•

上一章

8.2 卡片视图

8.2.1 创建卡片视图的数据模型

在项目中新创建一个Model文件夹,在该文件夹中新建一个Swift类型的文件,并将 其命名为InYourHeartModel,修改该文件的代码如下。

本节,我们将创建一个独立的视图页面,通过它来呈现Photos中的图片。

```
import SwiftUI

struct Destination {
  var place: String
  var country: String
  var image: String
}
```

Destination结构体用于定义图片的名称和所拍摄的国家、地区,该结构体涉及的三个属性均为字符串类型。

在项目中新创建一个Data文件夹,在该文件夹中新建一个Swift类型的文件,并将 其命名为InYourHeartData。我们在该文件中会创建与图片相关的模型数据信息。

修改文件中的代码如下。

```
import SwiftUI

var inYourHeartData: [Destination] =
[
   Destination(
    place: "定都阁",
    country: "中国",
    image: "building01"),
    ......
]
```

我们所定义的inYourHeartData数组会包含所有25张图片的数据信息,你可以直接从"项目资源/Data"文件夹中打开InYourHeartData.txt文件,直接将数据信息复制到数组中。

8.2.2 创建卡片视图

在创建好数据模型和相关的数据信息后,就可以在ContentView中创建用户界面了。在项目中新创建一个View文件夹,然后将ContentView.swift文件拖曳到该文件夹中。

在View文件夹中新建一个SwiftUI类型的文件,将其命名为CardView。

首先修改Properties部分的代码如下。

```
//MARK: - Properties
let id = UUID()
var photo: Destination
```

id属性用于标识每一张照片,Destination类型的photo属性则是照片的数据信息。

在Preview中,我们需要为CardView添加photo参数值。另外,需要指定预览窗口的尺寸为固定值375×600。

```
//MARK: - Preview
struct CardView_Previews: PreviewProvider {
  static var previews: some View {
    CardView(photo: inYourHeartData[0])
    .previewLayout(.fixed(width: 375, height: 600))
  }
}
```

继续修改Body部分的代码如下。

```
//MARK: - Body
var body: some View {
  Image (photo.image)
    .resizable()
    .cornerRadius(24)
    .scaledToFit()
    .frame(minWidth: 0, maxWidth: .infinity)
    .overlay(
     VStack(alignment: .center, spacing: 12, content: {
       Text (photo.place)
         .foregroundColor(.white)
         .font(.largeTitle)
         .fontWeight(.bold)
         .shadow(radius: 1)
         .padding(.horizontal, 18)
         .padding(.vertical, 4)
         .overlay(
          Rectangle().fill(Color.white).frame(height: 1),
           alignment: .bottom
       Text (photo.country)
         .foregroundColor(.black)
         .font(.footnote)
         .fontWeight(.bold)
         .frame (minWidth: 85)
         .padding(.horizontal, 10)
         .padding(.vertical, 5)
         .background(Capsule().fill(Color.white))
      .frame(minWidth: 280)
      .padding(.bottom, 50),
     alignment: .bottom
```

在body部分,我们添加了一个Image控件用于在视图中呈现照片,然后设置其圆角属性和最大最小宽度。通过overlay修饰器,我们在照片的上面添加了一个VStack容器,容器中包含两个文本,一个用于显示拍照的地区,另一个则显示国家。这个浮动层的最小宽度为280点,并且与照片底部有50点的距离,最后设定这个浮动层的对齐方式为与照片底部对齐。

在VStack容器中的第一个Text同样具有一个overlay浮动层,我们通过它让地区文字的底部出现一个白色的横线。下面的Text控件则利用background修饰器,在文字的后面添加了一个胶囊形状的白色背景,在预览窗口的效果如图8-4所示。



图8-4 卡片视图在预览窗口中的效果

最后让我们回到ContentView,在Body部分替换一行代码,在预览窗口的效果如图 8-5所示。

```
//MARK: - Body
var body: some View {
   CardView(photo: inYourHeartData[0])
}
```

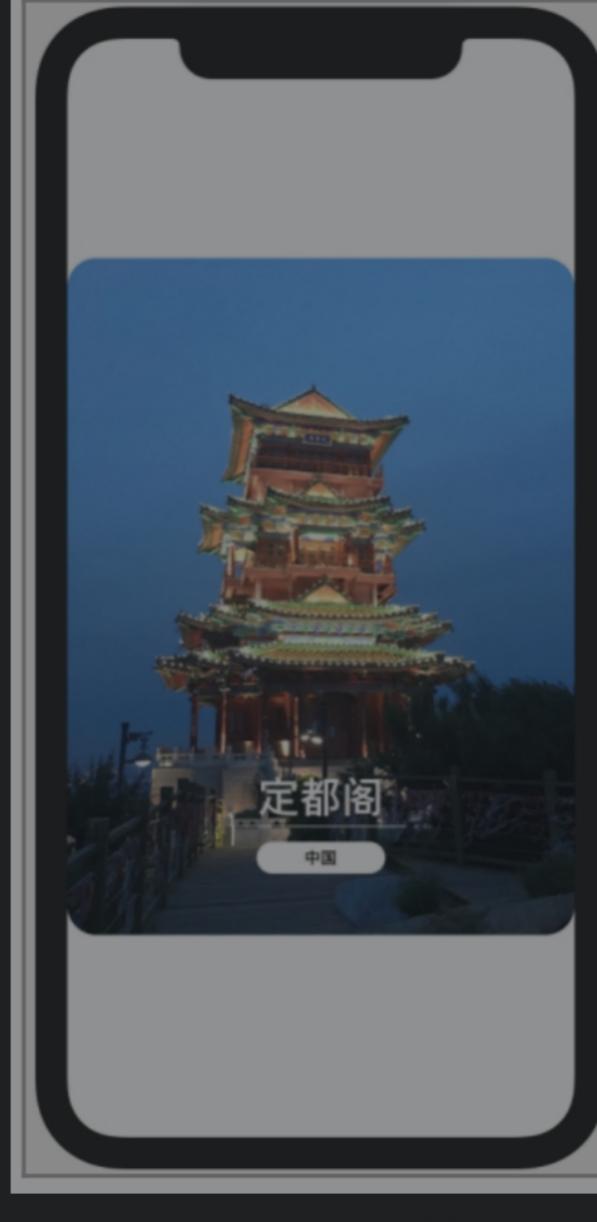


图8-5 ContentView在预览窗口中的效果

下一章