UICollectionView in SWIFT

Workshop im Modul FSIOS

von Josina Zotzmann

Inhalte des Workshops

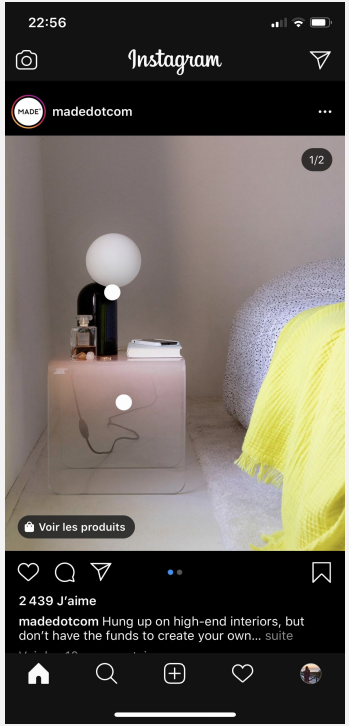
- Einführung in die Theorie
  - Compositional Layout
  - Diffable Data Sources
- Beispiele
- Übung

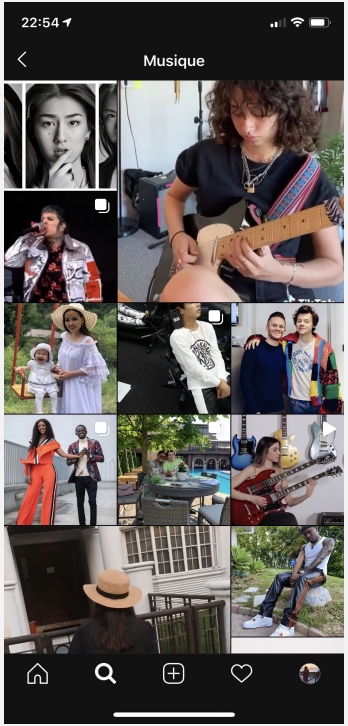
Git Repository

https://github.com/JosinaZ/FSIOSWorkshop.git

Beispiel UICollectionView - Fotos App

Beispiel UICollectionView - Instagram

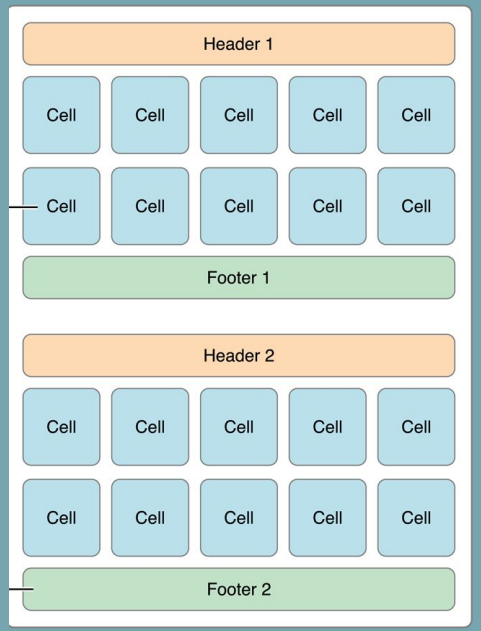




UICollectionView

- *“An object that manages an ordered collection of data items and presents them using customizable layouts.”*
- ähnlich zur UITableView:
  Die App verwaltet die mit der CollectionView verknüpften Daten
- class UICollectionView : UIScrollView

UICollectionView



- individuelle Items werden in Sections gruppiert
- Items werden in Zellen angezeigt
  - Instanz der Klasse **UICollectionViewCell**
  - konfiguriert und bereitgestellt von der data Source
- Daten können auch in header/footer angezeigt werden
  - Definition im **CollectionViewLayout** Objekt

Layouts

Visuelle Anordnung

- Ein Layout-Objekt der Unterklasse **UICollectionViewLayout** definiert die Organisation und Position aller Zellen und weiteren Ansichten in der Collection View

- **UICollectionFlowLayout** als ein einfaches layout
    - Scolling Direction
    - header/footer Views
    - Size inspector

- **Custom Layout** → Instanziierung CollectionViewLayout als Unterklasse

Cells & supplementary Views

Views aus der “Warteschleife” holen:

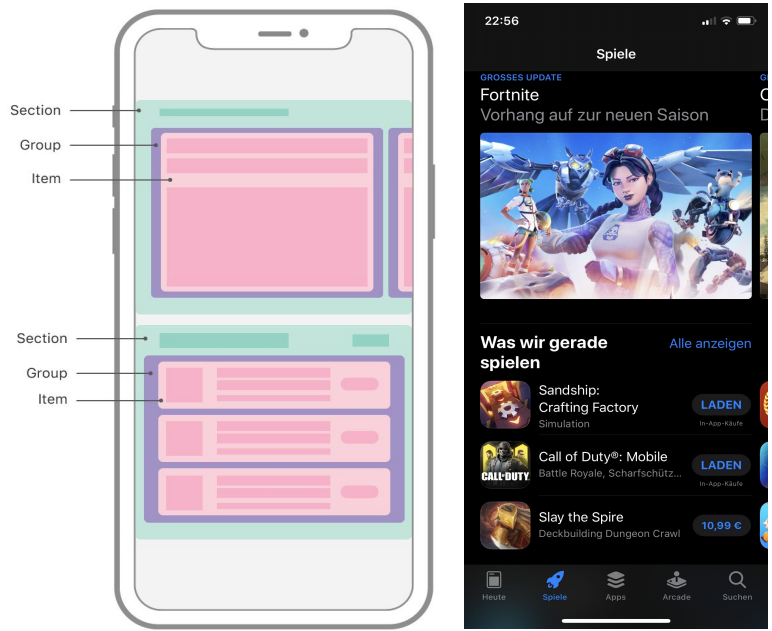
- dequeueReusableCell (withReuseIdentifier: for :)
- dequeueReusableSupplementaryView (ofKind: withReuseIdentifier: for :)

