Mars

Earth

26,000 ml

Earth

iPhone User Interface Anti-Patterns

Eric Hackathorn
UCAR Software Engineering Assembly
February 23, 2012

Anti-Patterns

UI Anti-Pattern: A design that causes at least as many problems as it solves, despite better options being available.
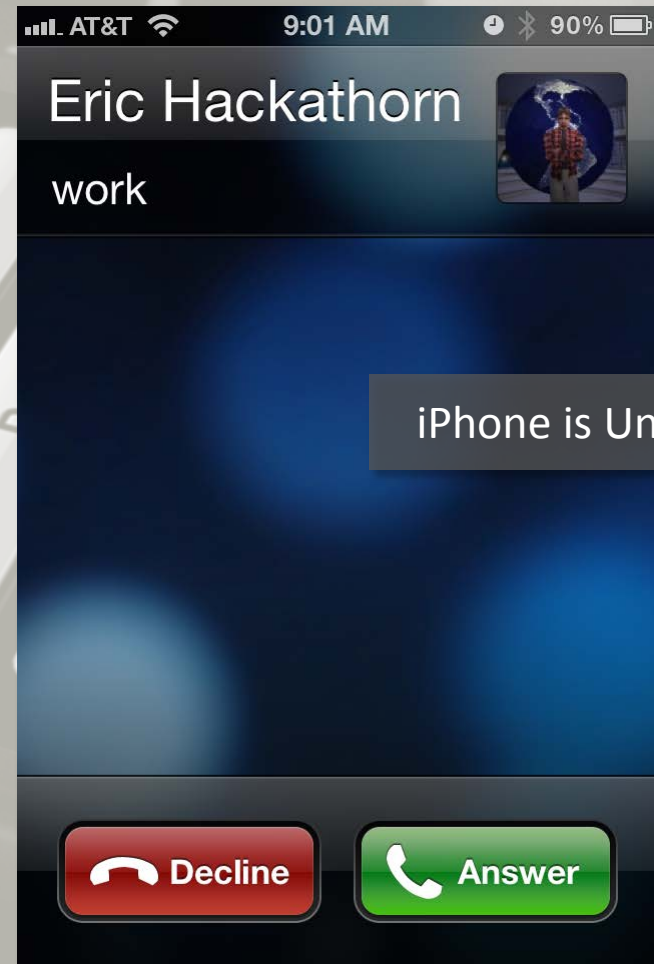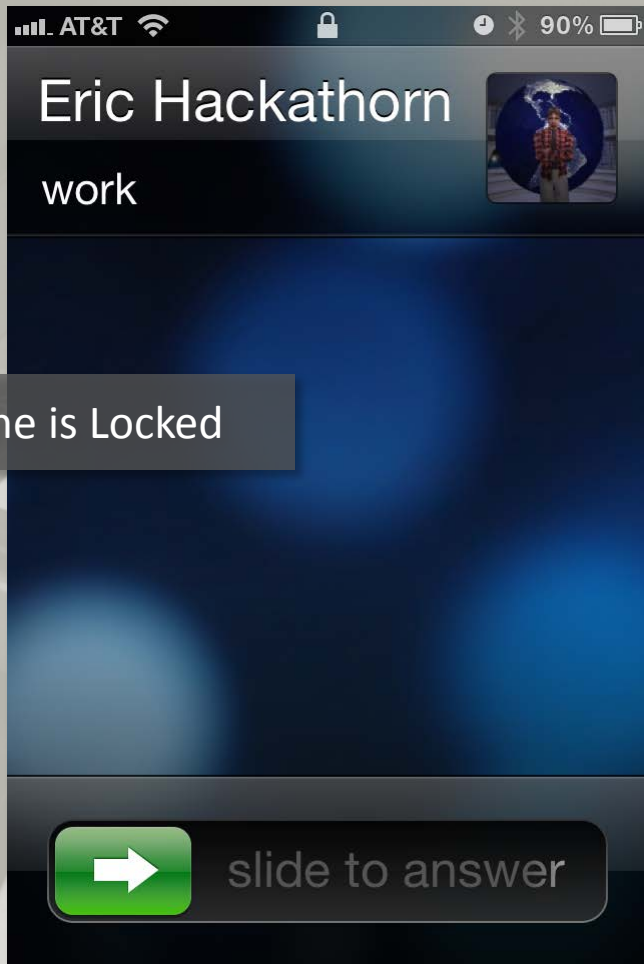
Ideas "borrowed" from:

