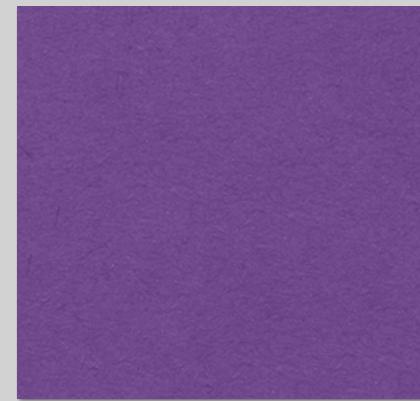
# Transition Coordinator Completion Block



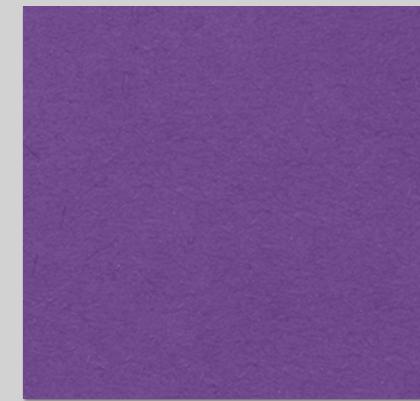
# Timing

---

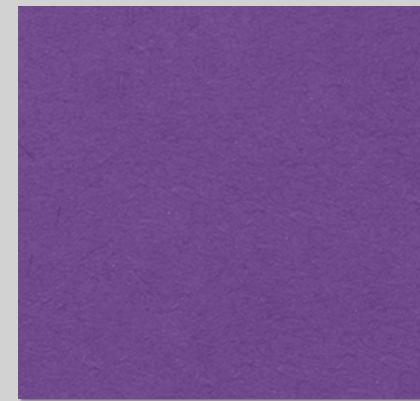
- Experiment Experiment Experiment



1.0 sec



