# 输入法前台设计文档

傅承彦 2011/5/31

目录

[输入法前台设计文档 1](#_Toc294624714)

[1. 整体架构 1](#_Toc294624715)

[com.aeviou: 1](#_Toc294624716)

[com.aeviou.keyboard 1](#_Toc294624717)

[com.aeviou.candidate 2](#_Toc294624718)

[com.aeviou.utils 3](#_Toc294624719)

[2. 运行流程: 4](#_Toc294624720)

[输入流程 4](#_Toc294624721)

[绘制流程 5](#_Toc294624722)

## 整体架构

### com.aeviou:

#### AeviouInputMethodService:

整个输入法程序的入口,负责处理系统输入法接口与aeviou内部架构的转换。

为Android输入法框架的一部分

### com.aeviou.keyboard

#### AeviouKeyboardView:

同为Android输入法框架的一部分，系统提供负责屏幕的绘制和接受屏幕输入的函数。

本类中使用了strategy模式，具体的绘制和输入处理有AbstractKeyboard的子类来完成。

在用户切换不同语言键盘时，替换掉currentKeyboard指针，实现不同输入逻辑的转换。

#### AbstractKeyboard：

触摸屏显示和输入接受的抽象接口，具体的实现包括：

滑行键盘:HexKeyboard，此键盘的绘制先是渲染到一张buffer中，随后在显示到屏幕上。这张buffer同时被Tipview用来进行手指遮挡区域的重绘。

一般的方形按键键盘：SquareKeyboard

#### AbstractKey:

屏幕上单个键盘的抽象类，实现单个按键的绘制与是否被点中的逻辑判断。具体的实现类包括：

HexKey, SquareKey.

#### InputLogic:

对键盘上各种不同类型的按键的处理逻辑的封装。

包括：一般按键，退格，空格，回车等。

#### SlidePath:

HexKeyboard中滑行路径的封装，利用PinyinTree对路径是否有效的判断。同时标记已有的路径和可能后继，使得HexKeyBoard能绘制出不同的颜色。

本类还负责对路径的模糊处理，在路径中加入可能被屏幕滑动事件所遗漏的按键。在每两个已知的滑动路径点或路径上的有效按键之间，找到可能经过但是遗漏的按键，对路径进行补全。

#### TipView：

将键盘在手指周围的一小块区域在屏幕上方重绘，即截取并显示手指附近区域的小块区域中的HexKeyboard对象内的 buffer贴图。使得不熟练的用户可以看到被手遮挡的区域。

#### KeyboardFactory:

从配置文件中读入键盘布局，并生产相应的AbstractKeyboard键盘对象。

### com.aeviou.candidate

#### CandidateBar：

输入拼音的中文候选框，最多显示6个字或包含6个字的词组。显示时会覆盖键盘切换的位置

#### CandidateKey：

CandidateBar上面显示的元素，可以是单个字，也可以是左中右三种词组中的一个字。组成词组时，通过改变背景图片使得几个字的背景可以连成一个词组的背景。

#### SentenceView:

CandidateBar上方的拼音显示，半透明效果。在有拼音输入是才显示。

### com.aeviou.utils

#### AeviouConstants:

保存全局变量，贴图常量，界面位置信息的常量，按钮标示符，以及用于在实际手机分辨率和内部逻辑分辨率之间转换的矩阵。

#### BitmapUtils：

从本地存储中读取各种图片的工具类

#### FileUtils：

负责读取词库文件，并且在首次运行中把词库合并放置在手机ROM中。

#### SoundUtils：

读取并且播放键盘的效果声音

#### Vector2F:

用于模糊路径计算的向量类

## 运行流程:

现以中文滑行输入为例

### 输入流程

用户点击CandidateBar，对应的单词背提交给系统

用户抬起手。同以上的传递顺序，SlidePath调用输入法后台，传入path里含有的字母。后台生成候选词，绘制到CandidateBar上

SlidePath判断输入的有效性：

* 如果是新的输入，计算已有path中的最后一点到新的输入点间的所有有效按键。把有效按键加入path。
* 如果是回退到路径中已经有的点，删除路径中被回退的点。

HexKeyboard把输入传递给SlidePath，并且通知TipView重绘

用户输入被转换成逻辑坐标，传递给HexKeyboard

AeviouKeyboardView 检测到用户输入

### 绘制流程

AeviouKeyboardView 绘制背景。

* 如果没有用户输入，在背景最上方绘制键盘和设置切换按钮。
* 如果有用户输入，在背景最上方绘制CandidateBar。

在AeviouKeyboardView上面建立两个新的window，绘制SentenceVIew和Tipview

TipView按照用户输入的位置，计算出相应bufferA中的位置。

* 截取bufferA的绘制区域，绘制在屏幕的上方。
* 绘制用户手指的位置

AeviouKeyboardView通知内嵌的HexKeyboard重绘

将绘制对象设置为一张与HexKeyboard等大的贴图bufferA

* 遍历其中所有的按钮，每个按钮按照自己不同状态绘制不同的字符和背景
* 通知SlidePath绘制滑行路径

HexKeyboard将bufferA绘制在自己的表面