- Sleight of Hand
- Billboards
- Bullhorns
- The Bouncer
- Gesture Hijacking

...and Designing Cocoa Touch Applications

Programming the

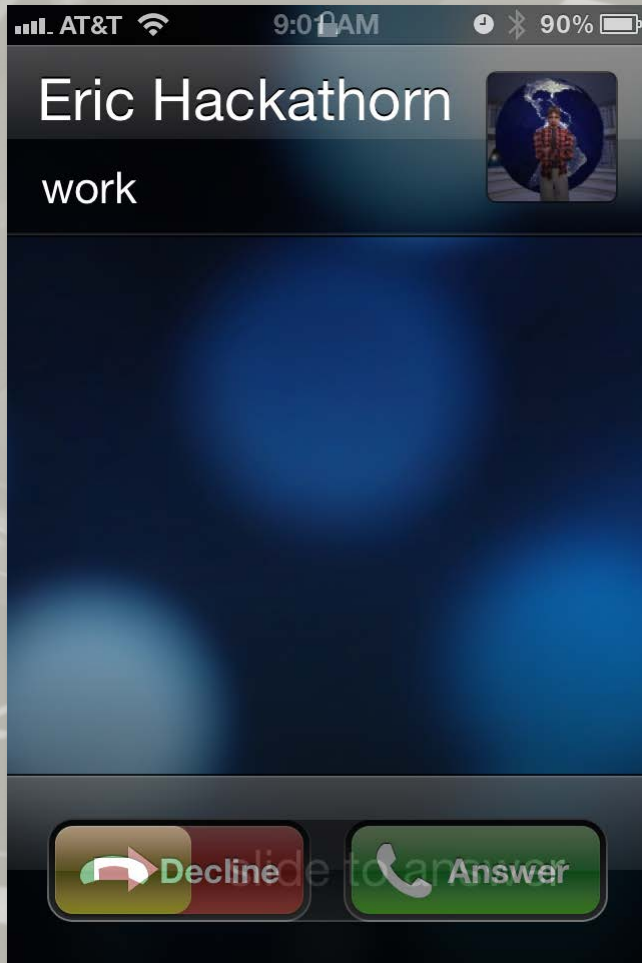iPhone User Experience

O'REILLY®

Toby Boudreaux

Sleight of Hand
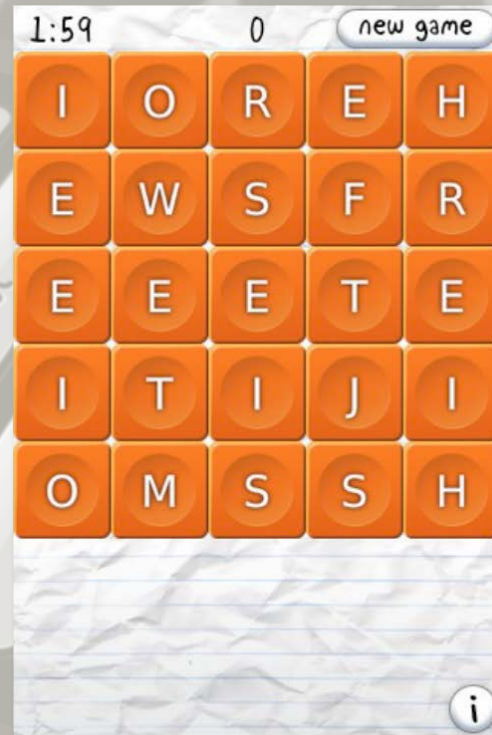## Swapping Meaning for Hot Areas

