

iOS 세미나 3

2023 와플스튜디오 루키 세미나

2023.10.29, 박신홍

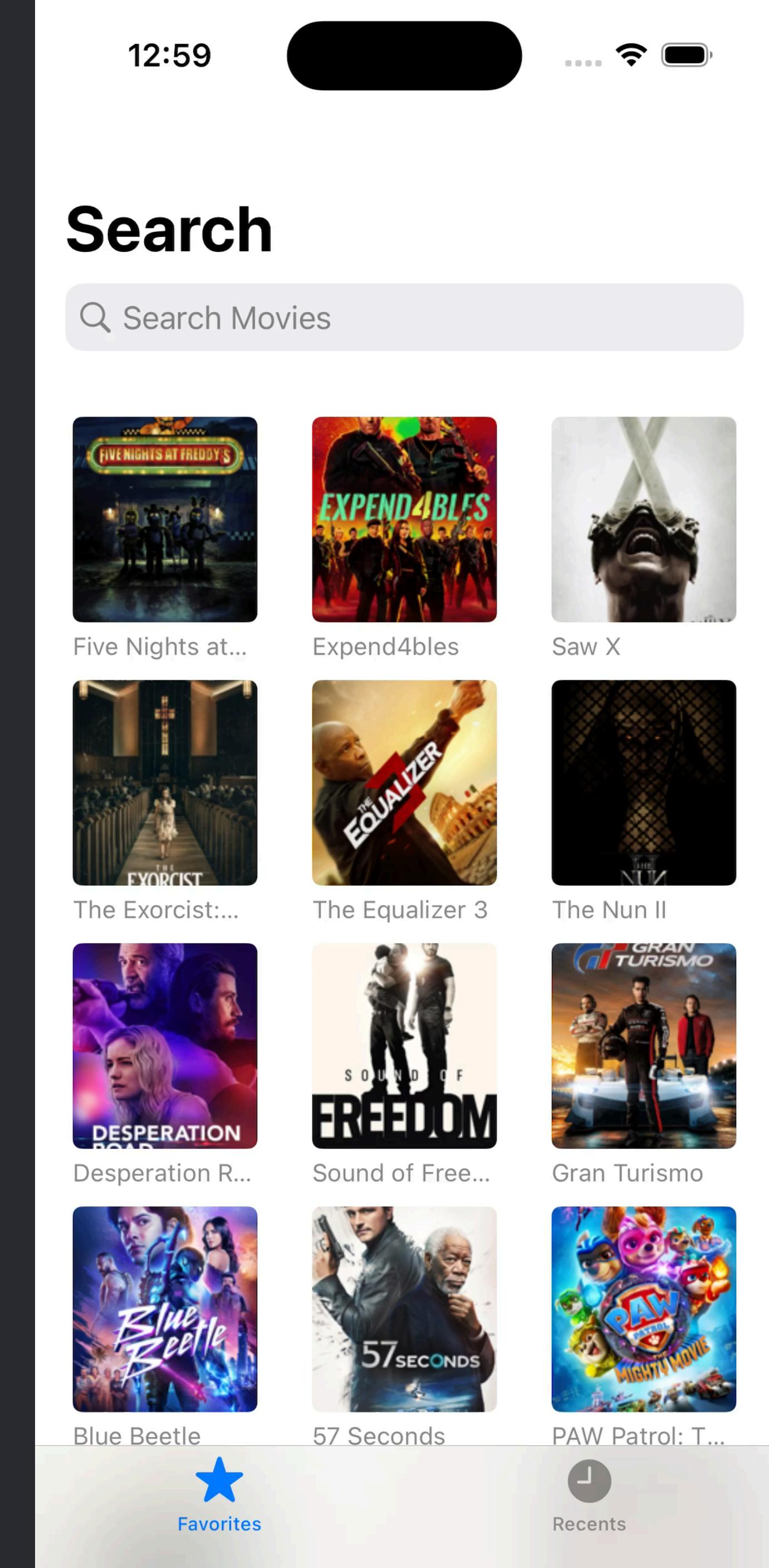
오늘 배울 내용

- 과제 1 리뷰
- Combine (feat. 반응형 프로그래밍)
- UICollectionView
 - DiffableDataSource
- UISearchController

Spoiler

과제 3: 넷플릭스 홈탭 만들기

- 첫 번째 탭 구현하기
- UICollectionView
 - DiffableDataSource
 - UICollectionViewFlowLayout
- Combine
- UISearchController



