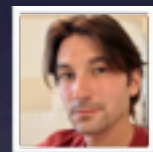


Responsive Touch and Transitions



Jonathan Badeen
@badeen

Transitions



Importance of Transitions

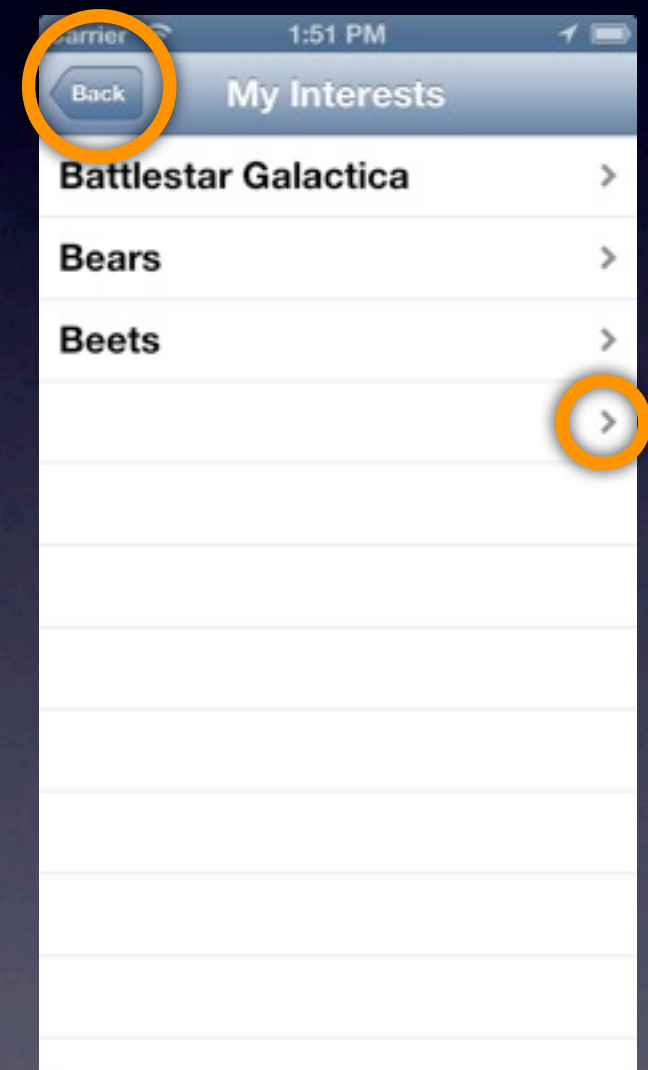
- Connects multiple views
 - Small screens can't maintain multiple views
- Shows the relationship between screens
- Gently prepares the user for a change
- Delights the user

Questions Transitions Answer

- How does the new information relate to the previous information?
- What am I about to see?
- What should I stop looking at?
- Can I get back to the last screen and how?
- Where am I in the application?

