

# Advanced Collection Views and Building Custom Layouts

Session 219

Luke the Hiesterman

iOS Frameworks

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

# What You Will Learn

# What You Will Learn

- Deeper architectural understanding

# What You Will Learn

- Deeper architectural understanding
- Go far with `UICollectionViewFlowLayout`

# What You Will Learn

- Deeper architectural understanding
- Go far with `UICollectionViewFlowLayout`
- Build your own layout

# Concepts and Architecture

# Views in a Collection View

# Views in a Collection View

- Cells



# Views in a Collection View

- Cells
- Supplementary Views

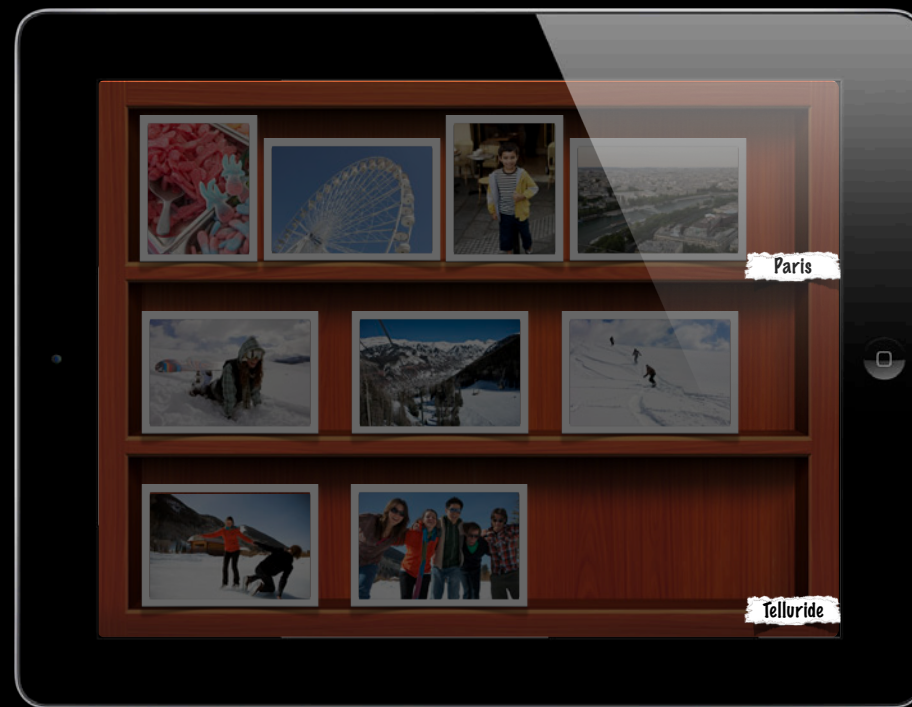
# Views in a Collection View

- Cells
- Supplementary Views
- Decoration Views