Sleight of Hand
Swapping Meaning for Hot Areas

- Consider muscle memory and habit
- Overlay screens and consider proximity
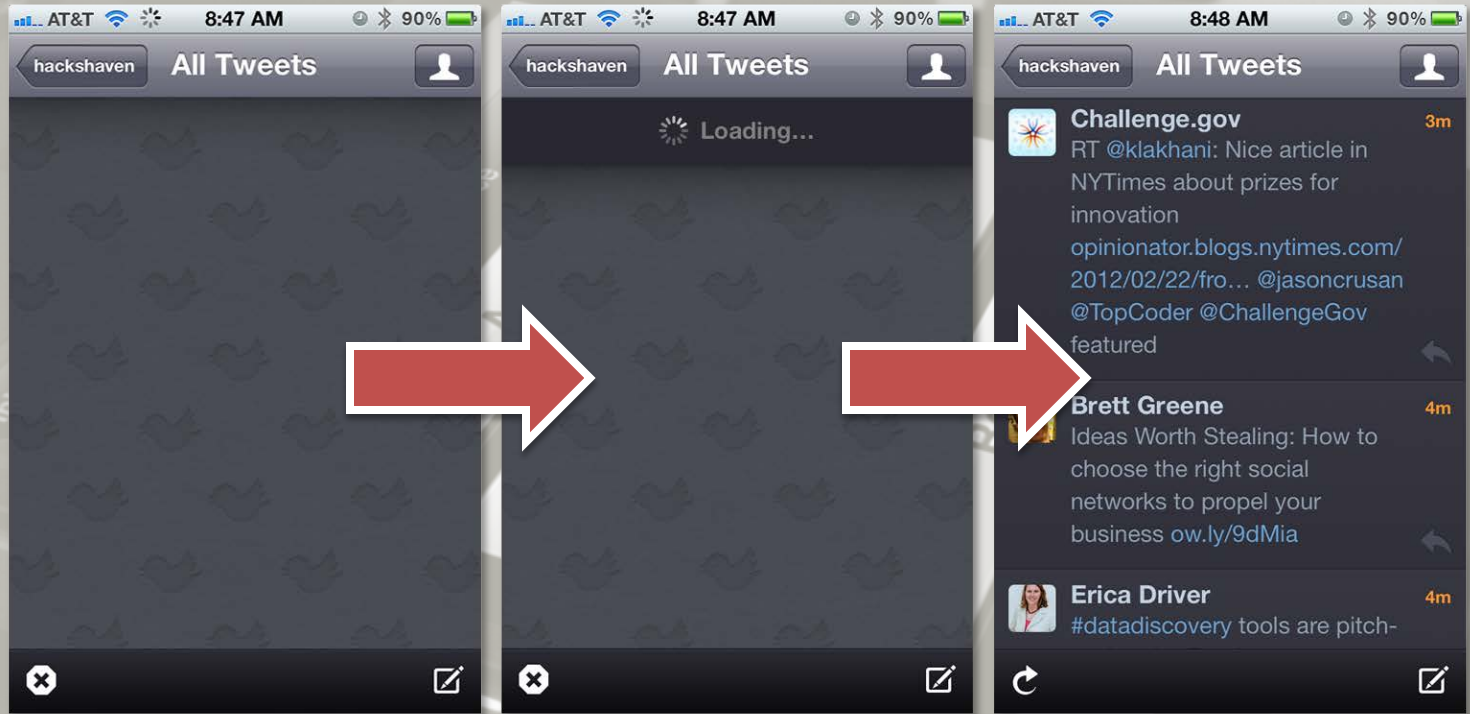- Account for an extra and accidental touch

Billboards

Avoid Displaying An About or Splash Screen

- Forget "quit" and "launch"
- Replace with "pause" and "un-pause"
- Think about fast application cycling
- Don't put branding ahead of users

