PERSONAL RESUME

个人概况

姓名：黄智慧　　籍贯：广东

学历：本科　　外语：英语 6 级

邮箱：944195790@qq.com　　手机：13528790106

工作经验：7 年　　期望薪资：面议

求职意向

Android 开发

专业技能

1)，熟练掌握 Java 语言，熟悉常用的设计模式(代理、单例、工厂、建造者、观察者、策略等)、多线程并发，有良好的编程习惯和面向对象编程思想；
2)，熟练掌握 JVM 原理，反射原理，动态代理以及对 ClassLoader 热修复、增量更新、APT、资源加载动态换肤有比较深的理解；
3)，熟练掌握常用数据结构和算法；
4)，熟练掌握 Android 消息机制、事件传递机制，熟练自定义 View 控件及常用动画，能利用事件分发原理解决 UI 交互问题、对 UI 界面设计和优化有实际工作经验；
5)，有阅读 FrameWrok 层源码来解决实际问题的经验，熟悉 AMS，WMS，PKMS, Binder 等原理；
6)，对内存优化，启动速度优化、包体积优化、crash 监控等性能优化有实际调优经验；
7)，研究过 Glide/OkHttp/Retrofit/RxJava 等第三方框架以及 Jetpack 常用组件(LifeCycle、LiveData、DataBinding、ViewModel、WorkManager 等)源码，熟悉 TCP/IP,HTTP, HTTPS 协议，并具备相关性能调优能力；
8)，有 MVP/MVVM 架构搭建项目的实际开发经验；
9)，对模块化，组件化，插件化开发架构有深入的研发经验；
10)，熟悉 Kotlin, JS, C/C++, groovy 语言，有实际 NDK 层代码开发的经验；熟悉小程序开发以及 Java 后台开发；了解 Flutter 开发。

工作经验

时间：2021.6—至今　　单位：深圳依时货拉拉科技有限公司

性质：互联网/IT　　部门：小拉出行

规模：　5000人　　职位：高级Android工程师

时间：2017.9—2021.5　　单位：深圳市云辉牧联科技有限公司

性质：互联网/IT　　部门：研发部

规模：　100人　　职位：高级Android工程师

时间：2015.3—2017.9　　单位：深圳市蝶讯网科技有限公司

性质：互联网/IT　　部门：产研部

规模： 500-2000人　　职位：Android工程师

项目经验

项目一： **小拉出行**
项目名称： **小拉出行用户端APP**
开发周期： 2021.06/至今
使用技术： MvvM架构 + 组件化 + hook + mqtt通讯 + IM通讯 + 地图 + ktlint + x5 + Jetpack + 性能优化
责任描述：

1,主要功能模块开发，以及研发技术选型

2,启动速度优化、包体积优化、内存泄漏、布局优化等性能优化

技术描述：

1、在项目中引入代码格式检查ktlint，在commit 或者 push 的时候，对命名风格、代码格式、注释规范等进行检查，有效提高项目代码的规范程度。

2、性能优化：1）启动速度，编写启动框架，将每个功能的初始化放到单独的task中，并分配初始化的依赖顺序以及工作线程，使得启动速度从原来的5秒以上提升到3秒以内。但配置相对低的机型仍然很慢，将SplashActivity中的大部分逻辑移动到主页并延迟，在生产环境中可看到启动速度在3s以内用户占到93.9% 以上。 2）、包体积优化，从原来的90多MB缩减到50MB左右 3）、内存泄漏治理 4）、crash监控，目前控制在0.01%左右 5）、卡顿追踪分析等。

3、在使用神策SDK的时候，其内部用LruCache自动缓存Activity中的Fragment，当Activity销毁的时候，被缓存的Fragment依然存活，从而导致内存泄漏，但sdk 并没有提供清除LruCache的方法。于是使用Hook方式，在监听到Activity销毁的时候，判断如果当前FragmentActivity，就遍历当前Activity中的Fragment，如果存在就通过反射拿到SDK的LruCache，并移除对应的Fragment 有效解决神策SDK导致的内存泄漏。

4、UI设计的时候要求将文字填充在一张背景图的两个气泡中，由于不同的机型可能有对不齐的情况，于是使用constraintDimensionRatio来设置UI稿上的宽高比，在通过Guideline作为TextView的基准线，有效地适配在不同机型上也能够居中于两个背景图气泡中。

