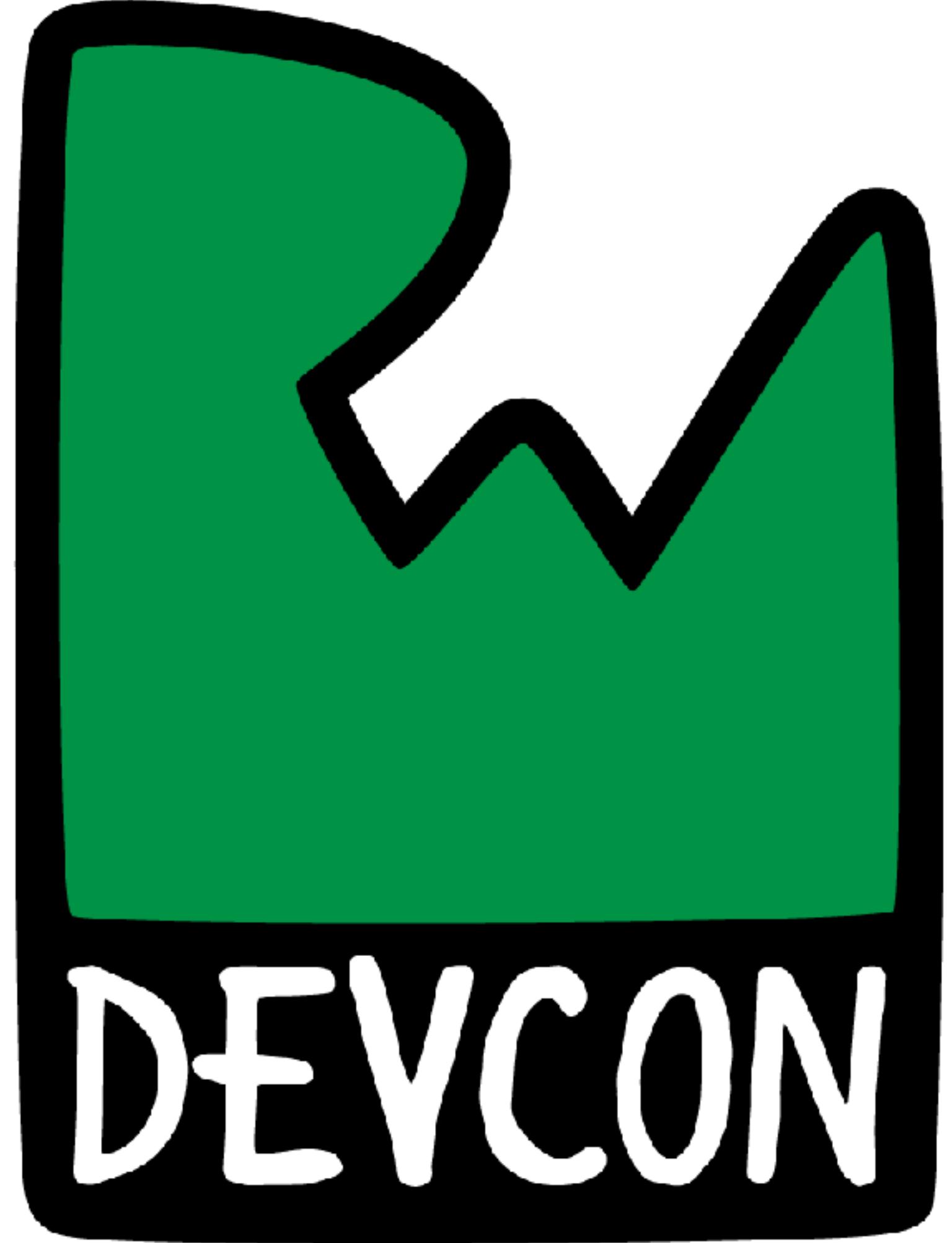


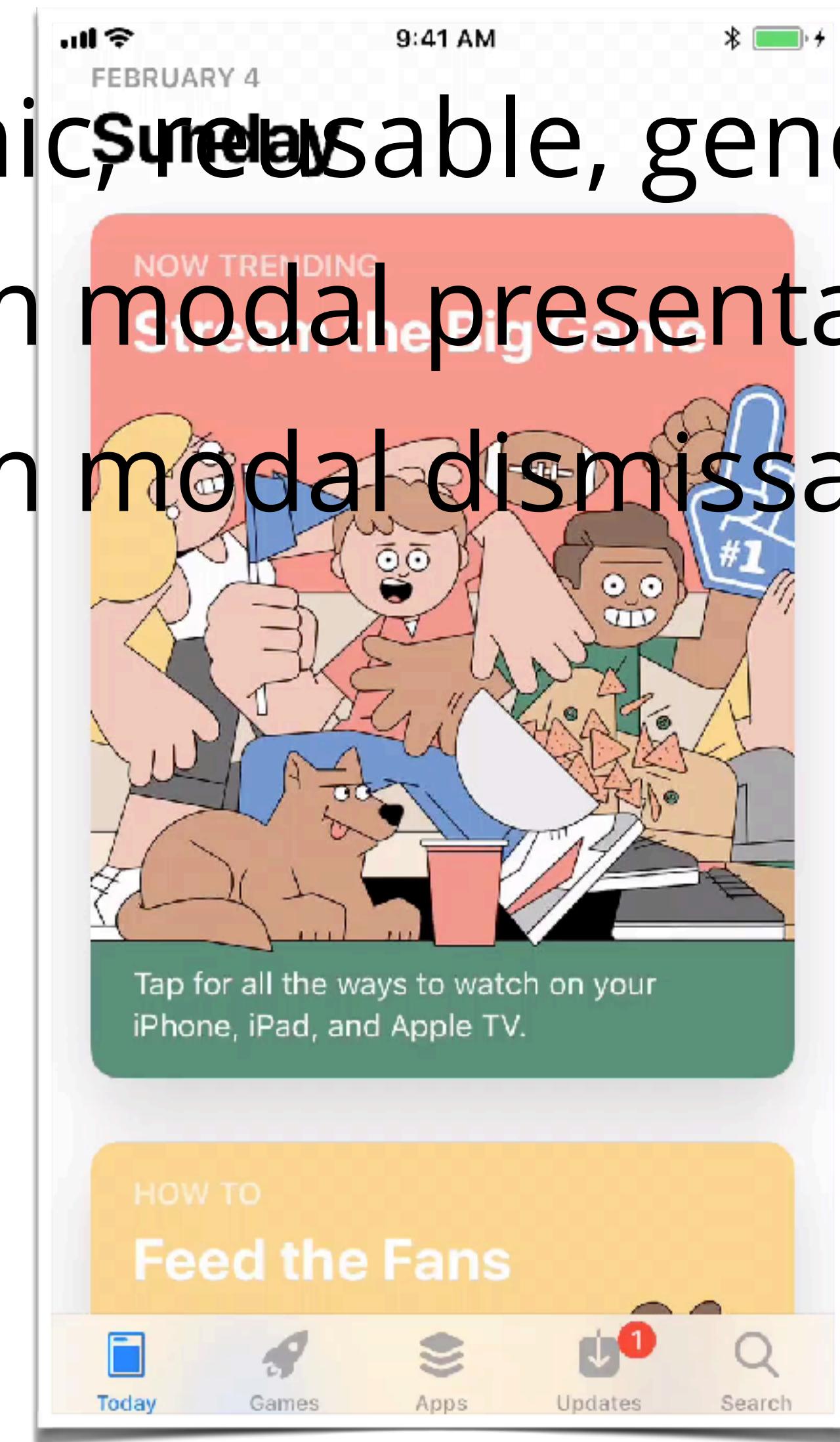
Session 14: Custom Views



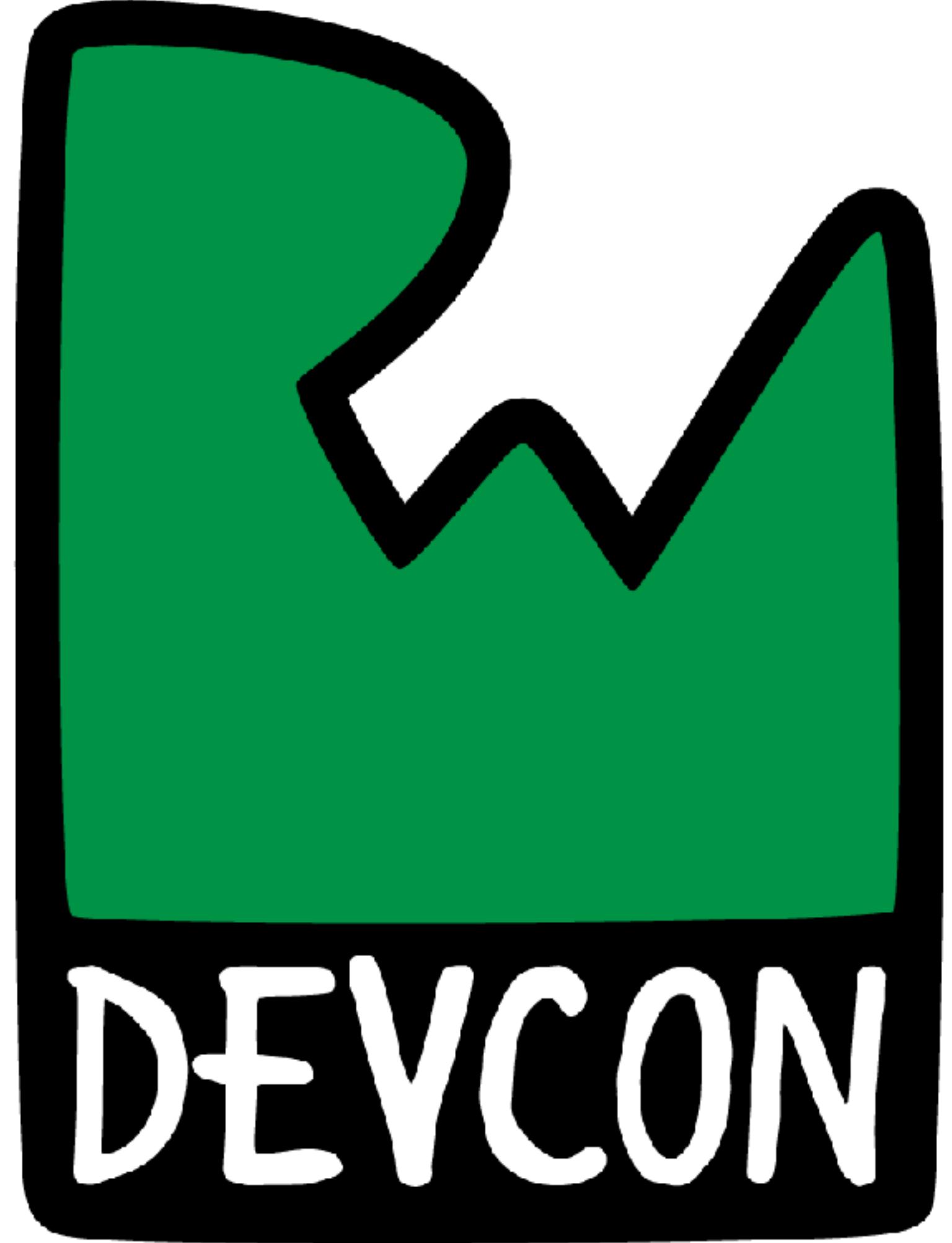
TUTORIAL OVERVIEW

OVERVIEW

- ▶ **Demo 1:** Dynamic, reusable, generic
- ▶ **Demo 2:** Custom modal presentation
- ▶ **Demo 3:** Custom modal dismissal



Session 14: Custom Views



DYNAMIC, REUSABLE, GENERIC

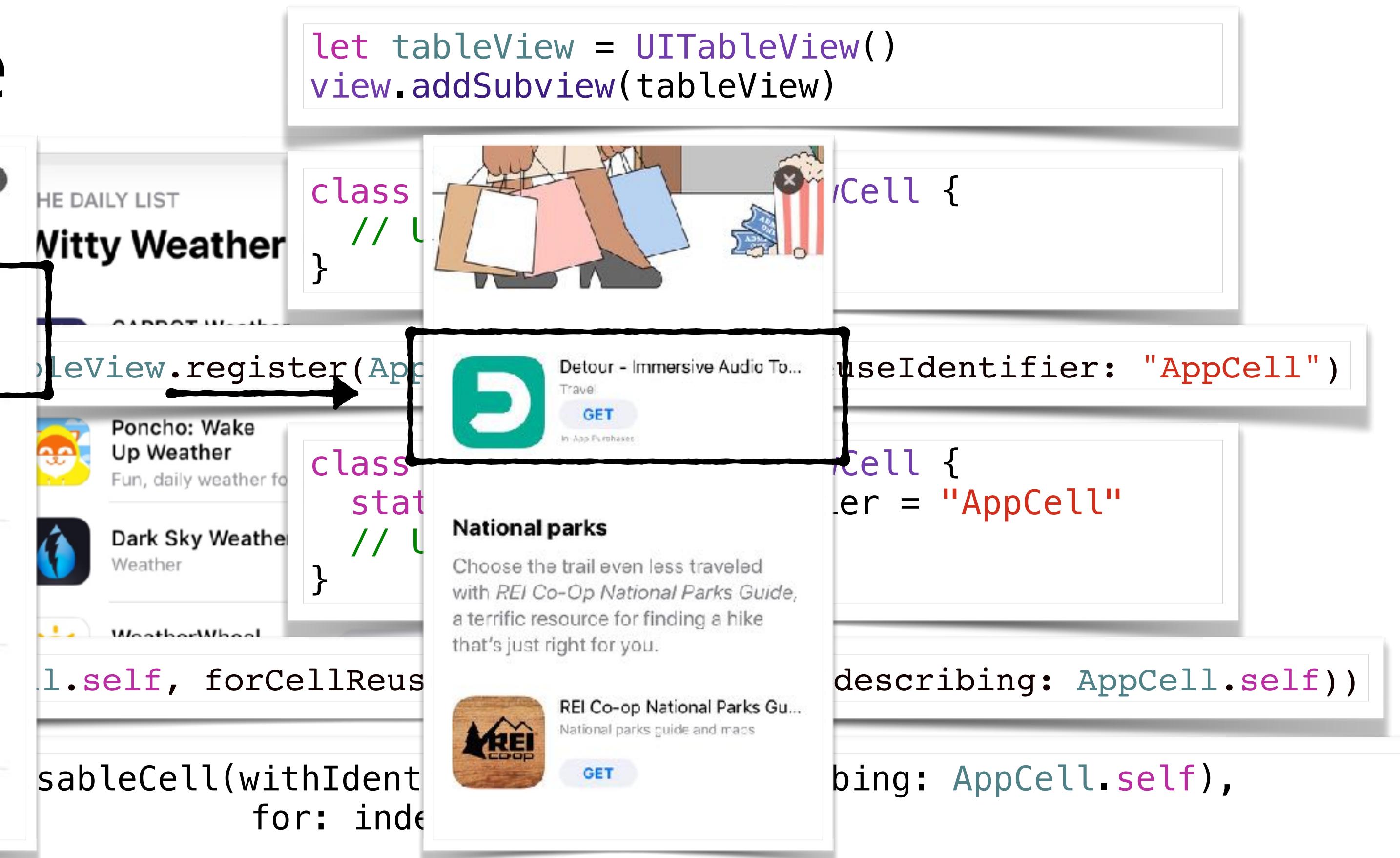
CUMBERSOME UITABLEVIEWS

- ▶ UITableViews and UITableViewCellCells are great
- ▶ But cumbersome

R
W



```
guard let appCell =  
else { return UITableViewCell() }
```



UITABLEVIEWS + GENERICS = ❤



```
tableView.register(AppCell.self, forCellReuseIdentifier: "AppCell")
```



```
tableView.registerCell(AppCell.self)
```



```
guard let appCell = tableView.dequeueReusableCell(withIdentifier: String(describing: AppCell.self),  
                                              for: indexPath) as? AppCell  
else { return UITableViewCell() }
```



```
let appCell = tableView.dequeueReusableCell(forIndexPath: indexPath) as AppCell
```