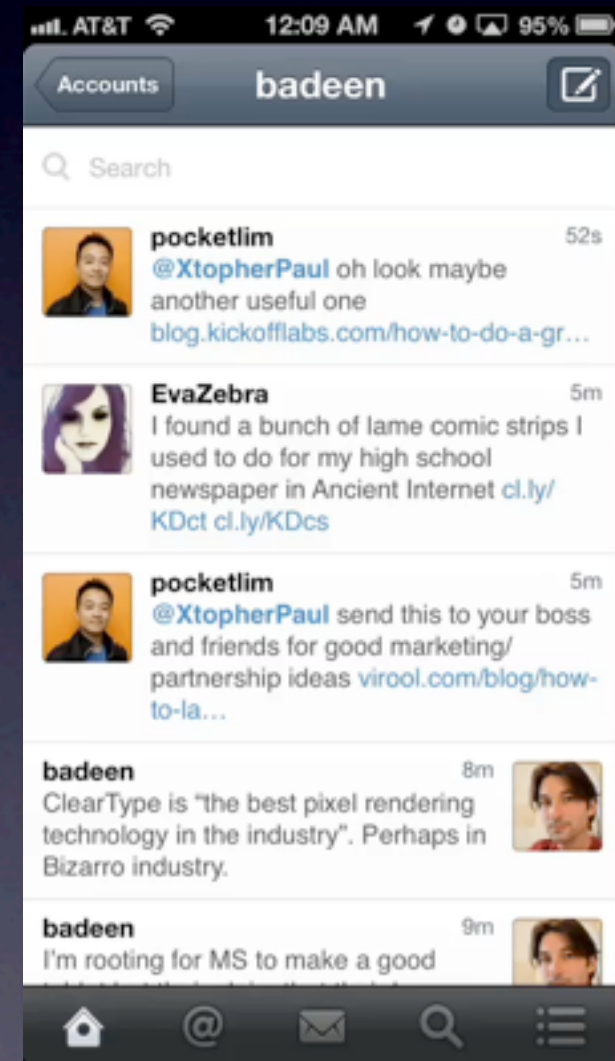
UINavigationController

- Transitions reflect interface elements
- Disclosure indicator implies a shift to the right
- “Back” button implies shift to the left.



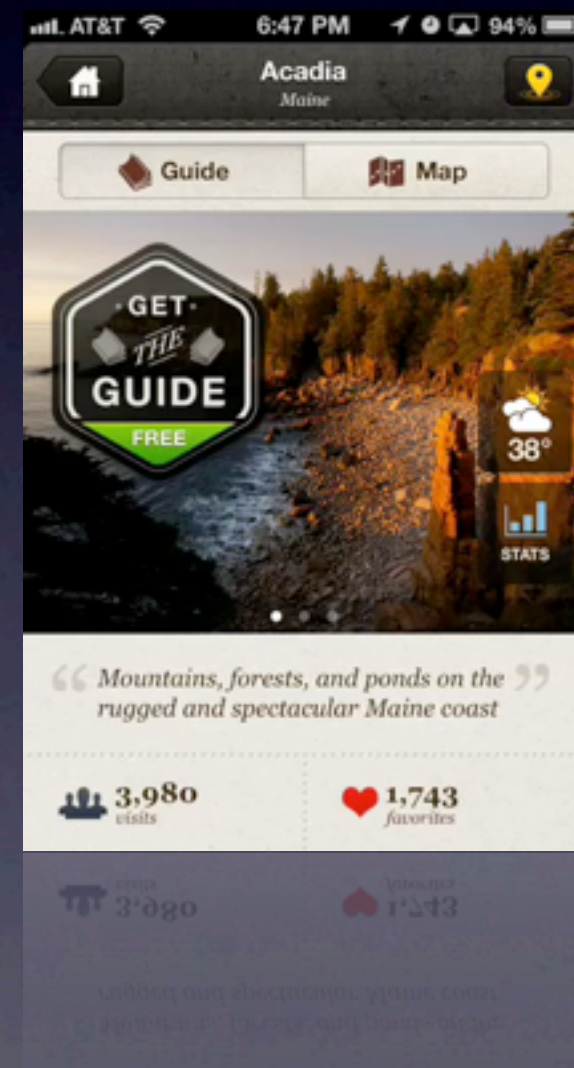
Flurry for Twitter

- New tweet pushes main interface away and shifts focus
- Fly out animation implies success