A Better Method: Progressive Reveal

- Show a structured screen without the user data
- Give the impression that your application un-pauses
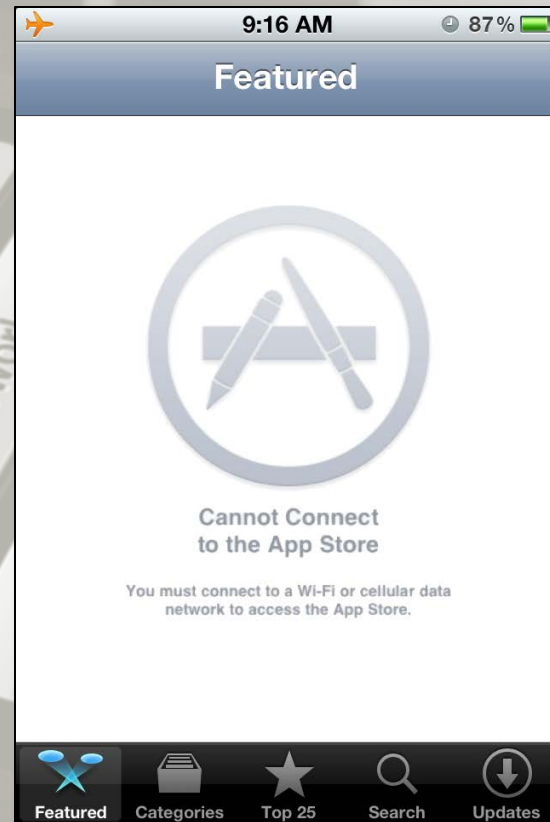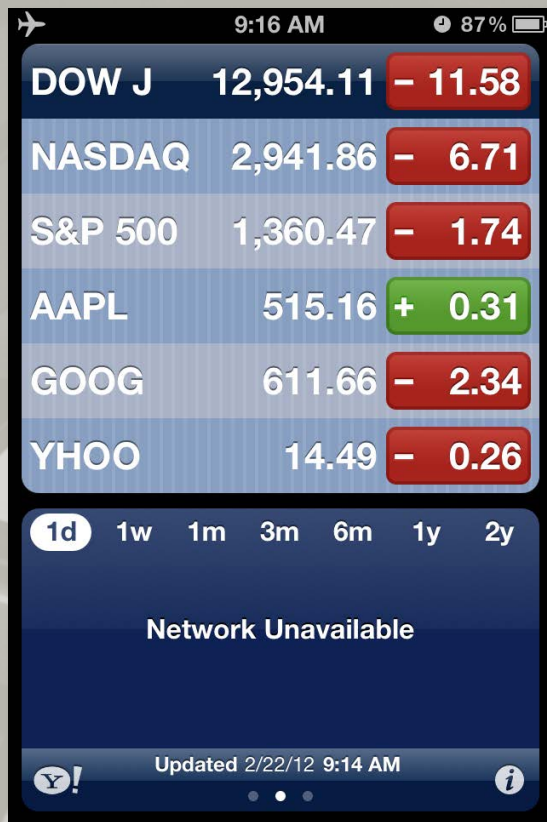- Make application cycling addictive

Bullhorns

Notification mechanisms that are disproportional to the message
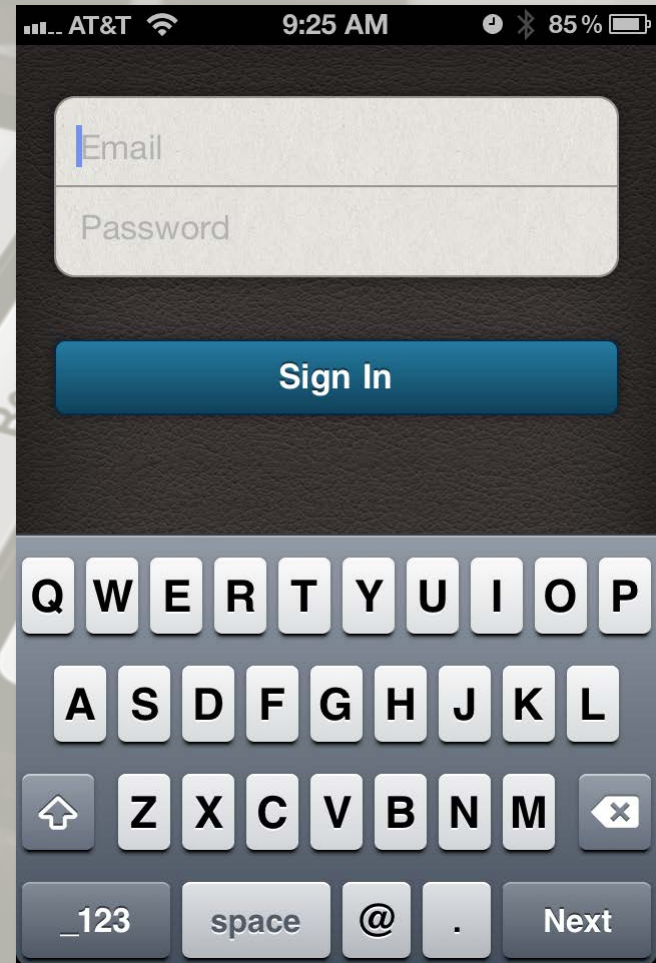

