Coordinatorパターンの実践

@yoshikuni_kato

potatotips #38
2017/03/22

Who am I?

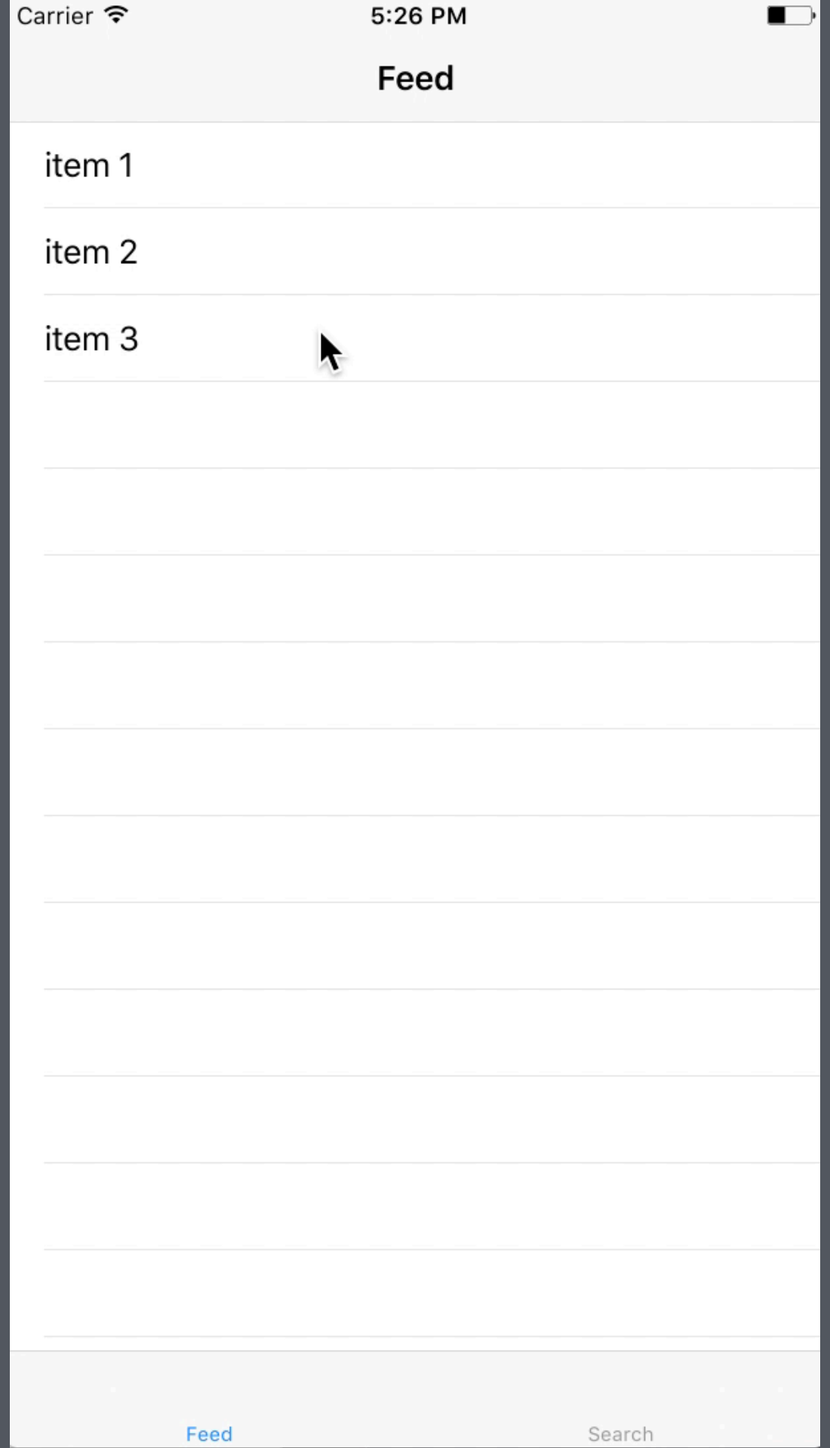
- 加藤由訓（Yoshikuni Kato）
  @yoshikuni_kato
- iOSエンジニア
- Yahoo! Japan -> オハコ
- 「ラジへぇ」くん →
- Interests: 設計 / FRP / Coordinator Pattern / UI実装

