上一章

5.3 创建图像滑动视图

在这一节中我们将使用SwiftUI框架创建一个图像滑动视图。如果你已经学习完了 之前章节的内容,并且没有任何疑惑,那么本节所要完成的任务对于你来讲不会存 在任何压力。我们还是需要先载入JSON格式的数据,然后针对Bundle类创建一个 decode扩展方法,最后创建图像滑动视图。

5.3.1 创建数据模型

这里所创建的数据模型文件,应该与Data文件夹中的header.json文件内容相匹 配。在Model文件夹中新建一个Swift类型的文件,将其命名为HeaderModel。修 改文件中的代码如下。

```
import Foundation
struct Header: Codable, Identifiable {
  let id: Int
  let image: String
```

作为数据模型,它需要符合Codable协议,否则无法将解析后的JSON格式数据写 入该模型中。Identifiable则用于确定每个数据的唯一性,所以必须有一个id属性。

5.3.2 创建JSON解析方法

接下来我们需要为Bundle创建一个扩展方法,我们通过该方法来解析JSON格式的 数据文件。

在 Extension文 件夹中新建一个 Swift类型的文件,将其命名为 CodableBundleExtension。 该文件中涉及的代码,与第4章的 CodableBundleExtension.swift文件一样,所以直接将代码复制过来即可。

让我们回到Constant.swift文件中,在//Data注释语句的下面添加一行新的常量定 义。这样,我们就可以直接在程序中使用这些图像了。

```
// Data
let headers: [Header] = Bundle.main.decode("header.json")
```

5.3.3 创建用于滑动的图像视图

我们将数据准备好后,就可以为滑动视图创建相应的图像视图了。在View/Home 文件夹中新建一个SwiftUI类型文件,将其命名为HeaderItemView。修改文件中的 代码如下。

```
struct HeaderItemView: View {
  // MARK: - Properties
  let header: Header
  // MARK: - Body
  var body: some View {
   Image(header.image)
     .resizable()
      .scaledToFit()
      .cornerRadius(12)
// MARK: - Preview
struct HeaderItemView_Previews: PreviewProvider {
  static var previews: some View {
   HeaderItemView(header: headers[1])
     .previewLayout(.sizeThatFits)
     .padding()
      .background(colorBackground)
```

在预览窗口中可以看到图5-15所示的效果。

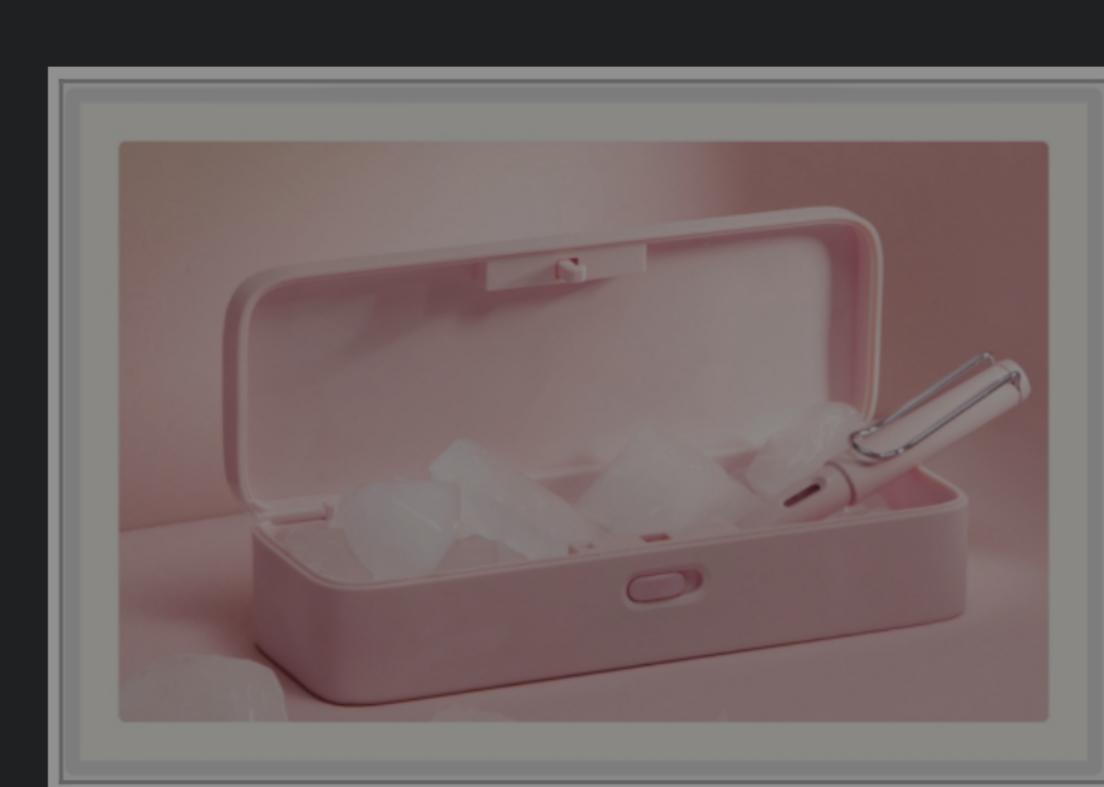


图5-15 HeaderItemView在预览窗口中的效果

接下来,我们创建一个独立的图像滑动视图,在该视图中呈现图片滑动的效果。

在View/Home文件夹中新建一个SwiftUI类型文件,将其命名为HeaderTabView。 修改文件中的代码如下。

```
// MARK: - Body
var body: some View {
  TabView {
    ForEach(headers) { item in
     HeaderItemView(header: item)
       .padding(.top, 10)
       .padding(.horizontal, 15)
   } //: Loop
  } //: TabView
  .tabViewStyle(PageTabViewStyle(indexDisplayMode: .always))
```