0.2 sec



0.5 sec

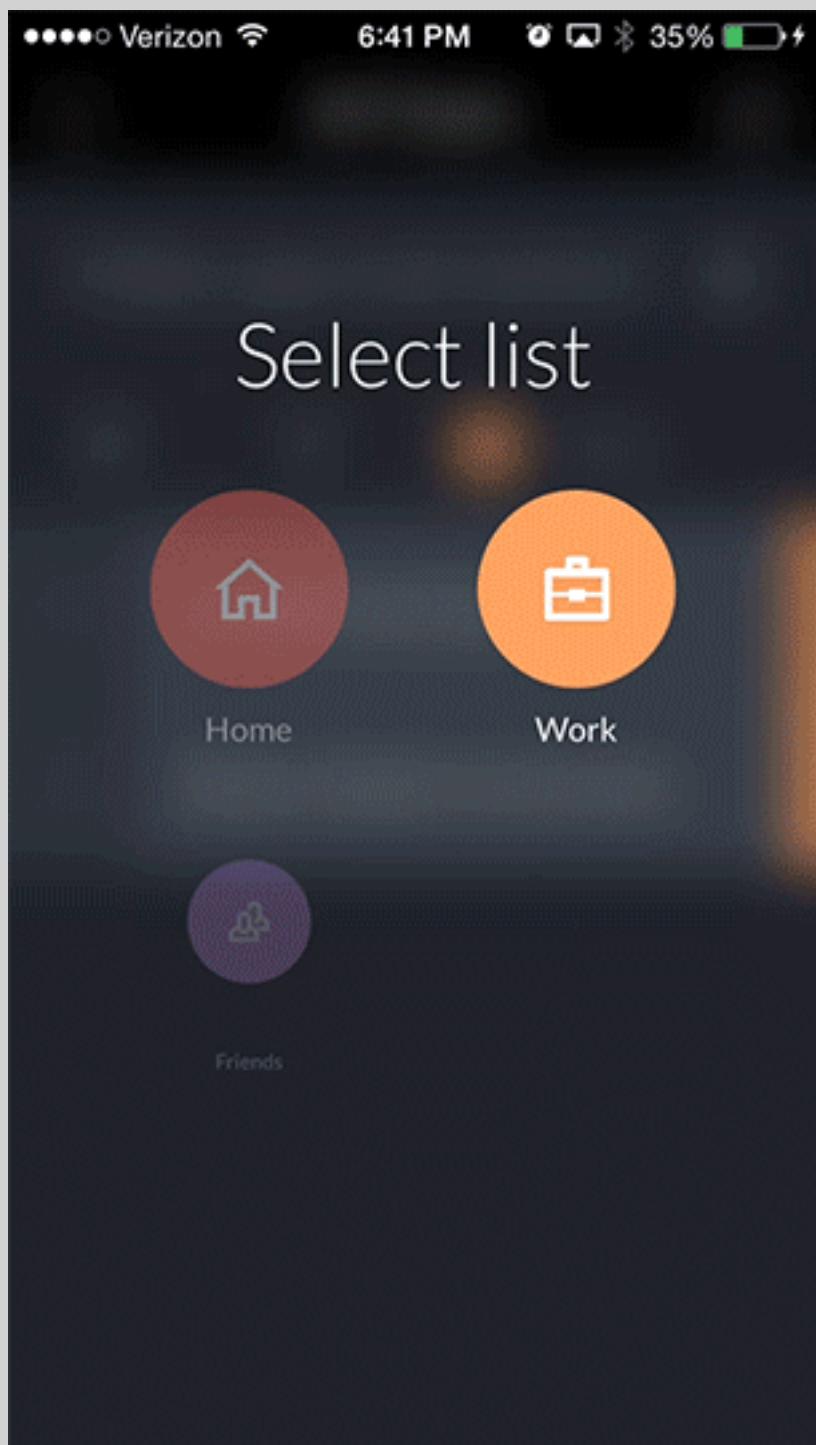
# Timing

---



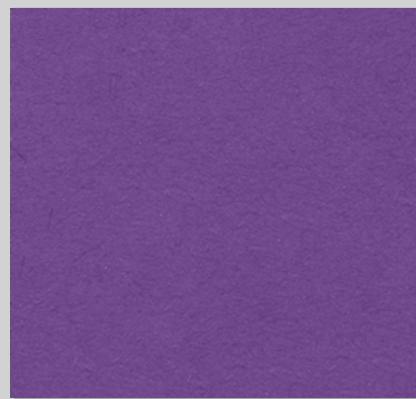
[print version](#)

# Timing



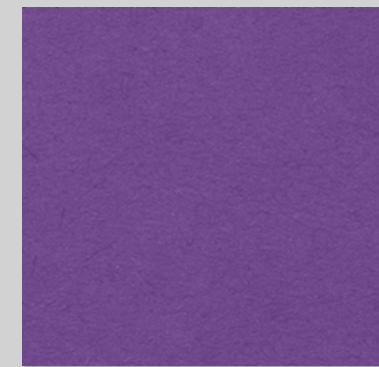
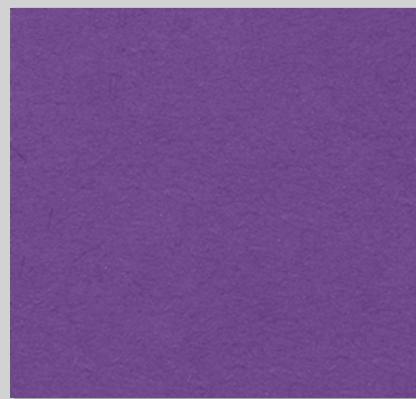
# Arc

