

Diffable Data Sources

Péter Krassay - [@farkasseb](#)

Goals

- **Give a taste of Diffable Data Sources**
- Show some iOS 13 novelties



Agenda

- Motivation
- Diffable Data Sources
 - Snapshots
 - Live Coding
 - Problems, Considerables, Performance
 - Conclusion
- iOS 13 stuff



Motivation

- TableView / CollectionView?
 - Simple and flexible: 👍 ...right?
 - With constant static data: 😊
 - Constant static data, Core Data, filtering: 😊
 - Dynamic data from network with search/filtering, drag and drop, prefetching, editing mode, reordering, dynamic header and footer views: 💥🤔

NSInternalInconsistencyException

What's the problem?

- Data source vs. UI
 - State must always agree
- No single source of truth
- Who's responsibility?



Diffable Data Sources

- iOS, tvOS and macOS
- "A Declarative Approach to UI State"
- Simple, automatic diffing
 - *apply(_:animatingDifferences:completion:)*

Snapshots

- The UI always has a *snapshot* of truth
- Changes? Create and apply a new snapshot
 - Feed it to the UI and let it take care of the rest
- UI needs **unique identifiers** for *sections* and *items*
 - Conform to the **Hashable** protocol
 - No more IndexPaths 🎉 (well actually...)*

Classes

- Common
 - **NSDiffableDataSourceSnapshot**
- iOS, tvOS
 - **UICollectionViewDiffableDataSource**
 - **UITableViewDiffableDataSource**
- macOS
 - **NSCollectionViewDiffableDataSource**

Working with IndexPathes

- *Well actually...
 - We do have to keep interacting with IndexPath APIs
 - The conversion is usually simple and straightforward
 - Can be challenging and frustrating

Problems



Photo by Tim Gouw on Unsplash

What about updates?

- What should we do if the id is not. changing, only the content?
 - Ask for update manually!
 1. Ask for current snapshot: *dataSource.snapshot()*
 2. Ask to reload changed items: *reloadItems(withIdentifiers:)*
 3. Apply snapshot: *apply(_:animatingDifferences:completion:)*

What about section headers/footers?

- Subclass and Override
 - **UITableViewDiffableDataSource**
 - *tableView(_:titleForHeaderInSection:)*
 - **UICollectionViewDiffableDataSource**
 - *collectionView(_:viewForSupplementaryElementOfKind:at:)*
- Implement
 - **TableView**
 - *tableView(_:viewForHeaderInSection:)* and
 - *tableView(_:heightForHeaderInSection:)*
 - **Collection View**
 - *collectionView(_:viewForSupplementaryElementOfKind:at:)*

Documentation?

- *apply(_:animatingDifferences:completion:)*
 - What does *animatingDifferences* do? Seems straightforward enough? Give it a try!
- "No overview available."
 - nooverviewavailable.com

Performance and Considerables

- $O(n)$
- Can be called from a background thread!
 - BUT never mix: always call from the same thread (warning in log)
- Never use it with *performBatchUpdates(_:completion:)*! (warning in log)
- Share Sheets in iOS 13 uses Diffable Data Sources 🎉
 - As well as SwiftUI deep down somewhere

Conclusion

- Diffable Data Sources are no magic bullets
 - iOS 13+
 - It provides solution for a specific problem
 - Your mileage may vary
- Didn't cover *Compositional Layout* or *Swift UI*