과제 1 공통 피드백

- 타이밍 이슈
 - 내 데이터가 어떤 플로우로 넘어가는지, 어떤 타이밍에 UI가 업데이트되는지 완벽히 이해한 채로 구현해야 합니다.
 - Ex) 데이터를 업데이트하고 UI를 업데이트해야 하는데, UI를 업데이트하고 다음 Cycle에 데이터를 업데이트를 해서 예상치 못하게 동작하는 경우
 - 이런 문제에 대해 가이드를 제공하는 것이 MVVM, 아키텍쳐, 디자인 패턴 등등의 개념
- 디버깅
 - Xcode LLDB po 명령어

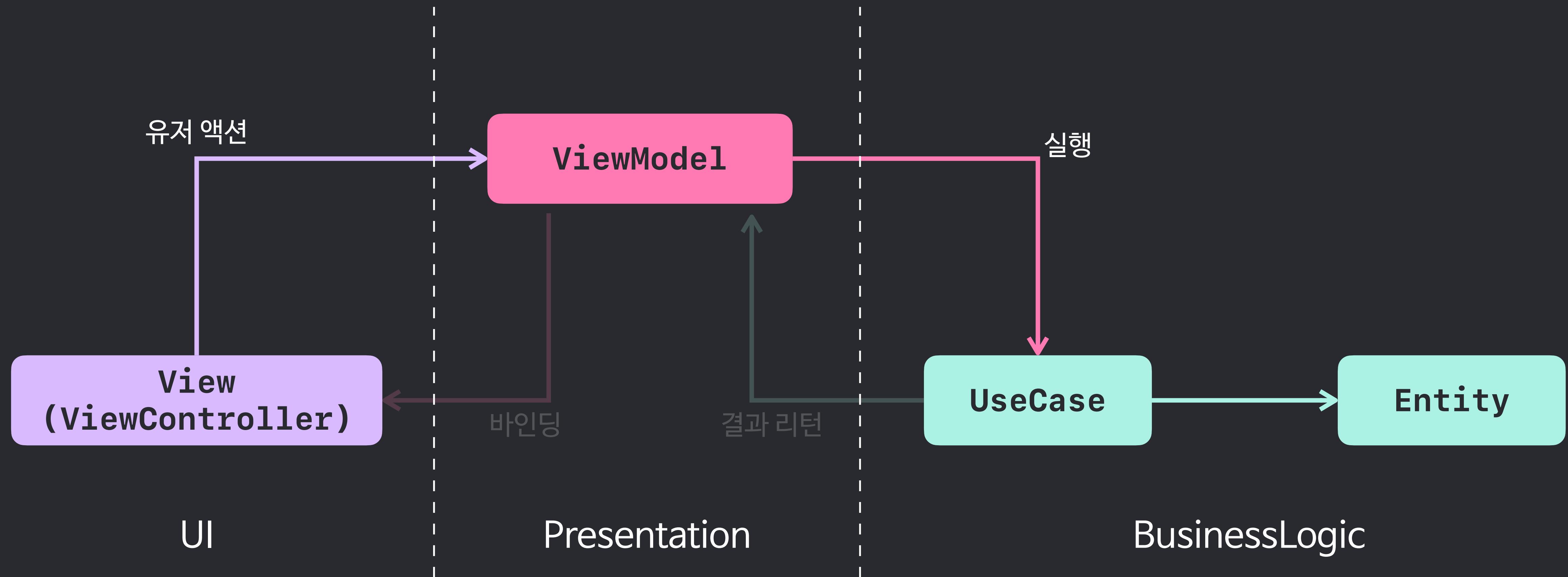
과제 1 공통 피드백

솔루션 코드 봐주세요!

seminar-2023 / ios / seminar1 / Seminar1Solution / Seminar1Solution /			Add file	...
Name	Last commit message	Last commit date		
..				
Assets.xcassets	Add Seminar1Solution			3 weeks ago
Base.lproj	Add Seminar1Solution			3 weeks ago
BusinessLogic	Add Seminar1Solution			3 weeks ago
Infra	Add Seminar1Solution			3 weeks ago
Presentation	Add Seminar1Solution			3 weeks ago
UI	Add Seminar1Solution			3 weeks ago
AppDelegate.swift	Add Seminar1Solution			3 weeks ago
Info.plist	Add Seminar1Solution			3 weeks ago
SceneDelegate.swift	Add Seminar1Solution			3 weeks ago