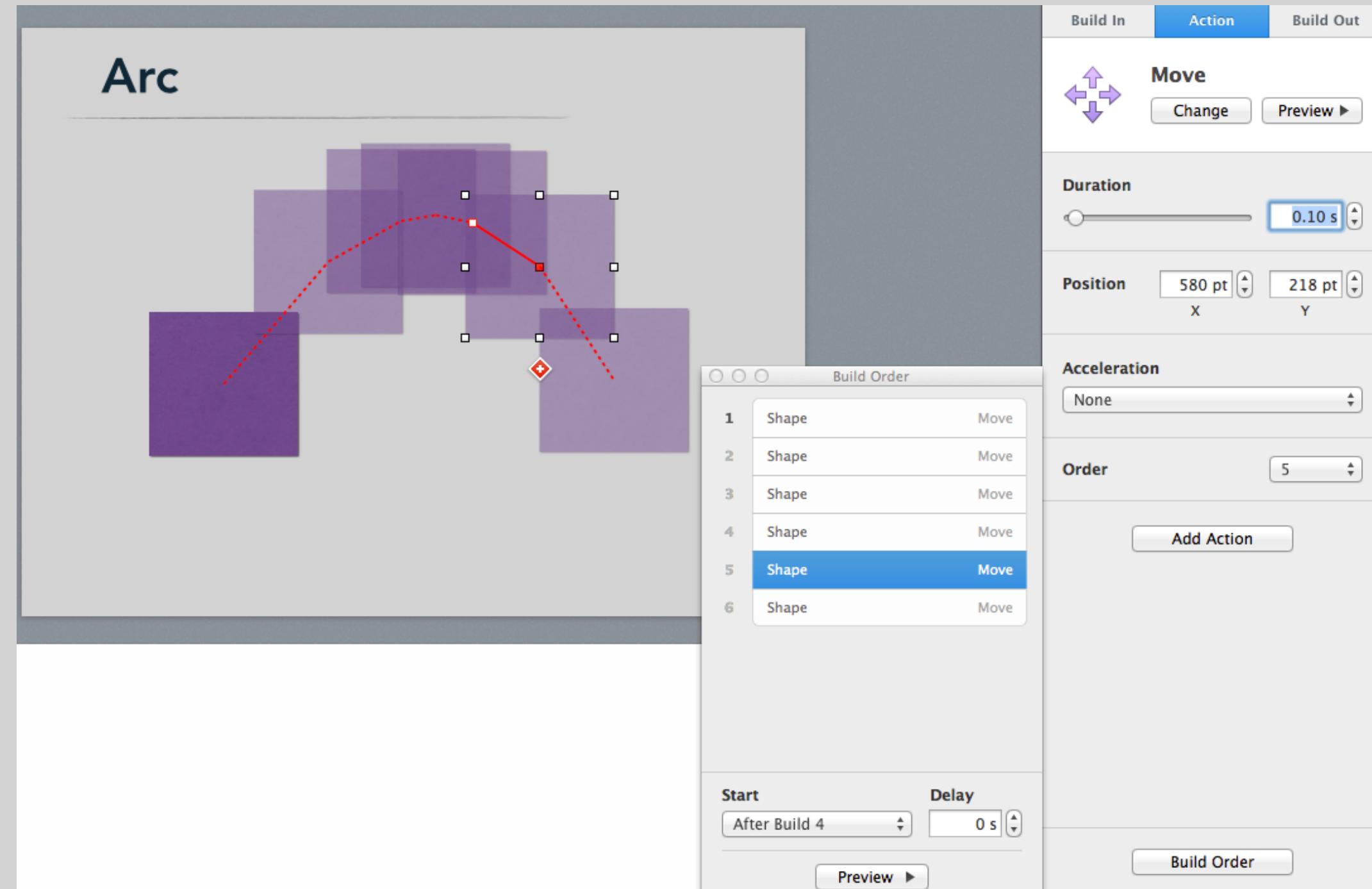
---



# Arc

---





 Hardcover

 Softcover

 Wire-bound

Journal 



## Journal

Size: 11 x 8½ inches

Price: \$29.99 each

\$0.99 each extra page



Cancel

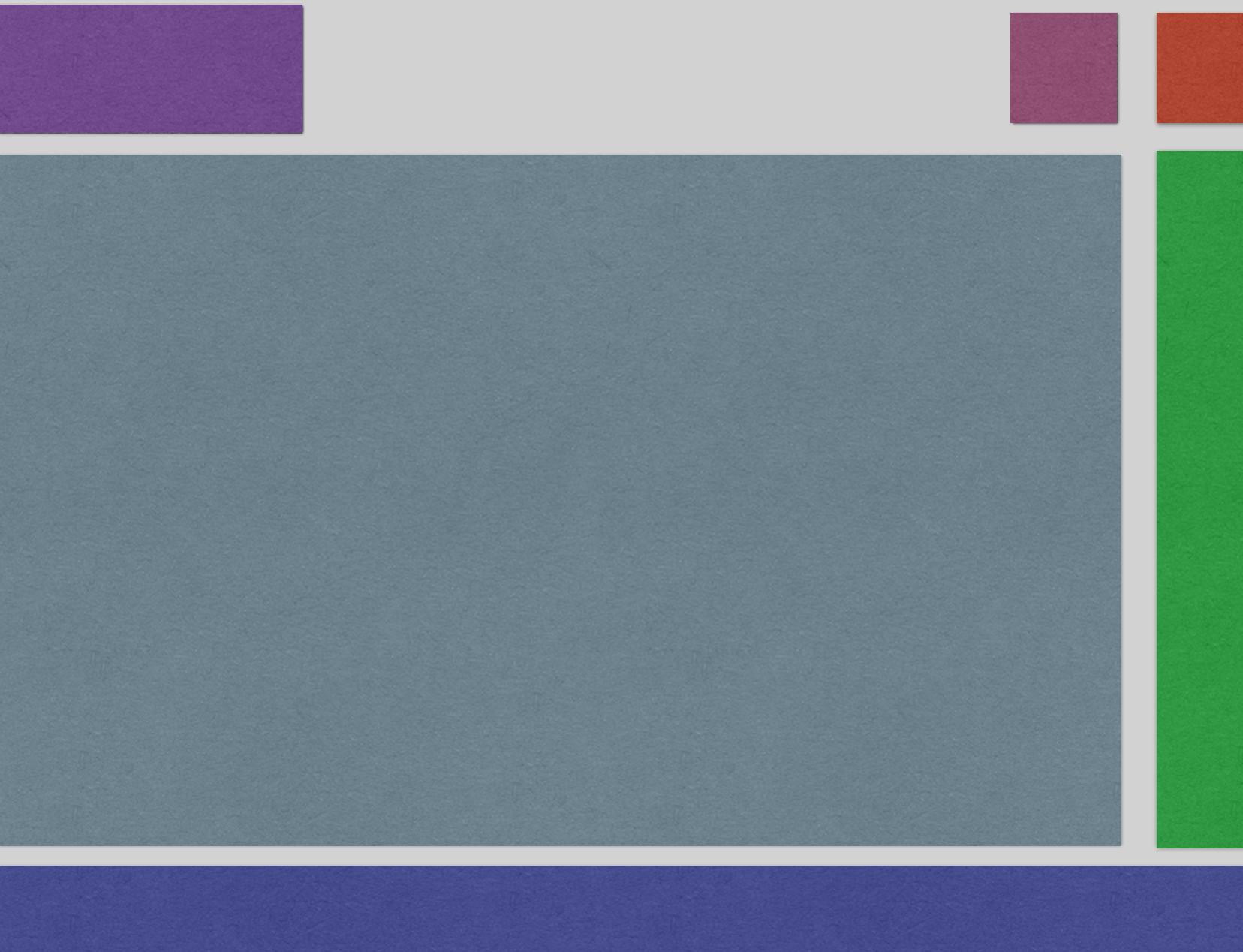
Create

# Keyframe Animations

```
[UIView animateWithDuration:[self  
transitionDuration:transitionContext]  
                      delay:0  
                    options:0  
      animations:^{
        [UIView addKeyframeWithRelativeStartTime:0.0  
                                     relativeDuration:0.5  
                                   animations:^{
          ...
        }];
        [UIView addKeyframeWithRelativeStartTime:0.5  
                                     relativeDuration:1.0  
                                   animations:^{
          ...
        }];
      } completion:^(BOOL finished) {
        ...
      }];
}
```