# Supplementary vs. Decoration Views



# Supplementary vs. Decoration Views



# Supplementary vs. Decoration Views

- Supplementary views are data driven

# Supplementary vs. Decoration Views



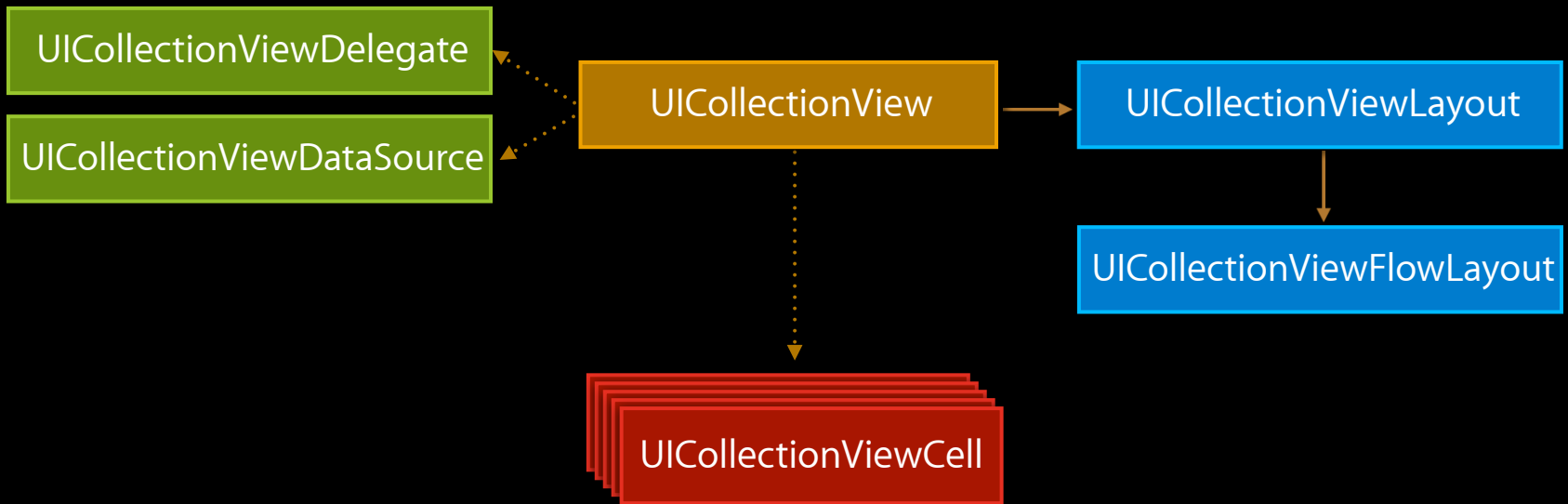
# Supplementary vs. Decoration Views

- Supplementary views are data driven
- Decoration views are layout driven

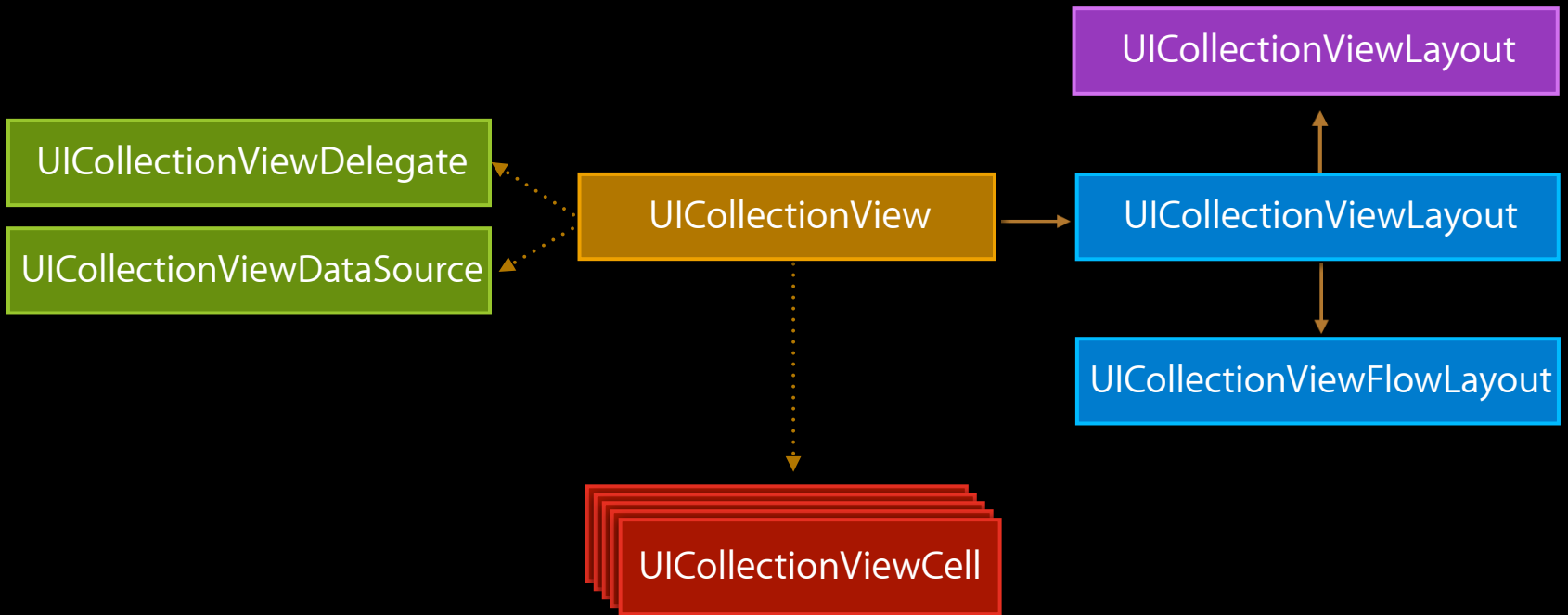
# UICollectionViewLayoutAttributes



# UICollectionViewLayoutAttributes



# UICollectionViewLayoutAttributes



# UICollectionViewLayoutAttributes



UICollectionView

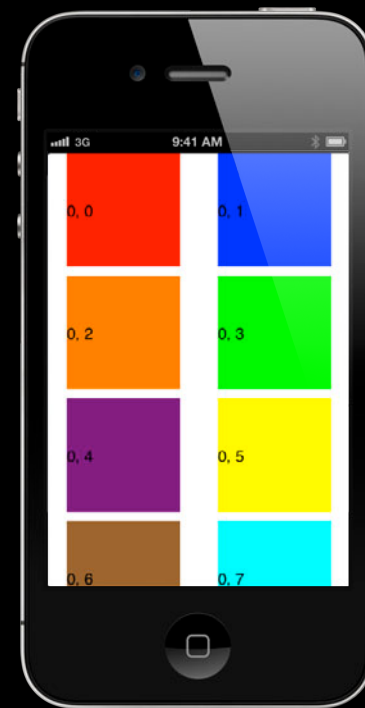


UICollectionViewLayout

# UICollectionViewLayoutAttributes

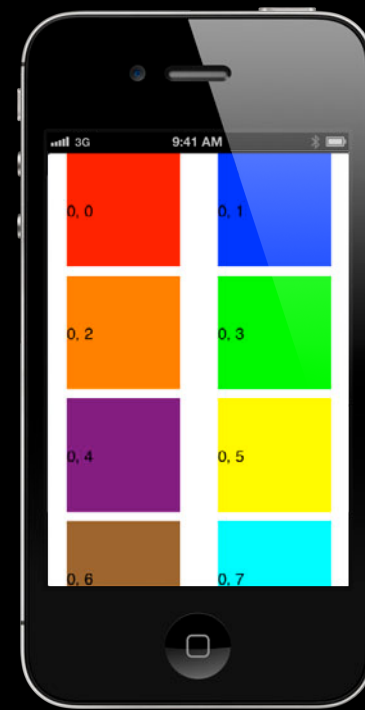


# UICollectionViewLayoutAttributes



# UICollectionViewLayoutAttributes

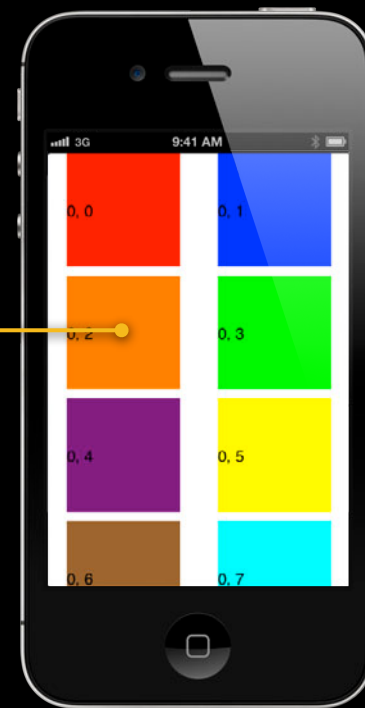
center and size



# UICollectionViewLayoutAttributes

center and size

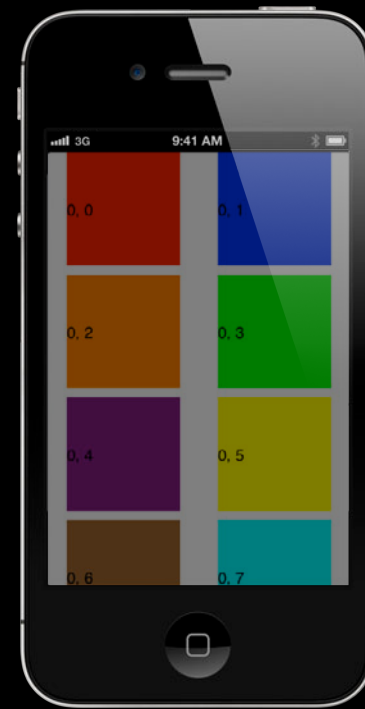