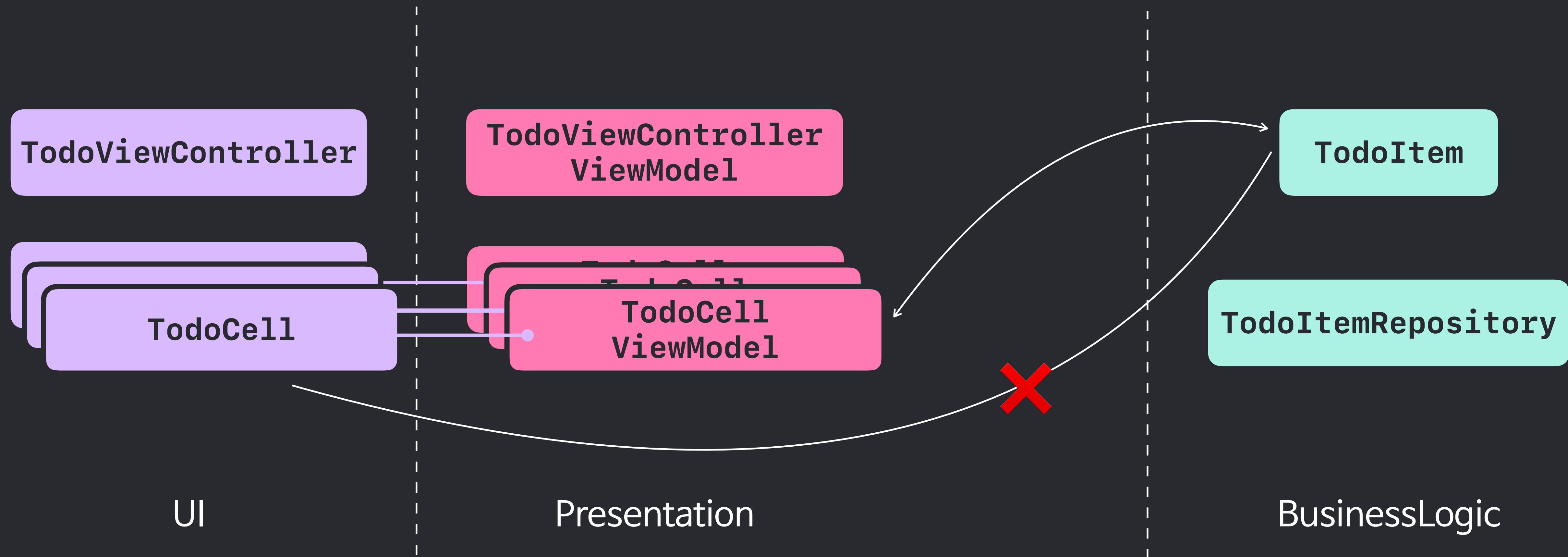
과제 1 공통 코드 리뷰

코드 아키텍쳐



과제 1 공통 코드 리뷰

코드 아키텍쳐



과제 1 공통 코드 리뷰

각 컴포넌트의 역할

- TodoViewControllerViewModel
 - TodoCellViewModel의 리스트를 관리하고,
 - 리스트에서 추가/제거가 발생했을 때 UI 레이어에게 알려주고,
 - TodoTableView를 구성할 때 필요한 데이터를 UI 레이어에게 제공한다.
- TodoCellViewModel
 - TodoCell 하나 하나를 그리기 위해 필요한 UI 로직들
 - TodoCell이 BusinessLogic 레이어에 있는 Todoltem과 직접 상호작용하지 않도록, 중간 다리 역할을 해준다.
 - TodoCell을 configure할 때는 Todoltem이 아니라 TodoCellViewModel을 사용한다.
 - 각 TodoCell은 Todoltem이 아니라 TodoCellViewModel을 참조하고 있다.