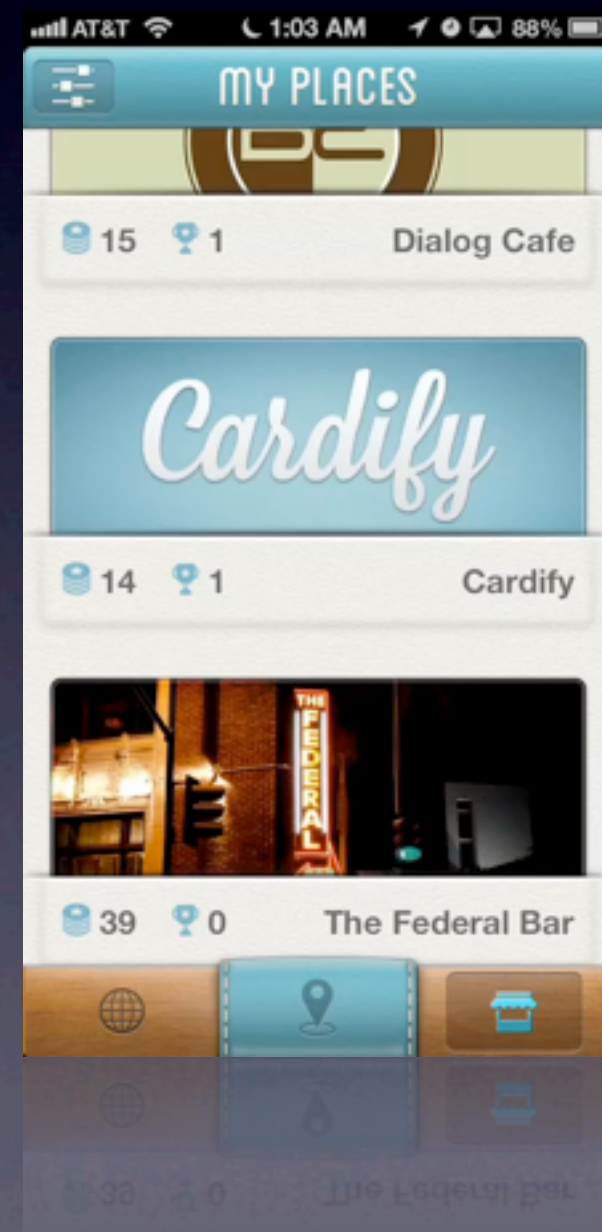
National Geographic Park Guides

- Main interface moves back to shift focus to new information
- Main interface remains visible to maintain relationship with new data
- Placement in the application is easily identifiable
- Return by tapping “Done” or the background interface



Cardify

- Incorporates elements from the previous interface
- Can be closed by tapping the card or “Done” button



Responsive Touch



Copy Microsoft

Copy Apple

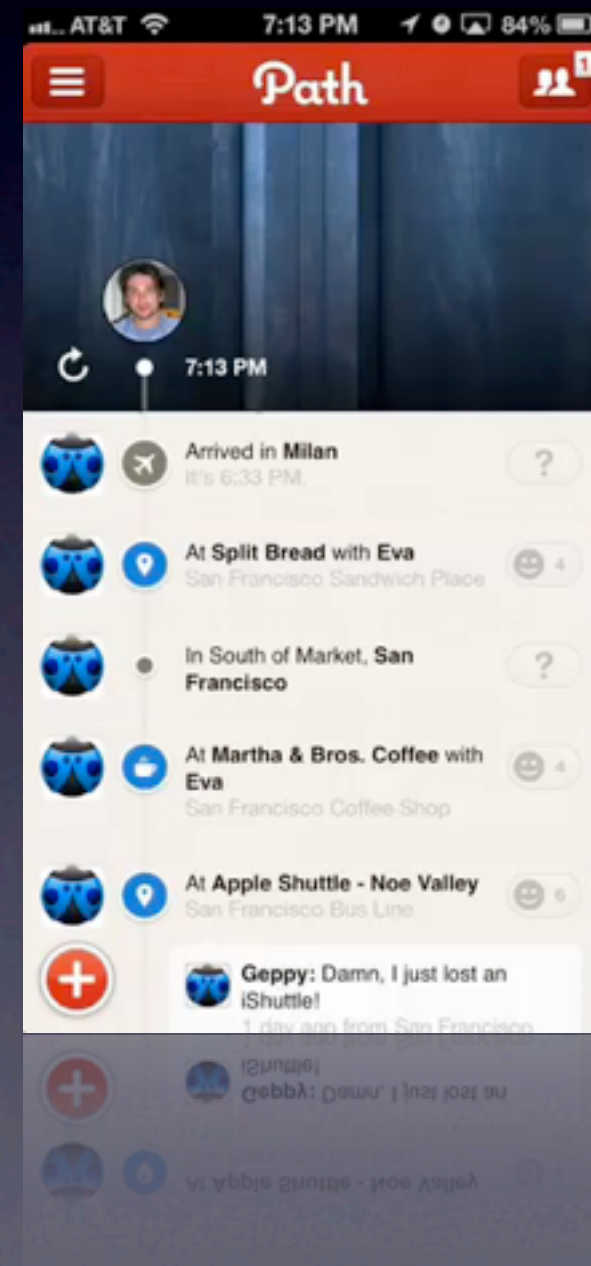
- UITapGestureRecognizer = Lonely
- UISwipeGestureRecognizer = Evil
- Obey physics
- Follow through after gesture

