



Wordle

Main Activity

Scaffold

- TopBar顶部导航栏
 - 提供与侧边导航栏功能一致的快捷按钮
 - 应用名称
- Drawer侧边导航栏
 - 打开侧边导航栏的按钮
 - 可透过点击和滑动进入
- Navigation导航
 - 提供导航不同页面的详细信息
 - 提供不同页面的route、图标、标题
 - 实现根据route值导航页面

游戏界面

- 图形化输入方格
 - 状态、颜色
 - Empty 白色
 - Input 白色
 - Correct 绿色
 - Partial 黄色
 - Incorrect 红色
- 图形化键盘
 - 状态、颜色
 - 退格键
 - 删除当前位置输入
 - 确认键
 - 检查输入条件后提交结果
 - 字母键
 - 输入字母
- 作弊键
 - 弹出非强制性消息，透露此次答案
 - 取消此次游戏记录
 - 让玩家产生楚之不爽的愧疚感

词库界面

- 根据用户的尝试输入结果更新可行候选词
 - 含用户输入中得到“红色”反馈的字母的单词不会出现在词库中
 - 用户输入中得到“绿色”反馈的字母不在该位置上的单词不会出现在词库中
 - 用户输入中得到“黄色”反馈的单词在报错位置上的单词不会出现在词库中
- 提供手势快捷操作
 - 单击：为了防止误触，单击为提醒玩家手势操作
 - 双击：快速将此单词填入方格中
 - 长按：访问网络，得到此单词的中文翻译
- 提供横屏适配，横屏状态下每行能显示更多的单词，充分利用屏幕空间
 - 使用LazyNxCircularGrid，虽然是实验性API，但性能也不错

统计界面

- LazyColumn
 - 展示玩家成功通关游戏的记录
 - 时间
 - 游戏模式
 - 尝试次数
 - 答案
 - 详情按钮：可查看当时的每一步操作
 - 分享按钮，可以把结果分享给社交平台的好友
 - 垃圾桶按钮：删除此条记录
 - 删除时弹出警告框，防止用户手滑删除心爱的in 2 tries记录
- 数据保存着本地数据库，可在应用外清除

设置界面

- 个性化主题设置
 - “跟随系统”：会跟随系统深色浅色色调调整
 - “舒适护眼”：适合用户在夜间使用
 - “闲聊论坛”：灵感来源于上世纪BBS论坛
 - “戴克斯格”：配色来自推特公开的配色方案
 - “百特丹斯”：配色来自抖音视频
 - “随机选择”
 - 屏幕快速闪动，最后随机选择
- 难度设置
 - 随机输入 (Easy)：玩家可以输入一个不是单词的字母，可以更快地小选择范围
 - 单词输入 (Medium)：玩家只能输入合法的单词 (合法还有另一层含义)
 - 随机词库 (Hard)：答案本身将不一定是一个合法的单词，玩家输入也可以任意输入。

关于界面

- 一些关于游戏的文本内容
- 制作了动画过程，增强窗口扩展、缩小时的视觉效果
- 使用LazyColumn提高性能

退出界面

- 制作了横屏适配，且用户在横屏切换的过程中，窗口之间的打开关闭延迟并不会丢失状态，而且横屏状态下每行多显示一个元素，充分利用屏幕空间
- 随机展示程序员冷笑话 (目前量不多)
- 致取一个测试工程师走进一家酒吧点了乱七八糟的东西，所以进入这个界面，程序会“闪退”

Compose

主题设置

- 不同主题搭配的顏色
 - 跟随系统模式
 - 黄昏撞色潮流模式
 - PIT论坛模式
 - 仿微信模式
 - 仿字节跳动模式
 - 小、中、大三种圆角
- 自定义几何元素
 - 普通阅读字体
- 字体
 - 适于显示英语单词的等宽编程字体JetBrains Mono

Room数据库

- Data数据类
 - @PrimaryKey 时间
 - 游戏模式
 - 尝试次数
 - 单词
- DbUtil工具类
 - 拓展Date类，增加一个获取当前时间并转为“年-月-日 时-分-秒”的函数
- Database数据库
 - 从WordleViewModel中得到字段值，创建一个数据库实例并返回
 - 声明数据表
 - 声明数据库版本号
- Dao接口
 - 得到Dao接口的方法
 - @Insert 提供添加数据操作
 - @Delete 提供删除数据操作
 - @Update 提供更新数据操作
 - @Query("select * from mydata") 提供查询返回所有数据操作
 - @Query("select * from mydata where time=:time") 提供查询相应主键的数据 (编写但暂未使用)

DI 依赖注入

- AppModule: BaseApplication: Application
 - AppModule 提供Context as BaseApplication
 - BaseApplication 程序入口，并注入到Main Activity中
- NetworkModule
 - 提供网络组件支持
 - 解析Url，获取API