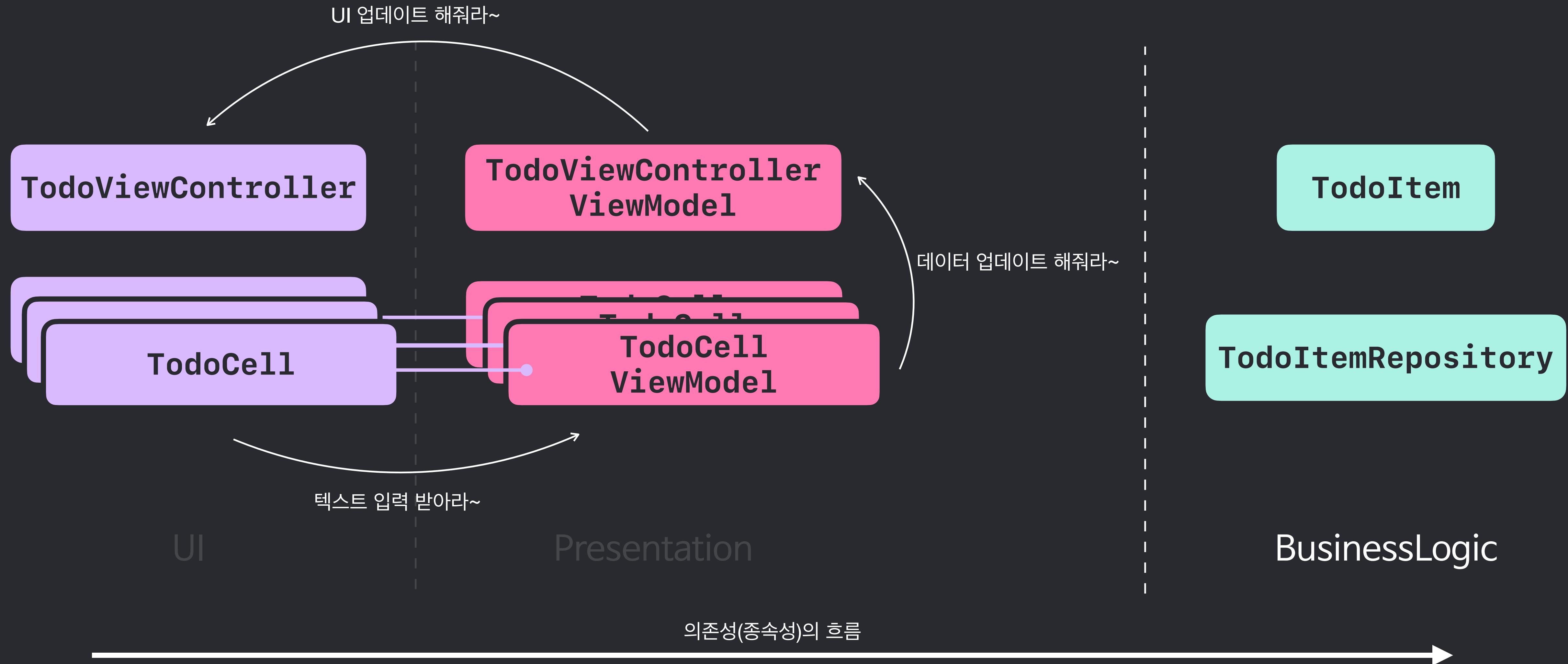
과제 1 공통 코드 리뷰

데이터 플로우

- TodoCell
 - 유저 인풋 받음
 - TodoCellViewModel에 입력받은 유저 인풋 전달 (e.g. 업데이트된 title 값 전달)
- TodoCellViewModel
 - 새로운 인풋으로 저장하고 있던 Todoltem 업데이트
 - TodoCellViewModelDelegate에게 Todoltem 업데이트되어야 한다고 알려줌
- TodoCellViewModelDelegate(=TodoViewControllerViewModel)
 - Todoltem의 id 값을 기준으로 TodoCellViewModel의 indexPath를 알아냄
 - TodoViewController에게 해당 indexPath 값을 업데이트하라고 알려줌

