内蒙古农业大学

学生实验报告

院 系： 计算机与信息工程学院

专 业： 计算机科学与技术

年 级： 2021级

课程名称： 嵌入式应用开发

学 号： 2021122156404

姓 名： 周烨

指导教师： 阿斯雅

2024年 5 月 8 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 院系 | 计算机院信息工程学院 | 年级 | 2021级 | 学号 | **2021122156404** |
| 专业 | 计算机科学与技术 | | | 姓名 | **周烨** |
| 实验名称 | 实验三 电影排行榜 | | | 实验室 | 414 |
| 实  验  目  的 | 1. 数组的使用 2. ForEach用法 3. 数据源使用 4. List组件 5. 应用程序上下文 6. 组件封装 | | | | |
| 实验题目 | 电影排行榜  扩展内容：1.增加登录界面。  2.增加数据库存储。 | | | | |
| 实验代码 | import { MovieBean } from '../common/been/MovieBeen'; import { ListHeaderComponent } from '../view/ListHeaderComponent' import { ListItemComponent } from '../view/ListItemComponent' import { TitleComponent } from '../view/TitleComponent' import { MovieRankViewModel } from '../viewmodel/MovieRankViewModel'; let movieRankModel: MovieRankViewModel = new MovieRankViewModel() @Entry @Component struct Index {  @State dataScoure1: MovieBean[] =[  // {'id':'1','movieName':'满江红','tickets':'54321'},  // {'id':'2','movieName':'阿凡达','tickets':'44321'},  // {'id':'3','movieName':'你好，李焕英','tickets':'34321'},  // {'id':'4','movieName':'火星救援','tickets':'24321'},  // {'id':'5','movieName':'星球大战','tickets':'14321'}  ];  @State dataScoure2: MovieBean[] =[];  @State isUpdataDataSource:boolean = true;   @Builder MovieRankList(){  Column(){  List(){  ForEach(this.isUpdataDataSource?this.dataScoure1:this.dataScoure2,(item,index) =>{  ListItem(){  ListItemComponent({index:index+1,  movieName:item.movieName,  tickets:item.tickets})  }  },item =>JSON.stringify(item))  }  .width('100%')  .height('70%')  .divider({strokeWidth:1})  }.padding(15)  .borderRadius(15)  .alignItems(HorizontalAlign.Center)  .backgroundColor(Color.White)  .width('90%')  }  aboutToAppear(){  this.dataScoure1 = movieRankModel.loadMovieRankDataSource1();  this.dataScoure2 = movieRankModel.loadMovieRankDataSource2();  }  build() {  Column() {  TitleComponent({isUpdataDataSource:$isUpdataDataSource})  ListHeaderComponent()  .margin({top:10,bottom:10})  //ListItemComponent()  this.MovieRankList()  }.width('100%').height('100%').backgroundColor('#ffa3b8cd')  } }  @Preview @Component export struct ListHeaderComponent{  build(){  Row(){  Text('排名')  .fontSize(20)  .width('30%')  .fontWeight(400)  .fontColor(0xd3d3d3)  Text('影片名')  .fontSize(20)  .width('50%')  .fontWeight(400)  .fontColor(0xd3d3d3)  Text('票房')  .fontSize(20)  .width('20%')  .fontWeight(400)  .fontColor(0xd3d3d3)  }  } }  @Preview  @Component  export struct ListItemComponent {  index: number = 1  movieName:string = '满江红'  tickets:string = '12345'   isCircleText(): boolean {  if(this.index === 1  || this.index === 2  || this.index === 3 ){  return true  }  return false  }   @Builder CircleText(index:number){  Row(){  Text(this.index.toString())  .fontSize(15)  .fontWeight(400)  .fontColor(Color.White)  }.justifyContent(FlexAlign.Center)  .borderRadius(25)  .width(25)  .height(25)  .backgroundColor(Color.Blue)  }  build(){  Row(){  Column(){  if(this.isCircleText()){  this.CircleText(this.index)  }else{  Text(this.index.toString()).lineHeight(25)  .textAlign(TextAlign.Center)  .fontSize(15)  .width(24)  .fontWeight(400)  }  }.width('30%')  .alignItems(HorizontalAlign.Start)  Text(this.movieName)  .width('50%')  .fontWeight(500)  .fontSize(16)  .fontColor(Color.Blue)  Text(this.tickets)  .width('20%')  .fontWeight(500)  .fontSize(16)  .fontColor(Color.Blue)  }  } }  import Brightness from '@system.brightness' //@Preview @Component export struct TitleComponent{  @Link isUpdataDataSource:Boolean;  build(){  Row(){  Row(){  Image($r("app.media.arrowcircleleft"))  .width(30)  .height(30)  .margin({right:5})  Text("电影排行榜").fontSize(20)  }.width('50%').height(30).justifyContent(FlexAlign.Start)  Row(){  Image($r('app.media.redo'))  .width(25)  .height(25)  .onClick(()=>{  this.isUpdataDataSource = !this.isUpdataDataSource;  })  }.width('50%').height(30).justifyContent(FlexAlign.End)  }.width('100%')  .padding({left:25,right:25})  .margin({top:10})  .height(45)  .justifyContent(FlexAlign.SpaceAround)  }  } | | | | |

|  |  |
| --- | --- |
| 实  验  步  骤 | 运行截图：  29f5c316b0c5e27003b0d01de4d98e9de84908ff65f9b24042b48fbb359967 |
| 心  得  体  会 | 1. Foreach是基于数组类型数据执行循环渲染的，参数itemgenerator生成子组件的lambda函数，为数组中的每一个数据项创建一个或多个子组件，实验中通过在list组件中使用Foreach来循环渲染加载每个电影数据从而在页面中实现一条一条的数据效果。 2. 实验中在etc下面创建的model文件夹中的MovieDataModel.ets模拟的两个数据源分别对应着刷新前后的电影显示列,之后通过onlick事件修改isUpdataDataSource来改变Foreach函数的实现条件，从而进行循环渲染加载数据的切换 |
| 成  绩  评  定 | 教师签名：  年 月 日 |