Agenda

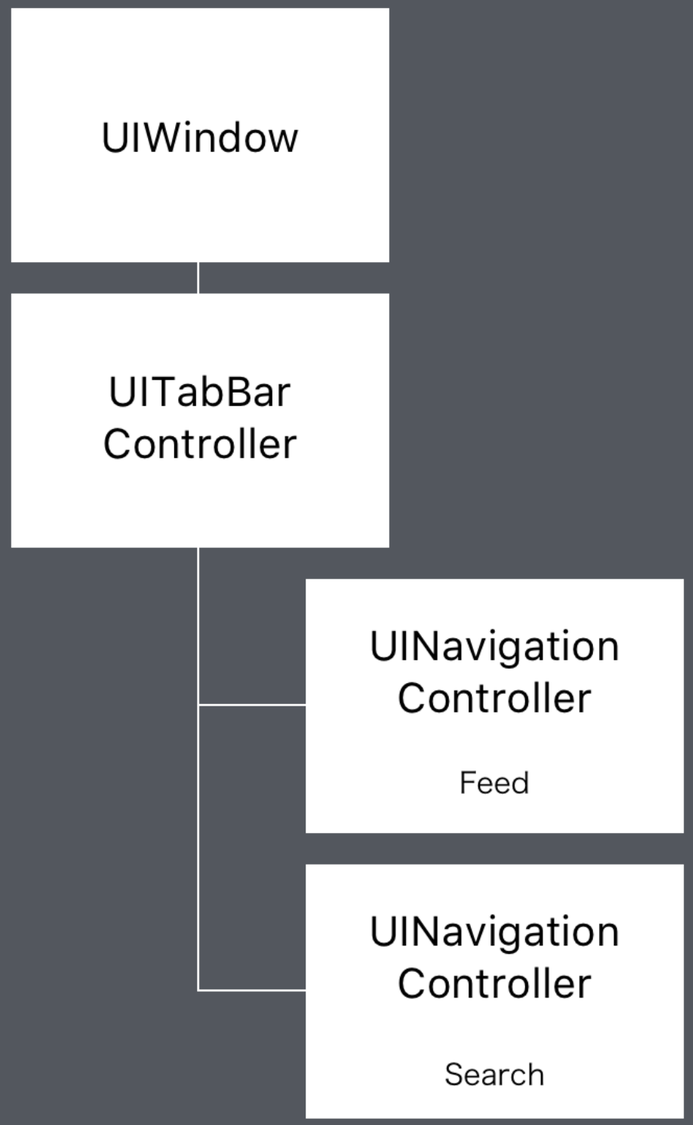
- Coordinatorパターンの実装例を紹介したい
- Coordinator: 画面遷移のロジックをVCから切り出したもの
  - 過去の資料（画面遷移の管理とMVVM）を参照ください

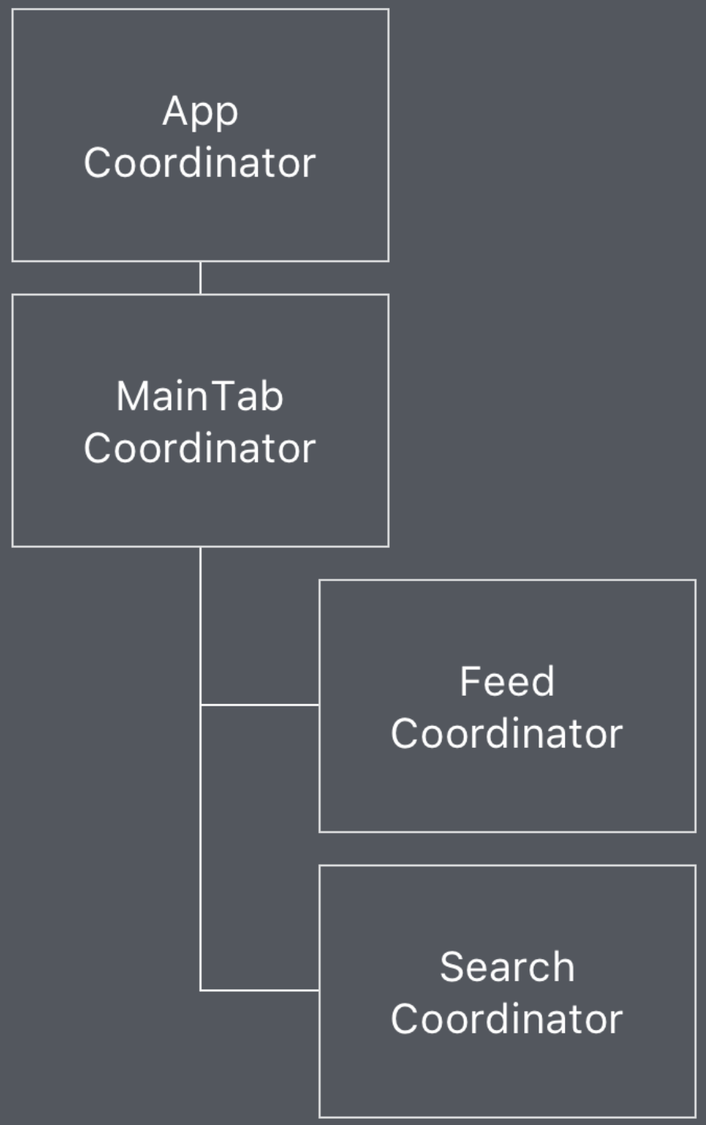
ソースコード

https://github.com/yoching/CoordinatorSample


Carrier
5:26 PM
Feed
item 1
item 2
item 3
Feed
Search

全体構成


UIWindow
UITabBar Controller
UINavigation Controller
Feed
UINavigation Controller
Search


