使用Scaffold, 定义TopBar, Drawer, 在Drawer 中为每个Item定义点击导航事件,在密封类中封 装每个页面的title, route, icon, ,最后在 导航栏 NacHost中实现导航功能 需要顶部导航栏,抽屉导航栏 移植后构建失败 注解上添加exportSchema,数据库版本手动加一 给Dao中的函数声明为挂起函数,把操作放在协程 使用数据类封装,用@StringRes注解标记字段是 一些需要搭配使用的内容,例如主题设置有相应 运行时闪退 字符串索引,并将文本内容存在String文件中,为 的代码,描述,预览图。关于界面的文本有标题 后期升级多语言做准备 和正文 数据接收容器设为状态,获取数据事件放在 内容的封装 操作后数据不更新 OnClick中触发 一些需要随机展示的字符串,如测试师进酒吧, 封装在数据类中,操作起来更方便 退出时的冷笑话 应用切至后台时意外退出 序列化自定义数据类 使用Room 由于数据的更新在协程中完成,而协程要放在点 击事件中触发,用户直接进入统计界面,不会触 传入List<List<T>>,使用forEach循环打印每个格 发任何数据更新。一开始我让用户每次查看数据 子,并根据格子值传入点击事件:输入,删除或 都要手动展开列表,这样展开列表的动作同时还 键盘 是提交。 包括了数据更新。但这样即便用户没有添加新数 一些手写组件 据进来还得忍受每次查看一个数据就要更新一次 传入SnapshotStateList<List<T>>,使用forEach 列表的情况, 于是引入一个图标用于在列表展开 循环打印每个格子,并根据格子自身的状态控制 时供用户刷新,同时原先是否状态的状态用 显示不同的颜色 格子 rememberSavable委托,即便屏幕方向变化,列 数据的更新不及时 表也不会莫名其妙收回了。 GitHub上公开了一个Wordle的API,可使用http. 获取单词 get("getword")随机得到一个单词 如何从网络拉取一个随机的单词 构建失败 viewmodel中没有注入依赖项 使用依赖注入 MainActivity没有添加@AndroidEntryPoint注解, 单击:提示用户使用方法 总之问题不来源于技术的复杂性而来源于对技术 仍然构建失败 的不熟练。 词库元素交互 双击, 快速填充候选词 使用detectTapGestures() 不仅仅是一个显示候选词的列表 长按: 查看单词中文释义 在程序的比较顶层的位置引入一个表示状态模式 的状态,并在WordleTheme中根据不同状态选择 不同primary, background等颜色。然后向下传递 改变此状态的动作,直到传入到"设置"中,当设置 当得到绿色反馈时:把Map对于索引的值改为该 中点击切换主题时就触发了在顶层改变主题状态 字母 如何在程序运行中切换颜色 的动作。即预先把"设置"中的状态提升了。 使用Map<Int,String>, Set<String>, List<List< String>>三种数据结构 当得到黄色反馈时,把该字母加到Set中,并加到 如果在设置中使用循环重复执行触发事件,似乎 List<List<T>>对应索引的List中 设置不同主题颜色搭配 这样的重组行为会被"优化"掉。于是我选择在程序 进入WordleTheme前设置一个状态用于检测是否 若得到绿色反馈:如果对应位置不是这个字母, 需要进行这样的闪烁,每次用户点击产生该闪烁 则筛选掉这个词汇 的按钮时,这个状态就会加10,并在每次闪烁后· 如何制作界面快速闪烁的视觉效果 红色反馈: 如果词汇中含有这个字母, 则筛选掉 筛选掉不可能的词汇 算法 更新候选词列表 黄色反馈:如果词汇对应位置含有List<List< 很显然,解决方法就是锁定屏幕方向。 Sttring>>中记录的字母,则筛选掉 因为屏幕比例的关系,手机屏幕方向发生变化 但这种做不到就干脆禁止用户进行这种操作的行 如果字母位置的字母是Map索引处的值,则显示 时,界面元素轻则发生畸变,重则不能在屏幕上 为是消极的开发行为、于是我在YouTube上学到 横屏适配 显示。 了使用LocalConfiguration.current获取设备当前 的状态,再判断是Configuration类下的 ORIENTATION_PORTRAIT还是ORIENTATION_ 如果字母在Set中,但不在List<List<String>>中, 显示颜色提示 LANDSCAPE,再把界面元素封装一下,为两种方 显示黄色 向分别设计布局。 