5、在项目需求中，产品要求在节假日期间根据节假的不同更换不同的UI皮肤，开始的做法是更换所有UI，重新打包发布升级。这样的做法导致每次更换皮肤都需要重新发布升级包，效率较低。于是编写自适应皮肤升级框架，主要利用View的实例化流程自定义SkinLayoutInflaterFactory，统计需要换肤的属性，接着制作皮肤包，主体APP下载皮肤包并读取皮肤包中的资源信息。

6、自定义数字滚动RollingTextView。计算原数值与目标数值之间的差值得出需要在滚动中显示的数字，再通过AnimatorUpdateListener动态更新每个数字的baseline数值，达到滚动显示的效果。

项目二： 牛业联盟APP
项目名称： 散户肉牛养殖管理平台
开发周期： 2019.05/2021.05
使用技术： MvvM架构 + 自定义组件化路由框架 + Retrofit网络框架 + Kotlin + Hilt + WebView + 增量更新 + Jetpack等 。
责任描述：

1,参与项目需求分析及评估，制定项目计划、框架搭建

2,封装公共模块功能，性能优化

3,相关文档及软著编写

技术描述：

1，原有项目使用的Arouter 框架无法支持远端路由表（如：某些原生页面线上的crash， 支持跳转到发布的H5页面；新版本增加了jsbridge，老版本对应页面路由到升级页面。如果用户想用新功能，提示升级APP）等功能。基于Arouter原有功能的基础上，自定义了路由框架，新增了远端路由表下发，多path对应同一页面，无运行时反射，支持使用路由打开第三方库页面等功能特性。

2，利用Groovy 编写Gradle task，如：在对apk加固的时候，往往需要启动其GUI客户端或者在网页上面操作进行加固，利用Gradle插件完成加固程序的调用，自动完成整个加固。以及在打渠道包的时候，同样利用Gradle task 完成渠道打包工作，极大简化过程的繁琐。

3, 在项目开发过程中，使用网络的地方都需要显式地去处理请求异常和重试， 使用 LiveData+Kotlin+Retrofit2+OkHttp 的组合，并定义公共的响应实体和请求状态显示的ViewGroup，使得使用网络时加载、错误、重试等状态只需要简单的数据绑定就可以完成。

4，在进行一些购物车数据缓存时，以往是采用SharedPreference进行缓存，但由于SharedPreference工作机制的限制（每次存入都需要先读取全部再整合写入，以及xml解析比较耗时），数据较多时写入数据比较耗时。使用mmkv，整体性能得到了很大提升。

5, 随着项目逐渐扩展，项目模块多且复杂，编译一次太慢，改了一行代码 或只调了一点 UI，就要 run 整个项目。采用组件化重构项目，功能得到了重用，并且加快编译速度，由于不同组件分别由不同的项目组成员开发，不会影响其他组件，降低团队成员熟悉项目的成本，提高协作效率。

6, 在进行一些 IO 等耗时操作，以往使用线程池，这样 Thread 只能交给系统内核来调度去竞争 CUP 时间片，不可控且造成不必要的开销。 使用 Coroutines ，在不使用时可以手动取消，有效解决了以上问题。

7，每次版本更新，我们都需要将新版本的安装包放到服务器，然后用户端将完整的 apk 包下载，这里存在一个问题，即在 APK 包比较大或者网络不是很好的情况下，下载会很慢，或者消耗用户很大的流量。于是在项目中引入增量更新（基于 bsdiff 算法），即在软件更新时只更新差异化的部分，以达到用最小的下载量完成软件更新的需求。

项目三: 云牛牛APP
项目名称: 云牛牛肉牛管理平台
开发周期: 2017.09/2021.05
使用技术: MvvM架构 + NDK + RxJava/Retrofit网络框架 + 热修复更新 + Jetpack + RFID扫描技术 + 百度语音+ ARouter+ Zxing+ Socket等。
责任描述:

1,框架设计搭建，以及研发技术选型
2,封装公共模块功能
3.主要功能模块开发

技术描述:

