

Α

-<u>Ö</u>-

## 上一章

## 8.8 移除和添加照片卡牌

本节,我们将通过编写一个函数来实现移除与添加照片卡牌的功能。

首先,在ContentView中的Properties部分添加一个变量lastCardIndex,然后为cardView计算属性添加@State封装,因为我们需要在cardViews数组发生变化的时候更新用户界面。

```
// MARK: - Properties
@State private var lastCardIndex = 1

// MARK: - CardViews
@State var cardViews: [CardView] = {
  var views = [CardView]()
  ......
}()
```

接下来,我们需要再添加一个moveCards()方法。

```
// MARK: - 移动卡片
private func moveCards() {
  cardViews.removeFirst()

  self.lastCardIndex += 1

  let inYourHeart = inYourHeartData[lastCardIndex % inYourHeartData.count]
  let newCardView = CardView(photo: inYourHeart)
  cardViews.append(newCardView)
}
```

在moveCards()方法中,先移除cardViews中位于顶部的卡片视图,然后将索引值加1。重点在于后面两行代码,先从inYourHeartData数组中取出索引值为lastCardIndex的照片信息,将其初始化为CardView类型的视图,再添加到cardViews数组中。

最后,我们还需要为gesture修饰器添加一个onEnded,一旦用户完成照片卡牌的 拖曳操作,就会进行卡牌移除或保持原样的相关操作。

当用户完成拖曳操作后,可以利用onEnded获取相关信息,一旦偏移量大于65或小于-65则执行moveCards()方法。

在预览窗口中启动Live模式,我们可以测试拖曳照片卡牌的运行效果。

通过本章的学习,我们继续巩固了使用Xcode 12创建iOS应用程序的方法,为项目创建启动画面,并构建独立的卡片视图页面。使用SwiftUI为项目创建Header和Footer视图,创建可复用视图组件。本章,我们还学会了有关Binding的知识,使用计算属性实现照片卡牌组,利用SwiftUI实现滑动手势,在滑动的过程中显示喜爱或者不喜爱的图标等。

全书完

已阅读 12 小时 17 分钟