App Coordinator
MainTab Coordinator
Feed Coordinator
Search Coordinator

Coordinatorの作成単位

- 「中に複数のVCを持つVC」単位でCoordinatorを作ると分かりやすい
  - UINavigationController
  - UITabBarController
  - containerViewを利用しているViewController
- UX的な分割とも一致するはず

Coordinator protocol

```
protocol Coordinator {
    var presenter: UIViewController { get }
    func start()
}
```

NavigationCoordinator protocol

```
protocol NavigationCoordinator: Coordinator {
    var navigationController: UINavigationController { get }
}

extension NavigationCoordinator {
    var presenter: UIViewController {
        return navigationController as UIViewController
    }
}
```

TabBarCoordinator protocol

```
protocol TabBarCoordinator: Coordinator {
    var tabBarController: UITabBarController { get }
}

extension TabBarCoordinator {
    var presenter: UIViewController {
        return tabBarController as UIViewController
    }
}
```

AppCoordinator

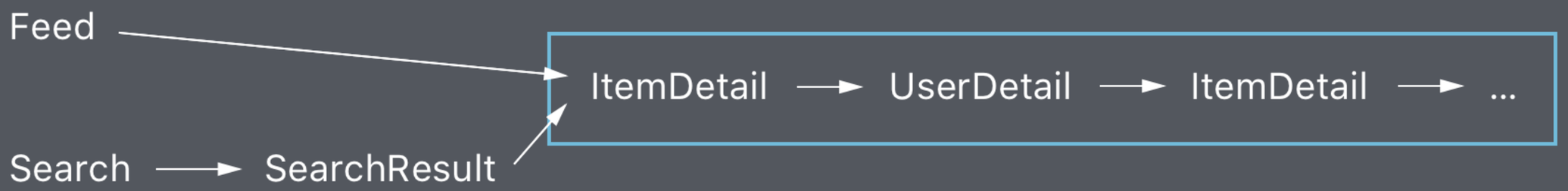
```
final class AppCoordinator {
    private let window: UIWindow
    private let rootCoordinator: Coordinator
    init(window: UIWindow, rootCoordinator: Coordinator) {
        self.window = window
        self.rootCoordinator = rootCoordinator
    }
    func start() {
        rootCoordinator.start()
        window.rootViewController = rootCoordinator.presenter
        window.makeKeyAndVisible()
    }
}
```

MainTabCoordinator

```
final class MainTabCoordinator: TabBarCoordinator {
    let tabBarController: UITabBarController
    private let childCoordinators: [Coordinator]
    init(presenter: UITabBarController, childCoordinators: [Coordinator]) {
        self.tabBarController = presenter
        self.childCoordinators = childCoordinators
    }
    func start() {
        childCoordinators.forEach { coordinator in
            coordinator.start()
        }
        tabBarController.setViewControllers(
            childCoordinators.map { $0.presenter },
            animated: false
        )
    }
}
```

2つのNavigationで共通する遷移がある

- 遷移



→ protocol-orientedに解決

DetailsPresentable protocol

```
protocol DetailsPresentable {
    func showItemDetail(item: Item)
    func showUserDetail(user: User)
}
```

DetailsPresentable extension

```
extension DetailsPresentable where Self: NavigationCoordinator {
    func showItemDetail(item: Item) {
        let itemDetailVC = StoryboardScene.ItemDetailViewController.initialViewController()
        itemDetailVC.item = item
        itemDetailVC.userTapped = showUserDetail
        navigationController.pushViewController(itemDetailVC, animated: true)
    }

    func showUserDetail(user: User) {
      // ...
    }
}
```

※ VC→Coordinatorの通信は、closureを介す / delegate / Observable(FRP) など

FeedCoordinator

```
final class FeedCoordinator: NavigationCoordinator, DetailsPresentable {

    let navigationController: UINavigationController

    init(presenter: UINavigationController) {
        self.navigationController = presenter
        presenter.title = "Feed"
    }

    func start() {
        let feedViewController = StoryboardScene.FeedViewController.initialViewController()
        feedViewController.itemSelected = showItemDetail
        navigationController.pushViewController(feedViewController, animated: false)
    }
}
```

Summary

- Coordinatorの作成例の紹介
  - https://github.com/yoching/CoordinatorSample
  - どう分割するか
- 共通処理をprotocol-orientedに解決する

Thank you!

参考

- 画面遷移の管理とMVVM

- Presenting Coordinators by Soroush Khanlou, NSSpain(2015)
- Boundaries in Practice by Nonaka Ayaka, try!Swift(2016)
- MVVM-C In Practice by Steve Scott, UIKonf(2016)
- Connecting View Controllers at Swift Talk(objc.io), 2016