UITABLEVIEWS + GENERICS = ❤

► Abstracted data types

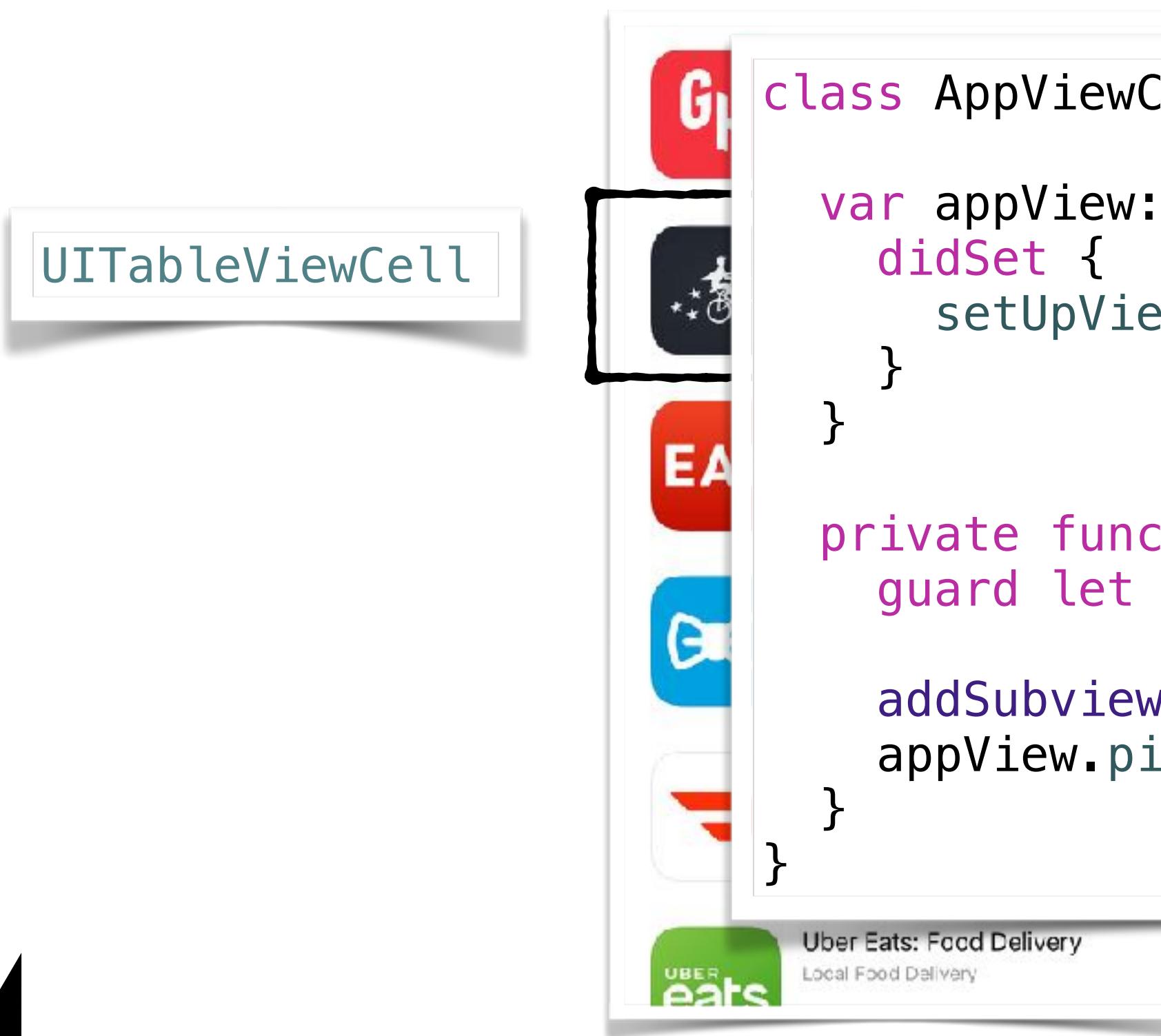
```
func intMin(a: Int, b: Int) -> Int {  
    return a < b ? a : b  
  
extension UITableView {  
    public func registerCell<Cell: UITableViewCell>(_ cellClass: Cell.Type) {  
        // ~Make some magic to register the cell~  
    }  
  
    public func dequeueReusableCell<Cell: UITableViewCell>(forIndexPath indexPath: IndexPath) -> Cell {  
        // ~Make some magic to dequeue the cell~  
    }  
}
```

```
func genericMin(a: String, b: String) -> String {  
    let string = genericMin(a: "swift", b: "objc") // objc  
    return string  
}  
  
let appCell = tableView.dequeueReusableCell(forIndexPath: indexPath) as AppCell
```



UITABLEVIEWS + GENERICS = ❤

► Dummy UITableViewCell?



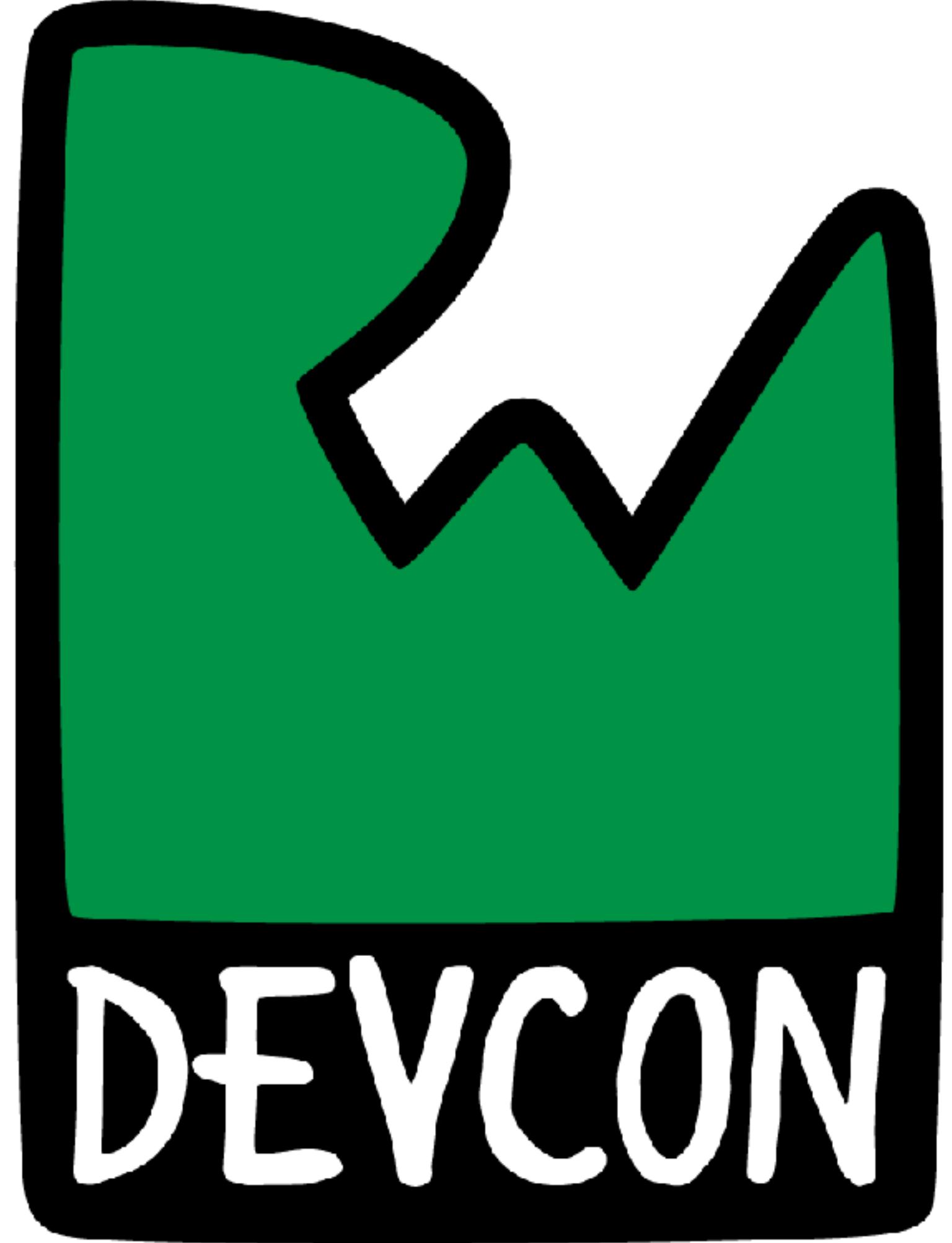
```
class AppTableViewCell: UITableViewCell {  
    var appView: AppView?  
    didSet {  
        setUpViews()  
    }  
    private func setUpView()  
    guard let appView = appView else { return }  
    addSubview(appView)  
    appView.pinEdgesToSuperview()  
}  
  
class GenericTableViewCell<View: UIView>: UITableViewCell {  
    // ~Make some magic to register the cell~  
}  
  
tableView.registerCell(GenericTableViewCell<AppView>.self)  
tableView.registerCell(GenericTableViewCell<CardView>.self)  
tableView.registerCell(GenericTableViewCell<AnyView>.self)
```

Uber Eats: Food Delivery
Local Food Delivery

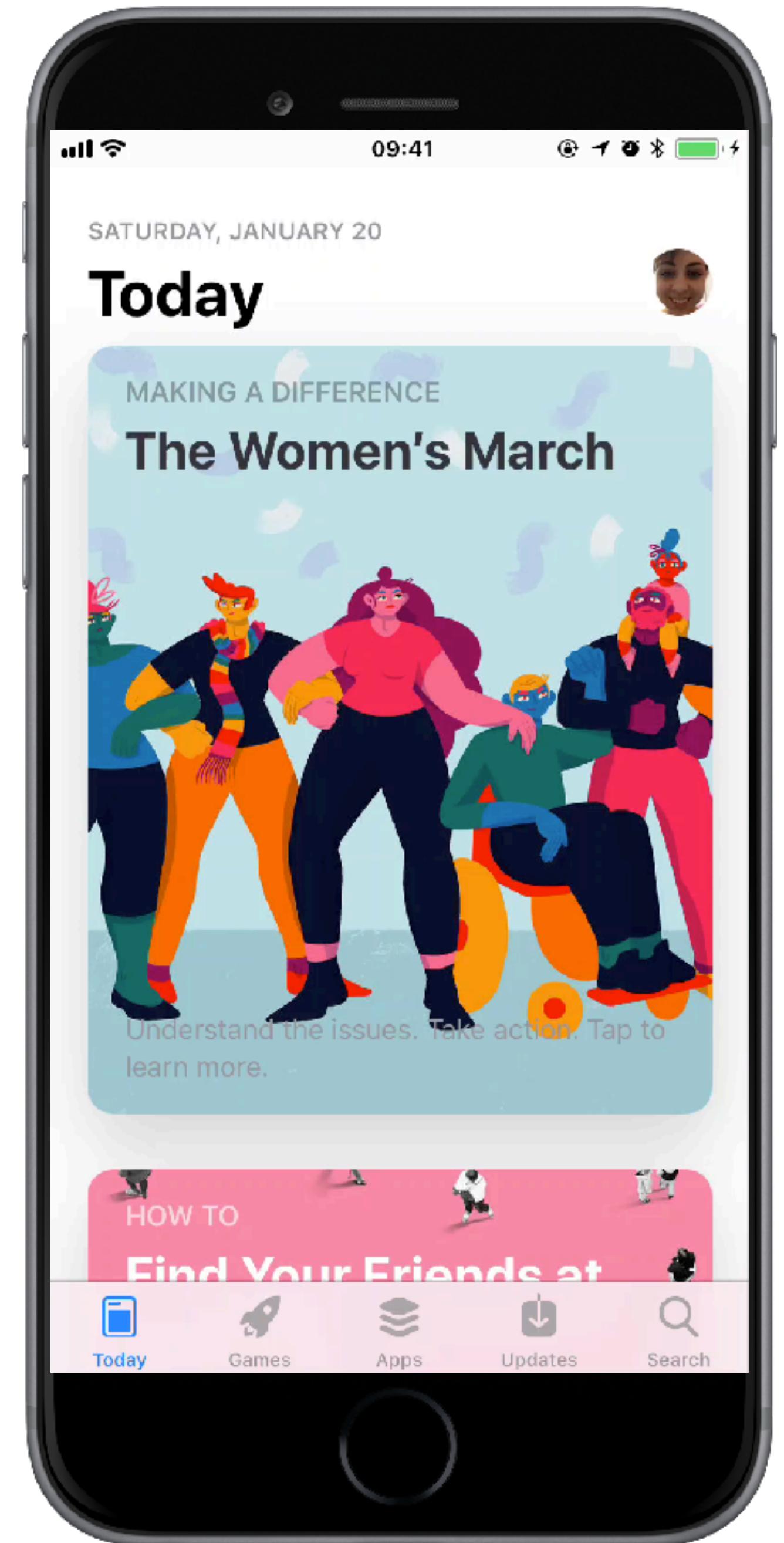
DEMO 1



Session 14: Custom Views



CUSTOM PRESENTATION



R
W

MODAL PRESENTATIONS

- ▶ Traditional modal view controller presentation

```
let detailViewController = DetailViewController(cardViewModel: cardViewModel)  
present(detailViewController, animated: true, completion: nil)
```

- ▶ “Advanced” presentation

```
let detailViewController = DetailViewController(cardViewModel: cardViewModel)  
detailViewController.modalTransitionStyle = .crossDissolve  
detailViewController.modalPresentationStyle = .overFullScreen  
present(detailViewController, animated: true, completion: nil)
```



