

## 第3章

# 屏幕和屏幕控制

### 【教学目标】

- (1) 掌握屏幕控制。
- (2) 熟练应用屏幕组件。



## 3.1 屏幕

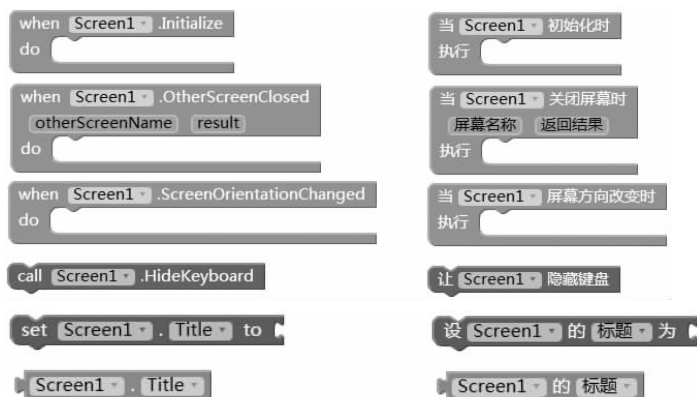
### 1. 屏幕的属性面板

屏幕(Screen)的属性面板主要有以下重要属性:

- (1) 应用名称(AppName)。
- (2) 图标(Icon): 设置 App 安装后显示的个性化图标。
- (3) 屏幕方向(ScreenOrientation)。
- (4) 显示状态栏(ShowStatusBar)。
- (5) 屏幕尺寸(Sizing)。
- (6) 标题(Title): 屏幕上方显示的标题。
- (7) 显示标题栏(TitleVisible)。

### 2. 屏幕的主要事件、方法和属性

屏幕的主要事件、方法和属性如下:



## 3.2 屏幕控制

屏幕控制(Control)包括以下 5 个功能。

(1) 打开另一个屏幕：

open another screen screenName 打开屏幕

(2) 打开另一个屏幕并传递初始值：

open another screen with start value screenName  
startValue 打开屏幕  
并传递初始值

(3) 屏幕获取初始值：

get start value 屏幕初始值

(4) 关闭当前屏幕：

close screen 关闭当前屏幕

(5) 关闭屏幕并返回值：

close screen with value result 关闭屏幕并返回值

### 案例 切换屏幕并传值

参考源程序：Multi\_Sreen\_PassValue

#### 【功能描述】

设计一个 App, 切换两个屏幕并传递值。

#### 【组件设计】

(1) 屏幕 Screen1 组件设计如图 3-1 所示。按钮 Button\_NextScreen 用来打开屏幕 Screen2。

(2) 屏幕 Screen2 组件设计如图 3-2 所示。按钮 Button\_PreScreen 用来返回屏幕 Screen1, 标签 Label\_PassValue 用来显示屏幕 Screen1 传递的值。

#### 【逻辑设计】

(1) 单击 Screen1 中的“下一页”按钮, 打开 Screen2, 并传递初始值(startValue)。

```
when Button_NextScreen .Click
do
  open another screen with start value screenName " Screen2 "
  startValue " 欢迎您到此一游！ "
```

(2) Screen2 打开时, 获取 Screen1 传递的初始值, 赋给标签 Label\_PassValue 并显示; 单击“返回”按钮, 打开 Screen1。

```
when Screen2 .Initialize
do
  set Label_PassValue . Text to get start value

when Button_PreScreen .Click
do
  open another screen screenName " Screen1 "
```

