

KanaAnki 日语假名记忆工具

KanaAnki 是一款日语假名记忆工具，旨在帮助正在学习日语五十音的同学利用碎片时间进行记忆和复习。应用的主要模块是“识字辨音”和“听音认字”，即屏幕上方随机出现日语的假名或者罗马音，用户在屏幕下方出现的四个选项中选择正确的选项，以此加深对五十音的记忆。应用还附带了一个易于查询的五十音图。



应用功能

个性化设置



程序允许用户在首页上设定日语假名的样式、罗马字转写法的样式和自动发音选项。



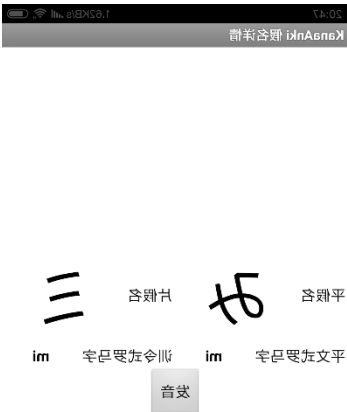
程序允许设定的假名样式为“显示平假名”、“显示片假名”以及同时“显示平假名和片假名”。

程序目前支持“平文式罗马字转写法”和“训令式罗马字转写法”。

如果用户开启了“自动发音”选项，则用户在点击“识字辨音”或“听音认字”屏幕下方的选项时，程序自动朗读选项上的假名或罗马字。

五十音图

当用户点击首页上的“五十音图”按钮时，屏幕跳转到五十音图界面。程序以五段十行的形式展现日语的假名学习中常用到的五十音图。当用户点击五十音图中的假名按钮时，屏幕跳转到该假名的详情界面。



假名详情

假名详情屏幕可以显示某一特定假名的平假名写法和片假名写法、平文式罗马字转写法和训令式罗马字转写法；用户点击“发音”按钮时，程序为用户朗读该假名的发音。



识字辨音

当用户点击著屏幕上的“识字辨音”按钮时，屏幕跳转到识字辨音界面。屏幕上方显示罗马字，屏幕下方出现假名的四个选项，用户需要在屏幕下方的四个假名按钮中选择与屏幕上方罗马字对应的正确选项；若用户选择正确，选项按钮的背景颜色以绿色反馈，并在下一秒后进入切换到下一题。当用户选择错误是，选项按钮以红色反馈，并等待用户更正选择。屏幕的左上角记录了用户选择正确和错误的次数。当用户不确定

