

《嵌入式应用开发》

实验三

计算器界面布局实验指导手册

版本：V 1.0

目录

(一) 实验目的	3
(二) 实验涉及知识点	3
(三) 实验准备	3
(四) 实验内容及课时分配	3
(五) 实验过程	4
1. 实验功能描述	4
2. 创建项目	5
3. 修改 Index.ets 页面定义按键数组	6
4. 定义计时器按钮样式	6
5. 编写网格布局	7
6. 顶部功能组件	8
7. 迭代数字按钮	10
8. 测试效果	12

(一) 实验目的

1. 了解自适应布局
2. 计算器页面布局

(二) 实验涉及知识点

1. 自适应布局
2. 常见布局种类
3. Grid 组件
4. GridItem 子组件

(三) 实验准备

1. 技能要求：

操作此实验需要具备基本的 TS 语法知识；

2. 实验环境要求：

基于 Windows10 或者 MacOS 操作系统，安装了 DevEco Studio 开发工具。

(四) 实验内容及课时分配

序号	实验内容	实验课时	对应核心知识点	对应实验目标点
1	了解自适应布局	0.5	1、2、3、4	1、2
2	计算器页面布局	0.5	1、2、3、4	1、2
	总计	1		

(五) 实验过程

1. 实验功能描述

完成计算器布局

界面：

竖屏

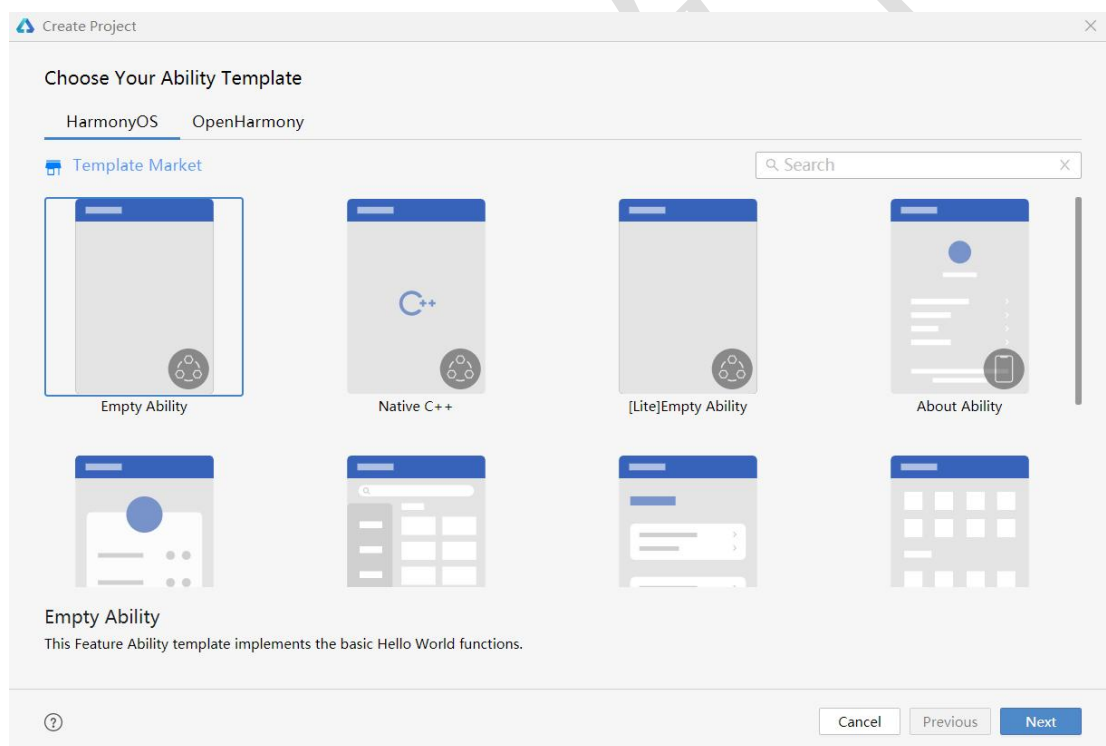


横屏：

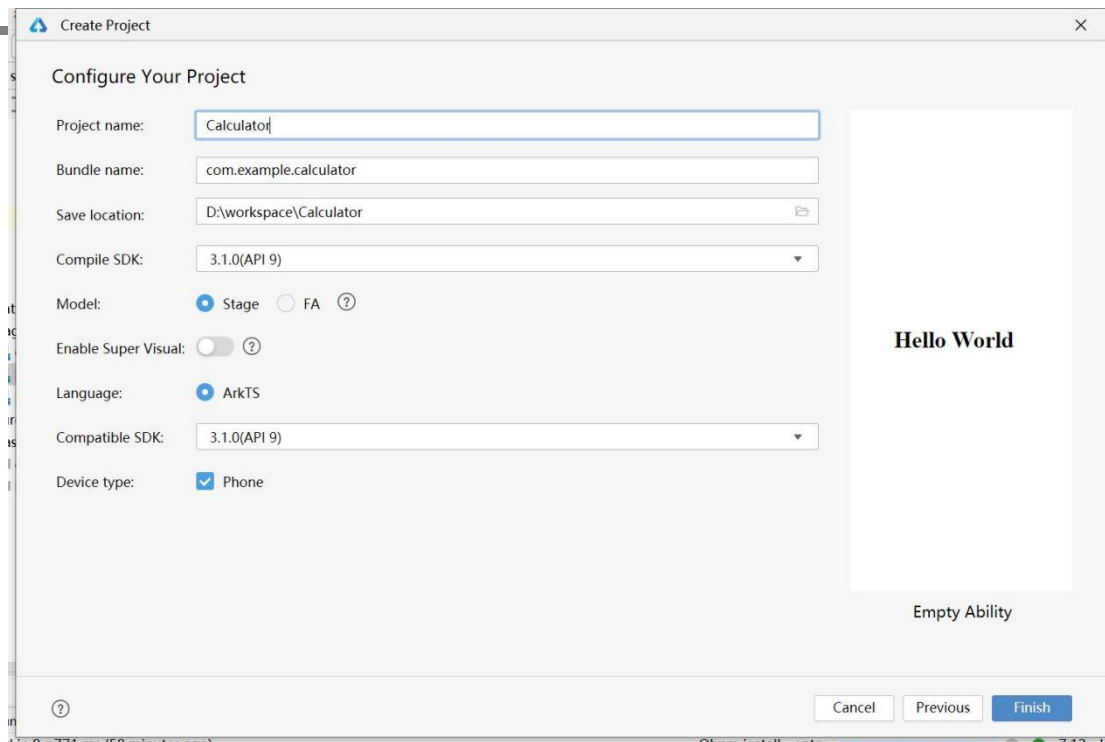


2. 创建项目

从 file>new>create project 菜单下新建项目：



点击【Next】按钮，输入项目配置信息：



点击【Finish】按钮完成项目创建。

3. 修改 Index.ets 页面定义按键数组

修改 Index.ets 文件（目录：entry > src > main > ets > pages”，打开“Index.ets”文件）：

删除原有代码，并定义数组：

```
@State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']
```

当前 Index.ets 中的代码如下：

```
@Entry
@Component
struct Index {
  @State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']

  build() {
    Column() {

    }.width('100%).margin({ top: 5 })
  }
}
```

4. 定义计时器按钮样式

在 Index.ets 文件中添加如下代码：

```
@Style textStyle(){
  .backgroundColor(0xd0d0d0)
```

```
.width('100%')
.height('100%')
.borderRadius(5)
}
```

当前 Index.ets 代码如下：

```
@Entry
@Component
struct Index {
    @State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']

    // 定义计时器按钮样式
    @Styles textStyle(){
        .backgroundColor(0xd0d0d0)
        .width('100%')
        .height('100%')
        .borderRadius(5)
    }

    build() {