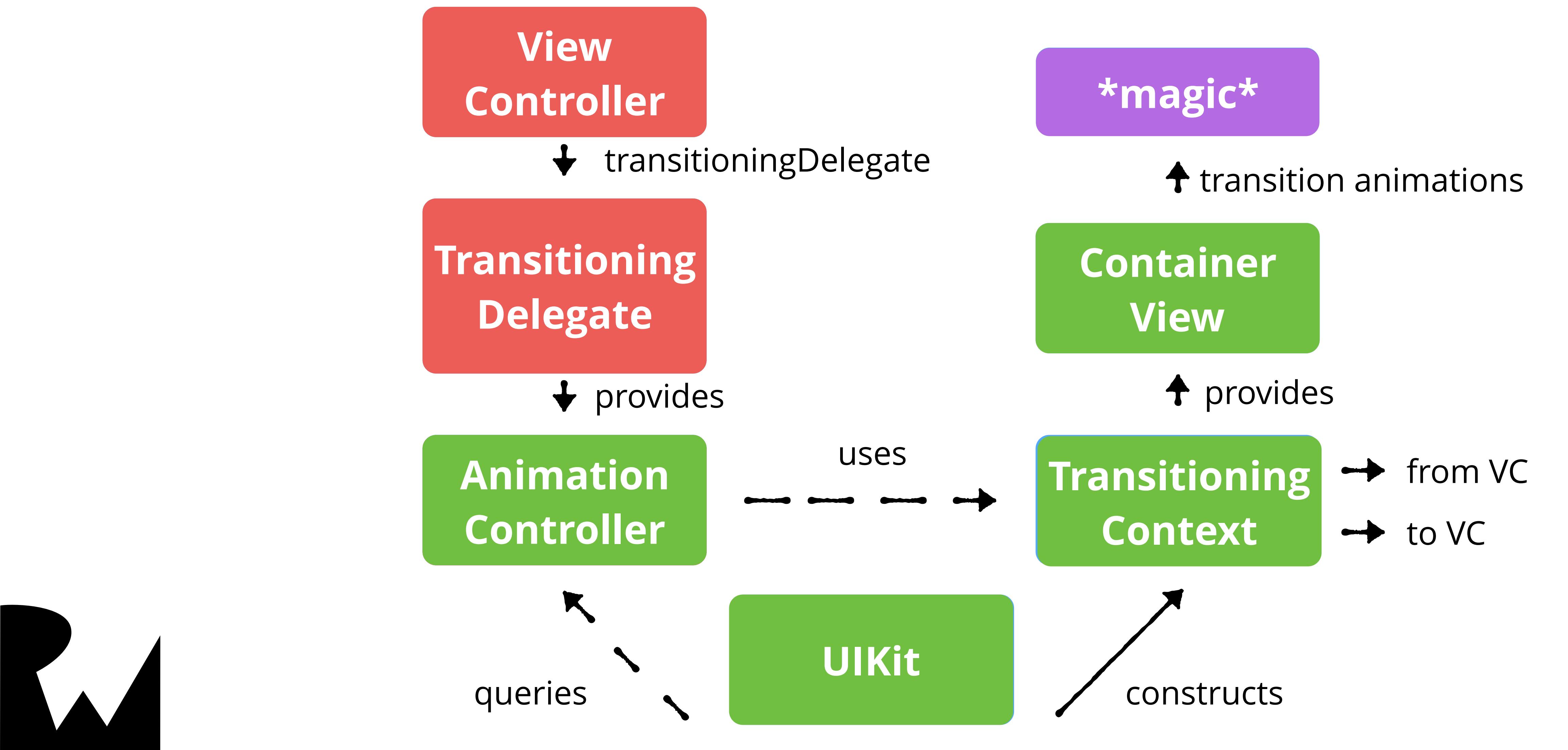
CUSTOM MODAL PRESENTATIONS

- ▶ Presentation transitions can also be completely custom

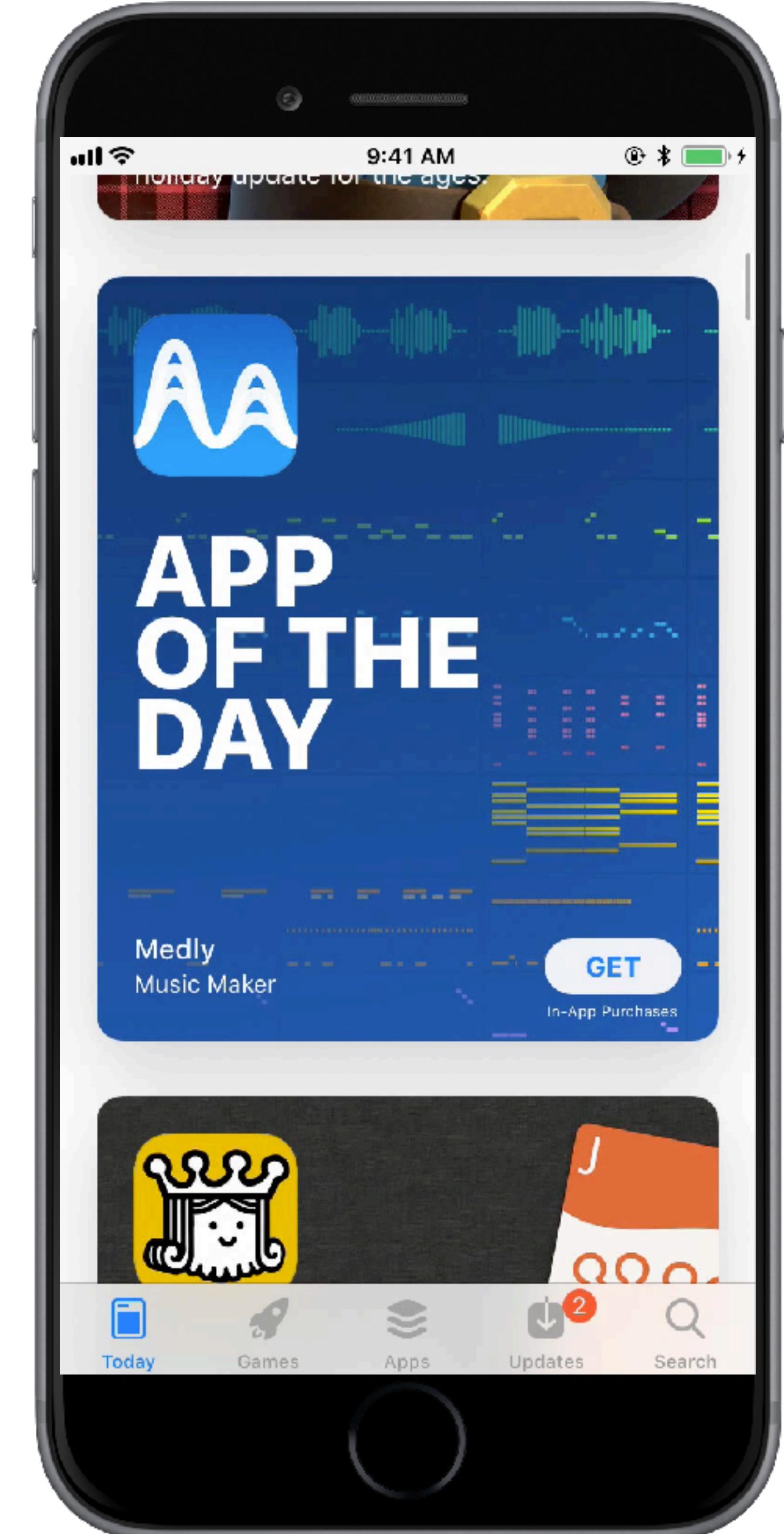
```
class CustomTransitioning: NSObject, UIViewControllerAnimatedTransitioning {  
    func transitionDuration(using transitionContext: UIViewControllerContextTransitioning?) -> TimeInterval {  
        return 0.5  
    }  
  
    let detailViewController = DetailViewController(cardViewModel: cardViewModel)  
  
    detailViewController.transitioningDelegate = transitionManager.oning {  
        let containerView = transitionContext.containerView  
  
        present(detailViewController, animated: true, completion: nil)  
        let key = transitionContext.viewController(forKey: .to)  
  
        transitionContext.completeTransition(true)  
    }  
}
```



CUSTOM MODAL PRESENTATIONS



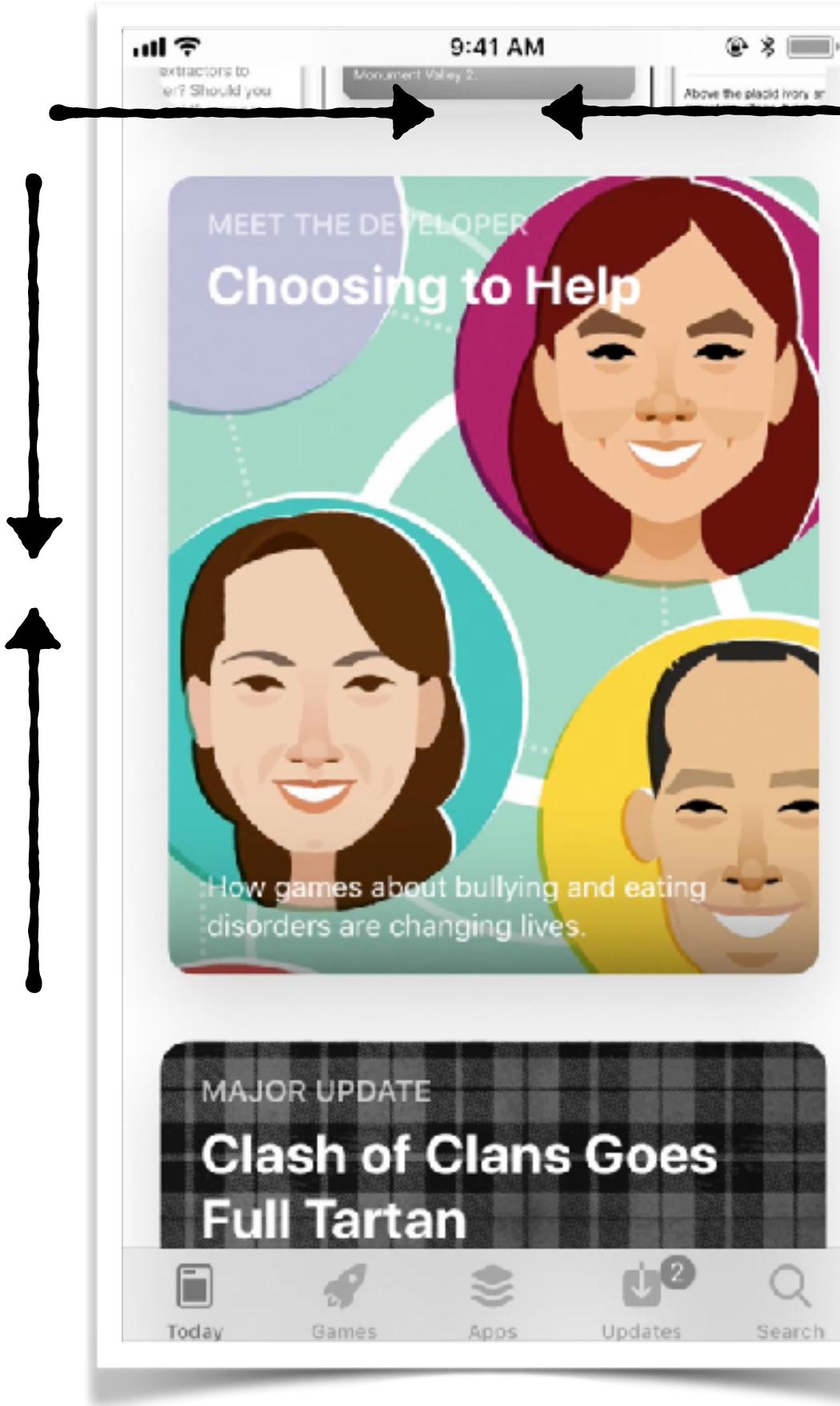
Dissecting the animation



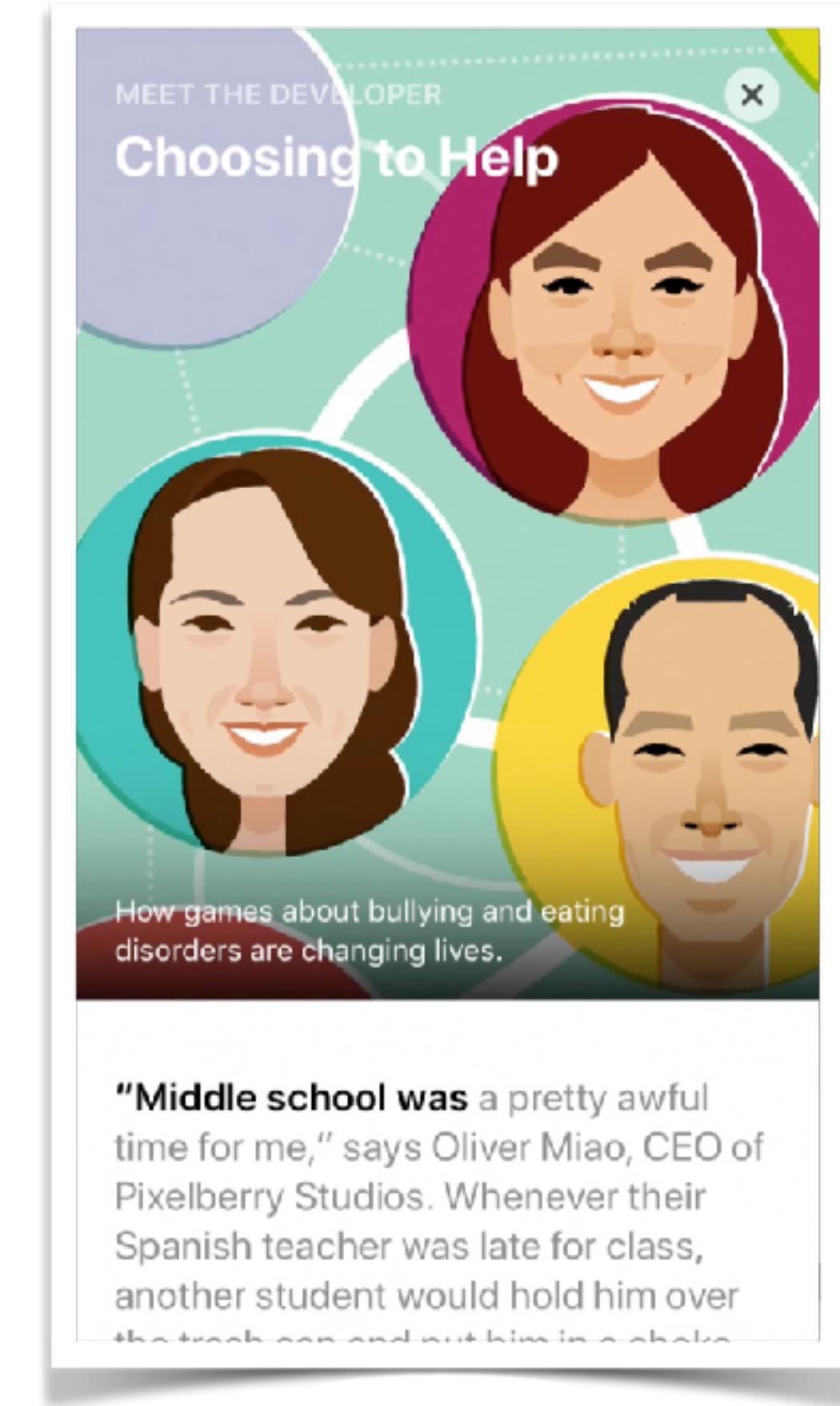
DISSECTING THE ANIMATION



Original



Scale down



\$#%?X



DISSECTING THE ANIMATION



DISSECTING THE ANIMATION

- ▶ Dim the background
- ▶ Blur the background
- ▶ Resize and straighten
- ▶ Unscroll the text