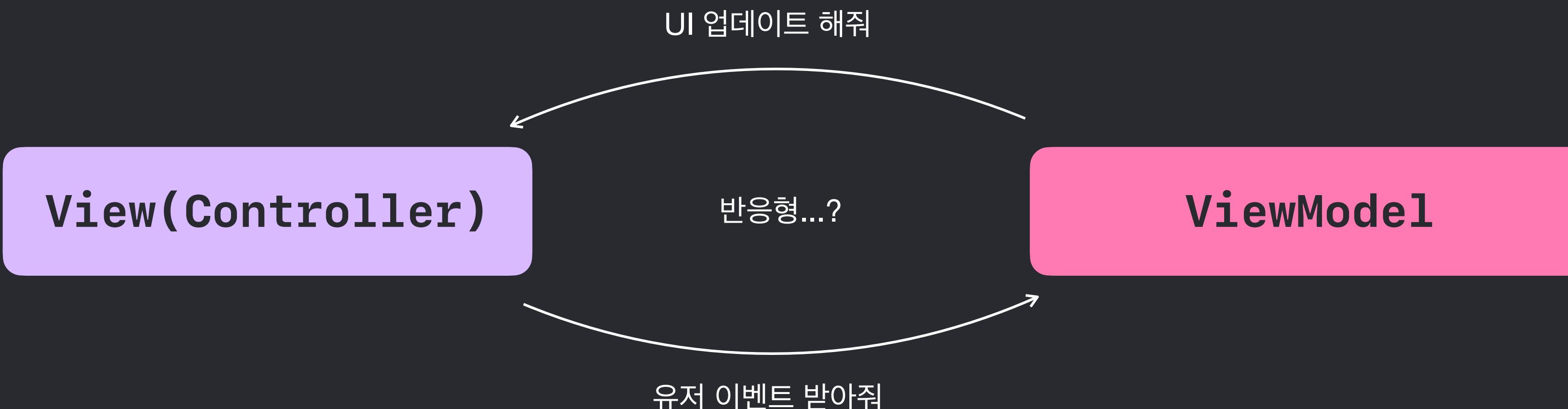
과제 1 공통 코드 리뷰

데이터 플로우



과제 1 공통 코드 리뷰

- UI가 유저 이벤트를 받아서, ViewModel에 전달해주는 건 지극히 자연스럽다.
- 그런데 ViewModel이 UI 레이어에 이벤트 전달하는 게 좀 이상한 것 같다.. 꼭 delegate 패턴을 사용해야만 하는걸까?
- 참고로 지금까지 보여드린 delegate 기반 데이터 플로우도 협업에서 자주 사용되는 근본 패턴이긴 합니다.