        Column() {

            }.width('100%').margin({ top: 5 })
        }
    }
}
```

5. 编写网格布局

首先设置组件间隙：

```
Column({ space: 5 })
```

然后添加 Grid 容器及 GridItem 子组件，完成网格拆分：

```
Grid() {
    GridItem() {
    }

    }.columnsTemplate('1fr 1fr 1fr 1fr')

    }.rowsTemplate('2fr 1fr 1fr 1fr 1fr')

    }.columnsGap(10)

    }.rowsGap(15)

    }.width('90%')

    }.backgroundColor(0xF0F0F0)
```

```
.height('70%')
```

当前 Index.ets 文件中的代码：

```
@Entry
@Component
struct Index {
    @State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']

    // 定义计时器按钮样式
    @Styles textStyle(){
        .backgroundColor(0xd0d0d0)
        .width('100%')
        .height('100%')
        .borderRadius(5)
    }
    build() {
        Column({ space: 5 }) {
            Grid() {
                GridItem() {
                }
            }
            .columnsTemplate('1fr 1fr 1fr 1fr')
            .rowsTemplate('2fr 1fr 1fr 1fr 1fr')
            .columnsGap(10)
            .rowsGap(15)
            .width('90%')
            .backgroundColor(0xF0F0F0)
            .height('70%')
        }.width('100%').margin({ top: 5 })
    }
}
```

6. 顶部功能组件

在网格布局子组件中定义计算器顶部的功能组件，包括：输入显示窗口，清空按钮，回退按钮：

```
GridItem() {
    Text('0')
    .fontSize(30)
    .textStyle()
}.columnStart(1).columnEnd(4)

GridItem() {
    Text('清空')
    .fontSize(16)
```



```

        .textAlign(TextAlign.Center)
        .textStyle()
    }.columnStart(1).columnEnd(2)

    GridItem() {
        Text('回退')
        .fontSize(16)
        .textAlign(TextAlign.Center)
        .textStyle()
    }.columnStart(3).columnEnd(4)