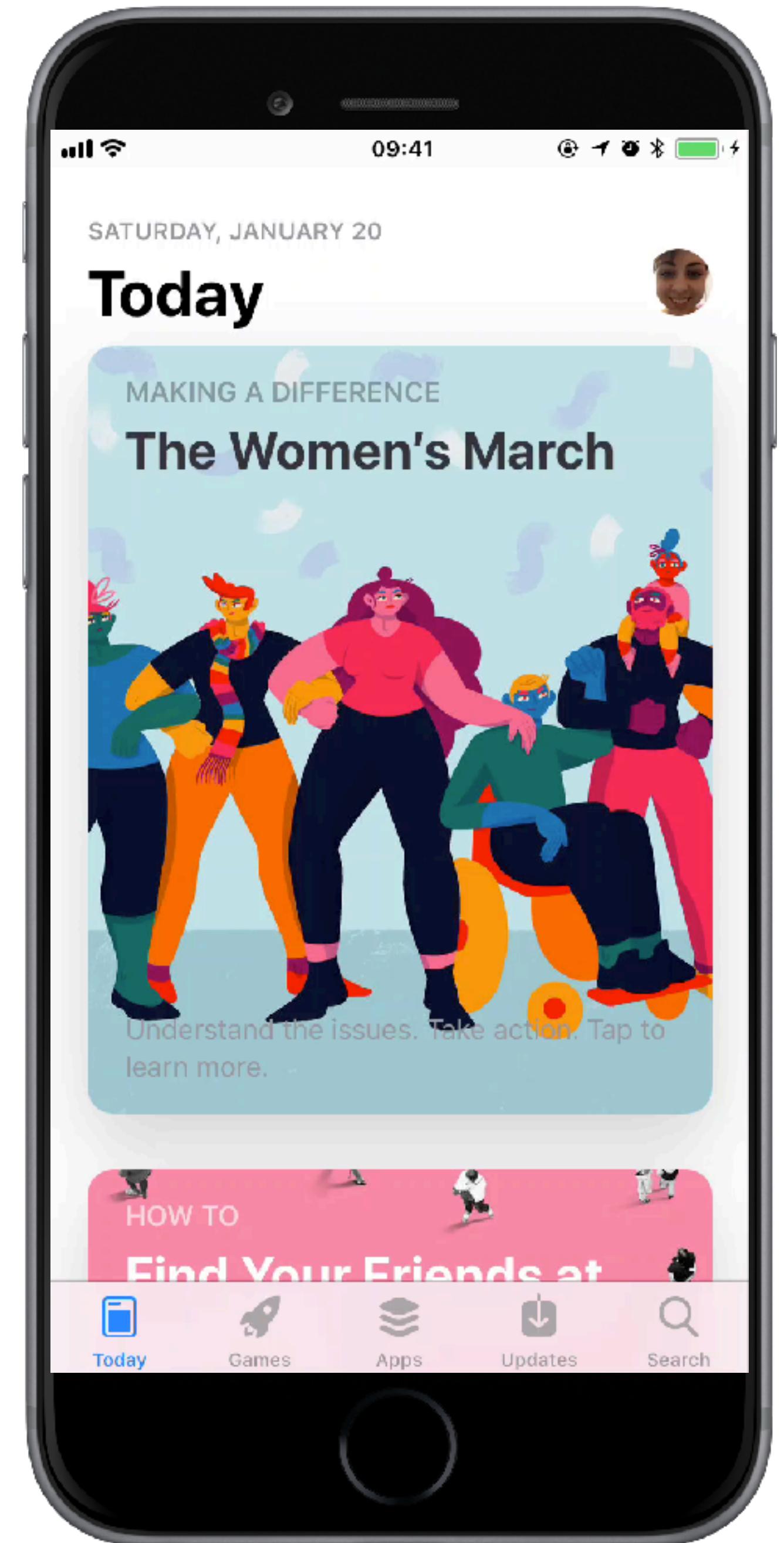


ANATOMY OF THE CARDVIEW



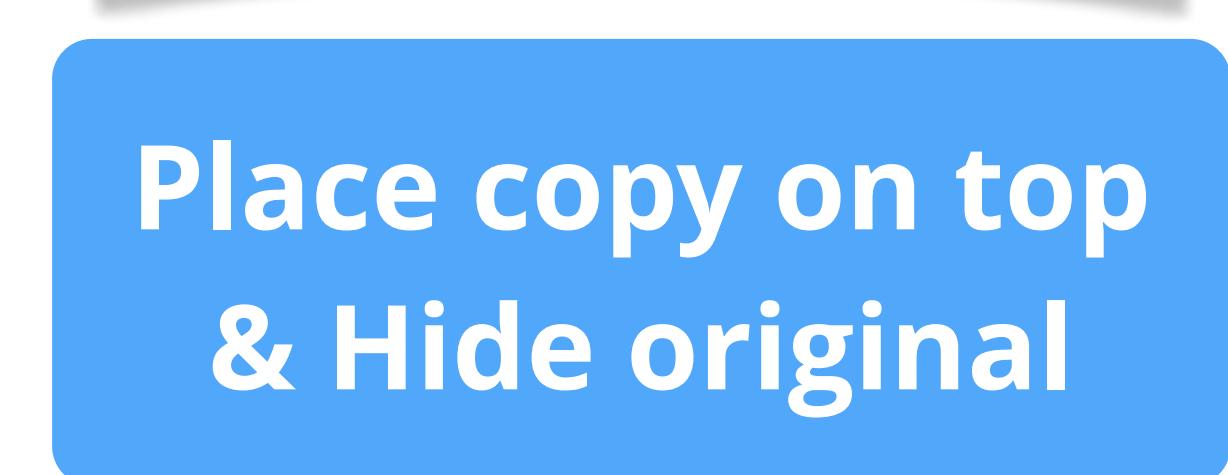
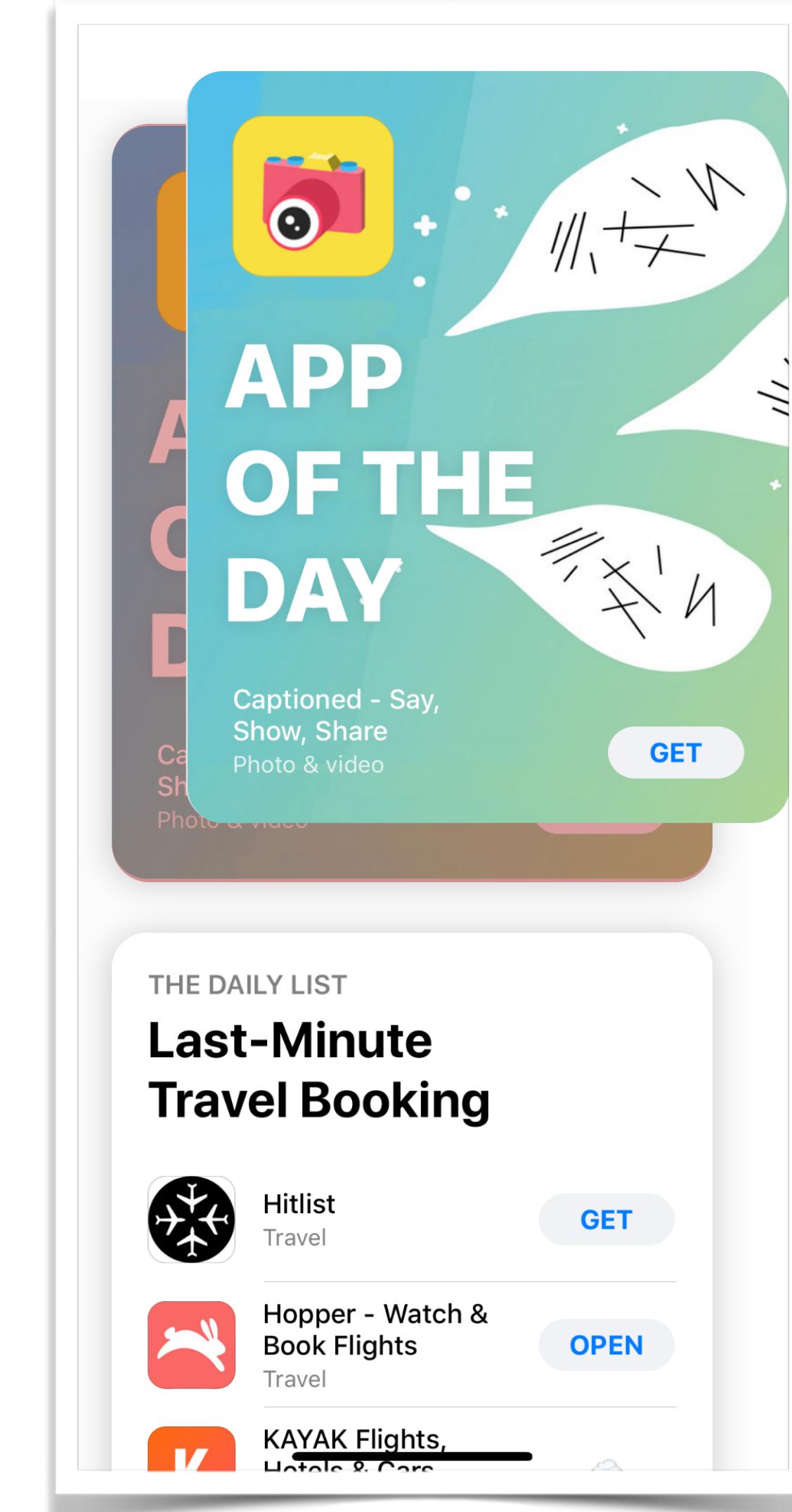
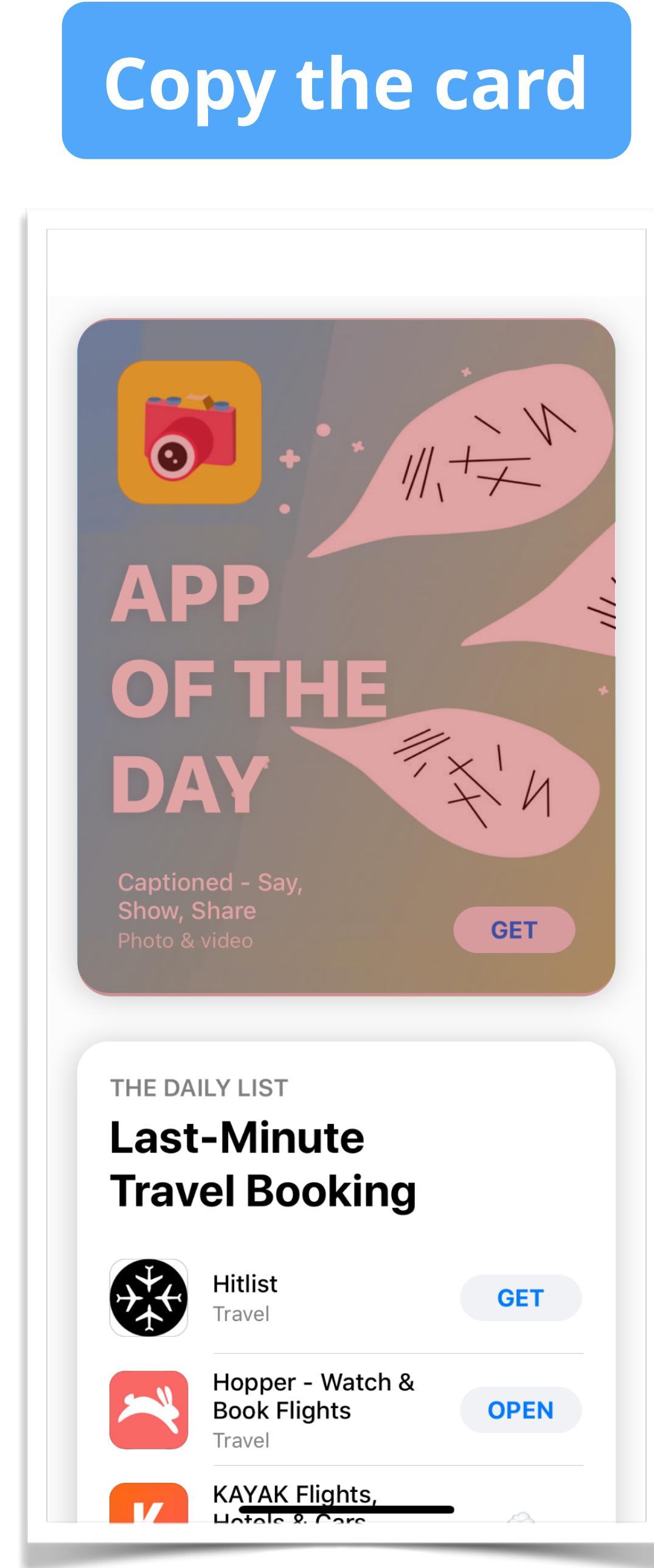
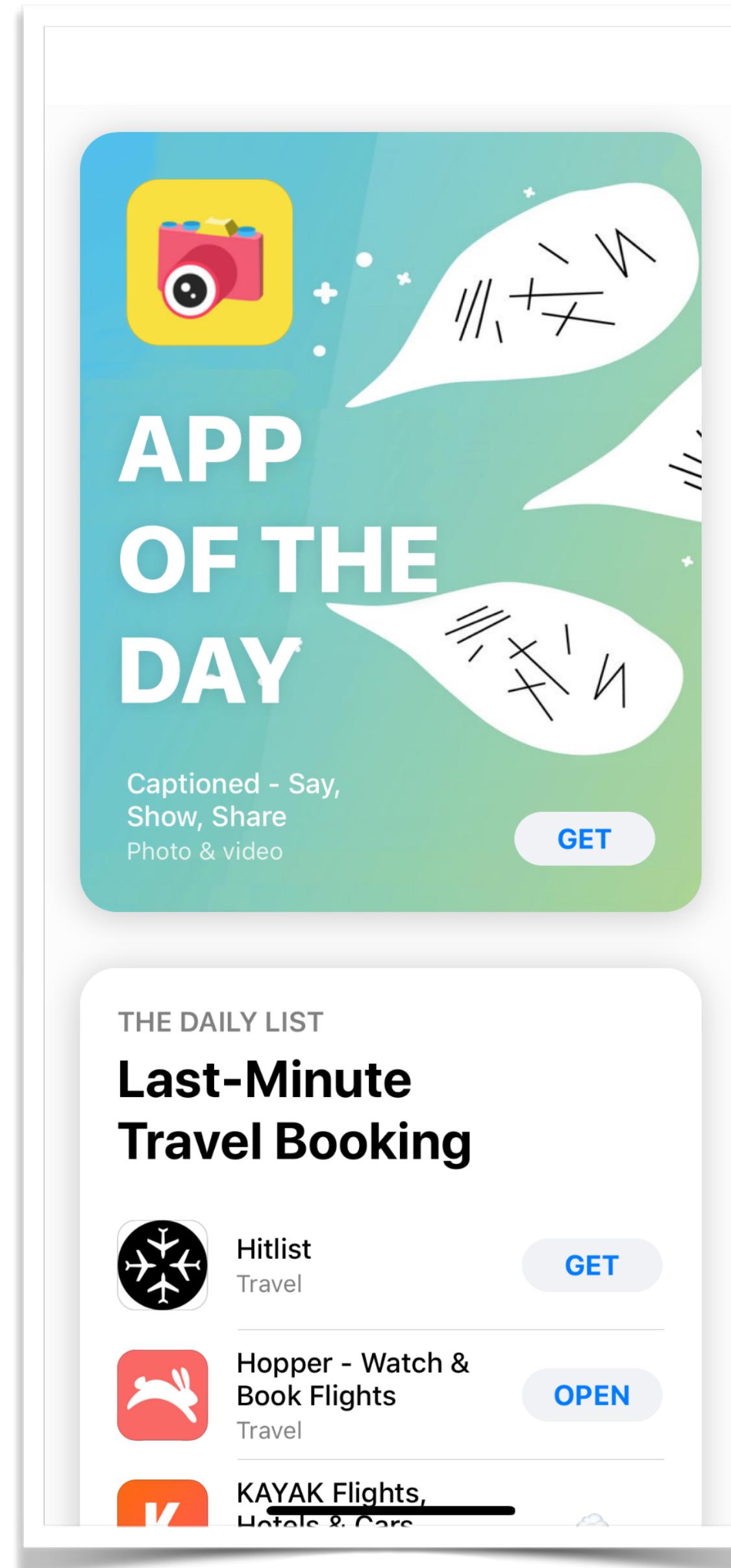
DEMO 2