Center



# UICollectionViewLayoutAttributes

center and size

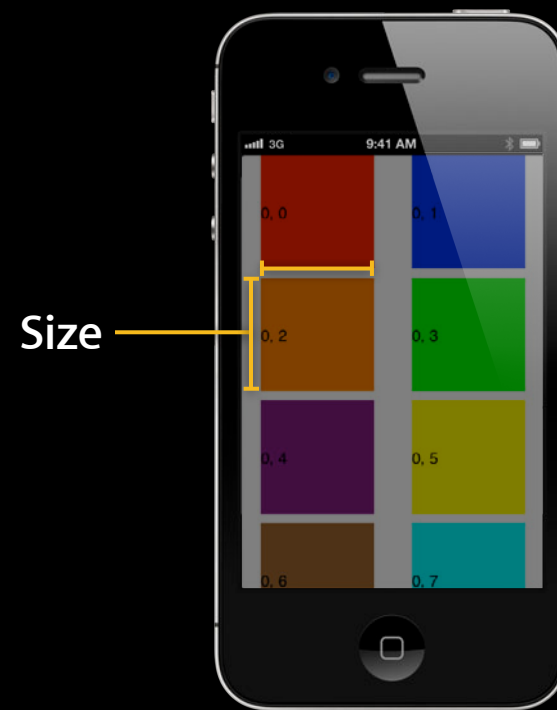
Size





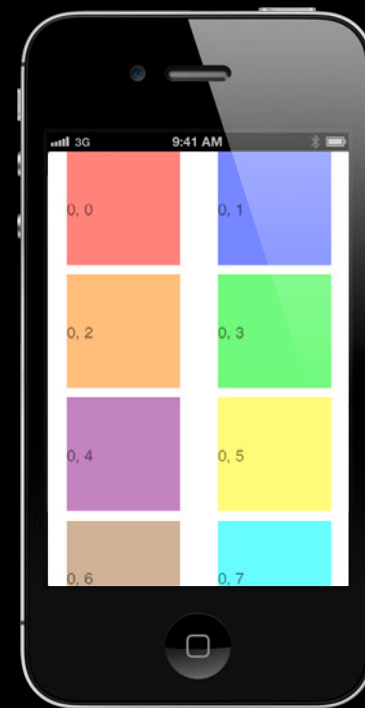
# UICollectionViewLayoutAttributes

center and size



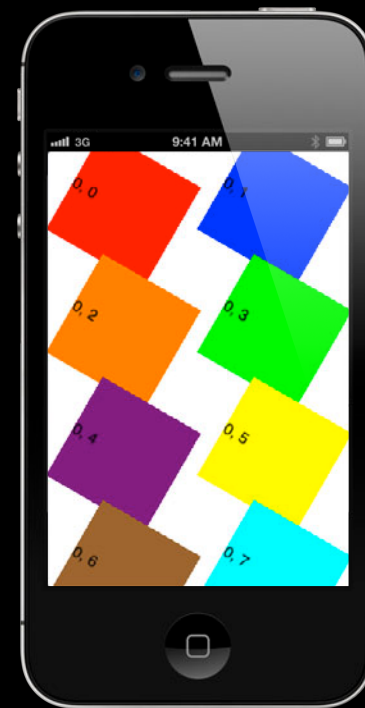
# UICollectionViewLayoutAttributes

center and size  
alpha



# UICollectionViewLayoutAttributes

center and size  
alpha  
transform



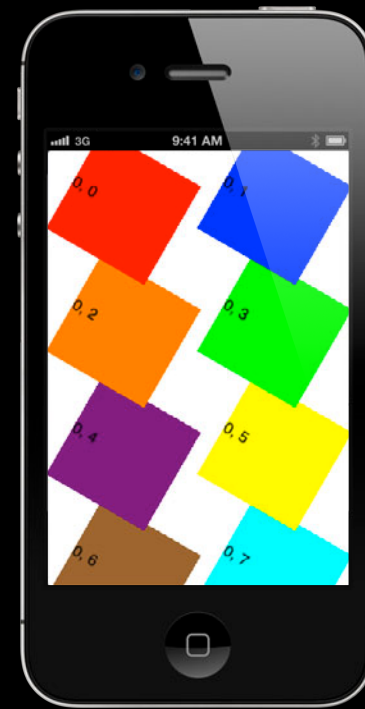
# UICollectionViewLayoutAttributes

center and size

alpha

transform

zIndex

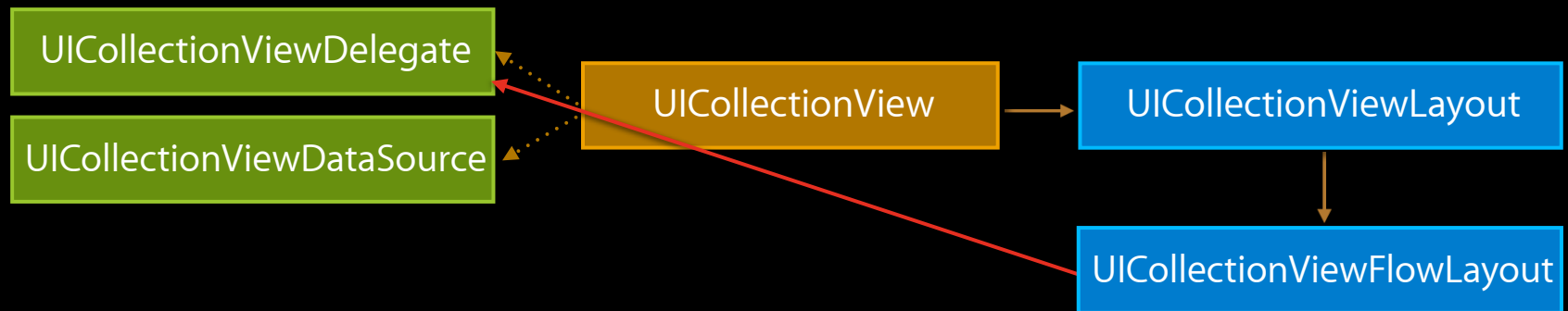


# Layout Delegation Pattern

# Layout Delegation Pattern



# Layout Delegation Pattern



# The Extensible Delegate

Not your granddaddy's protocol



# The Extensible Delegate

Not your granddaddy's protocol

```
UICollectionViewDelegate : UIScrollViewDelegate
```