ta



哪一项是正确的选项时，可以点击屏幕右上方的“查看详情”按钮跳转到“假名详情”屏幕，查看罗马字对应的假名信息。

听音认字

功能上与“识字辨音”类似，但屏幕上方出现的是假名，屏幕下方的选项是罗马音，此处不再赘述。

代码实现

KanaAnki 是一个开源项目，源代码可参阅项目 [GitHub](#)。

假名数据

KanaAnki 的假名数据以键值对的形式存放在数据库中。如图所示，每条假名数据包含四个键：hebon、kunrei、kira 和 kata，分别储存对应信息。

数据的储存和读取实现请参考主屏幕和其他相关屏幕代码。

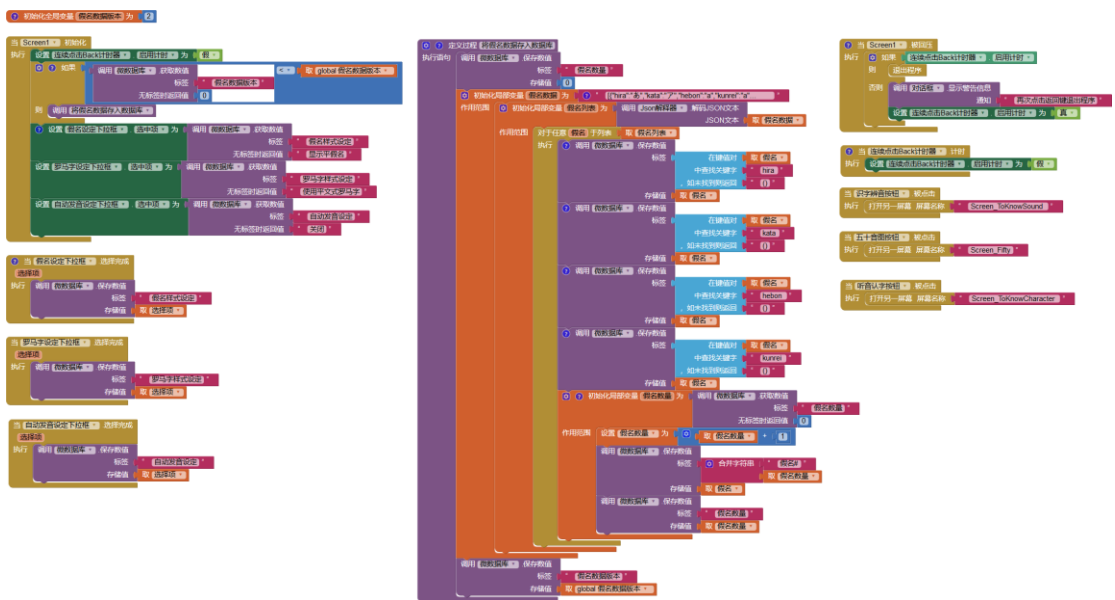
数据库的使用

KanaAnki 的重要储存介质是数据库。数据库储存的信息分为“系统设定”和“假名数据”两大类，结构如图所示。