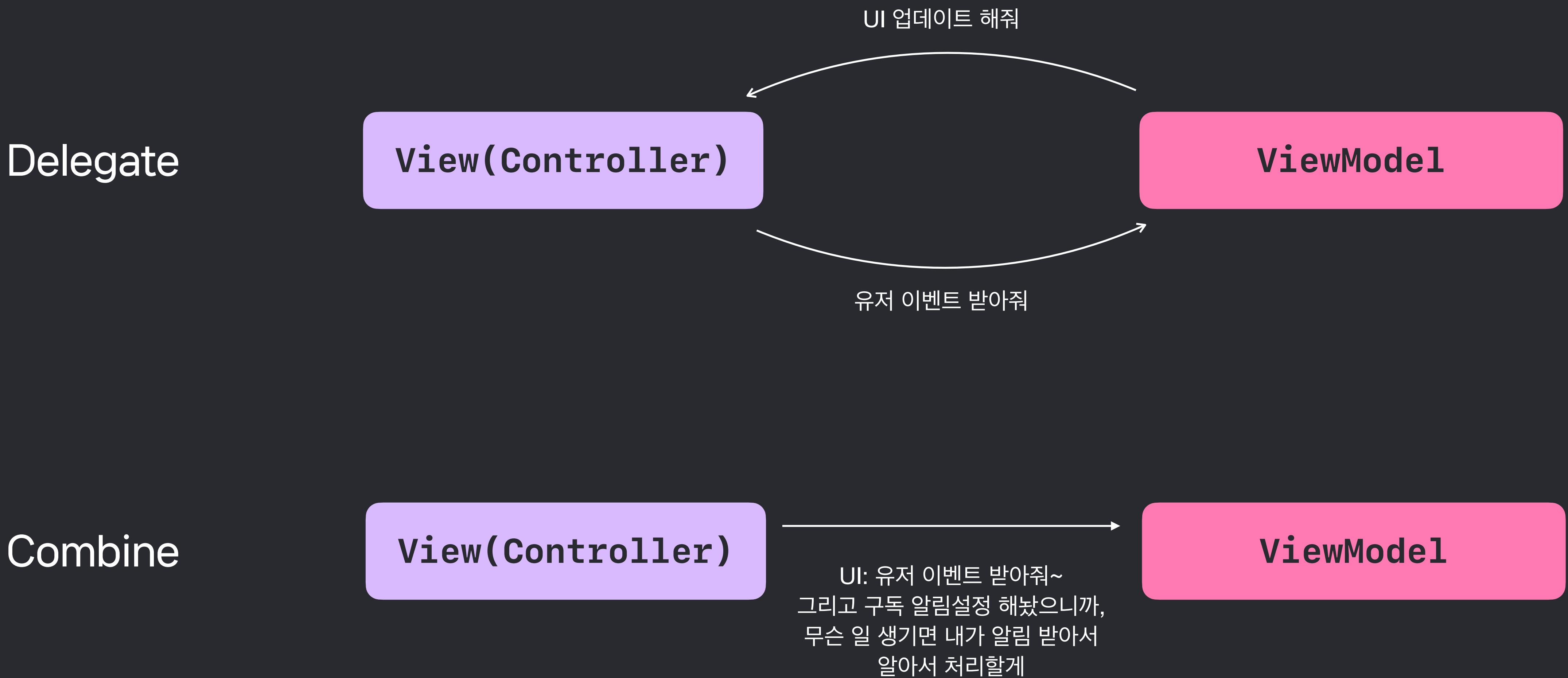
Combine

반응형 프로그래밍을 더 쉽게



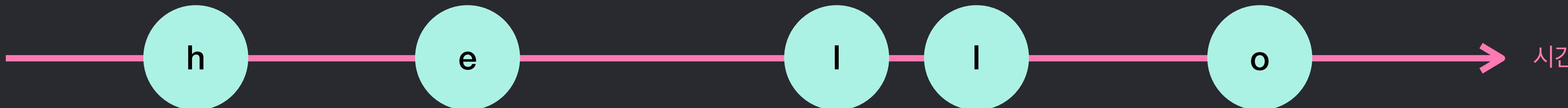
- Combine은 애플에서 반응형 프로그래밍을 쉽게 할 수 있도록 만들어놓은 퍼스트파티 프레임워크
- SwiftUI가 내부적으로 Combine을 많이 씀
- 원조는 서드파티인 RxSwift 인데, 애플이 좀 더 최적화해서 출시
 - 성능은 Combine이 더 좋지만, RxSwift가 UIKit 관련 편의기능이 많아서 RxSwift 쓰는 회사도 많음
 - 라떼는 RxSwift로 배움

그래서 Combine이 뭔가요?



구독? 뭘 구독하나요?

- 이벤트 스트림을 구독합니다
 - 이벤트 스트림: 시간의 흐름에 따라 발생하는 일련의 사건들
 - 예시: 유저가 검색창에 "hello"라고 입력할 때





