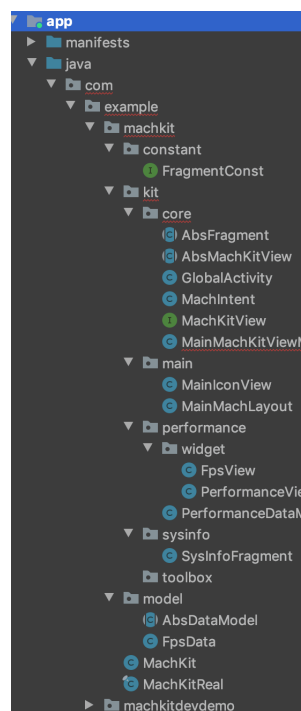


项目进展

- 总体概括：
 - 建立了项目初步的架构
 - 实现了view悬浮方案
 - 基本确定了获取接入方APP运行参数的实现
 - 建立绝大部分基类
-

- 项目初步架构：

-



- 概括为下面五大模块：
 - core模块包含项目基类，接口，分离所有view的通用逻辑，以及事件通信的类。
 - main模块主要是处理悬浮方案，作为MachKit的入口。
 - performance模块包含所有显示APP runtime参数的view控件和获取信息的核心方法。
 - toolbox作为额外为接入方提供功能的模块，可能可以提供的功能包括：一键跳转开发者模式，沙盒显示文件结构，显示SP文件，显示数据库。
 - MachKit，暴露向接入方的入口。
-

- view悬浮方案：

- 用户传进Application参数后，注册监听生命周期的。在onActivityResumed环节，在R.android.content控件上attach一层FrameLayout，基于入口Icon拖动的逻辑处理都基于FrameLayout来实现。（重写点击事件的处理方法等）
-

- 获取接入方APP运行时参数：
 - FPS: Choreographer
 - 内存：以cmd的方式从linux内核读取
 - 网络：接入方设置网络请求的节点，例如请求发出，请求收到，开始json解析，json解析结束等。并且提供常用框架，如okhttp的一键化接入。
 - 冷启或其他运行时间：接入方设置。
 - 图片：提供常用框架如Fresco的图片显示占用内存的功能，由MachKit hook框架来完成。
 - 异常捕捉：支持对RxJava框架未能消费的Exception的捕捉，打印。