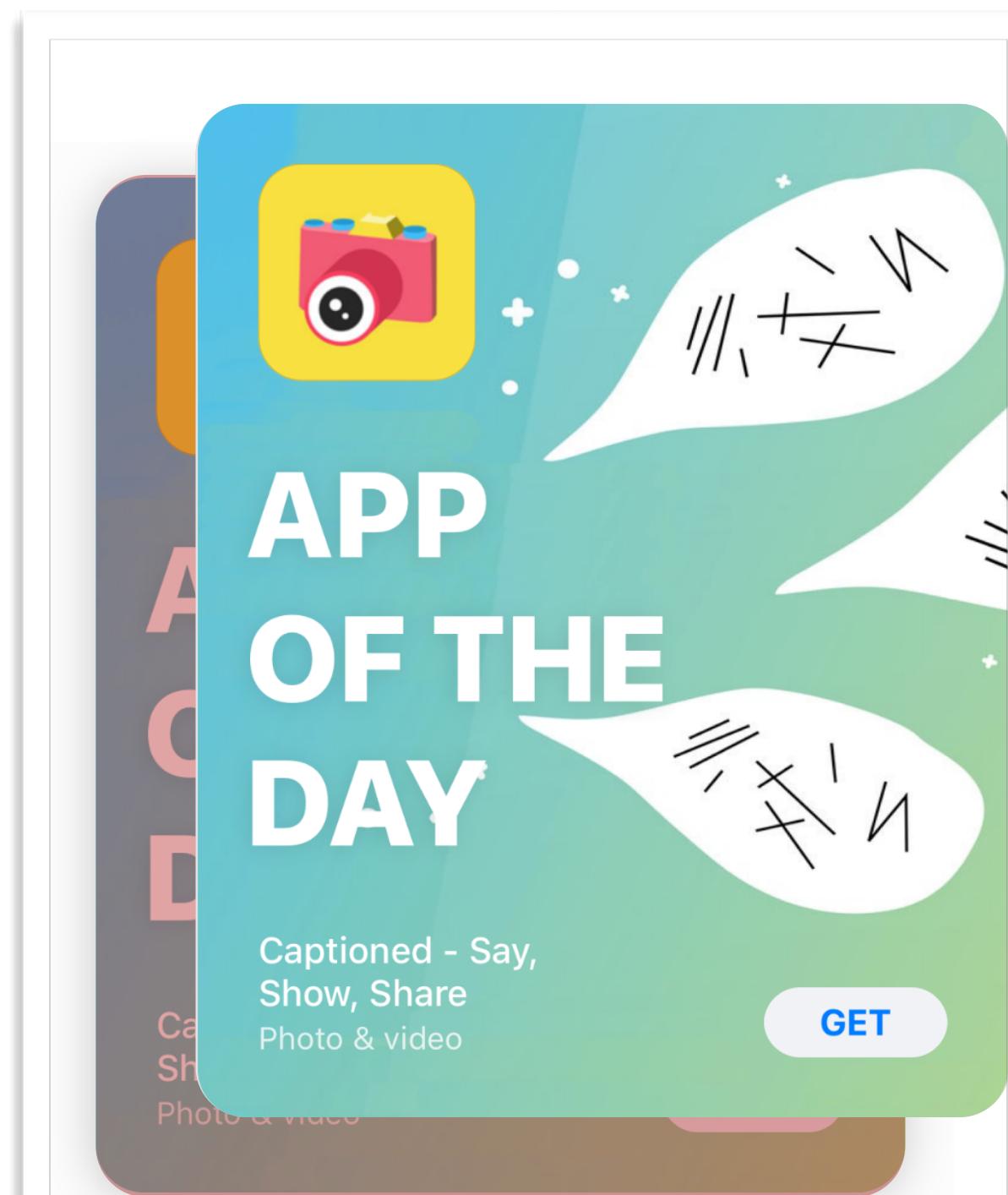




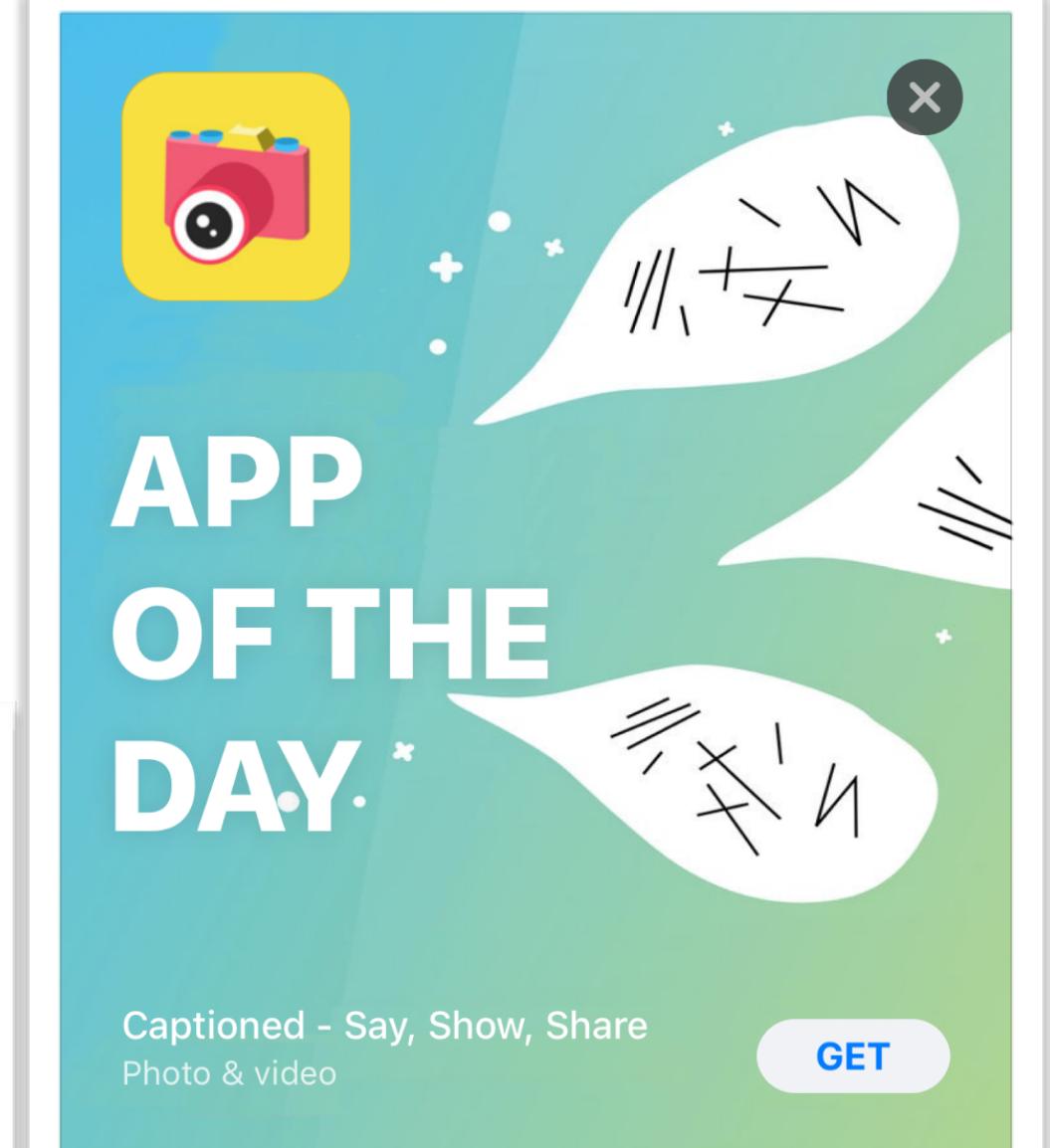
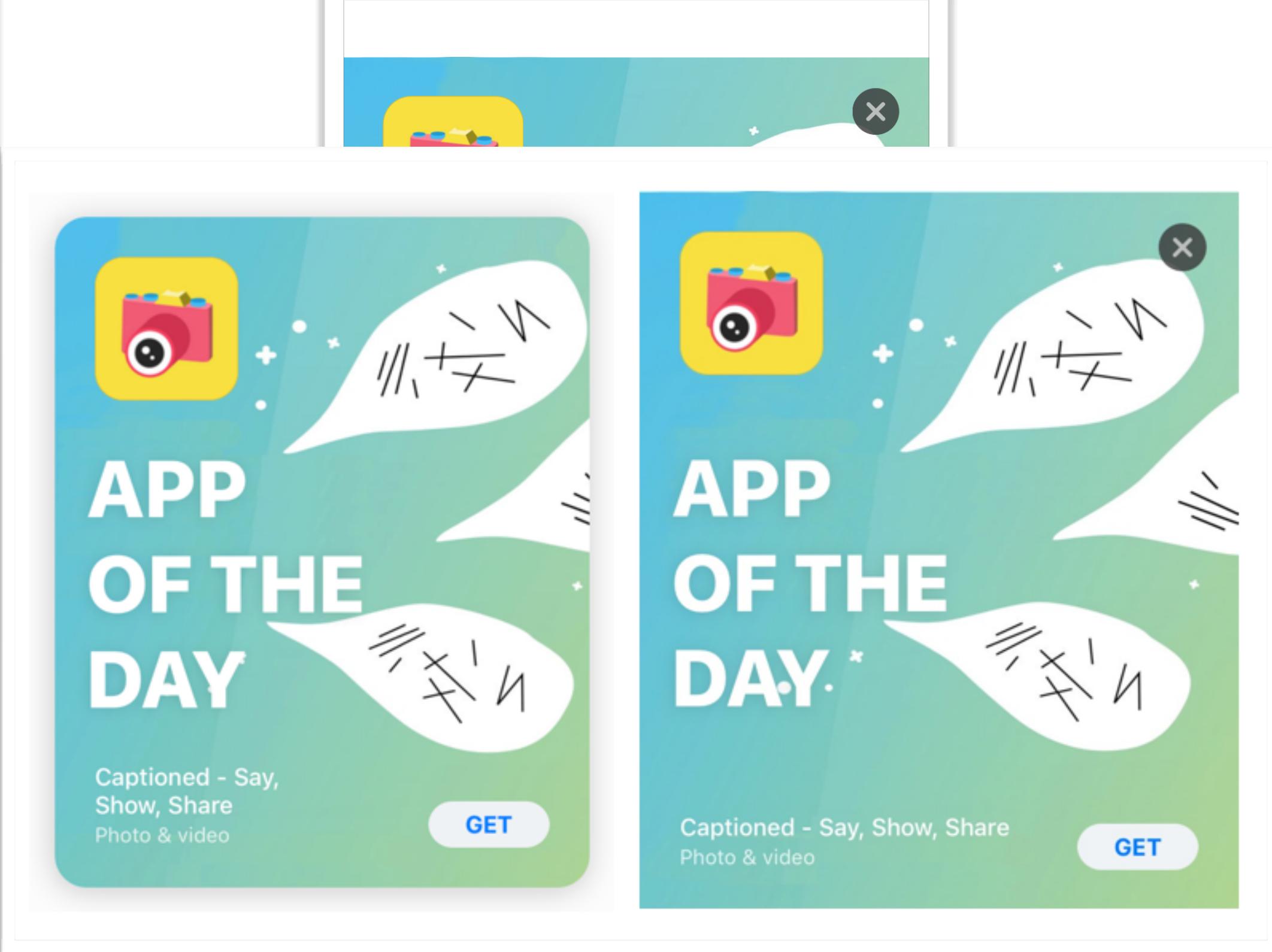
R
W

R
W



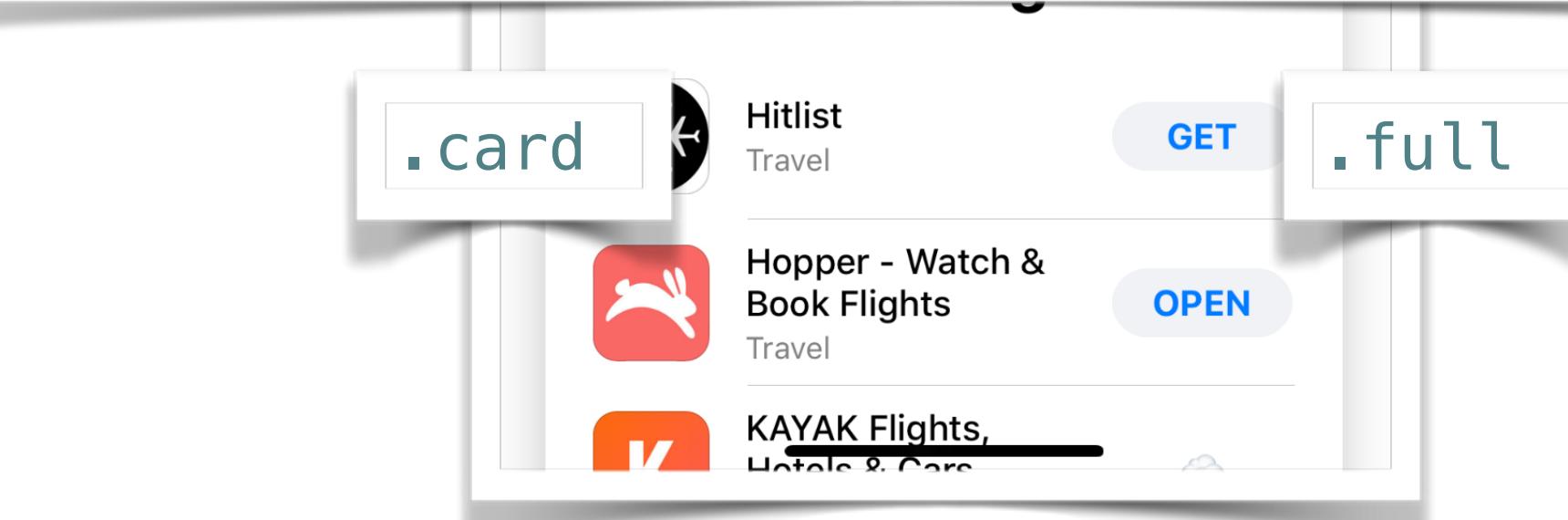


