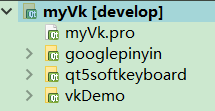
### 工程目录结构



工程目录如上图：

* googlepinyin工程：谷歌拼音输入法，c语言编写，只有功能，没有界面，项目输出静态库到qt5softkeyboard工程pinyin目录中（.pro中定义）。
* qt5softkeyboard工程：软键盘界面，使用了qt提供的插件类，加载了该插件的qt软件，会自动执行对应的函数，工程输出动态库到xdebug或xrelease中。
* vkDemo工程：示例软件。

### 拼音输入法的googlepyinyin库

采用的安卓底层的谷歌拼音输入法，在Qt5.7之后，Qt官方也出了一款输入法（支持中文拼音输入），基于QML设计的，其中的拼音输入法实现也是拿了安卓底层的谷歌拼音。对外提供的接口在pinyinime.h文件里面。

### 实现软件盘qt5softkeyboard动态库

**工程目录如下：**

* dict：存放最后要使用的二进制字典文件，此字典文件是谷歌拼音输入法根据最原始的汉字文件生成的，大家到时候直接使用就行了，不用深究此处。
* font：存放图标字体，FontAwesome.otf。Font Awesome 是一款可缩放的矢量图标字库。相信很多人对这个感到陌生，请大家自行搜索FontAwesome，以及如何在Qt中运用FontAwesome。
* images：按钮图片背景。
* pinyin：刚才编译出来的谷歌拼音库和头文件。
* googlepinyin.h，googlepinyin.cpp：谷歌拼音库的接口使用文件。
* keyboardform.h，keyboardform.cpp：键盘布局文件。
* keyeventdispacher.h，keyeventdispacher.cpp：键盘按键的事件处理文件。
* inputcontext.，platforminputcontextplugin.，Qt5Input.json，virtualkeyborad\_global.h：这几个文件不做赘述。
* res.qrc：资源文件。

**程序原理：**

使用了qt提供的输入法插件类进行了编写。

Qt应用启动之前会根据环境变量QT\_IM\_MODULE的值加载匹配的输入法插件，此过程可简单理解为：

1. Qt扫描当前目录下的platforminputcontexts或则是QT\_INSTALL/plugins/platforminputcontexts/路径下的.dll文件，并尝试加载。
2. 得到插件中实现了QPlatformInputContextPlugin类的实例。
3. 调用步骤2中实例的create方法，create的传入参数key为QT\_IM\_MODULE的值，如果该值与此插件所定义的值相匹配则生成QPlatformInputContext对象。
4. QPlatformInputContext对象即为输入法上下文对象，相当于输入法的入口。不过具体到实际开发中，我们是定义一个QPlatformInputContext的子类来实现自定义的输入法。

**输入法的工作原理：**

1. QWidget得到焦点。
2. Qt框架将该事件告知输入法上下文对象。
3. 输入法上下文对象查看有焦点对象的类型，如果需要弹出输入窗口，则将自身设置为有焦点对象的事件过滤器并根据有焦点对象的位置显示出输入法面板。
4. 根据需要，拦截发送给有焦点对象的相关按键事件，根据键值从输入法引擎中取得结果字符串。
5. 在适当时间点（比如检测到按回车键）将结果字符串发送给有焦点对象实现。
6. 如果焦点切换到其他非可输入Widget，会触发输入法上下文对象的QPlatformInputContext的hideInputPanel方法，自定义输入法类只需重载该方法即可知道何时隐藏输入法面板。

### demo示例

最终生成了qt5softkeyboard.dll，已经自动输出到vkDemo程序中运行环境xdebug或xrelease的platforminputcontexts中。

将qt5softkeyboard工程目录中 dict 整个文件夹复制到vkDemo.exe所在目录；

最后运行QtDemo.exe 即可。