Interactive Transitions

- Real-time gesture transition
- Involves the user (more interactive)
- Strengthens the connection between interfaces

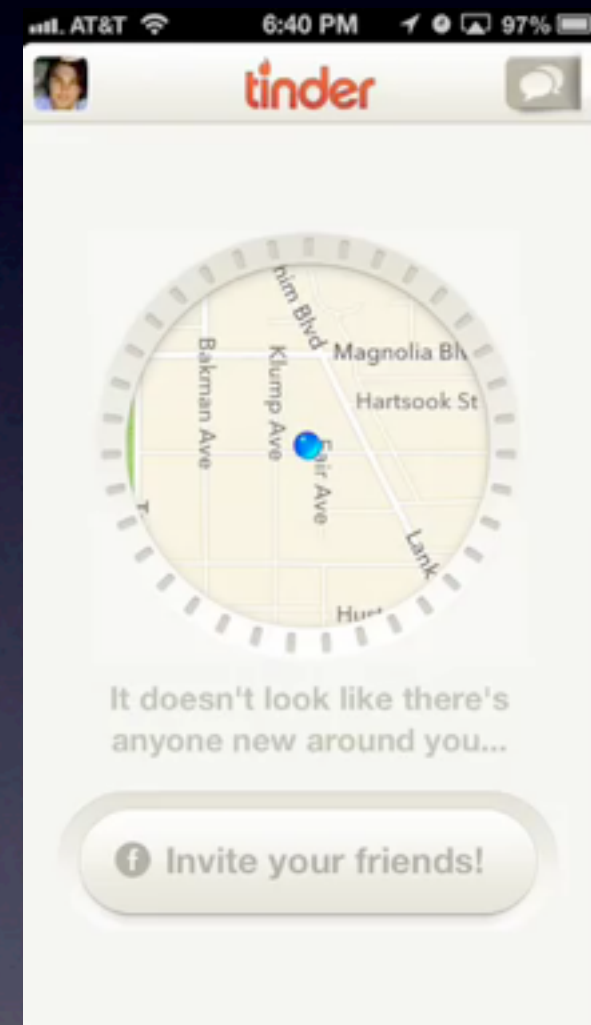
Transition Examples: Path

- Tap the navigation buttons to reveal menus
- Pan horizontally to reveal menus
- Overlapping screen remains to identify location
- Subtle realism in physics



Transition Examples: tinder

- Control placement indicates direction of interaction
- Control placement indicates relative placement in application



Transition Examples: Our Choice

- Reuses elements from the source view
- Slight resizing for focus
- Fun physics



JLBPartialModal

Mimics National Geographic's Park Guides
iPhone application.

<https://github.com/badeen/JLBPartialModal>



JLBPModal.m

```
- (void)presentViewController:(UIViewController *)viewControllerToPresent
    dismissal:(void (^)(void))block
{
    Create a subclassed view controller which will use UIViewController Containment

    self.containerViewController = [[JLBPModalContainerViewController alloc] init];
    self.containerViewController.contentViewController = viewControllerToPresent;

    Create a new window for our modal

    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    [self.window makeKeyAndVisible];

    ... Setup window styles and styling

    Add gesture recognizer for dismissing when tapping outside the modal view

    [self.window addGestureRecognizer:[[UITapGestureRecognizer alloc] initWithTarget:self
                                          action:@selector(windowTapped:)]];

    [self.containerViewController showContentWithAnimationDuration:0.4f completion:nil];

    Animate our other windows into the background (there should only be 1)

    for (UIWindow *window in [[UIApplication sharedApplication] windows]) {
        if (window != self.window) {
            [window.layer addAnimation:[self pullBackAnimation] forKey:@"pullBackAnimation"];
        }
    }
}
```