Klasse oder nib-File registrieren:

- register (_: forCellWithReuseIdentifier :)

Compositional Layout

- UICollectionViewLayout Objekt
  - Item>group>section>layout
- 4 Core Types:
  - NSCollectionLayoutSize
    - NSCollectionDimension
  - NSCollectionLayoutItem
  - NSCollectionLayoutGroup
  - NSCollectionLayoutSection



Beispiel iOS App Store

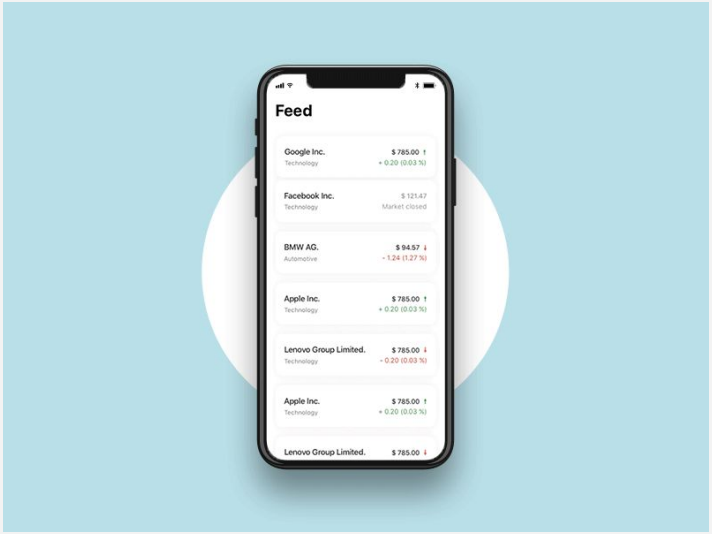
Compositional Layout - Beispiel Code

```
private func createLayout() -> UICollectionViewLayout {

//1
let itemSize = NSCollectionLayoutSize(widthDimension:
.fractionalWidth(1.0),heightDimension: .fractionalHeight(1.0))

let item = NSCollectionLayoutItem(layoutSize: itemSize)

//2
let groupSize = NSCollectionLayoutSize(widthDimension:
.fractionalWidth(1.0),heightDimension: .absolute(44))

//3
let group = NSCollectionLayoutGroup.horizontal(layoutSize:
groupSize,subitems: [item])

let section = NSCollectionLayoutSection(group: group)

let layout = UICollectionViewCompositionalLayout(section: section)

return layout

}
```

```
collectionView = UICollectionView(frame: view.bounds,
collectionViewLayout: createLayout())
```

```
let group = NSCollectionLayoutGroup.horizontal(layoutSize:
groupSize,subitem: item, count: 2)
```

Libraries



Quelle: https://medium.com/better-programming/spice-up-your-apps-collection-views-with-these-7-libraries-cda2379ce4d7

Data Source - Diffable Data Source

- neue API: **DiffableDataSource**
- **automatische Updates** zwischen den States
- kein reloadData() & performBatchUpdates()
- eine Methode: apply()

- **4 Klassen:**
  - UICollectionViewDiffableDataSource
  - UITableViewDiffableDataSource
  - NSCollectionViewDiffableDataSource (Mac)
  - NSDIffableSourceSnapshot

Diffable Data Source

Prozess in drei Schritten:

1. Wird ein neues Datenset oder Änderungen in die UICollectionView mit DiffableDataSource einfügt, muss ein Snapshot erstellt werden
2. Snapshot werden die Beschreibungen der Items (den Daten) zugewiesen, die im Update angezeigt werden sollen
3. Snapshot auf das UI angewendet werden

- Snapshots vertrauen nicht auf index Paths
    - für die Items wird ein typ-sicherer unique identifier value erstellt

      → Hashable Protocol

Diffable Data Source

Um Data Source zu updaten/befüllen, werden die benötigten Sections & ihre Items zum Snapshot hinzugefügt und angewendet:

```
func updateDataSource(animated: Bool) {

var snapshot = NSDiffableDataSourceSnapshot<Section, Movies>()

snapshot.appendSections(Section.allCases)

snapshot.appendItems([Movies(name: "Inception")], toSection: .one)
snapshot.appendItems([Movies(name: "War")], toSection: .one)
snapshot.appendItems([Movies(name: "Departed")], toSection: .one)

snapshot.appendItems([Movies(name: "Departed")], toSection: .two)

dataSource.apply(snapshot, animatingDifferences: animated)
```

Übung

Compositional Layout

- Layout mit 3 verschiedenen Sections
    - 25 Items in einer Section (1 Spalte)
    - 3 Spalten
    - 5 Spalten, dessen Zellen 10% der Breite einnehmen
- Instagram Stories
    - Circle Images in UICollectionView

Beispiele Compositional Layout & Diffable Data Source

https://developer.apple.com/documentation/uikit/views_and_controls/collection_views/using_collection_view_compositional_layouts_and_diffable_data_sources

Vielen Dank für eure Aufmerksamkeit!

Quellen

- https://developer.apple.com/documentation/uikit/uicollectionview
- https://developer.apple.com/documentation/uikit/views_and_controls/collection_views/layouts
- https://developer.apple.com/videos/play/wwdc2019/215
- https://medium.com/better-programming/spice-up-your-apps-collection-views-with-these-7-libraries-cda2379ce4d7
- https://medium.com/better-programming/ios-13-compositional-layouts-in-collectionview-90a574b410b8
- https://www.youtube.com/watch?v=EcffhZbHHmk
- https://developer.apple.com/videos/play/wwdc2019/215