网络

- WordResponse数据类
 - val word: String, 用于保存得多的单词
- WordleService接口
 - 提供挂起函数@CCT(getword)得到包装成WordResponse的单词 (同时也包含单词和获取状态)

Repository仓库

- WordleInterface接口提供挂起函数获取数据源加载状态
- WordleRepo接收注入的WordleService，由接口的方法获取一个单词并返回成功状态，若捕获异常，打包错误信息返回错误状态

ViewModel

- 状态
 - 接收注入WordleRepo，该类实现了WordleService接口的getWord()函数
- MutableState<T>类型
 - gameModel: Difficulty: 默认值: Free: 游戏模式状态
 - cheatOpen: Boolean: 默认值: false: 是否点了作弊按钮
 - won: Boolean: 默认值: false: 是否赢得了胜利
 - gameOver: Boolean: 默认值: false: 游戏是否结束，此与won并非相互关系
 - word: Resource<String>: 默认值: Resource.Empty(): 当前游戏的词汇。
 - answer: String: 默认值: "": 即word的字面值，工具人。word的替身，用于简化一些操作
 - _userGrid: 输入方格的状态，默认全为Empty()
 - _userKeyboard: 键盘的状态，默认为Input0, 字母量为键对应的大写字母
 - 一个Map<Int,String>, 用于筛选候选词，筛选对应位置不是“绿色”的单词。若用户在尝试中得到了“绿色”反馈，就把字母填入到Map键所指的值中
 - 一个List<List<String>>, 用于筛选候选词，筛选对应位置是“黄色”的单词。若用户在尝试中得到了“黄色”反馈，就把字母填入到与得到“黄色”反馈相同位置索引的List<String>中
 - 一个Set<String>, 用于提示候选词，一个折中的方法，避免位置置上出现“黄色”字样的单词。若用户在尝试中得到了“黄色”反馈，就把字母填入Set中
- 普通类型
 - tries: Int: 默认值: 0: 表示玩家尝试次数，其实默认值最少都没有关系，但 Kotlin 不写这个 0 要多写延迟初始化和类型判断

函数

- 复位输入方格
- 复位输入键盘
- 获取单词，在ViewModelScope协程中运行
 - 获取word值和状态
 - 若当前模式为随机词库模式，将word替换为一个随机生成的乱七八糟的字符串
 - 如果word状态非异常，将值传给answer
 - 调用复位方格和复位键盘的函数
- 更新方格
 - 复位输入位置的行列为0: 复位词库候选词: 复位Map, Set, List, 复位作弊开关为false和尝试次数为0
- 更新键盘
 - 响应输入行为，将字母显示在方格中
 - 更新输入位置
 - 使键盘格子颜色与方格集一致，辅助玩家进行输入判断
- 更新方格集
 - 判断输入是否合法 (不同游戏模式下法则不一)
 - 对玩家提交的词案做出反馈，并给方格和键盘涂上辅助色
- 退格
 - 更新Map, Set, List
 - 判断游戏是否结束
 - 更新输入位置
- 更新候选词库
 - 复位方格
 - 更新输入位置
- 判断输入合法性

一些数据类

- 自定义主题
 - 主题代号
 - 主题描述
 - 主题预览图
- 难度 (枚举类)
 - FREE (随意输入)
 - RESTRICTED (单词输入)
 - RANDOM (随机词库)
- 测试工程师进酒吧
- 冷笑话
 - 用于在退出界面展示
- 问题
 - 关于界面给用户提供帮助

资源文件

- 原创App logo
 - 运用游戏的三种主色调与方格构成，红色的方格就够引起你的注意一种键盘的感觉，组成的W为Wordle的字母
- 原创WordLE图标
 - 根据不同主题选取不同颜色组成，能和页面和谐融合，设计理念大胆、创新，风格美观，大方
- 原创版权图标
 - 内容是我GitHub昵称，至于为什么是这个昵称，我也不知道，也许GitHub用户不要这么会抢名字，也许我搞错其他搞错了，就设计这个与WordLE图标一致，共同组成主题的一部分
- 原创主题预览图
 - 用于在主题设置页面给用户提供视觉参考
- 来自阿里适重图标库的图标
 - 用于辅助设置导航栏中的导向
- Jetbrains的开源等宽字体
 - 其显示候选词和历史记录时视觉效果整齐统一
- 从网页HTML获取的，根据网上公开资料提取的，根据Android Material规范组合的颜色
 - 用于主题搭配
- 深色模式
 - 使应用显示适应用户的手机的颜色深浅
- 字符串
 - 各种数据类中的字符串都引用至此，为之后提供多语言支持做准备，保证良好的可维护性

需要的权限

- 网络权限
- 获取网络状态
- 读写外部存储

Cradle配置

- kotlin相关
- 依赖注入相关
- Compose相关
- Room相关
- 协程相关
- jetpack相关