JLBPartialModal.m

```
- (CAKeyframeAnimation *)pullBackAnimation
{
    CATransform3D startTransform = [self perspectiveTransform];
    CATransform3D middleTransform = CATransform3DConcat([self windowRotationTransform],
                                                         startTransform);
    CATransform3D endTransform = CATransform3DConcat([self windowScaledTransform],
                                                       startTransform);

    CAKeyframeAnimation *anim = [CAKeyframeAnimation animationWithKeyPath:@"transform"];
    anim.duration = 0.4;
    anim.calculationMode = kCAAnimationCubic;
    anim.removedOnCompletion = NO;
    anim.fillMode = kCAFillModeForwards;
    anim.values = @[ [NSValue valueWithCATransform3D:startTransform],
                     [NSValue valueWithCATransform3D:middleTransform],
                     [NSValue valueWithCATransform3D:endTransform] ];

    return anim;
}

- (CATransform3D)windowRotationTransform
{
    return CATransform3DMakeRotation(M_PI_4 / 1.7f, 1, 0, 0);
}

- (CATransform3D)windowScaledTransform
{
    return CATransform3DMakeScale(0.8f, 0.8f, 1.0f);
}
```

```
- (void)showContentWithDuration:(NSTimeInterval)duration
{
    Position the view so it sits at the bottom.

    UIViewController *contentVC = self.contentViewController;
    CATransform3D viewHeight = CGRectGetHeight(contentVC.view.frame);
    contentVC.view.center = CGPointMake(CGRectGetMidX(self.view.bounds),
                                         CGRectGetMaxY(self.view.bounds)
                                         - (viewHeight / 2.0f));

    Apply a transform to move it offscreen.

    contentVC.view.transform = CGAffineTransformMakeTranslation(0.0f, viewHeight);

    [self addChildViewController: contentVC];
    [self.view addSubview:contentVC.view];

    Fade in our dark background (overlay) and animate the view in from the bottom

    [UIView animateWithDuration:duration
        delay:0.0f
        options:UIViewAnimationOptionCurveEaseOut
        animations:^(
            contentVC.view.transform = CGAffineTransformIdentity;
            self.view.backgroundColor = [UIColor colorWithWhite:0.0f
                                                            alpha:0.6f];
        ) completion:^(BOOL finished) {
        [contentVC didMoveToParentViewController:self];}]];
}
```

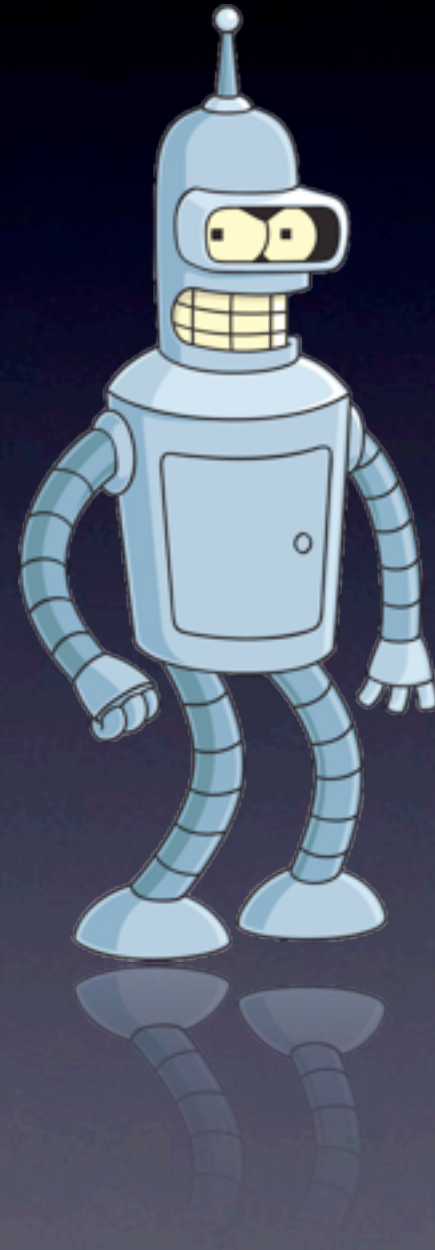

JLBPartialModal

Mimics National Geographic's Park Guides
iPhone application.

<https://github.com/badeen/JLBPartialModal>



The Future



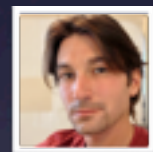
The Not So Distant Future

- Easier 3D manipulation (Scene Kit)
- Physics-based animation APIs
- GUI timeline editing for animation

The More Distant Future

- 3D screens
- Tactile screens
- Pressure-sensitive screens

Responsive Touch and Transitions



Jonathan Badeen
@badeen