De-round the corners
& pin to top

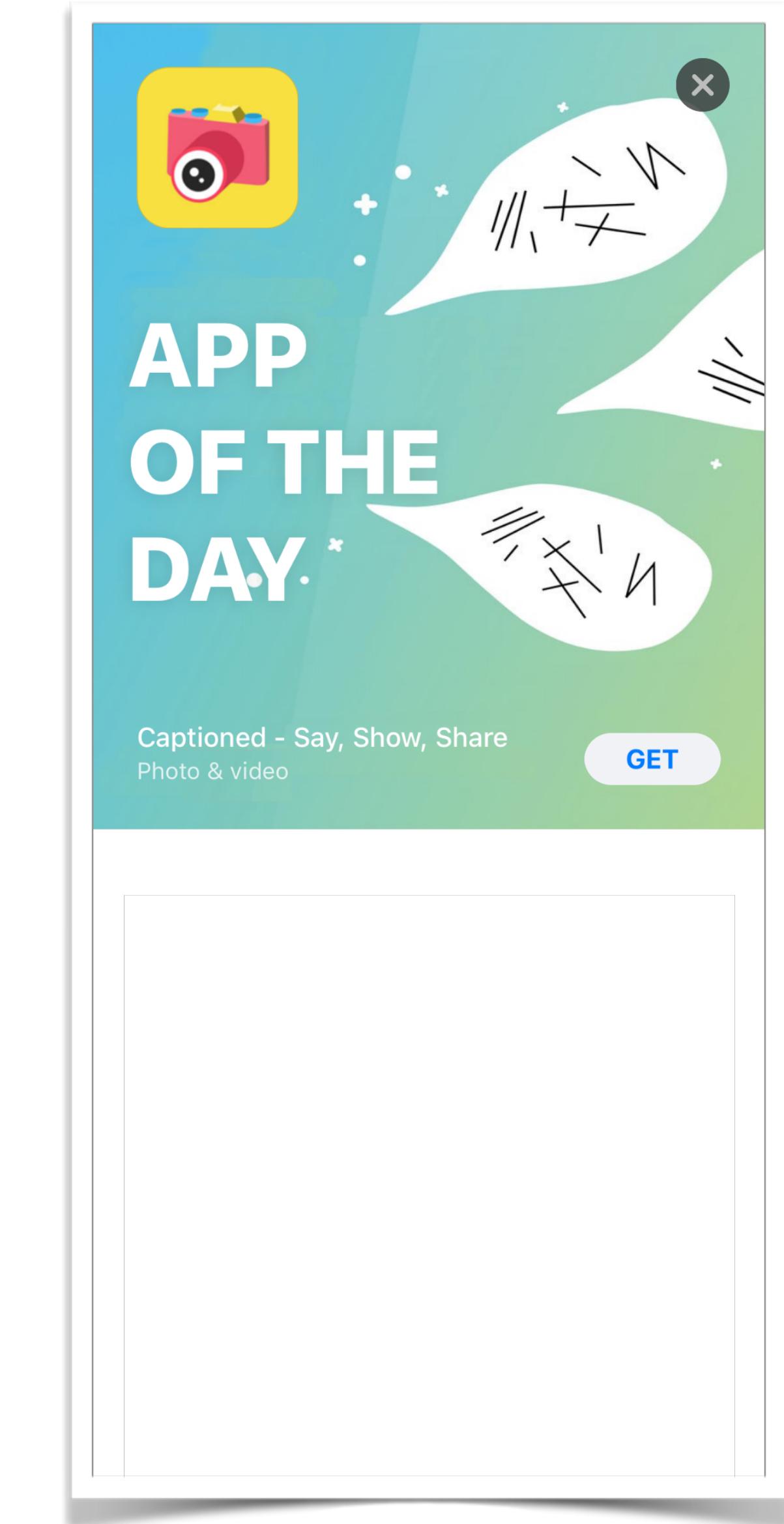
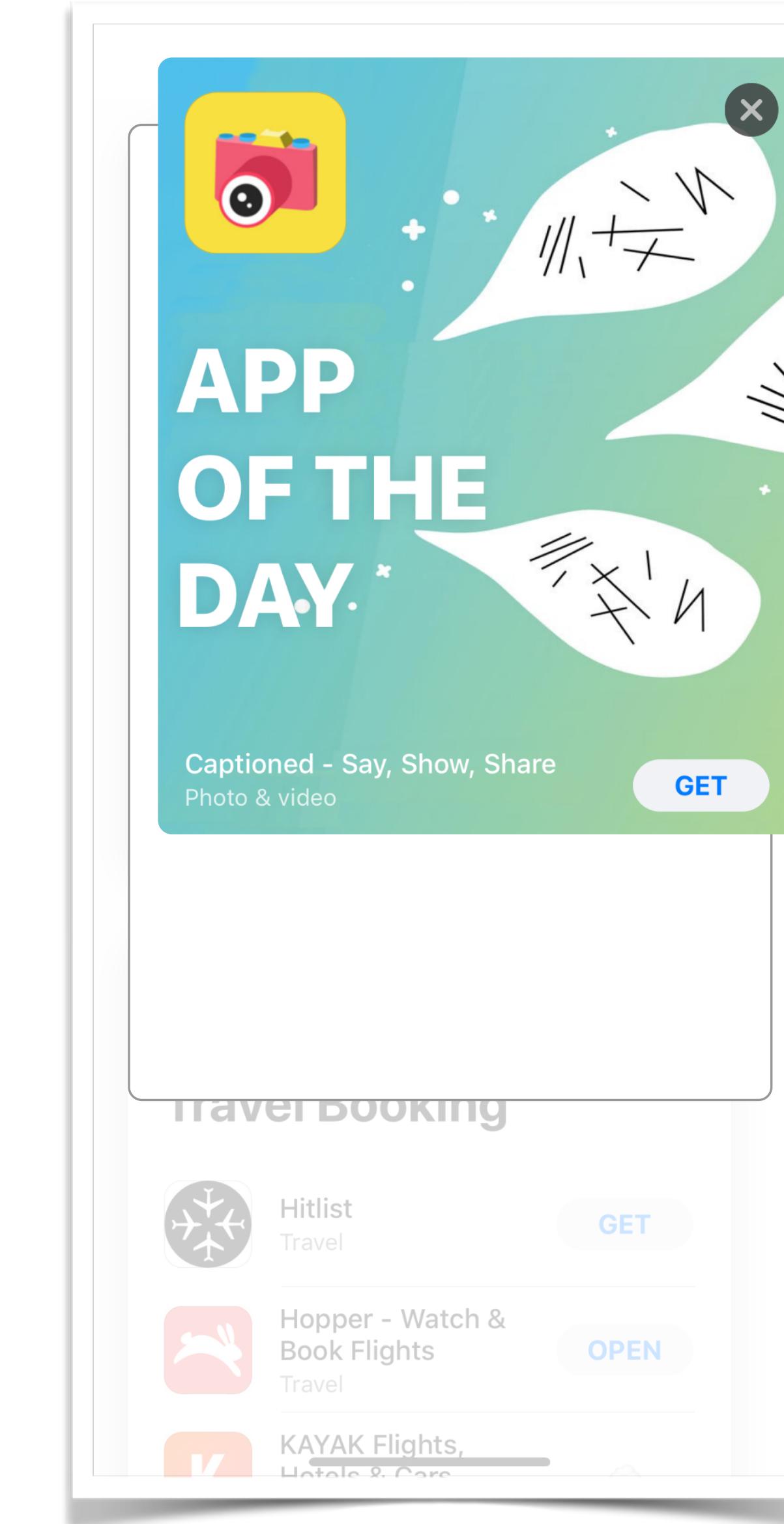
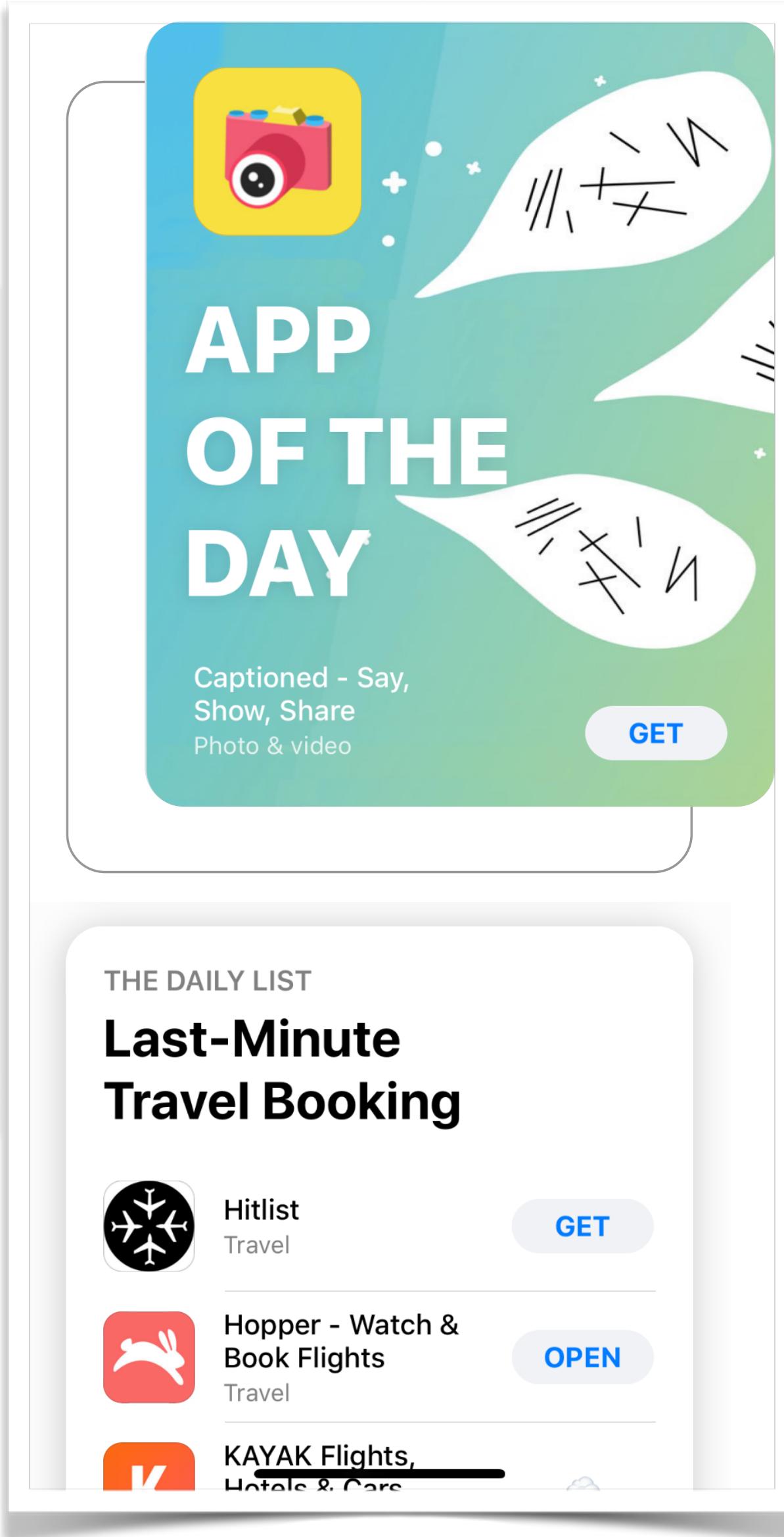
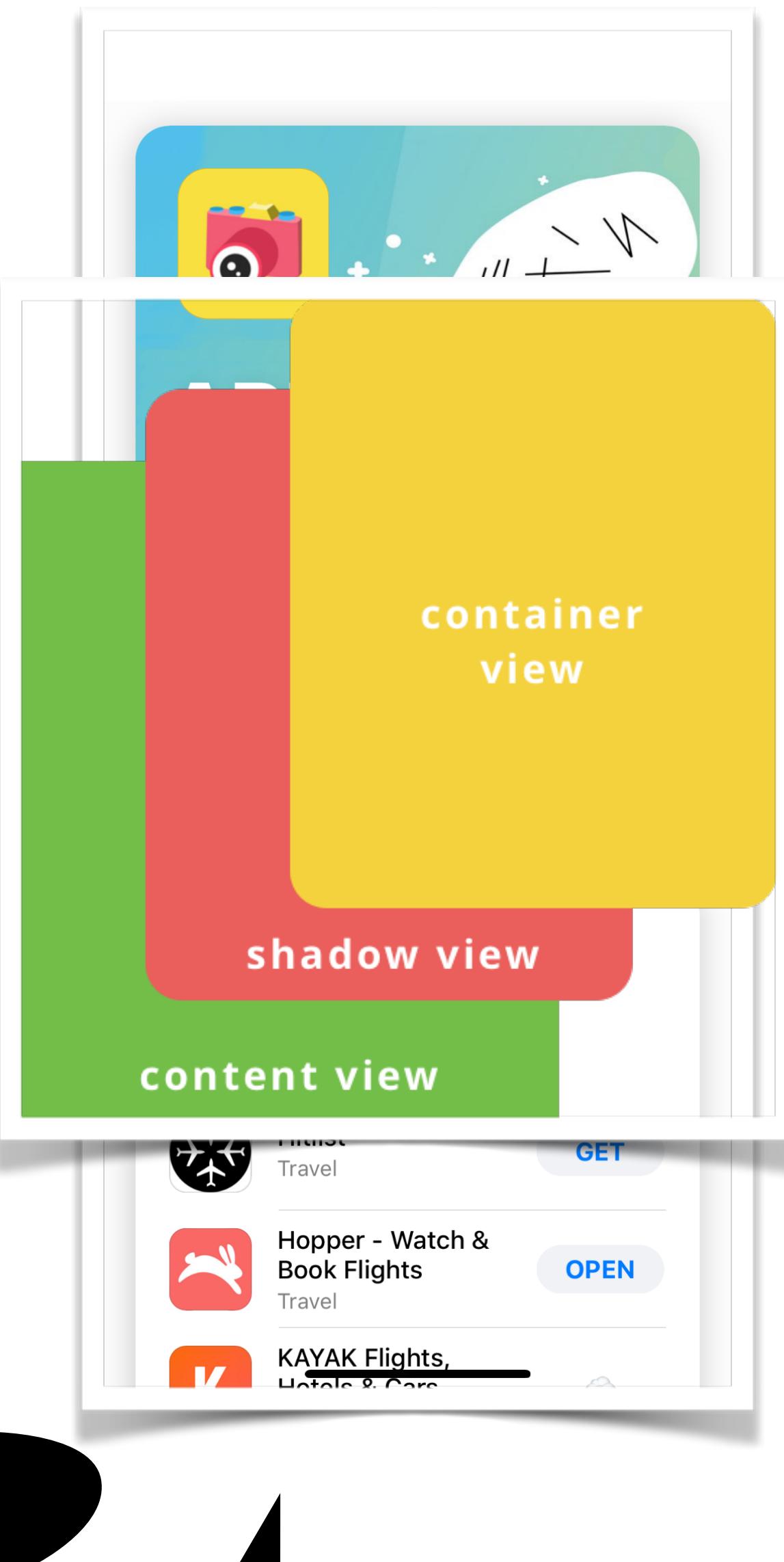


Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset



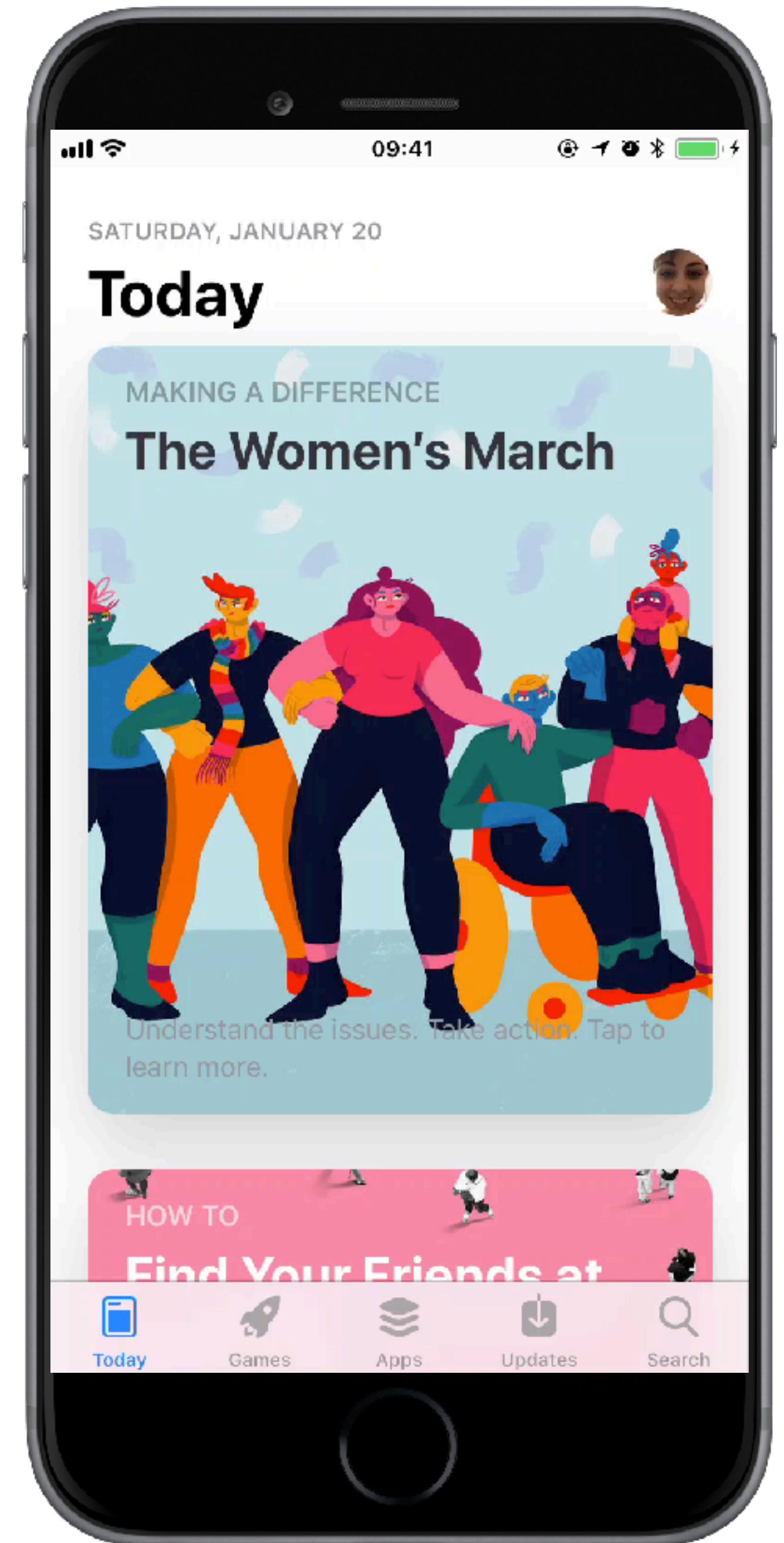
Show the full view
controller



Session 14: Custom Views



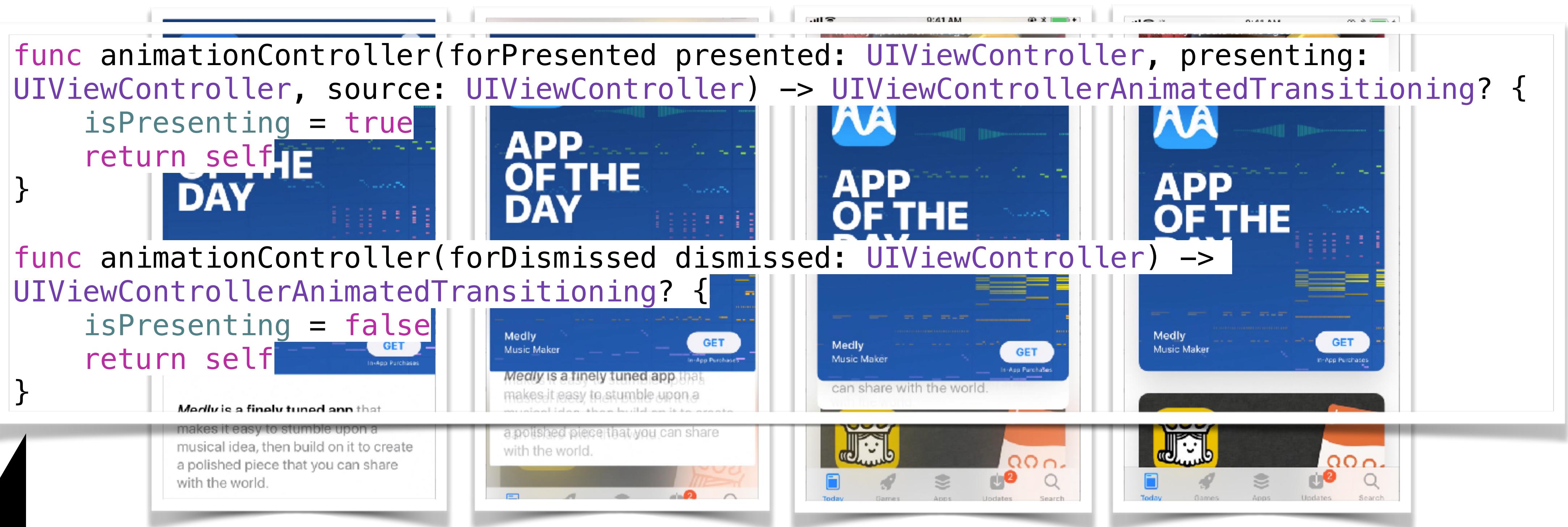
CUSTOM DISMISSAL



R
W

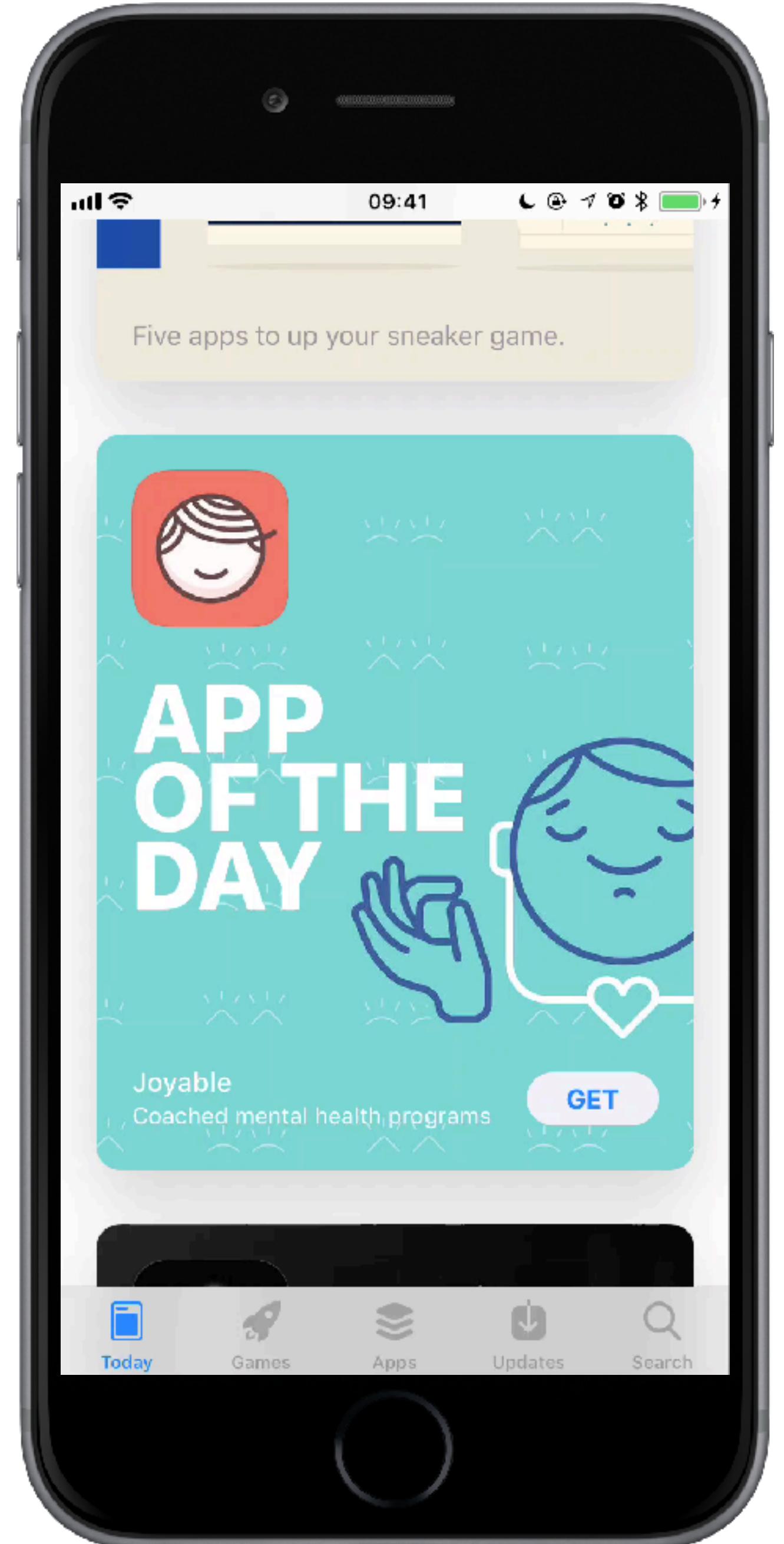
CUSTOM DISMISSAL

- ▶ The opposite of the presentation
- ▶ Reuse the groundwork for dismissal

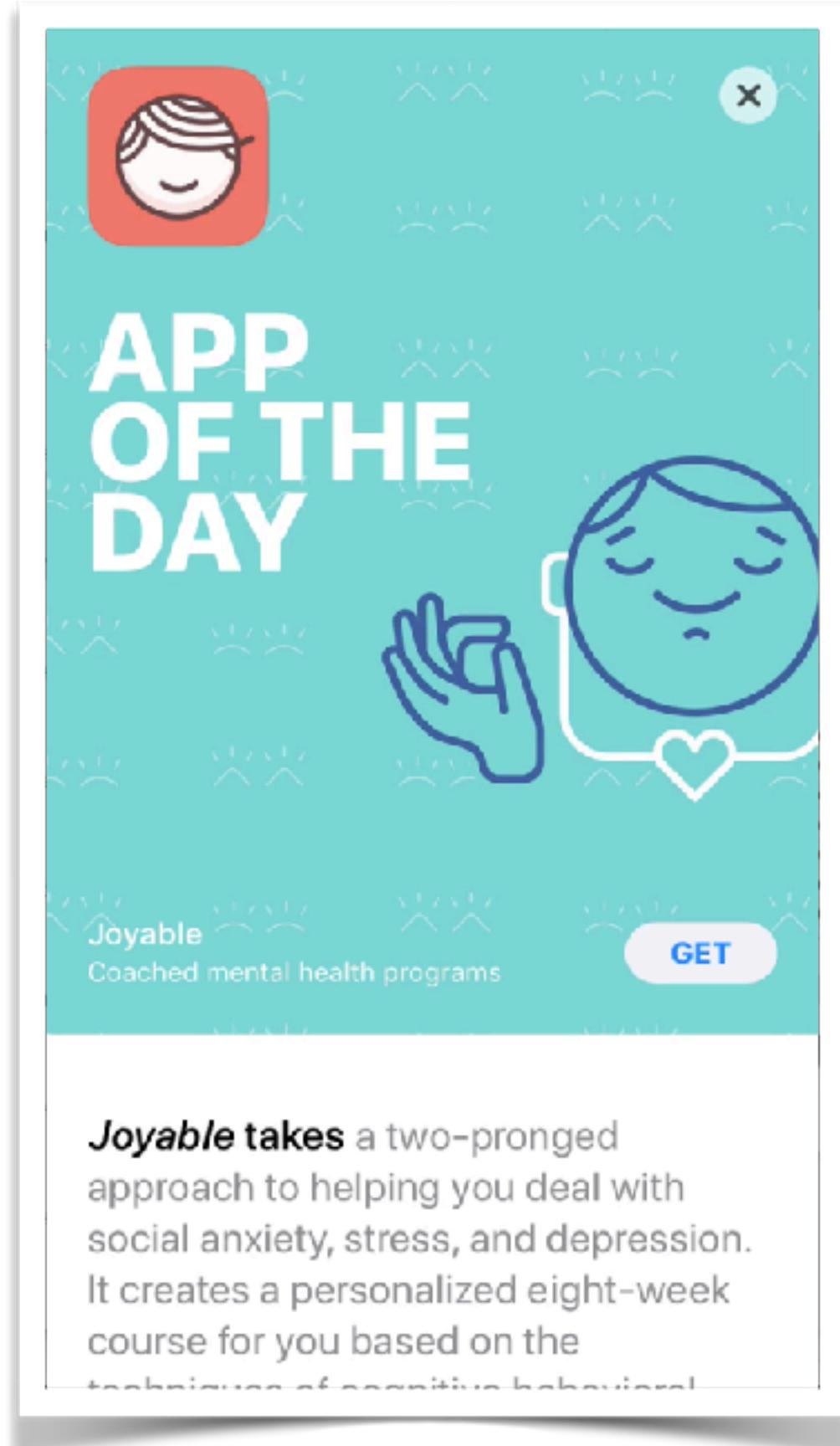


R
W

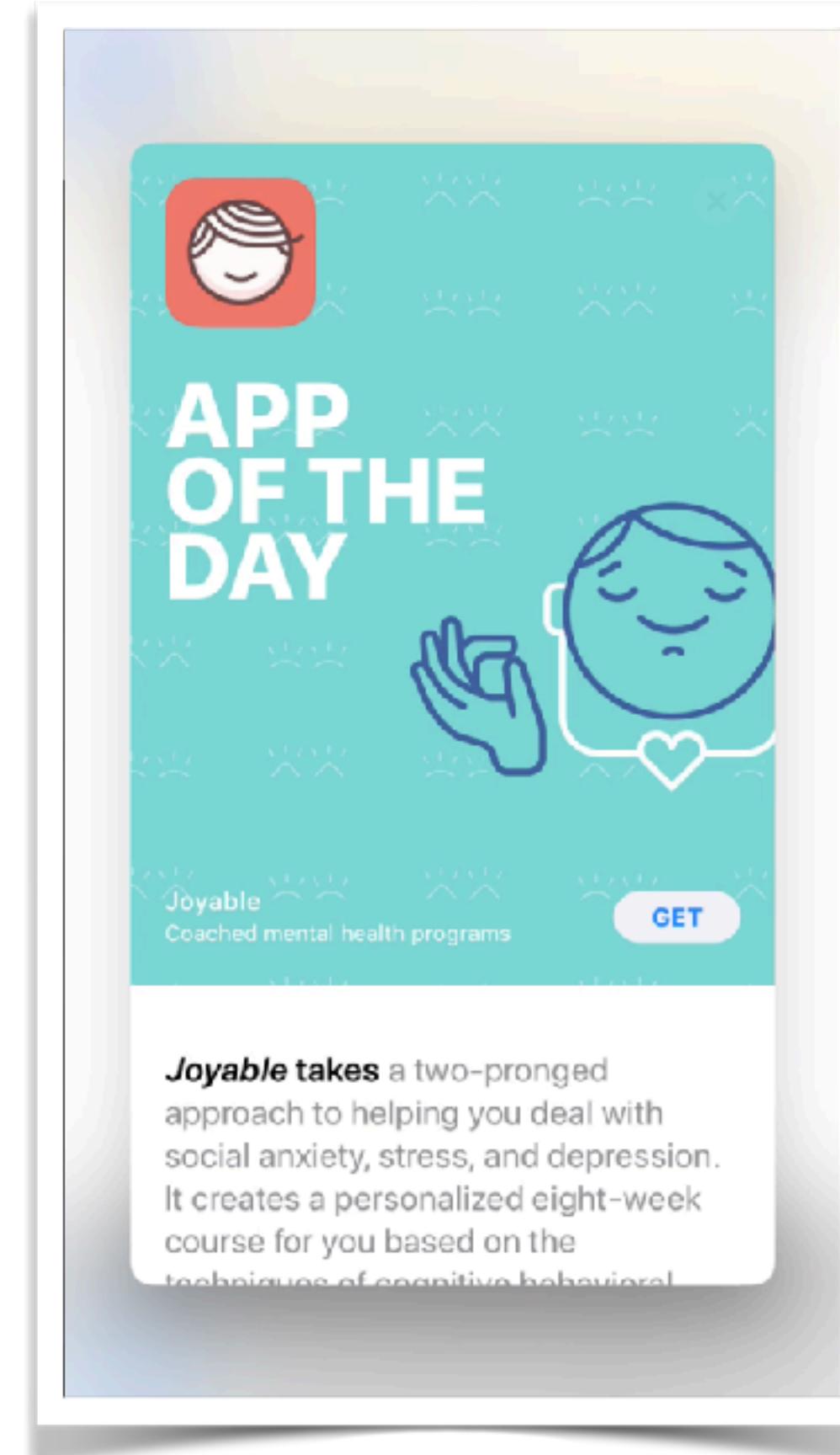
A new way to dismiss



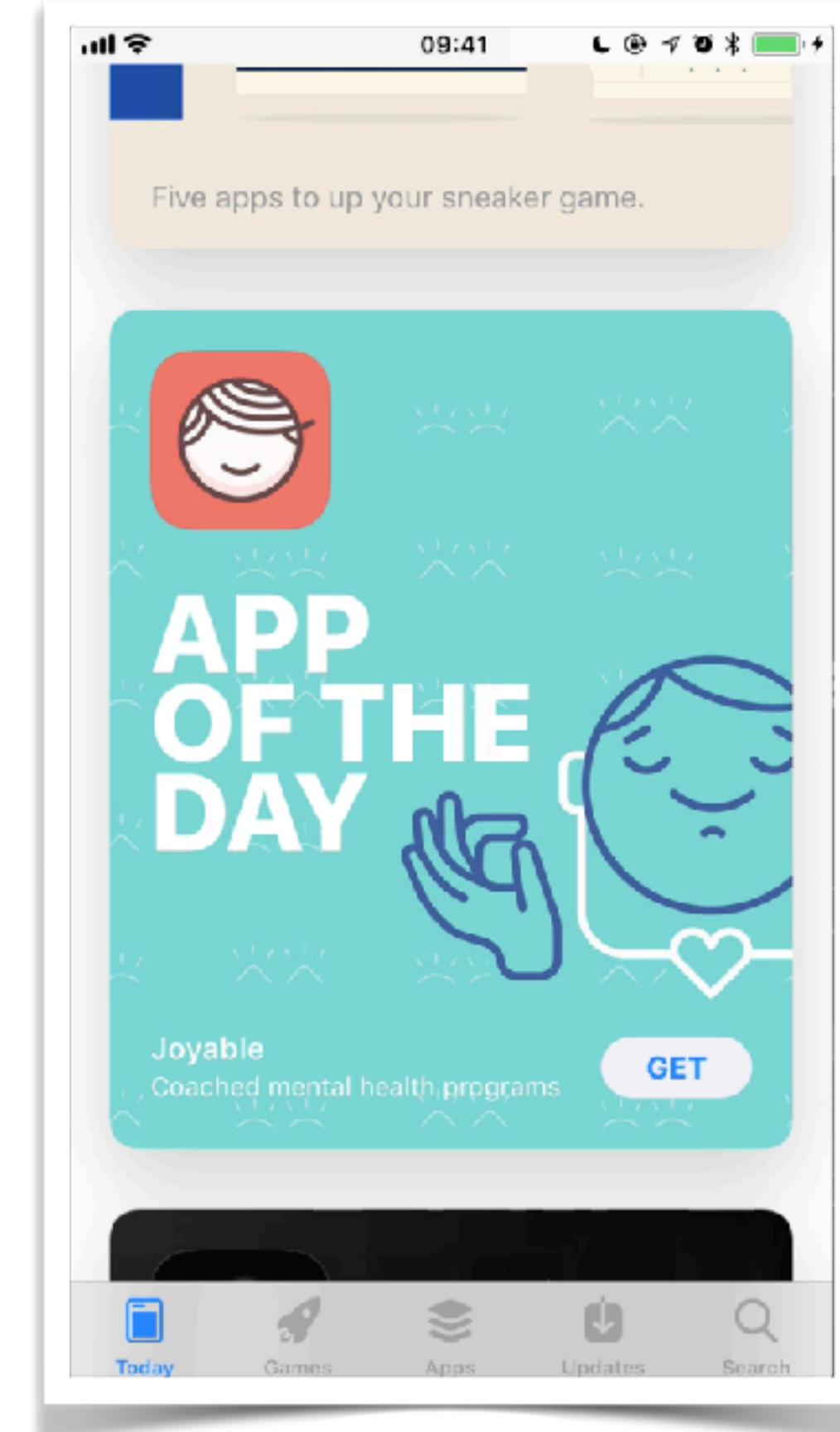
DISSECTING THE ANIMATION



Original



Scale down,
round & blur



Complete



CREATING SNAPSHOTS

```
UIGraphicsBeginImageContextWithOptions(_ size: CGSize, _ opaque: Bool, _ scale: CGFloat)
```

```
let frame = UIScreen.main.bounds  
  
UIGraphicsBeginImageContextWithOptions(frame.size, false, 0.0)  
let image = UIGraphicsGetImageFromCurrentImageContext()  
UIGraphicsEndImageContext()  
  
return image
```



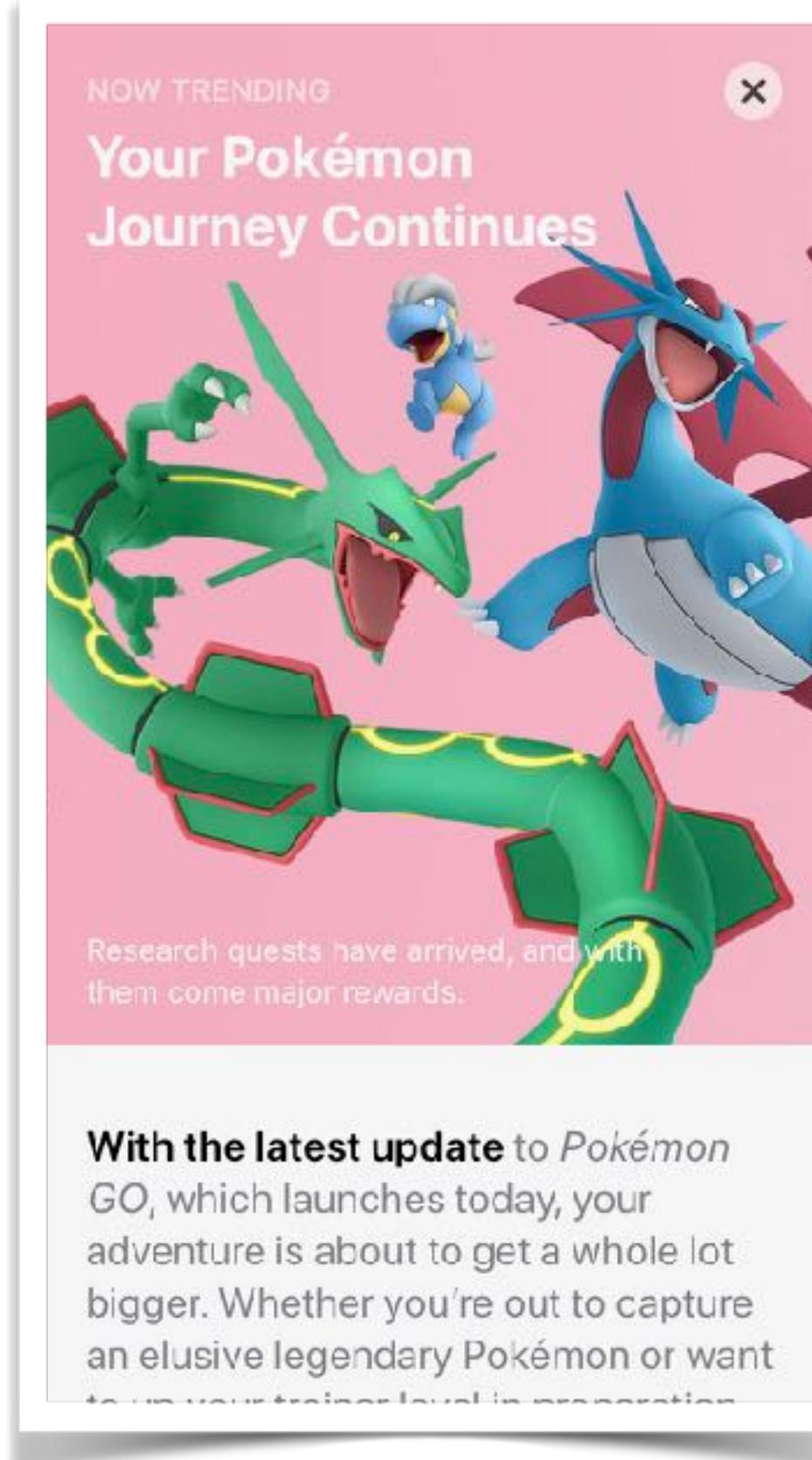
LISTEN TO THE SCROLLVIEW

- ▶ Create a snapshot when the user scrolls down at the top
- ▶ Conform to the UIScrollViewDelegate
- ▶ Use the y contentOffset to drive animation and dismissal