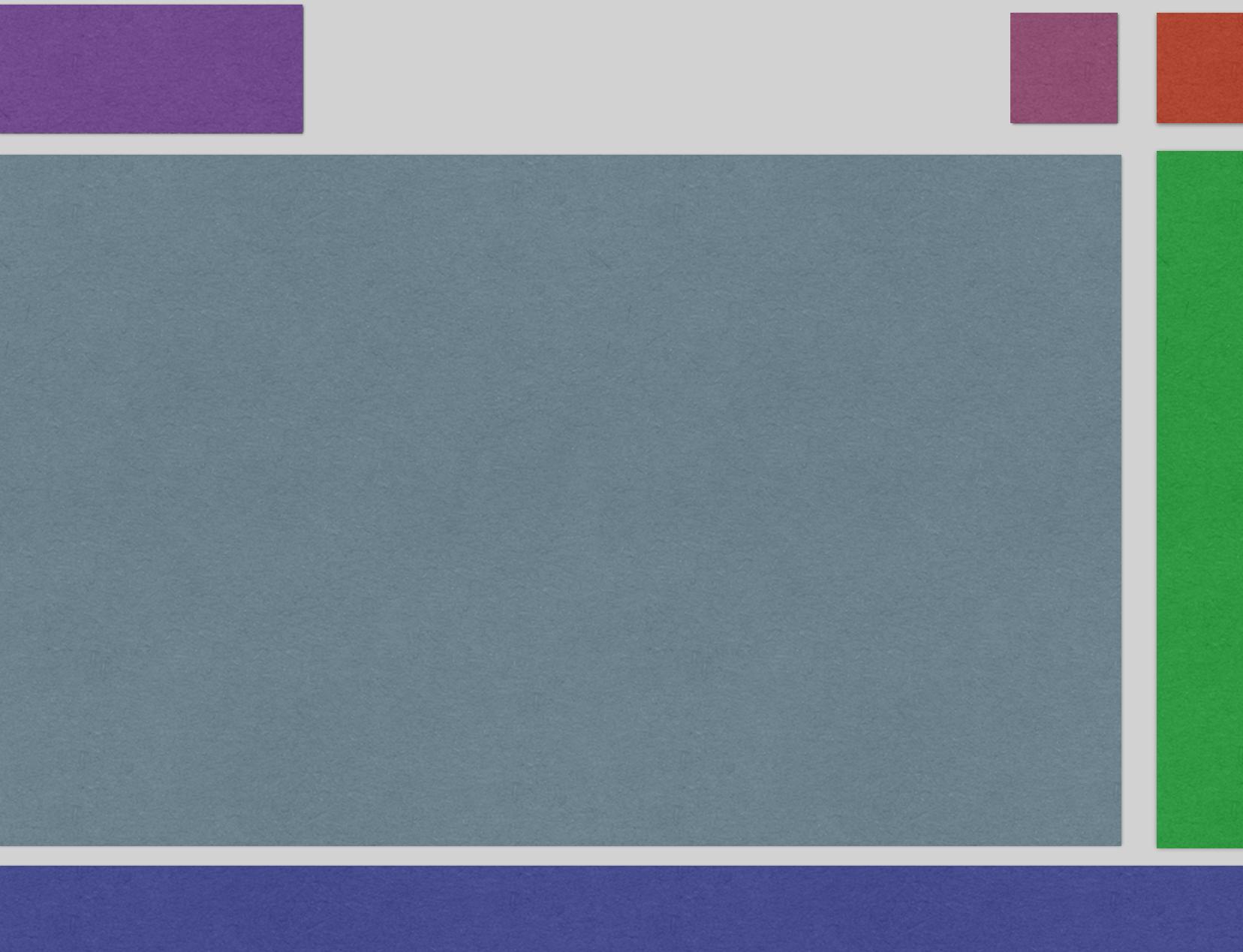
# Arc

---



# Arc

---



# Solidarity

---

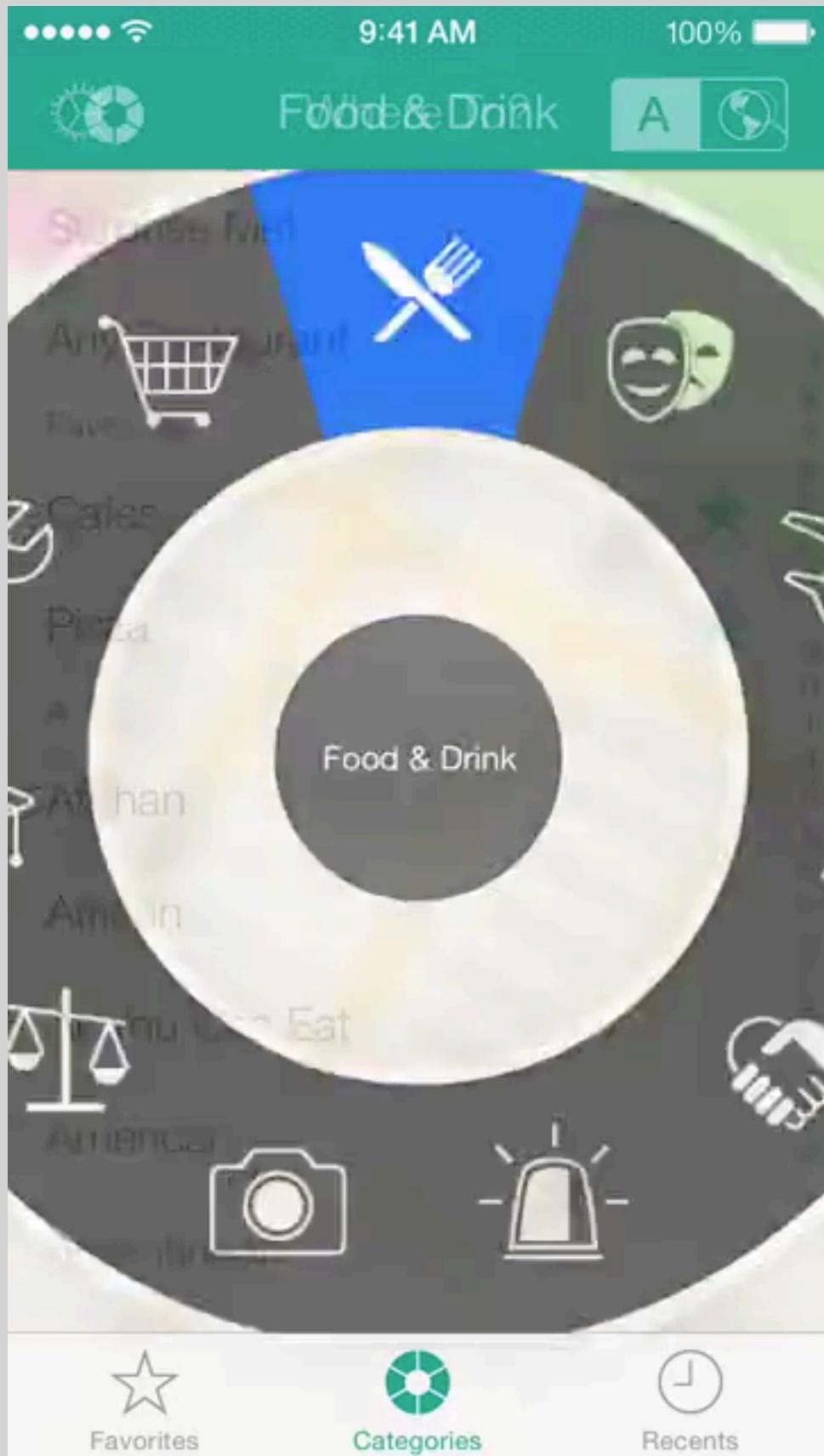
- Dynamic Animator



# Appeal

---



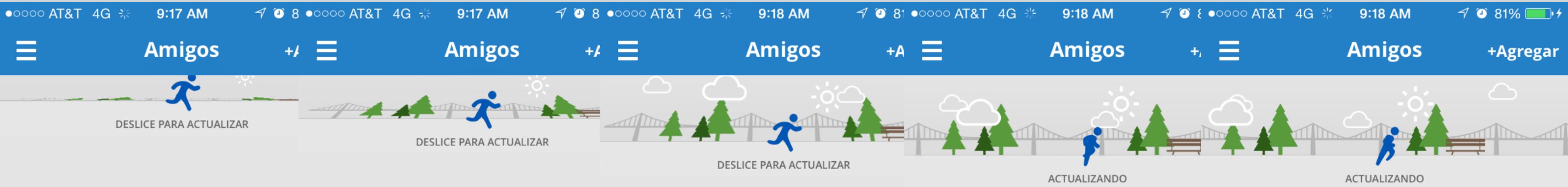


# Partial View Snapshots

---

- `(UIView *)resizableSnapshotViewFromRect:(CGRect)rect  
afterScreenUpdates:(BOOL)afterUpdates  
withCapInsets:(UIEdgeInsets)capInsets;`
- Great for effects / cool animations

# Secondary Action



# Review

---

1. Animate.

2. Spring.

3. Anticipate.

4. Continuous.

5. Interactive.

6. Timing.

7. Arc.

8. Solidarity.

9. Appeal.

10. Secondary Action.

# 2 more Steps

---

Prototype

then

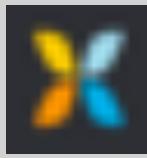
Implement

# Prototype



pop

Canvas  
Animate in Xcode without code



# Further Reading

---

- WWDC 2014 Session 236 - "Building Interruptible and Responsive Applications"
- Also WWDC 2014 211, 223, 228, 230, 235
- Every Scroll View talk
- Tech Talk 2013 - Custom Transitions Using View Controllers
- Tech Talk 2013 - User Interface Design for iOS 7 Apps
- WWDC 2013 Session 226,  
"Implementing Engaging UI"
- Our iOS 8 demos at: <https://github.com/bignerdranch/iOS8Demos>
- <http://ronnqvi.st>

# Thanks!

---

- Take a class from the Big Nerd Ranch
- Big Nerd Ranch can also help build your app
- Talk to me



@randomstep



step@bignerdranch.com

- Turn in your evaluation

# Questions?

# Thank You!