```

当前 Index.ets 中的代码:

```

@Entry
@Component
struct Index {
    @State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']

    // 定义计时器按钮样式
    @Styles textStyle(){
        .backgroundColor(0xd0d0d0)
        .width('100%')
        .height('100%')
        .borderRadius(5)
    }

    build() {
        Column({ space: 5 }) {
            Grid() {
                GridItem() {
                    Text('0')
                    .fontSize(30)
                    .textStyle()
                }.columnStart(1).columnEnd(4)

                GridItem() {
                    Text('清空')
                    .fontSize(16)
                    .textAlign(TextAlign.Center)
                    .textStyle()
                }.columnStart(1).columnEnd(2)

                GridItem() {
                    Text('回退')
                    .fontSize(16)
                    .textAlign(TextAlign.Center)
                    .textStyle()

```

```

        }.columnStart(3).columnEnd(4)
    }
    .columnsTemplate('1fr 1fr 1fr 1fr')
    .rowsTemplate('2fr 1fr 1fr 1fr 1fr 1fr')
    .columnsGap(10)
    .rowsGap(15)
    .width('90%')
    .backgroundColor(0xF0F0F0)
    .height('70%')
    }.width('100%').margin({ top: 5 })
}
}

```

7. 迭代数字按钮

利用 `ForEach` 迭代数字按钮部分：

```

ForEach(this.Number, (day: string) => {
    if (day === '0') {
        GridItem() {
            Text(day)
                .fontSize(16)
                .textAlign(TextAlign.Center)
                .textStyle()
        }.columnStart(1).columnEnd(2)
    } else {
        GridItem() {
            Text(day)
                .fontSize(16)
                .textAlign(TextAlign.Center)
                .textStyle()
        }
    }
})

```

最终 `Index.ets` 完成后代码：

```

@Entry
@Component
struct Index {
    @State Number: Array<string> = ['1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '*', '0', '.', '/']

    // 定义计时器按钮样式
    @Styles textStyle(){
        .backgroundColor(0xd0d0d0)
        .width('100%')
        .height('100%')
        .borderRadius(5)
    }
}

```

```

}

build() {
  Column({ space: 5 }) {
    Grid() {
      GridItem() {
        Text('0')
          .fontSize(30)
          .textStyle()
      }.columnStart(1).columnEnd(4)

      GridItem() {
        Text('清空')
          .fontSize(16)
          .textAlign(TextAlign.Center)
          .textStyle()
      }.columnStart(1).columnEnd(2)

      GridItem() {
        Text('回退')
          .fontSize(16)
          .textAlign(TextAlign.Center)
          .textStyle()
      }.columnStart(3).columnEnd(4)

      ForEach(this.Number, (day: string) => {
        if (day === '0') {
          GridItem() {
            Text(day)
              .fontSize(16)
              .textAlign(TextAlign.Center)
              .textStyle()
          }.columnStart(1).columnEnd(2)
        } else {
          GridItem() {
            Text(day)
              .fontSize(16)
              .textAlign(TextAlign.Center)
              .textStyle()
          }
        }
      })
    }

    .columnsTemplate('1fr 1fr 1fr 1fr')
    .rowsTemplate('2fr 1fr 1fr 1fr 1fr 1fr')
  }
}

```

```
.columnsGap(10)

.rowsGap(15)

.width('90%')

.backgroundColor(0xF0F0F0)

.height('70%')

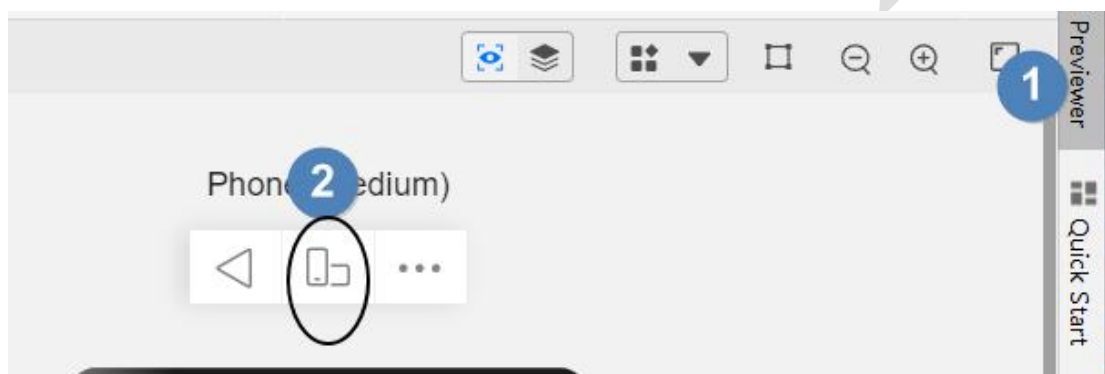
}.width('100%').margin({ top: 5 })

}

}
```

8. 测试效果

在预览器中预览计算器运行效果，点击预览器中屏幕切换测试效果：



竖屏：



横屏:



城通教育