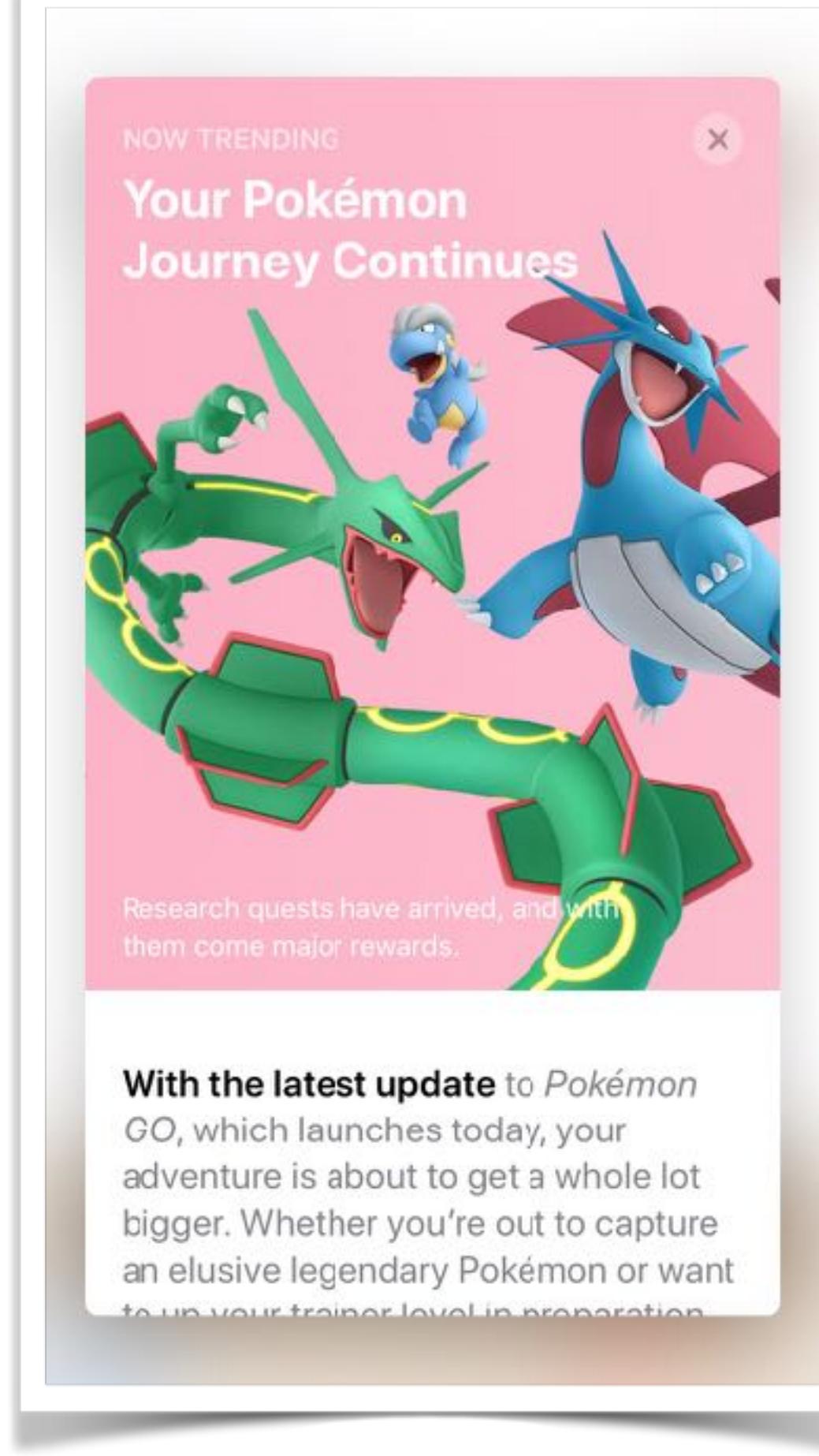
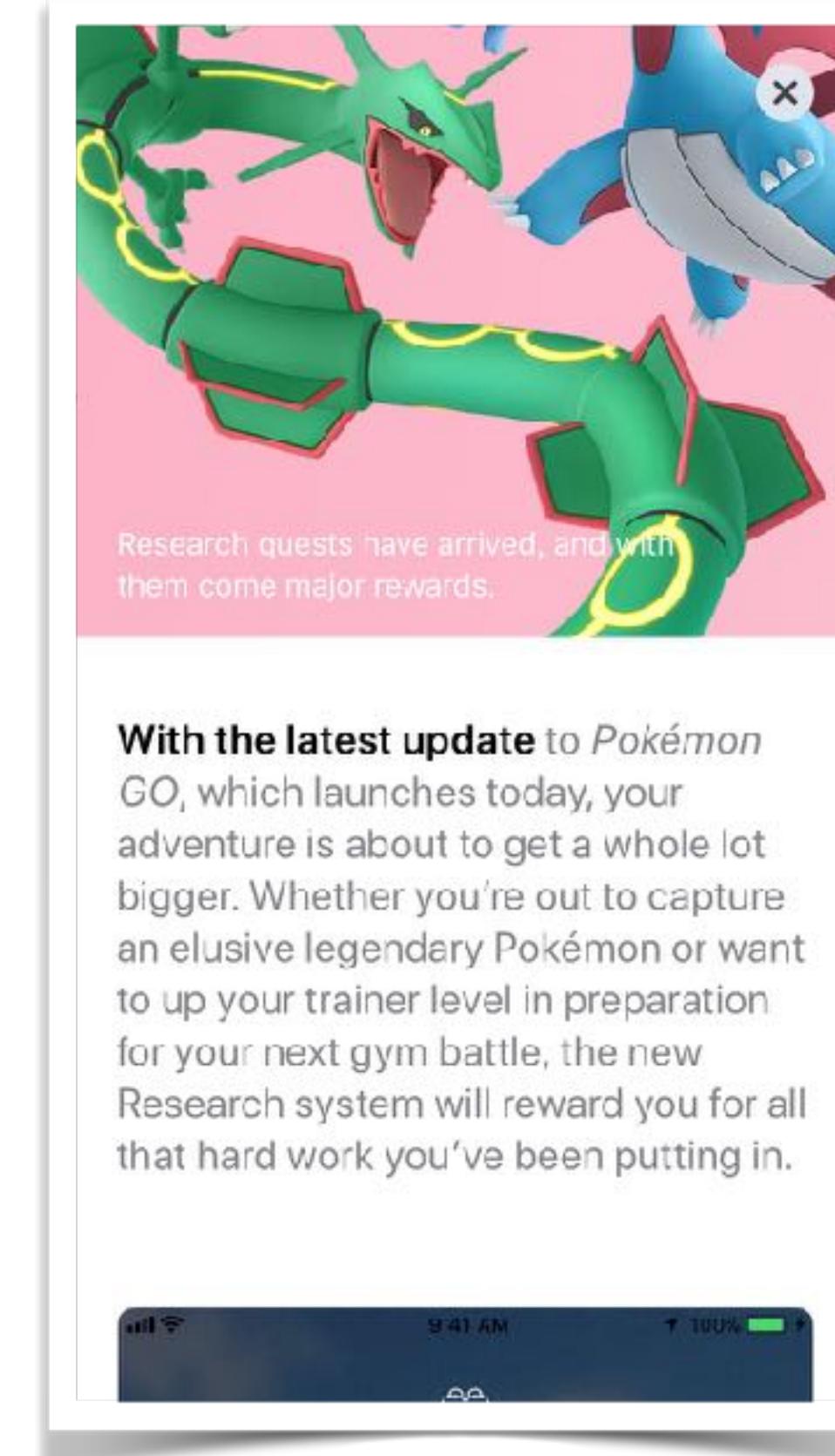
```
func scrollViewDidScroll(_ scrollView: UIScrollView) {  
    let yContentOffset = scrollView.contentOffset.y  
  
    // ~Make some magic~  
}
```



DISSECTING THE ANIMATION

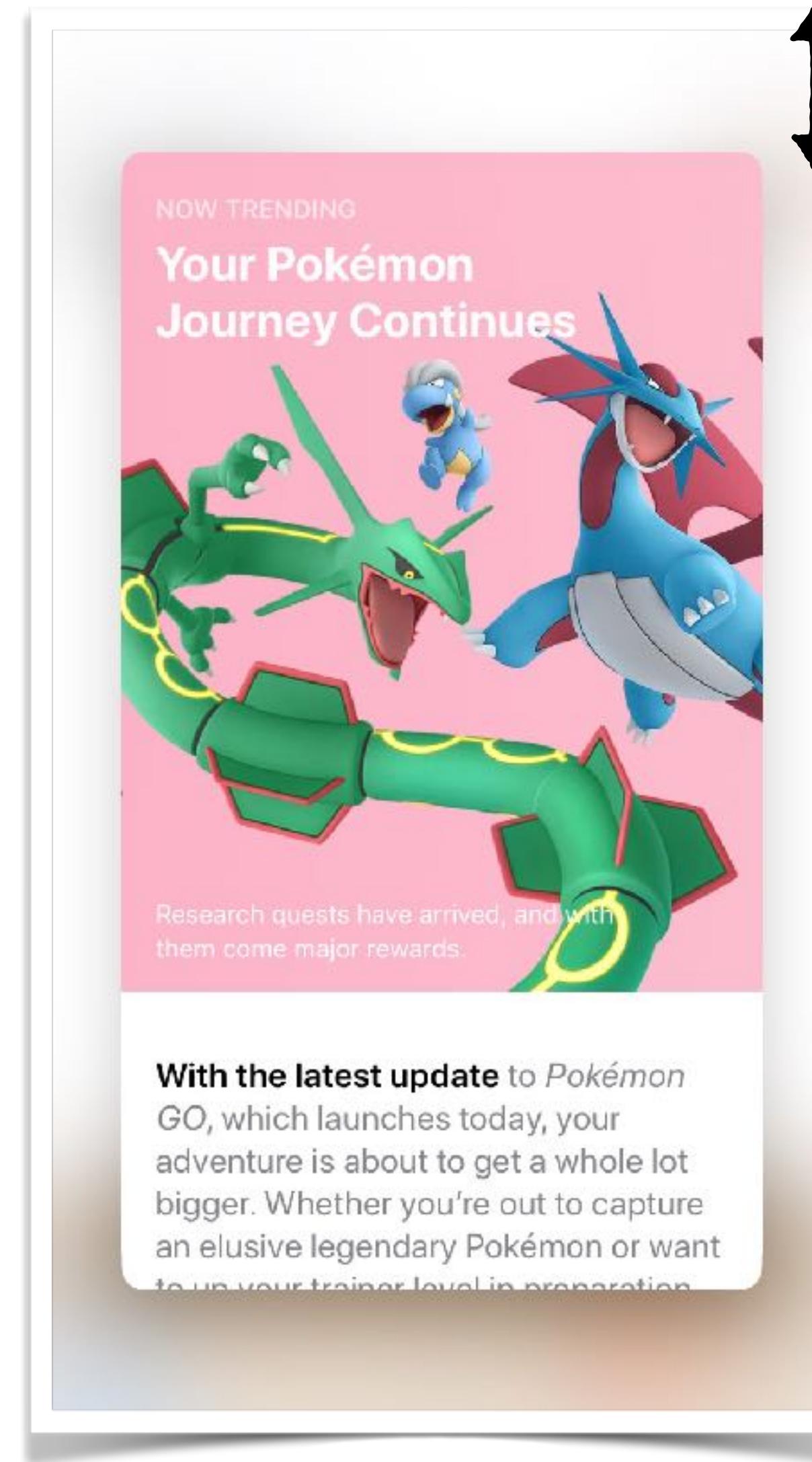
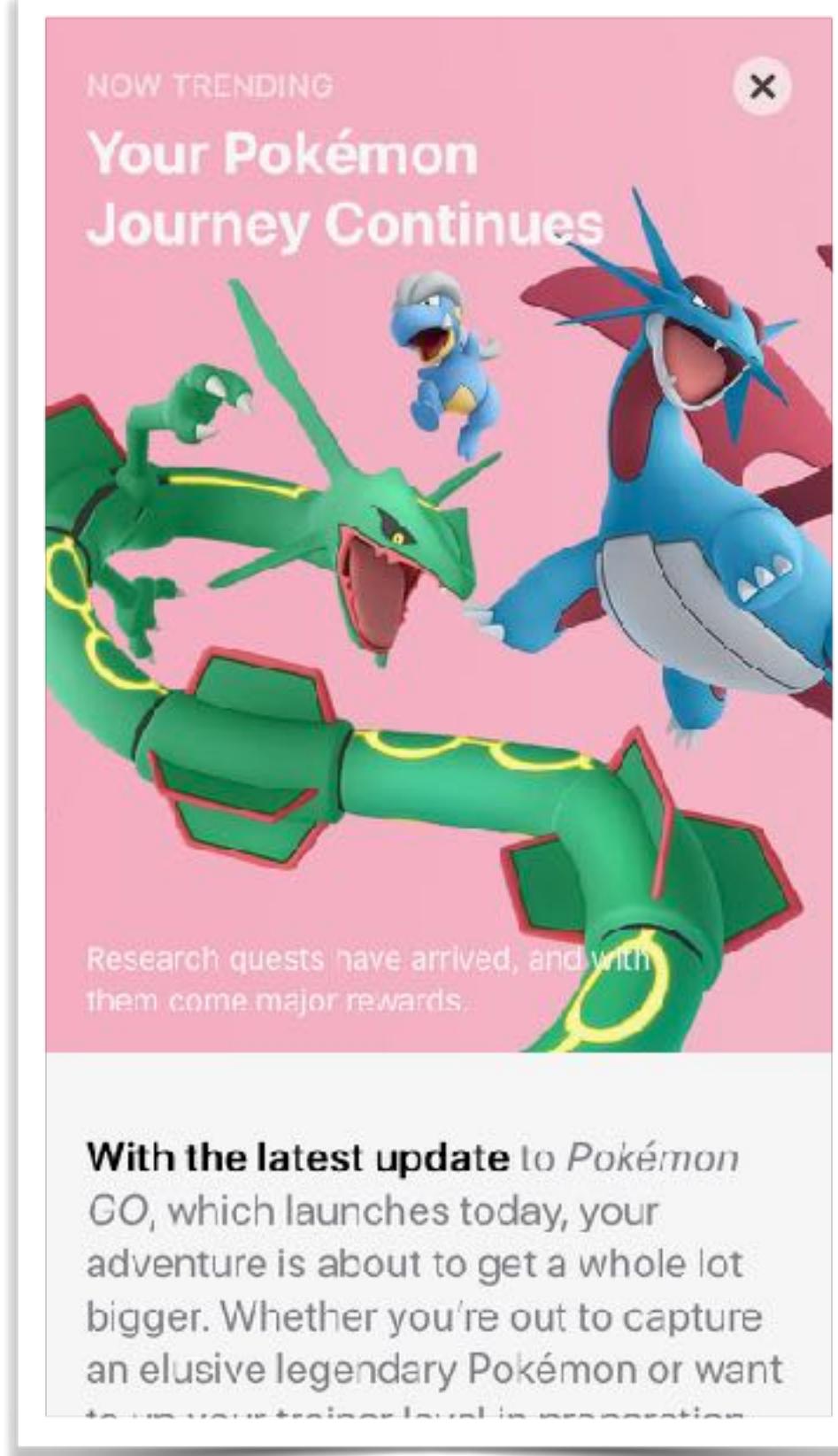


Scrolling down



Scrolling up

DISSECTING THE ANIMATION



20 pts

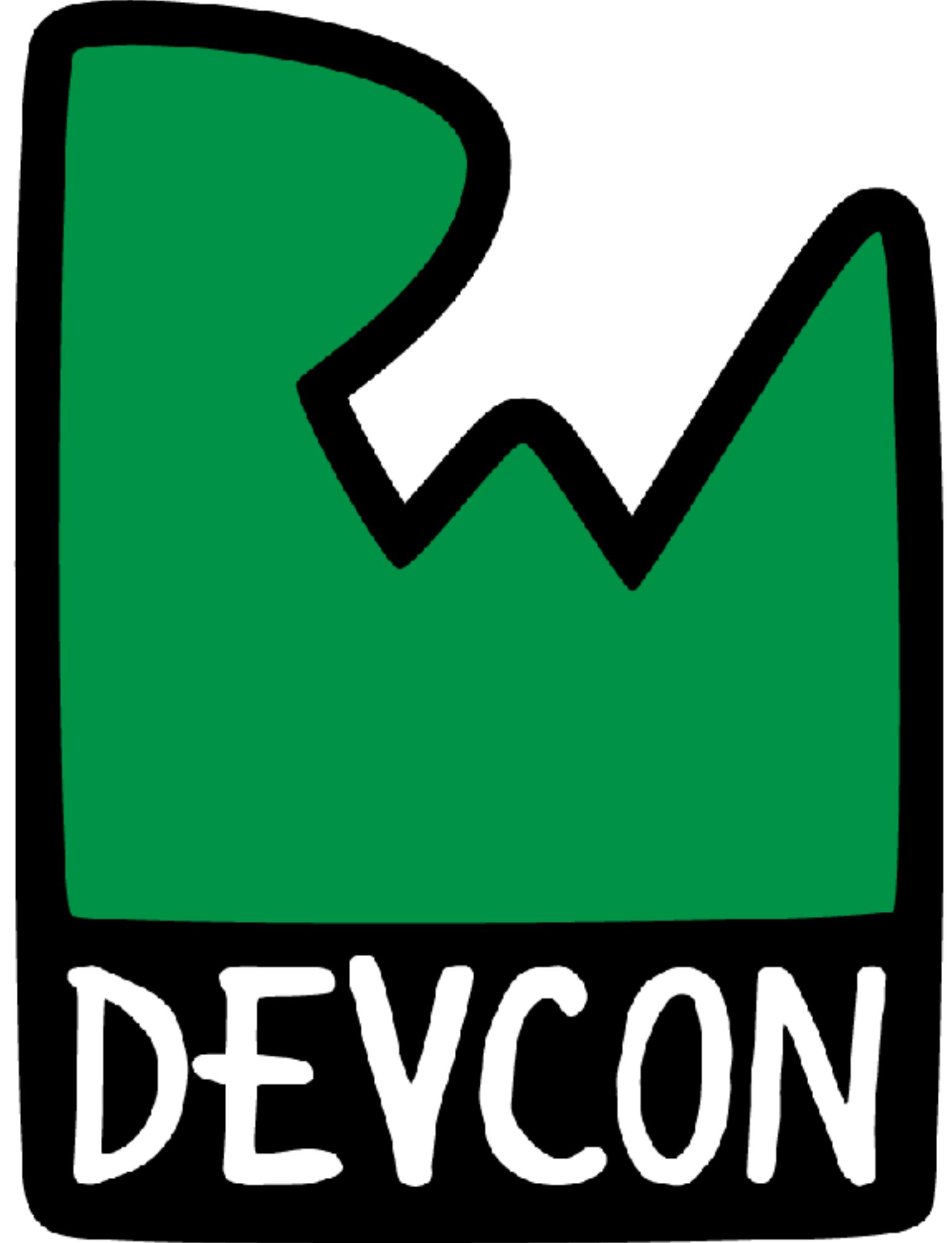
scalingFactor =
 $(100 + yContentOffset)/100$

if $yContentOffset > 20$
dismiss

DEMO 3



Session 14: Custom Views



RW CONCLUSION

WHAT You LEARNED

- ▶ **Demo 1:** Dynamic, reusable, generic
- ▶ **Demo 2:** Custom modal presentation
- ▶ **Demo 3:** Custom modal dismissal



WHERE To Go FROM HERE?

- ▶ [iOS Animations by Tutorials](#), by Marin Todorov
- ▶ [iOS Core Animation: Advanced Techniques](#)
- ▶ [UIPresentationController Tutorial: Getting Started](#)
- ▶ [Custom Container View Controller Transitions](#)
- ▶ [UI Movement](#)
- ▶ Twitter: [@hellosunschein](#)