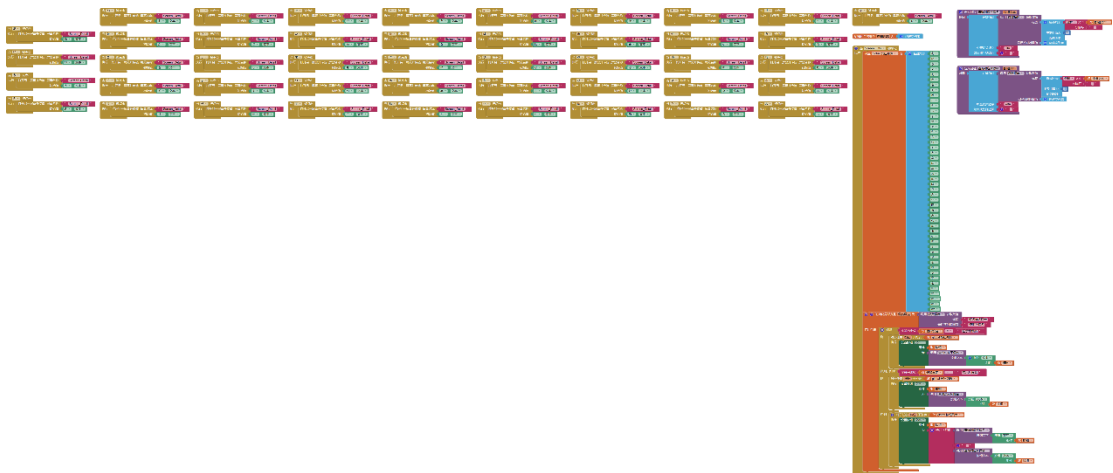
代码一览

源文件中有较为详细的注释，此处不再赘述。

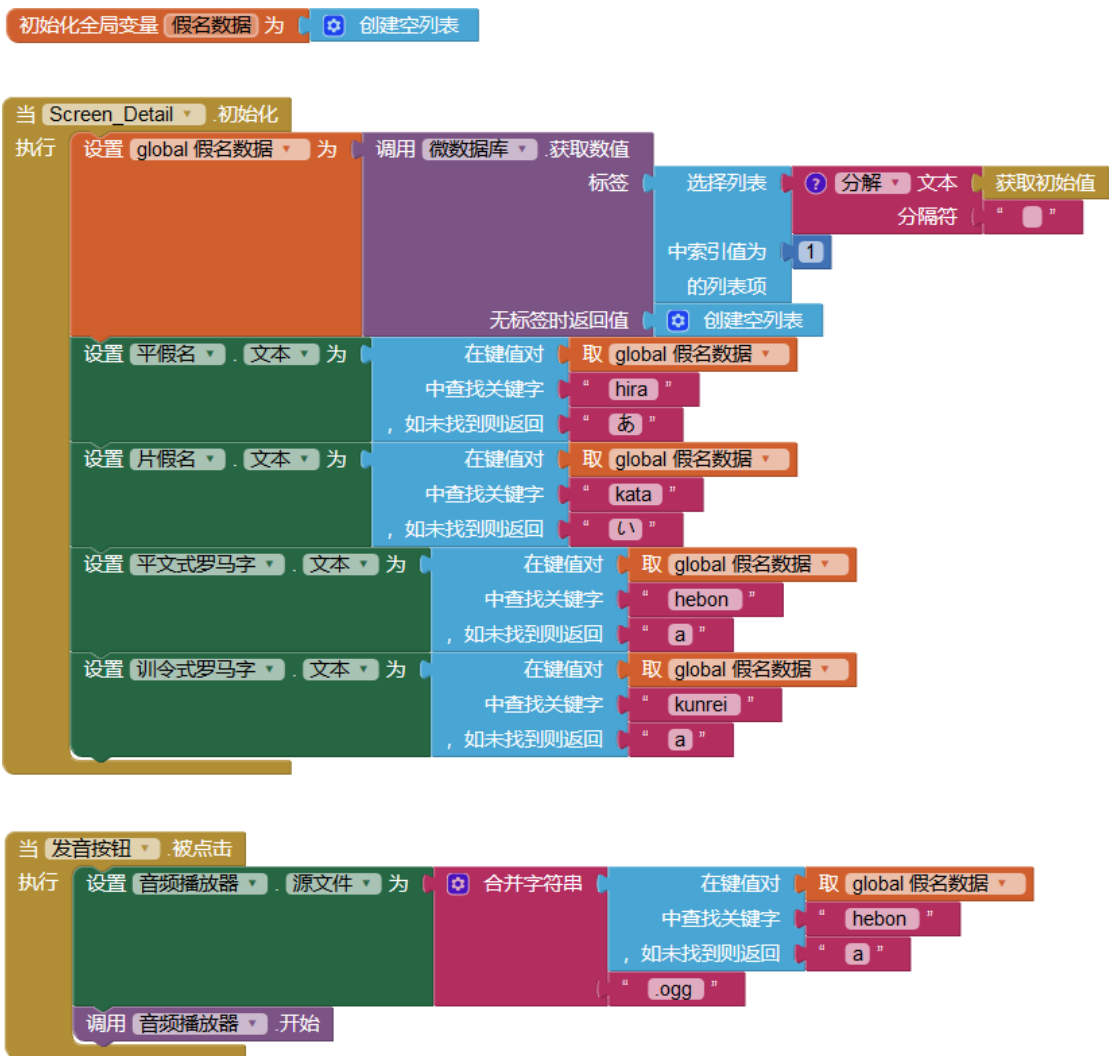
主屏幕代码一览



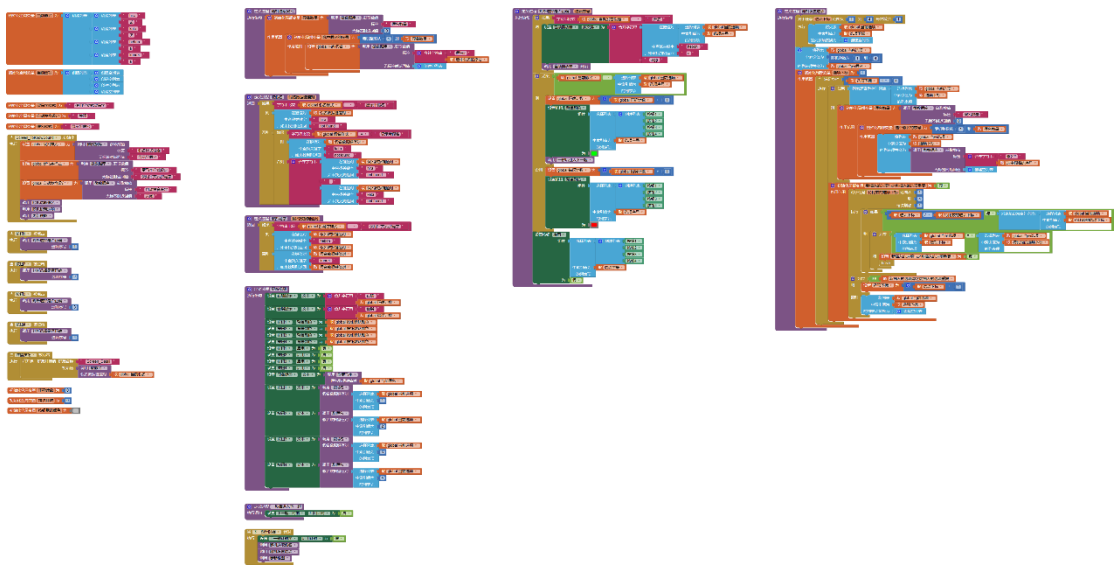
五十音图代码一览



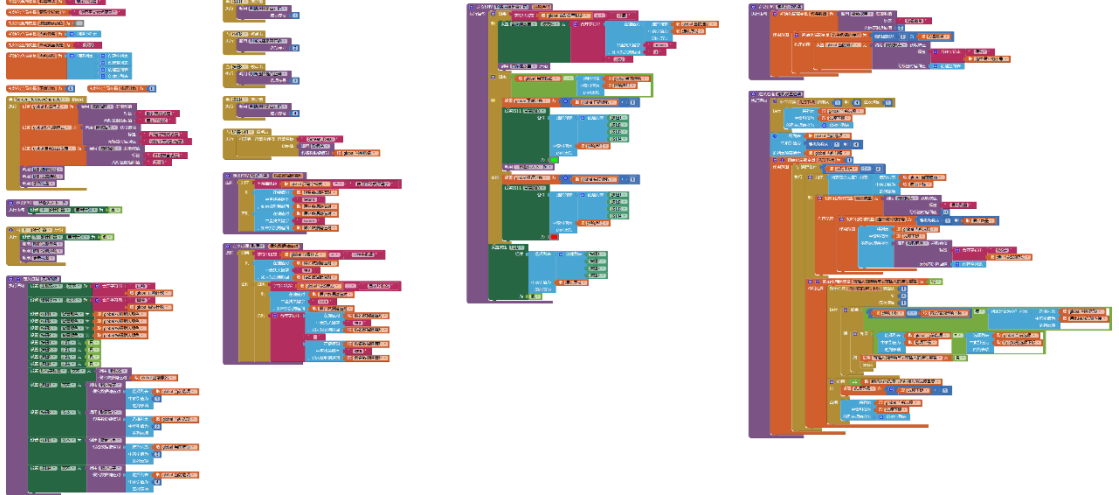
假名详情代码一览



识字辨音代码一览



听音认字代码一览



一点反思

使用 App Inventor 工具进行过一段时间的开发之后，我认识到 AI 在应用快速原型开发上面确实有很大的优势，特别是提供了原生应用开发中没有的实时调试功能，对生产效率的提高有所裨益。积木式的编程最大化地降低了学习的成本。

但是，不容忽视的是，当程序的体积膨胀到一定规模，AI 积木式的工具反而就没有那么直观了，尤其是当代码体积占据几个屏幕的大小的时候，由于 AI 没有提供搜索功能（如果是自定义的函数，有跳转到函数定义的功能），代码块自动排序功能也不如人意，有时候寻找一个特定的代码块就会花费许多时间，这就导致大规模的程序难于维护。

AI 的另一个限制是没有提供动态地修改应用布局的方法。这在应用使用的控件较多的情况下给开发者带来了比较大的麻烦。在五十音图代码的编写中，我拖动了五十个按钮，为

这五十个按钮编写功能完全相同的点击事件处理函数，这无疑是一件枯燥而容易出错的工作。

在图形界面的设计上，(III-ω-)。程序使用的是 AI 原生的界面，没有进行美化，仅仅具备一定的实用性，在“雅”的元素上有所欠缺。

在程序设计上，识字辨音和听音识字两个屏幕代码实现上的差异仅仅是更新视图函数中几个参数的差异，本该结合到一个屏幕中以减少代码的冗余。但初期设计中没有考虑到上面的情况，造成了现在程序冗余的问题。