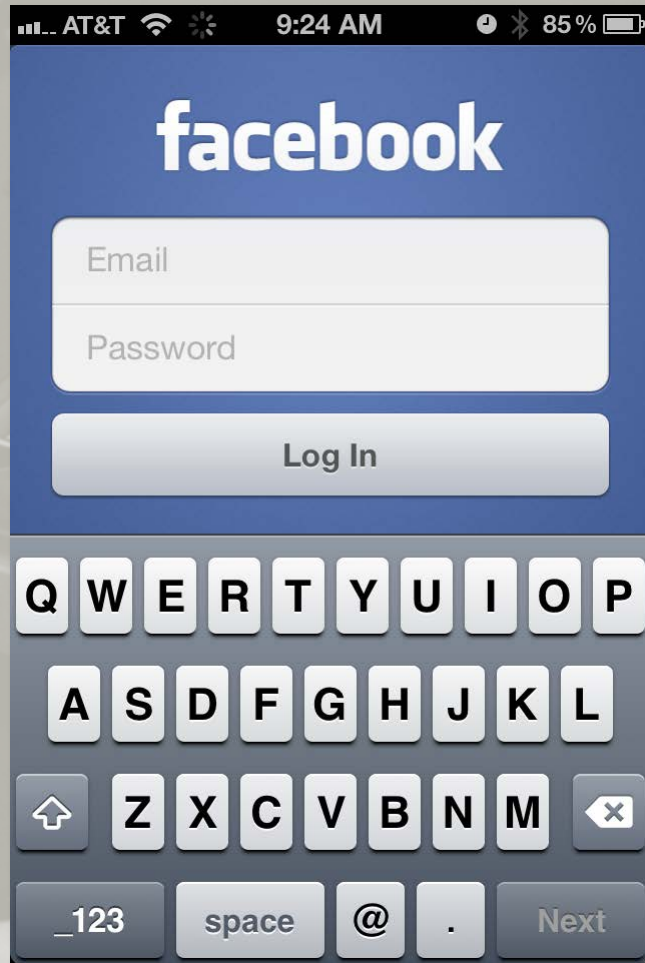It works, but it detracts from the immersive experience

