Gvoice安卓客户端说明

秦若溪

***申明：本说明不是客户端程序文档，并不覆盖包含程序的所有内容，只是就程序中部分代码进行说明。***

# 程序结构

客户端以MVC架构为设计目标，但由于Android本身并不很好地支持MVC设计，以及自身能力有限，因而并不完全满足MVC设计原则。

项目中Model部分在com.glodon.model及com.glodon.util中；Controller在com.glodon.gvoice中；View主要在/res/layout的各个xml文件中，但由于部分视图需要动态更新（如录音界面弹出的对话框），因此也有少量视图写在了MainActivity中。

## Model层

Model层主要执行业务逻辑。其中ChatEngine里包含主要的业务逻辑，包括接收用户提问、返回查询结果等工作。

Downloader负责异步下载，如下载图片等工作。

Recorder负责录音，并将音频存为wav文件。

com.glodon.util包中为使用到的底层工具，包括与服务器通讯等的程序在此包内，这些底层的信息被包装成WebMsg类，供ChatEngine使用。

## Controller层

Controller层主要工作为获取用户输入并将其传递给Model（主要是给ChatEngine），MainActivity负责主界面的显示与更新，按钮的响应等。

客户端主要需维护聊天列表，它由MsgAdapter作为Controller负责实时更新，聊天列表中的信息都为ChatMsg的子类，以满足进行显示的规范。

SettingPreferences负责将用户偏好设置存储于SD卡中。

## View层

View层主要为xml文件，但聊天界面中显示音量的对话框放在了MainActivity中，因为这是动态生成的。

# 可优化之处

本项目中将业务逻辑放在了Model中的ChatEngine里，本意为即使变更显示层，整个程序业务也无需变化。但是这样使得ChatEngine中产生了很多回调函数，用于跟Controller交互，使程序逻辑变复杂。而如果直接把业务逻辑放在Controller层，那么许多回调过程就可以省去，Controller就可以直接调用各种需要的工具来完成工作，不用再经ChatEngine转手。

因此或许本项目中的程序有更好的组织方式，能够既易于适应变化，又能使程序逻辑更清晰、简洁。