A Tour of UICollectionView

Session 225

Steve Breen, UIKit Frameworks Engineer
Mohammed Jisrawi, UIKit Frameworks Engineer
Agenda

• Build an app
  • Layouts
  • Updates
  • Animations
Key UICollectionView Concepts

Layout
Data source
Delegate
UICollectionViewLayout

Visual arrangement of content

UICollectionViewLayoutAttributes
- Bounds, center, and frame...

Invalidation

Animate between layouts
UICollectionViewFlowLayout

Concrete subclass of UICollectionViewLayout

UICollectionViewDelegateFlowLayout extends UICollectionViewDelegate

Line-based layout covers a wide range of designs
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout

Line Spacing
Flow Layout

Inter-Item Spacing
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout
Flow Layout

Line Orientation
Flow Layout

Line Spacing
Flow Layout

Inter-Item Spacing
UICollectionViewDataSource

Provides content

Section and item counts

optional func numberOfSections(in collectionView: UICollectionView) -> Int

func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int

func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell
Optional

Extends UICollectionViewDelegate

Fine-grained control

• Highlighting
• Selection

View appearance events

• willDisplayItem
• didFinishDisplayingItem
When to Go Custom?

Can we use UICollectionViewFlowLayout?
When to Go Custom?

Can we use UICollectionViewFlowLayout?
When to Go Custom?

Can we use UICollectionViewFlowLayout?
When to Go Custom?
When to Go Custom?
When to Go Custom?

Our fancy layout is not line-based
When to Go Custom?

Our fancy layout is not line-based

Start with flow before going custom!
Creating a Custom UICollectionViewLayout

Not complicated!

Four basic methods

...and one additional method to consider...
Providing Content Size

open var collectionViewContentSize: CGSize { get }

Size of bounds which contains all items

Needed for UIScrollView.contentSize
Providing Layout Attributes

- func layoutAttributesForElements(in rect: CGRect) -> [UICollectionViewLayoutAttributes]?
- func layoutAttributesForItem(at indexPath: IndexPath) -> UICollectionViewLayoutAttributes?

Query by geometric region

Query by IndexPath

Performance matters
Preparing the Layout

```swift
func prepare()
```

Called for every `invalidateLayout`

Cache `UICollectionViewLayoutAttributes`

Compute `collectionViewContentSize`
Handling Bounds Changes in Your Custom Layout

```swift
func shouldInvalidateLayout(forBoundsChange newBounds: CGRect) -> Bool
```

Called for every bounds change

- Size change
- Origin change

Called during scrolling 😳

Default implementation returns `false`
Demo

Our fancy custom UICollectionView Layout
Our Totally Cool Update Animation

1. Reload last item
2. Move last -> first
3. Delete 3rd item
Our Totally Cool Update Animation

1. Reload last item
2. Move last -> first
3. Delete 3rd item
*** Terminating app due to uncaught exception 'NSInternalInconsistencyException', reason: 'attempt to perform a delete and a move from the same index path (<NSIndexPath: 0xc000000000000016> {length = 2, path = 0 - 3})' ****
*** Terminating app due to uncaught exception 'NSInternalInconsistencyException', reason: 'attempt to perform a delete and a move from the same index path (<NSIndexPath: 0xc00000000000016> {length = 2, path = 0 – 3})' ****
Debugging Update Exceptions 😞

*** Terminating app due to uncaught exception 'NSInternalInconsistencyException', reason: 'attempt to perform a **delete** and a move from the same index path (<NSIndexPath: 0xc000000000000016> {length = 2, path = 0 - 3})' ****

Wait ... delete? 😳
performBatchUpdates()

func performBatchUpdates(_ updates: (() -> Void)?, completion: ((Bool) -> Void)? = nil)

Animate updates together

Perform data source updates and collection view updates in updates closure

Collection view updates ordering does not matter...

...data source updates ordering does matter
Data Source Updates: Order Matters

Delete 0 → Insert 1

Insert 1 → Delete 0

Delete 0 → Insert 1

Insert 1 → Delete 0
Data Source Updates: Order Matters

Delete 0 ➔ Insert 1

Insert 1 ➔ Delete 0
Data Source Updates: Order Matters

Delete 0 ➔ Insert 1

Inserted

Insert 1 ➔ Delete 0

0

1

2
Data Source Updates: Order Matters

Delete 0 ➔ Insert 1

1
Inserted
2

Insert 1 ➔ Delete 0

0
Inserted
1
2
Data Source Updates: Order Matters

Delete 0 ➔ Insert 1

- Inserted
- 2
- 1

Insert 1 ➔ Delete 0

- Inserted
- 1
- 2
Both of These Collection View Updates Are Equivalent!

```swift
let insertedIndexPath = IndexPath(item:0, section:0)
let deletedIndexPath = IndexPath(item:1, section:0)

let updates1 = {
    collectionView.insertItems(at: insertedIndexPaths)
    collectionView.deleteItems(at: deletedIndexPaths)
}
collectionView.performBatchUpdates(updates1)

let updates2 = {
    collectionView.deleteItems(at: deletedIndexPaths)
    collectionView.insertItems(at: insertedIndexPaths)
}
collectionView.performBatchUpdates(updates2)
```
# Collection View Updates Coalescing

<table>
<thead>
<tr>
<th>Action</th>
<th>Notes</th>
<th>Index Path Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Descending IndexPath order</td>
<td>Before updates</td>
</tr>
<tr>
<td>Insert</td>
<td>Ascending IndexPath order</td>
<td>After updates</td>
</tr>
<tr>
<td>Move</td>
<td>From: Before updates</td>
<td>To: After updates</td>
</tr>
<tr>
<td>Reload</td>
<td>Decompose: Delete and inserts</td>
<td>Before updates</td>
</tr>
</tbody>
</table>
Update Combinations That Don’t Make Sense

Exceptions will result from:
• Move and Delete the same location
• Move and Insert to the same location
• Move more than 1 location to the same location
• Referencing an invalid IndexPath
Applying Data Source Updates

Decompose Move into Delete and Insert updates

Combine all Delete and Insert updates

Process Delete updates first, in descending order

Process Insert updates last, in ascending order
What About reloadData()?

No updates required

Resync data source to collection view

Not animated

“The Sledgehammer”
Demo

Saving our totally cool update animation
Call to Action

- Build custom layouts
- Update animations
- Reload data
More Information

https://developer.apple.com/wwdc18/225