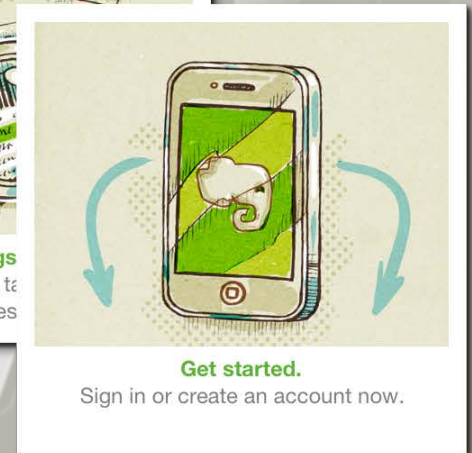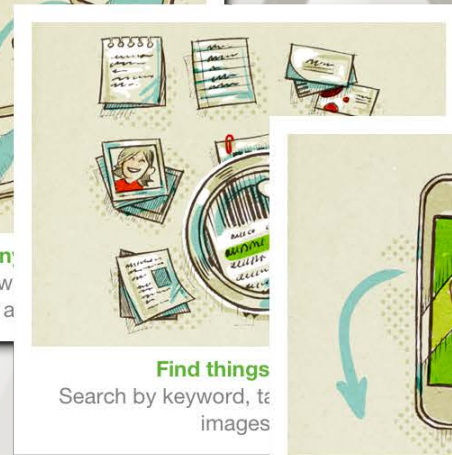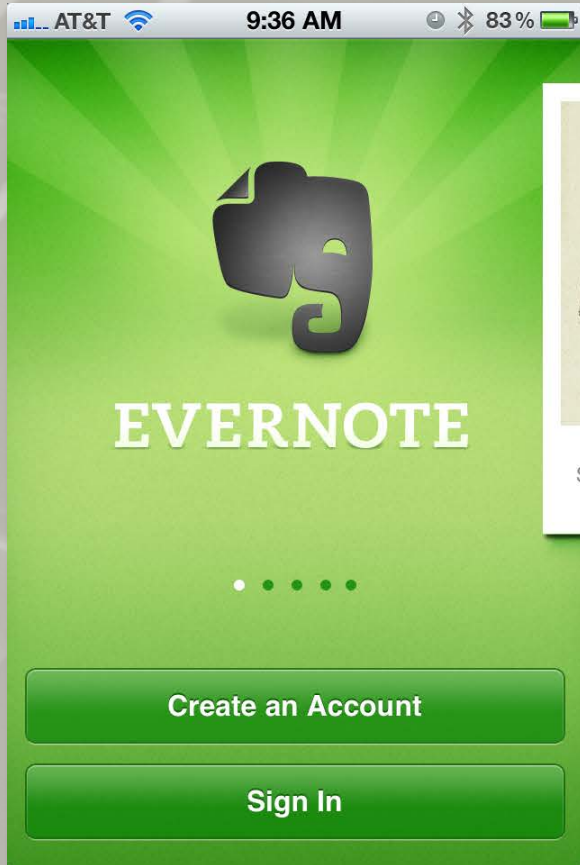# A Better Method: Appropriate Notifications

Keep the message as passive as the situation merits

The Bouncer
Providing Value only for Registered Users
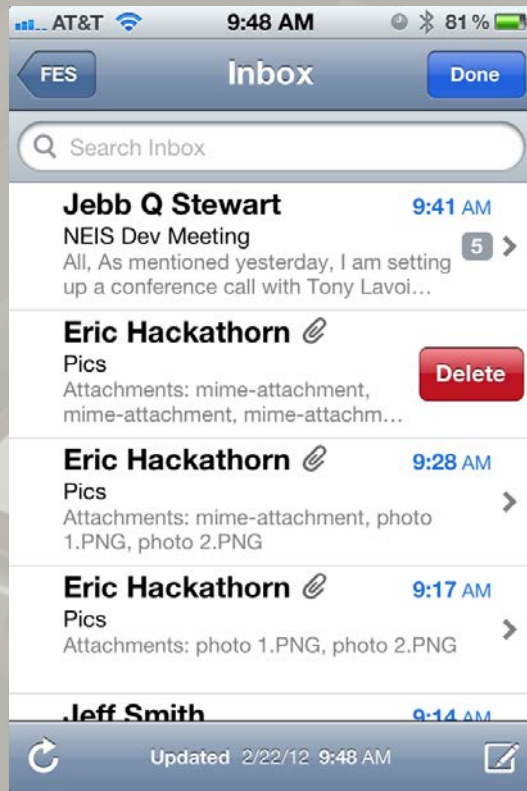
A Better Method

- If possible, allow users to register from the application
- Provide value and information
- Reward installs instead of penalizing or stonewalling

Gesture Hijacking
Using Established Gestures for Novel Behavior in a Single Application

- Gestures are learned, not genetically bestowed
- Hijacking learned gestures creates uncertainty, hindering the user experience

A Better Method: Details View

- Crowded interfaces are a real challenge
- Novel gestures might seem like a solution, but they too must be learned

Summary

- UI anti-patterns are not bugs, dumb features, or bad code
- Best practices (and the worst) are still emerging
- Compete in the App Store, cooperate on the device
- Differentiate, but not at the expense of usability

Questions?