我们使用TabView创建图像滑动视图,通过ForEach循环遍历出所有的Header图 像,然后为TabView添加tabViewStyle修饰器,参数PageTabViewStyle用于呈现 翻页风格,并且让翻页指示点永远显示在屏幕上,只不过目前的指示点在屏幕的底 部。在预览窗口中启动Live模式,可以看到如图5-16所示的效果。我们还可以滑动 鼠标让其左右翻页。

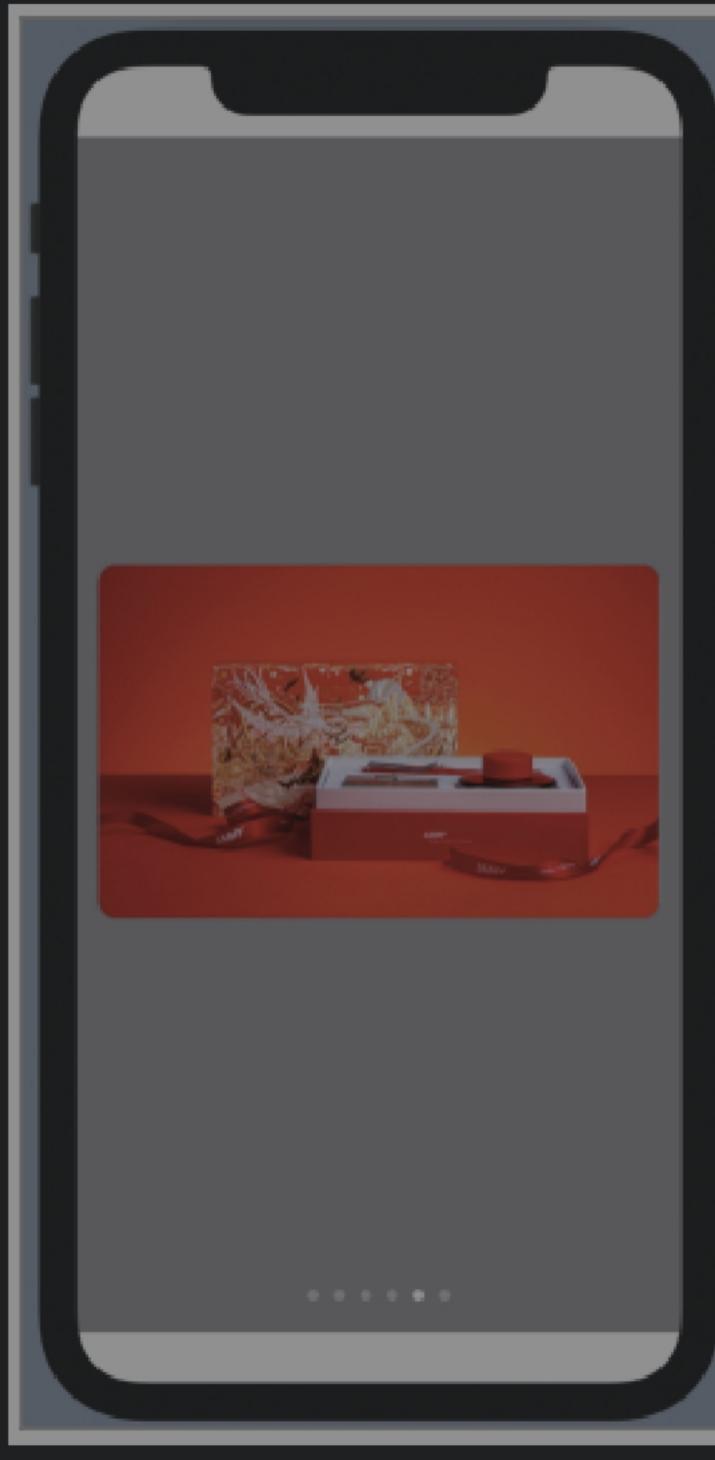


图5-16 图像滑动视图的效果

5.3.4 将图像滑动视图添加到主场景视图

最后,我们还需要将HeaderTabView添加到ContentView中,将ContentView代码 修改如下。

```
ZStack {
  VStack{
   NavigationBarView()
     .padding(.horizontal, 15)
     .padding(.bottom)
     .padding(.top, UIApplication.shared.windows.first?.safeAreaInsets.top)
     .background(Color.white)
     .shadow(color: Color.black.opacity(0.05), radius: 5, x: 0, y: 5)
   ScrollView(.vertical, showsIndicators: false) {
     VStack(spacing: 0) {
       HeaderTabView()
         .padding(.vertical, 20)
       FooterView()
         .padding(.horizontal)
      } //: VStack
    } //: ScrollView
  } //: VStack
  .background(colorBackground.ignoresSafeArea(.all, edges: .all))
} //: ZStack
.ignoresSafeArea(.all, edges: .top)
```

在ScrollView里有一个VStack容器,里面包含了图像滑动视图和FooterView,在 预览窗口中启动Live模式,效果如图5-17所示。



图5-17 图像滑动视图在ContentView中的效果