- Combine 실습
- CurrentValueSubject & PassthroughSubject
- eraseToAnyPublisher

Combine의 이름이 Combine인 이유

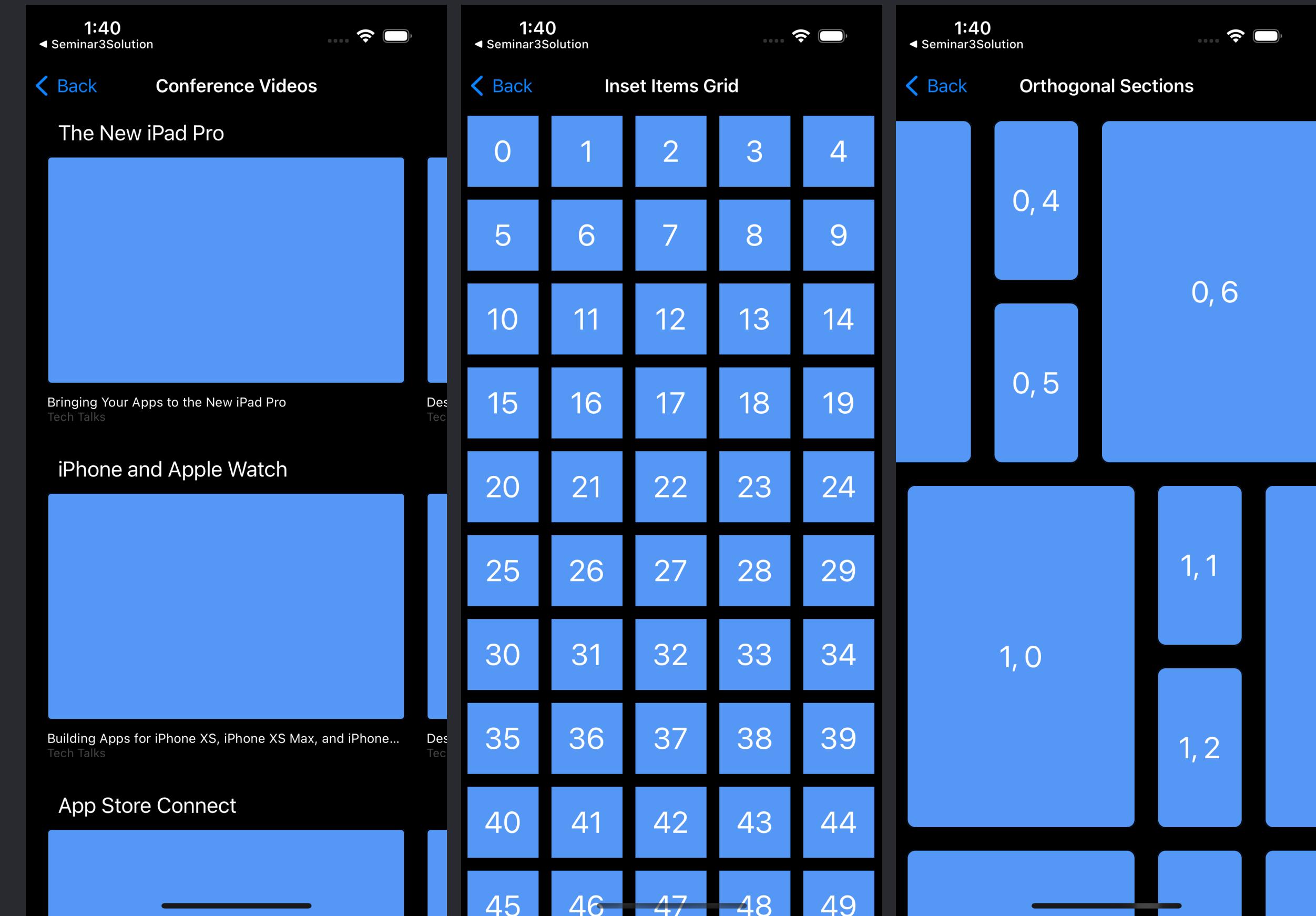
= 여러 Operator를 Combine할 수 있기 때문

- Operator를 중첩해서 사용함으로써 복잡한 로직을 간편하게 표현할 수 있음
 - debounce, filter, flatMap, ...
- 굉장히 많은 Operator가 있는데, 공식 문서에 적혀 있는 설명만 봐서는 이해하기 어려울 때가 있음
- 시각화 사이트: <https://rxmarbles.com/>
- Combine이 Rx 베낀거라 웬만하면 다 통한다!

UICollectionView

UITableView의 진화버전

- 별의별 레이아웃을 만들 수 있음
 - 자유도가 높은 만큼 익히기가 조금 복잡한 편
 - https://developer.apple.com/documentation/uikit/views_and_controls/collection_views/implementing_modern_collection_views
 - 샘플 코드가 아주 잘돼있어요



UICollectionView

- Delegate 메서드나 레이아웃 관련 코드는 조금 다르지만 UITableView와 원리는 결국 똑같다. (사실 모든 UIKit 컴포넌트는 사용하는 방법이 비슷비슷함)

collectionView(_:cellForItemAt:)

Asks your data source object for the cell that corresponds to the specified item in the collection view.

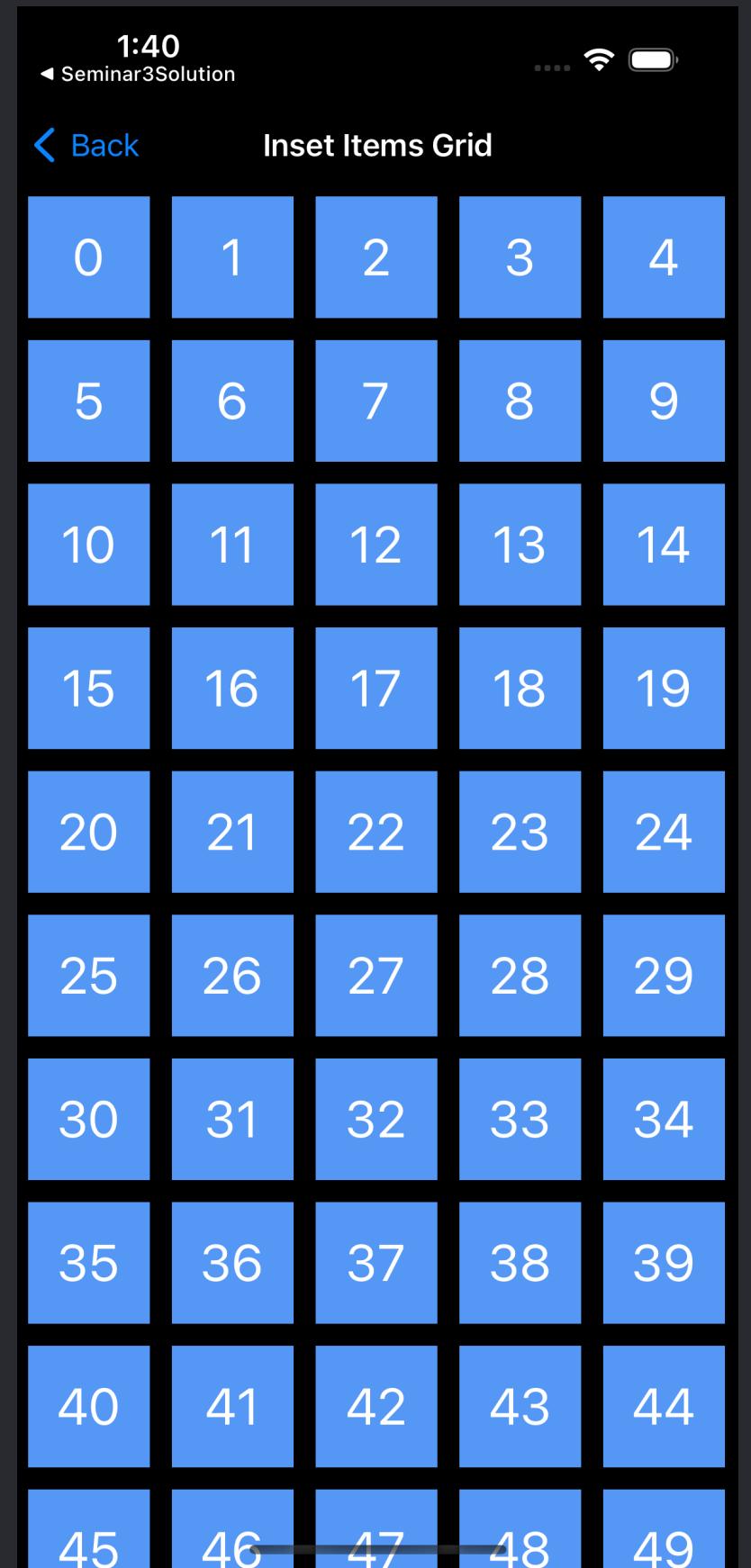
iOS 6.0+ iPadOS 6.0+ Mac Catalyst 13.0+ tvOS 9.0+ visionOS 1.0+ Beta