Novelties

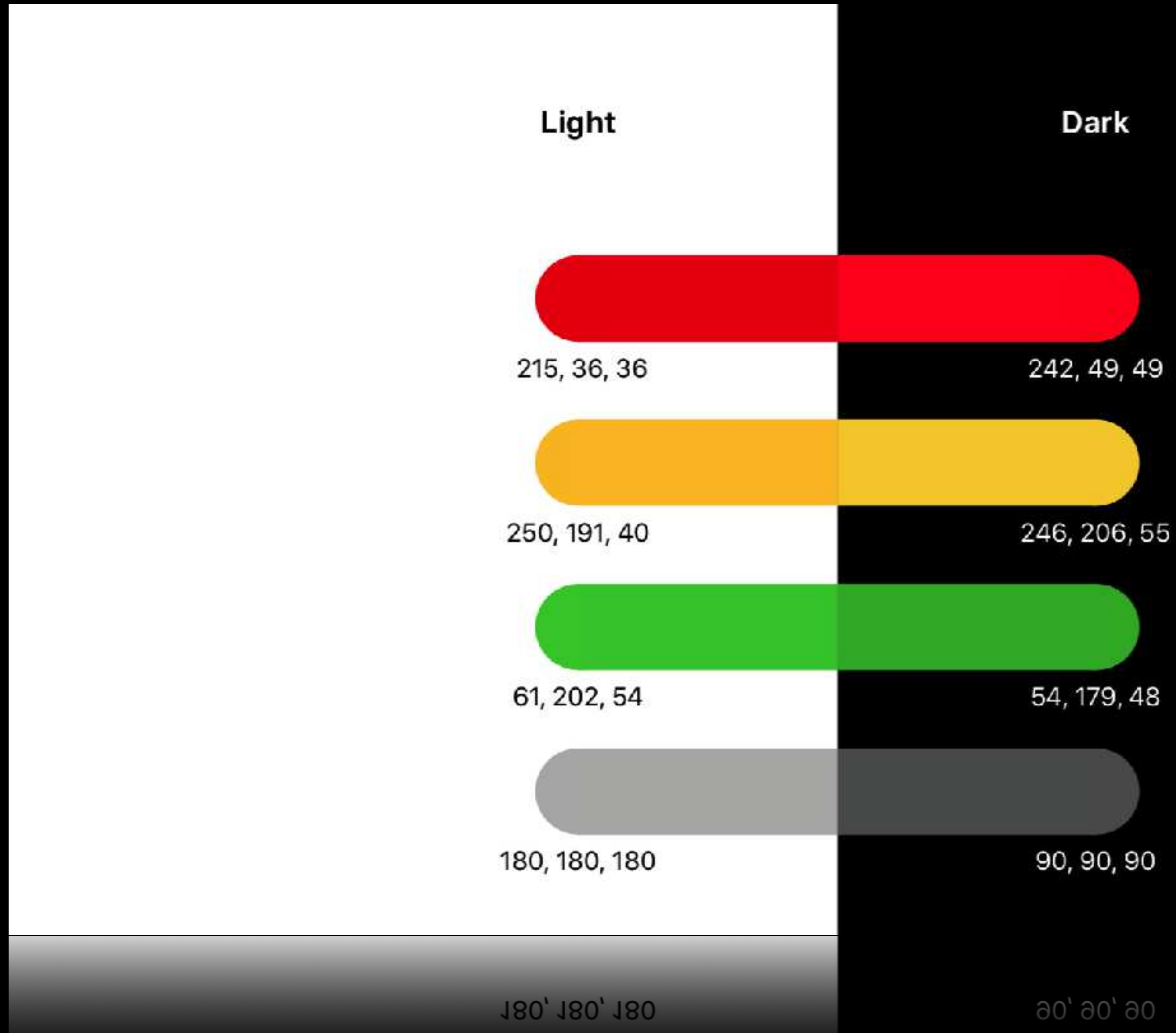


- Dark Mode
- SF Symbols
- Dependency Injection
- Relative Date Time Formatter

Dark Mode

- Use **semantic** colors!
 - *label, secondaryLabel, systemBackground* etc.
- Use **system** colors!
 - *systemRed* instead of *red*
 - Custom colors in Asset catalogs instead of fixed ones
- Prepare your **image assets** for dark Appearance!

System colors



SF Symbols

- 1500+ consistent, configurable symbols, ready to use
- Integrates seamlessly with San Francisco system font
- Vector assets with multiple weights
- *UIImage(systemName:)*

Dependency Injection

Remove Implicitly Unwrapped Optionals from View
Controllers Initialized from Storyboards

1. Change Implicitly Unwrapped Optional properties to normal properties
2. Implement the required initializer
3. Implement the custom initializer
4. *instantiateViewController(identifier:creator:)*

RelativeDateTimeFormatter

- **Localized**
- **DateTimeStyle**
 - *named*: prefers *deictic expressions*: "tomorrow", "yesterday", "next week"
 - *numeric*: "1 day ago"
- **UnitStyle**
 - *abbreviated*: "1 mo. ago" (may produce output distinct from *short* for non-English locales)
 - *full*: "1 month ago"
 - *short*: "1 mo. ago"
 - *spellOut*: "one month ago"

Additional resources

- WWDC 2019: [Advances in UI Data Sources](#)
- [Human Interface Guidelines - SF Symbols](#)
- Paul Hudson - Hacking with Swift Live!
- [NSHipster - Formatter](#)
- [NSHipster - Dark Mode on iOS 13](#)
- [NSHipster - iOS 13](#)

Thank you!

Please ask some questions!