# The Extensible Delegate

Not your granddaddy's protocol

```
UICollectionViewDelegate : UIScrollViewDelegate
```

```
UICollectionViewDelegateFlowLayout
```

# The Extensible Delegate

Not your granddaddy's protocol

```
UICollectionViewDelegate : UIScrollViewDelegate
```

```
UICollectionViewDelegateFlowLayout
```

```
... MyExtendedDelegate
```

# Changing the Layout

# Changing the Layout

- Change == layout invalidation

# Changing the Layout

- Change == layout invalidation
- Invalidate with `-invalidateLayout`

# Changing the Layout

- Change == layout invalidation
- Invalidate with `-invalidateLayout`
- Implicit invalidations with `performBatchUpdates:completion:`

# Changing the Layout

- Change == layout invalidation
- Invalidate with `-invalidateLayout`
- Implicit invalidations with `performBatchUpdates:completion:`
- Layout can dynamically invalidate with bounds changes



# Changing the Layout

- Change == layout invalidation
- Invalidate with `-invalidateLayout`
- Implicit invalidations with `performBatchUpdates:completion:`
- Layout can dynamically invalidate with bounds changes
  - `-shouldInvalidateForBoundsChange:`

# Advanced UICollectionViewFlowLayout

# What You Can Do With

`UICollectionViewFlowLayout`

# What You Can Do With UICollectionViewFlowLayout

- Any variation on a line-based breaking layout

# Reasons to Subclass

`UICollectionViewFlowLayout`

# Reasons to Subclass

`UICollectionViewFlowLayout`

- Add new kinds of views

# Reasons to Subclass

UICollectionViewFlowLayout



# Reasons to Subclass

`UICollectionViewFlowLayout`

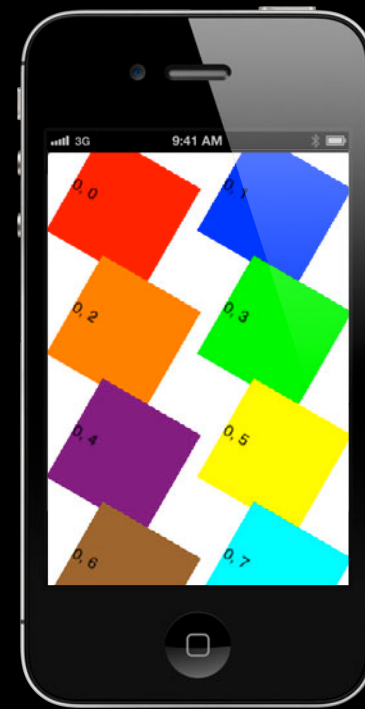
- Add new kinds of views
- Tweak layout attributes