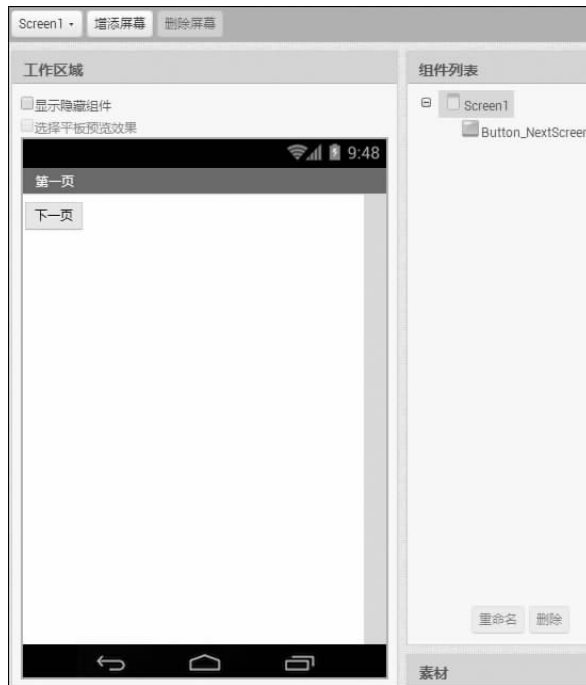


图 3-1 Screen1 组件设计



图 3-2 Screen2 组件设计

**【运行结果】**

运行结果如图 3-3 所示。



图 3-3 运行结果

## 思考与练习

- (1) 屏幕组件和屏幕控制有何区别？
- (2) 如何实现多个屏幕之间的切换？
- (3) 如何通过代码块在屏幕上创建多个按钮？

## 第4章

# 界面布局

### 【教学目标】

- (1) 了解水平滚动条布局组件和垂直滚动条布局组件。
- (2) 掌握表格布局组件。
- (3) 熟练应用水平布局组件和垂直布局组件。



界面布局(Layout)模块的功能是对界面中的所有组件进行合理布局。界面布局模块包括水平布局(HorizontalArrangement)、水平滚动条布局(HorizontalScrollArrangement)、表格布局(TableArrangement)、垂直布局(VerticalArrangement)和垂直滚动条布局(VerticalScrollArrangement)5个组件,如图4-1所示。



图 4-1 界面布局模块的组件

## 4.1 水平布局

水平布局组件可以实现其内部组件的水平排列,如图4-2所示。

水平布局组件的高度(Height)和宽度(Width)属性设置包括自动(Automatic)、充满(Fill parent)、像素(pixels)和百分比(percent)4个选项,如图4-3所示。

这4个选项的含义如下:

- 自动: 根据空间自动调整。
- 充满: 充满整个空间。



图 4-2 水平布局组件的属性面板



图 4-3 高度属性设置

- 像素：按像素确定大小。
- 百分比：按百分比确定大小。

## 4.2 水平滚动条布局

水平滚动条布局组件是水平布局组件的扩展,它带有水平滚动条,可根据需要自动调整宽度,其属性面板如图 4-4 所示。

## 4.3 表格布局

表格布局组件可以实现内部组件按照表格方式排列,它同时具有水平布局组件和垂直布局组件的特点,可设置行数(Rows)和列数(Columns)属性值,如图 4-5 所示。



图 4-4 水平滚动条布局组件的属性面板



图 4-5 表格布局组件的属性面板



## 4.4 垂直布局

垂直布局组件可以实现其内部组件的垂直排列,其属性设置和水平布局组件的属性设置类似。

## 4.5 垂直滚动条布局

垂直滚动条布局组件是垂直布局组件的扩展,它带有垂直滚动条,可根据需要自动调整高度,其属性面板如图 4-6 所示。



图 4-6 垂直滚动条布局组件的属性面板

## 案例 水平布局、垂直布局和表格布局的综合使用

参考源程序：Layout.aia

### 【功能描述】

综合使用水平布局、垂直布局和表格布局组件设计一个学生信息表的 App 界面。

### 【组件设计】

使用水平布局、垂直布局和表格布局组件完成组件设计,如图 4-7 所示。

- (1) 使用水平布局组件 HorizontalArrangement1 完成两个按钮组件的水平排列。
- (2) 使用表格布局组件 TableArrangement1 完成 2 行 2 列标签和文本输入框组件的排列。
- (3) 使用垂直布局组件 VerticalArrangement1 完成 3 个标签组件的垂直排列。



图 4-7 水平布局、垂直布局和表格布局综合使用

## 思考与练习

- (1) 水平布局组件和垂直布局组件有何区别?
- (2) 表格布局组件有何优势?
- (3) 水平布局组件和水平滚动条布局组件有何区别?