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

이제 익숙하죠?

UICollectionViewFlowLayout

- 오른쪽과 같이 단순한 레이아웃을 편하게 만들 수 있도록 제공되는 프로토콜
- 구현이 간편한 만큼 복잡한 커스터마이즈는 제한된다.
- 우리 과제에서는 이것을 사용할 예정!
 - 앱스토어나 넷플릭스같은 레이아웃을 구현하고 싶다면 UICollectionViewCompositionalLayout 사용 (예제 코드는 앞 슬라이드 링크에)
 - 이거 써서 멋진 레이아웃 구현하신다면 **과제 Grace day 3일 드립니다.**





- UICollectionViewFlowLayout
실습

DiffableDataSource

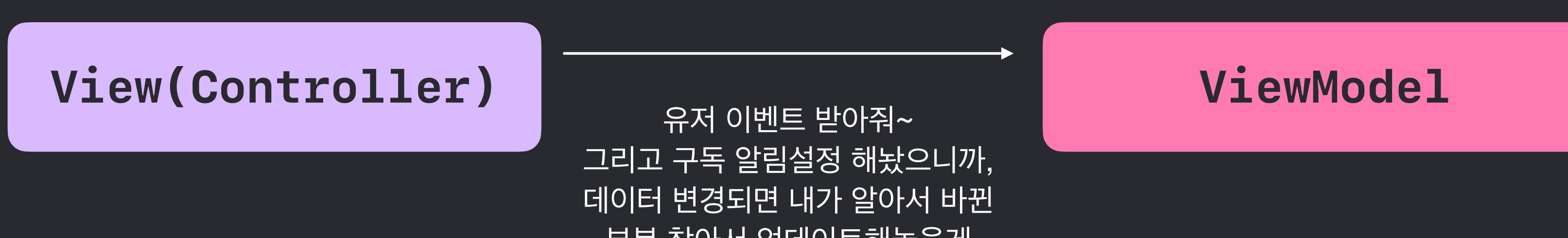
- 데이터가 바뀔 때마다 insertRows, reloadData, performBatchUpdates 적절히 호출해주는 건 생각보다 귀찮은 일
- 알아서 바뀐 부분 잘 찾아서 적당히 애니메이션 해줬으면 좋겠다!

```
var snapshot = NSDiffableDataSourceSnapshot<RecipeListSection, Recipe.ID>()
snapshot.appendSections([.main])
snapshot.appendItems(selectedRecipeIds, toSection: .main)
recipeListDataSource.apply(snapshot, animatingDifferences: true)
```

- 이렇게 snapshot을 준비해서 dataSource에 apply하면, O(1) 시간에 바뀐 부분을 찾아서 해당 부분만 업데이트해준다. 예쁜 애니메이션은 덤.

DiffableDataSource + Combine

- 둘의 궁합은 매우 좋음
- collectionView 또는 tableView를 어떻게 업데이트할지는 신경 안써도 되고, 데이터를 어떻게 불러올지만 집중하면 됨
- ViewModel에서 데이터를 잘 업데이트만 해주면, 별도 delegate 메서드를 호출해주지 않아도 collectionView가 자동으로 업데이트된다!





- Combine + UICollectionView + DiffableDataSource

Spoiler

과제 3: 넷플릭스 홈탭 만들기

- 첫 번째 탭 구현하기
- UICollectionView
 - DiffableDataSource
 - UICollectionViewFlowLayout
- Combine
- UISearchController

