上一章

## 6.3 创建游戏主界面

本节,我们将创建游戏主界面视图,该视图包含三个槽位和一个拉杆按钮。

## 6.3.1 设计水果机的槽位视图

在Views文件夹中新建一个SwiftUI类型文件,将其命名为ReelView。为Preview添加修饰器。

```
ReelView()
.previewLayout(.fixed(width: 220, height: 220))

修改Body部分的代码如下。
```

我们在Body部分添加了一个Image,并使用一些常规修饰器将其设置为合适的尺寸。在之后的界面设计中,我们会用到三个这样的槽位,所以这里再将相关的修饰器集成为一个自定义修饰器。

在Modifiers文件的底部添加一个新的自定义修饰器ImageModifier。

注意,在ImageModifier修饰器中,并没有将resizable添加进来,这是因为resizable是扩展(Extension),不属于修饰器。

回到ReelView中,将代码修改为下面这样,在预览窗口中的效果如图6–9所示。

```
//MARK: - Body
var body: some View {
    Image("槽位")
        .resizable()
        .modifier(ImageModifier())
}
```

图6-9 水果机的槽位视图效果

## 6.3.2 搭建游戏主界面视图

回到ContentView中,在//MARK: -FruitMachine注释语句的下面搭建游戏主界面的架构。

```
//MARK: - FruitMachine
VStack(alignment: .center, spacing: 0) {
  //MARK: - 槽位 #1
  ZStack {
   ReelView()
   Image("草莓")
     .resizable()
     .modifier(ImageModifier())
  } //: ZStack
  HStack(alignment: .center, spacing: 0) {
   //MARK: - 槽位 #2
   ZStack {
     ReelView()
     Image("柠檬")
       .resizable()
       .modifier(ImageModifier())
   } //: ZStack
   Spacer()
   ZStack {
     ReelView()
     Image("牛油果")
       .resizable()
       .modifier(ImageModifier())
   } //: ZStack
  } //: HStack
  .frame(maxWidth: 500)
} //: VStack
.layoutPriority(2)
```

在VStack容器中,我们先在上层添加一个槽位,利用ZStack容器将水果图片放置到槽位视图的上面,并应用了ImageModifier自定义修饰器。在VStack容器的下层添加一个HStack容器,左右放置两个ZStack容器,为了在iPad设备上也有完美的显示效果,为HStack容器添加frame修饰器,设置最大宽度为500点。为了保证整个VStack容器的空间尺寸,利用layoutPriority修饰器提高该容器的优先级。在预览窗口中的效果如图6-10所示。

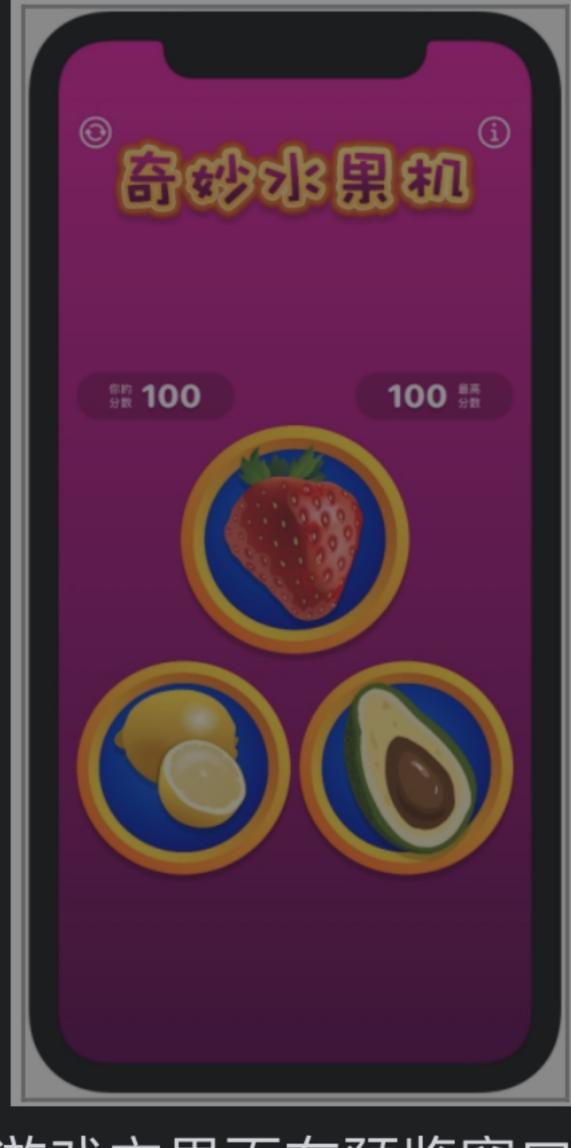


图6-10 游戏主界面在预览窗口中的效果

最后,我们还需要添加一个"拉杆"按钮。在包含两个槽位的HStack容器下面添加如下代码。

此时ContentView的效果如图6-11所示。



图6-11 ContentView在预览窗口中的效果