# Reasons to Subclass

## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes



# Reasons to Subclass

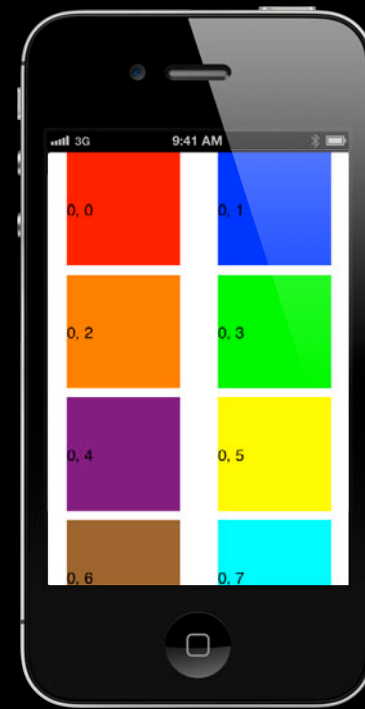
## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes

# Reasons to Subclass

## UICollectionViewFlowLayout

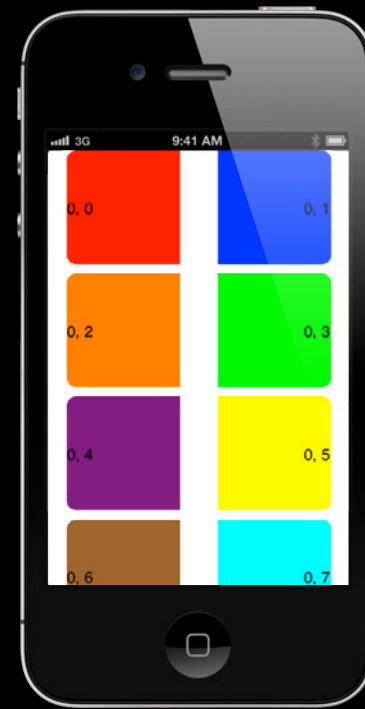
- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes



# Reasons to Subclass

## UICollectionViewFlowLayout

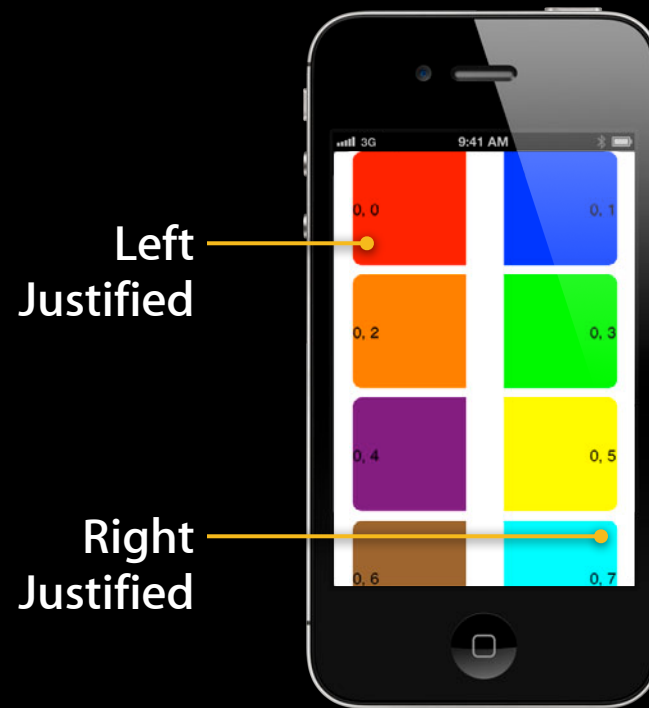
- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes



# Reasons to Subclass

## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes



# Reasons to Subclass

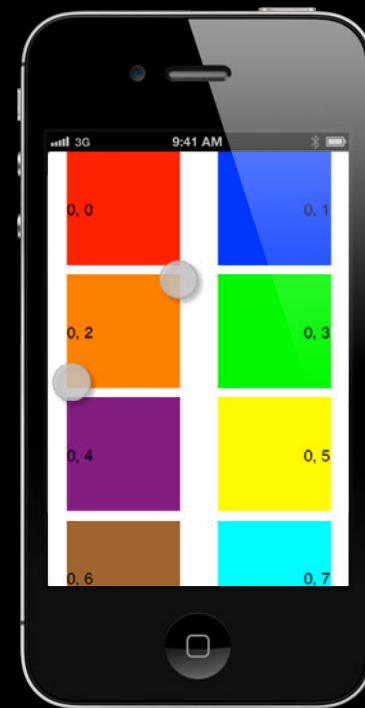
## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes
- Add gesture support