其他显示红色 WordLE 在LazyColumn的每个元素中添加一个按钮,其图 列表中每个元素都有相应的标题和内容,在不需 标是简易箭头,当点击时吧元素的扩展状态取 在viewmodel中储存:是否胜利,是否游戏结束, 要的时候,为了能看更多条消息,需要每条内容 反,并根据其值是否为真,决定是否显示内容, 是否作弊,候选词列表,本次答案,输入位置, 当其值为真时,显示详细内容。 能够单独控制扩展和收起 分享信息等其他部分需要了解的状态 储存"全局状态" 为Column的变化添加动画过程. 包括开始一局新游戏,更新格子,更新键盘,复 伸展与收缩过程非常突兀,就像手机卡住了一样 animateContenSize(...) 位格子,复位键盘,对候选词列表的更新,控制 输入位置的移动等对这些状态的操作 提供方法 ViewModel 容器由LazyColumn改成LazyVerticalGrid,并让横 屏状态下显示的格子数大致是竖屏下的1.8倍(16: 在统计界面数据库获取信息后,需要把相关信息 列表横屏时被拉伸,影响阅读体验 传递给分享界面,但是操作数据库要求比较高, 我希望最后是传递数据而不是传递动作,但分享 再换成多个LazyColumn,并把要展示的元素切片 LazyVerticalGrid中的元素张开时,同一行的其他 和统计两个功能本身不存在依赖关系,只能在统 独立元素大小也会发生变化,显得很奇怪 到多个列表中分别传到LazyColumn中 计界面把数据存入viewmodel,在分享界面从 viewmodel把数据读出来 担任信息沟通中间人 展示列表,LazyColumn,LazyRow, 状态提升到判断屏幕方向的可组合函数处 LazyVerticalGrid。 这样操作的结果是, 屏幕方向变化, 每个元素是 否展开的状态"不会被留下来",实际上这样操作 这样操作的结果状态可以保留了,但是每个元素 横竖屏使用的是两套状态 均不能单独控制了 把单个状态改成"状态字典",分别映射了每个元素 以何种方式分享?调用权限截屏分享?制作图 和是否展开 片? 一串文字? 要么技术太过棘手, 要么效果很 分享 由于没找到可以替代字典的数据类型,所以最初 的笨办法是新增一个状态,其值是字典的 toString()值,但显然这个String值没有任何含 使方格类型为密封类,其含有String字段表示其值 义,为了提升性能,将这个"附属状态"改成一个 和Color字段表示其颜色,并密封类Empty, Boolean值,每次键值对发生变化时,这个"附属 Input, Correct, Partial, InCorrect状态 需要特殊的方法表示输入格和键盘的颜色 状态"就取反 字典作为状态时,点击事件只改变字典的键值 表示游戏难度 枚举类 数据类,密封类,枚举类 对,字典真值没有改变,页面不重组 最终还要把这个"附属状态赋值给一个垃圾变量, 不然运行时会忽略这个"附属状态的变化",页面不 使用密封类表示Success, Error, Loading, 发生重组。" Empty状态,并携带数据消息或者异常信息 表示获取单词是否成功 强制回退时退出到主界面,防止产生大量堆栈 字母游戏特有的String, Char, 大小写问题, 有 无论其类型如何,为了后面的鲁棒性,需要String 时甚至会发生令IDE啼笑皆非的对String变量调用 数据类型 导航中令launchSingleTop和restoreState值为true 导航造成了堆栈错综复杂,用户回退可能出现不 的地方一律toString, uppercase, lowercase同理 toString()方法 导航 同寻常的结果 将MainActivity以外的导航事件由传入一个点击事 件改为传入一个navController,使其堆栈发生在 传统做法从MainActivity千里迢迢传一个this到很 当前界面而非主界面。 深的地方,改进为在Composable中使用 LocalContext.current得到Context, 此方法可解决 Toast缺失上下文 缺少上下文Context 对Toast函数来说,把唤起Toast事件放在Activity 再传入,避免在Composable中使用Toast。 Compose 尝试了好几种方法都不能在Composable中直接使

Presented with XMind

用,最后还是使用了Java中被废弃的Handler()

退出应用

来操作一个延时任务(退出)