ƛ 😍

Intro to functional reactive programming

FRP & Architectural Patterns
November 13, 2016

What’s *wrong* with MVC?

lots of **different communication patterns** (no coherent programming model)

massive view controllers - too easy to put code into the “controller”

too easy to **neglect failure scenarios**

bad testability when code is put into view controllers

The Rise of FRP

FRP on iOS has become vastly more **popular because of Swift**

multiple FRP frameworks (ReactiveCocoa, RxSwift, ReactiveKit,…)

premise of **simple and coherent programming model**

In an **ideal world**, we don’t need a Controller.

❤️

We only have **Data** and **Views.**

The main idea of FRP is that the

UI should be a constant reflection of the *application state*.

```
func createUI(appState: AppState) -> UI {
  //
}
```

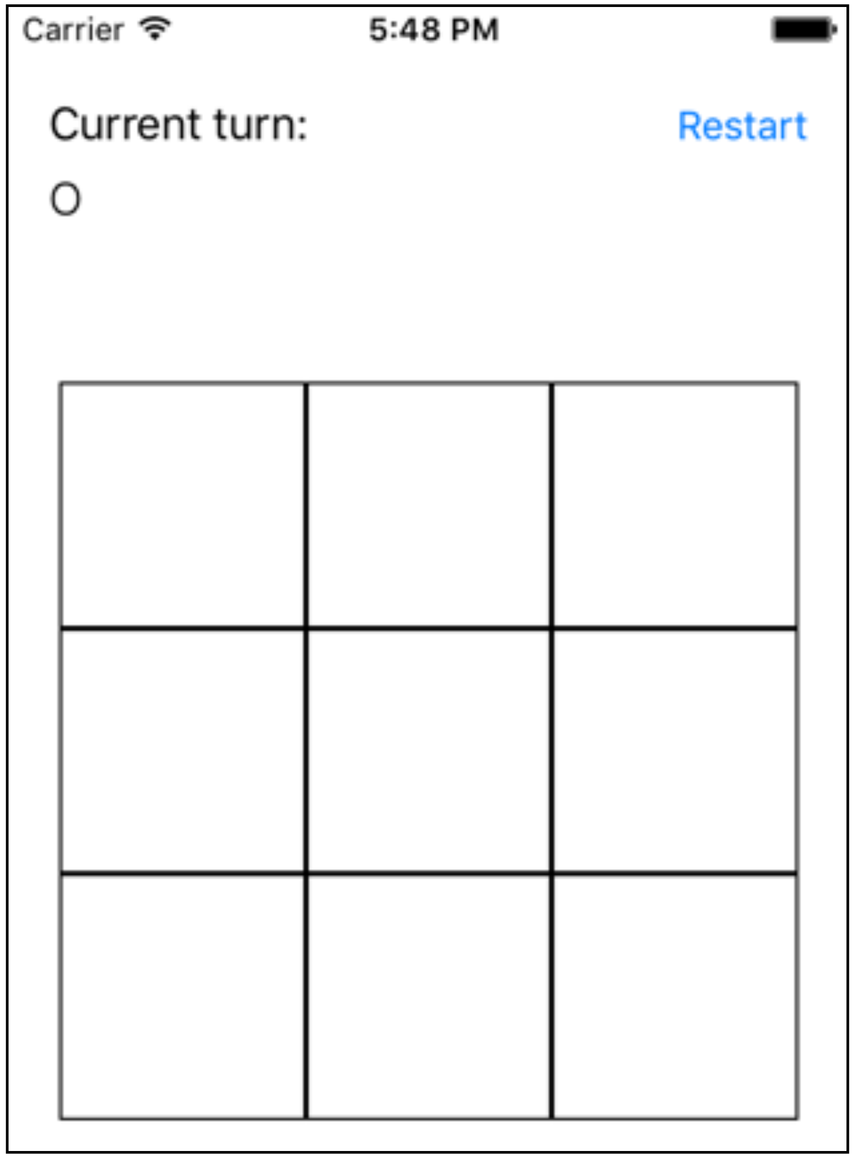
Application State

```
struct Board {

   var currentTurn: Marker = .circle

   var fields: [[Field]] = [
     [.free, .free, .free],
     [.free, .free, .free],
     [.free, .free, .free]
  ]

}
```



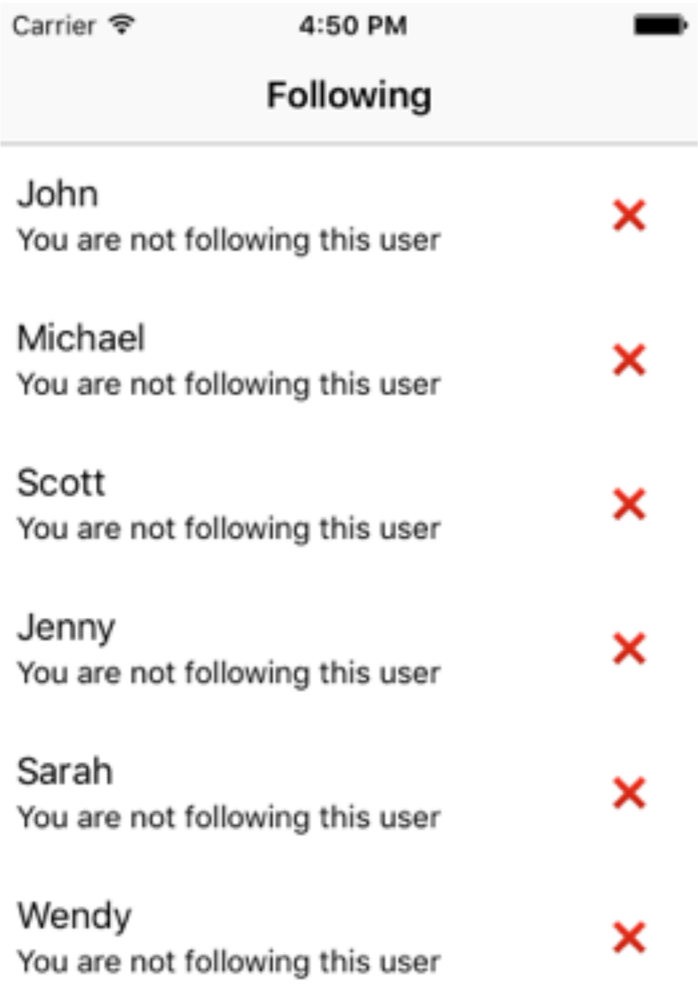
Carrier 5:48 PM

Current turn:

O

Application State

```
var users: [User] =  [
  User(name: "John", isFollowing: false),
  User(name: "Michael", isFollowing: false),
  User(name: "Scott", isFollowing: false),
  User(name: "Jenny", isFollowing: false),
  User(name: "Sarah", isFollowing: false),
  User(name: "Wendy", isFollowing: false),
]
```



Carrier 4:50 PM

Following

John
You are not following this user ✕

Michael
You are not following this user ✕

Scott
You are not following this user ✕

Jenny
You are not following this user ✕

Sarah
You are not following this user ✕

Wendy
You are not following this user ✕