# Reasons to Subclass

## UICollectionViewFlowLayout

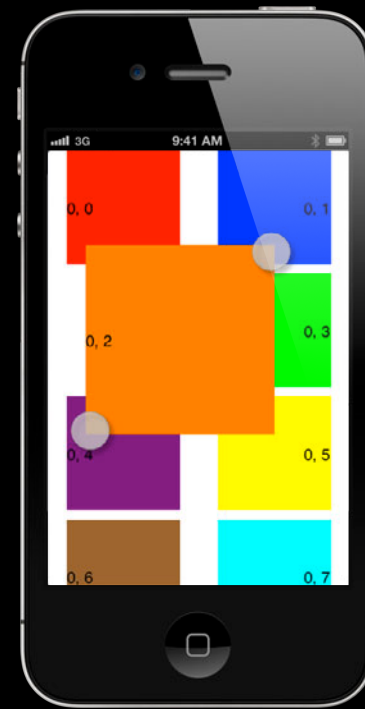
- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes
- Add gesture support



# Reasons to Subclass

## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes
- Add gesture support





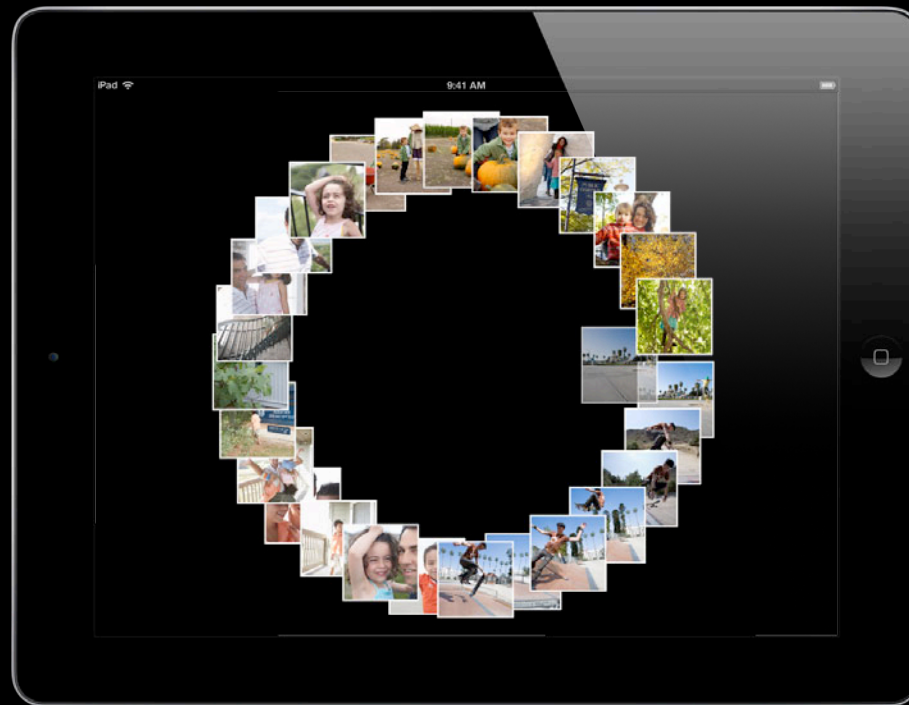
# Reasons to Subclass

## UICollectionViewFlowLayout

- Add new kinds of views
- Tweak layout attributes
- Add new layout attributes
- Add gesture support
- Customize insert and delete animations

# Reasons to Subclass

`UICollectionViewFlowLayout`



# Adding Extra Views

```
- (NSArray*)layoutAttributesForElementsInRect:(CGRect)rect
{
    NSMutableArray* attributes = [[super
    layoutAttributesForElementsInRect:rect] mutableCopy];

}
```

# Adding Extra Views

```
- (NSArray*)layoutAttributesForElementsInRect:(CGRect)rect
{
    NSMutableArray* attributes = [[super
    layoutAttributesForElementsInRect:rect] mutableCopy];
    [attributes addObject:[self myAdditionalElementAttributes];
}
}
```

# Adding Extra Views

```
- (NSArray*)layoutAttributesForElementsInRect:(CGRect)rect
{
    NSMutableArray* attributes = [[super
    layoutAttributesForElementsInRect:rect] mutableCopy];
    [attributes addObject:[self myAdditionalElementAttributes]];
    return attributes;
}
```