1、在项目引入Jetpack全家桶，采用MVVM架构，替换了原来MVP架构，DataBinding双向绑定解决了需要编写大量接口的痛点。引入Hilt依赖注入框架，降低低了代码的耦合度，使得对代码的可扩展性得到了极大的提升。

2、在使用开发兽药二维码扫码时，发现部分二维码出现中文乱码的情况，优化了zxing core 库中的编码猜测规则，从新生成jar包并引用，从而解决中文乱码问题。

3、由于APP需要上传本地缓存的数据或从服务器下载数据，项目原来使用的service 在应用退到后台的情况下，可能会导致数据同步失败或者无法执行。引入了WorkManager替换一些后台任务，可以很大程度提升任务完成度。

4、由于国家农业部二维码使用扫描枪的进行扫码，无法适应移动互联网时代，使用NDK将由激光扫码解码逻辑改写封装成图像解码，并封装到JNI层，从而适用于手机进行扫码。

5、业务查看采用View Pager 嵌套Fragment的方式，原来默认预加载数据，导致不必要的网络等开销。根据Fragment生命周期的可见性来判断界面的可见性来加载网络数据，进行lazyLoad，优化网络请求api的设置，减少数据解析时间和网络访问时间，提升性能。

项目四: Diction APP
项目名称: Diction 资讯展示平台
开发周期: 2015.03/2017.09
使用技术: MvP架构 + RxJava/Retrofit网络框架 +Dagger2+ 热修复更新+微信/支付宝+Zxing二维码+ Glide，插件化。

责任描述:

1,参与项目需求讨论，以及技术框架选型
2,封装公共模块功能，技术攻克
3,完成产品功能迭代与性能调优

技术描述:

1，由于项目中大部分都是图片的展示，图片展示在低端机型容易内存溢出，基于Glide做了图片显示的优化，在原生基础上进行了NDK层面优化，提升了Glide性能。

2， 开发资讯模块时，由于不同的主题显示不同的界面，布局嵌套较深，在性能较低的机型会出现明显的卡顿。使用约束布局优化布局层次，并使用 merge 和 ViewStub 来减少布局层次和布局的延迟显示。

3， 在项目开发过程中，根据节日的不同需要经常手动替换 APP 中的图标并更新安装包。 利用 View 的实例化流程自定义 SkinLayoutInflateFactory，让后统计需要换肤的属性，在需要换肤的时候，制作一个皮肤包，在 APP 启动的时候读取皮肤包的皮肤并执行换肤，极大提高了换肤的效率。

4, 在开发视频过程中，一些开源框架无法满足部分需求，如播放格式，加水印，加边框等。自定义 Ffmpeg 可以满足不同的需求。

项目五: Dictioin Mall APP
项目名称: Diction Mall 订购商城
开发周期: 2015.03/2017.09
使用技术: MvP架构 + ViewPager/Fragment嵌套UI框架 + RxJava/Retrofit网络框架 + Webview + 热修复更新 + FFmpeg视频展示 + 微信/支付宝 + Zxing二维码。

责任描述:

1,参与项目需求讨论，以及研发技术选型
2,完成产品功能迭代与性能调优

技术描述:

1,在打开活动报名等详情 WebView 界面时，往往在第一打开时需要等待较长时间。于是在客户端刚启动时，就初始化一个全局的 WebView 待用，并隐藏；当用户访问了 WebView 时，直接使用这个 WebView 加载对应网页，并展示。并自定义了一套 WebView 与 native 交互的系统，从而提升 Webview 性能，提升用户体验。

2, 在项目开发过程中，由于某些偶发的 bug 而导致应用崩溃，如果要更整个 apk，则需要用户重新下载安装，给用户带来不好的体验。 使用 Tinker 热修复框架，可以在用户不觉察的情况下进行下载修复，缩短了更新的时间，提高了效率和用户体验。

3，在开发消息聊天模块过程中，由于第一次加载时请求较多的接口和刷新界面，此时会出现卡顿现象。利用 IdleHandler 可以获取 Handler 的空闲状态，在主线程 Handler 空闲的时候再去做耗时的操作，可以有效解决卡顿问题。

4，在开发 Zxing 二维码扫码时，发现扫码速度较慢。扫码时，通过设置裁剪预览区大小范围，从而减小预览图的大小，同时设置 Camera 自动聚焦，有效提高二维码扫码的效率。