*Demo*

Pinch resizing gesture

**Mathieu Martin**

iOS Frameworks

# Writing Custom Layouts

# Going Fully Custom

Deciding when it's time



# Going Fully Custom

Deciding when it's time

- Changing all cell positions

# Going Fully Custom

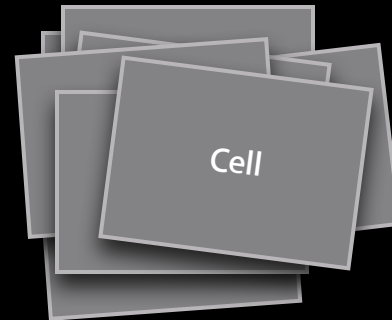
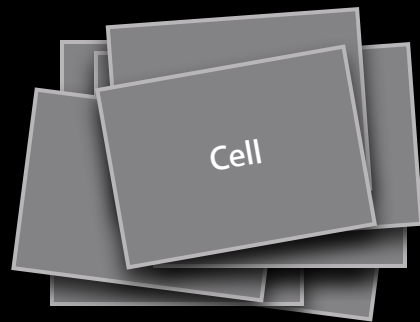
## Deciding when it's time

- Changing all cell positions
- Essentially, what you want looks nothing like a grid

# Going Fully Custom

## Deciding when it's time

- Changing all cell positions
- Essentially, what you want looks nothing like a grid



# Overriding UICollectionViewLayout

Required overrides

# Overriding UICollectionViewLayout

## Required overrides

`-collectionViewContentSize`

# Overriding UICollectionViewLayout

## Required overrides

-collectionViewContentSize

-layoutAttributesForElementsInRect:

# Overriding UICollectionViewLayout

## Required overrides

- collectionViewContentSize
- layoutAttributesForElementsInRect:
- layoutAttributesForItemAtIndexPath:

# Overriding UICollectionViewLayout

## Required overrides

-collectionViewContentSize

-layoutAttributesForElementsInRect:

-layoutAttributesForItemAtIndexPath:

-layoutAttributesForSupplementaryViewOfKind:atIndexPath:



# Overriding UICollectionViewLayout

## Required overrides

-collectionViewContentSize

-layoutAttributesForElementsInRect:

-layoutAttributesForItemAtIndexPath:

-layoutAttributesForSupplementaryViewOfKind:atIndexPath:

-layoutAttributesForDecorationViewWithReuseIdentifier:atIndexPath:

# Order of Operations

Know when your hooks get called

# Order of Operations

Know when your hooks get called

-prepareLayout

# Order of Operations

Know when your hooks get called

-prepareLayout

-collectionViewContentSize

# Order of Operations

Know when your hooks get called

-prepareLayout

-collectionViewContentSize

-layoutAttributesForElementsInRect

# Order of Operations

Know when your hooks get called

-prepareLayout

-collectionViewContentSize

-layoutAttributesForElementsInRect

- All other calls

# Order of Operations

Know when your hooks get called

-prepareLayout

-collectionViewContentSize

-layoutAttributesForElementsInRect

- All other calls

-invalidateLayout

# Order of Operations

Know when your hooks get called

- prepareLayout

- collectionViewContentSize

- layoutAttributesForElementsInRect

- All other calls

- invalidateLayout

- prepareLayout



# Order of Operations

Know when your hooks get called

- prepareLayout

- collectionViewContentSize

- layoutAttributesForElementsInRect

- All other calls

- invalidateLayout

- prepareLayout

- ... the cycle continues

*Demo*

Custom layouts

**Dr. Olivier Gutknecht**

iOS Frameworks

Recap

**In Review**

# In Review

- `UICollectionViewFlowLayout` is both subclassable and powerful

# In Review

- `UICollectionViewFlowLayout` is both subclassable and powerful
- `UICollectionView` supports arbitrary layouts built by you

# Apple Evangelists

**Jake Behrens**

UI Frameworks Evangelist

[behrens@apple.com](mailto:behrens@apple.com)

# Related Sessions

Introducing Collection Views

Marina  
Thursday 9:00AM



# Labs

Multithreading Lab

Essentials Lab A  
Thursday 2:00PM

 WWDC2012

The last 3 slides  
after the logo are  
intentionally left  
blank for all  
presentations.

The last 3 slides  
after the logo are  
intentionally left  
blank for all  
presentations.

The last 3 slides  
after the logo are  
intentionally left  
blank for all  
presentations.