###### int main(int argc, char \*argv[]) { QApplication a(argc, argv); QQmlApplicationEngine engine; engine.load(QUrl(QStringLiteral("qrc:/main.qml"))); if (engine.rootObjects().isEmpty()) return -1; return a.exec(); }QT学习笔记

QmlBook-In-Chinese.pdf

Qt5.12 3.9

Qml调用C++接口

动态加载

Qml和C++ 如何通信（q-》C 信号， q<-C C++传递）

QML，声明式脚本语言，QML（Qt Meta Language 或者 Qt Modeling Language），作为 C++ 语言的一种替代。而 Qt Quick 就是使用 QML 构建的一套类库。

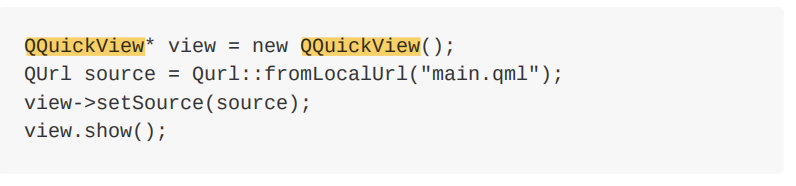
anchors.fill: *parent*

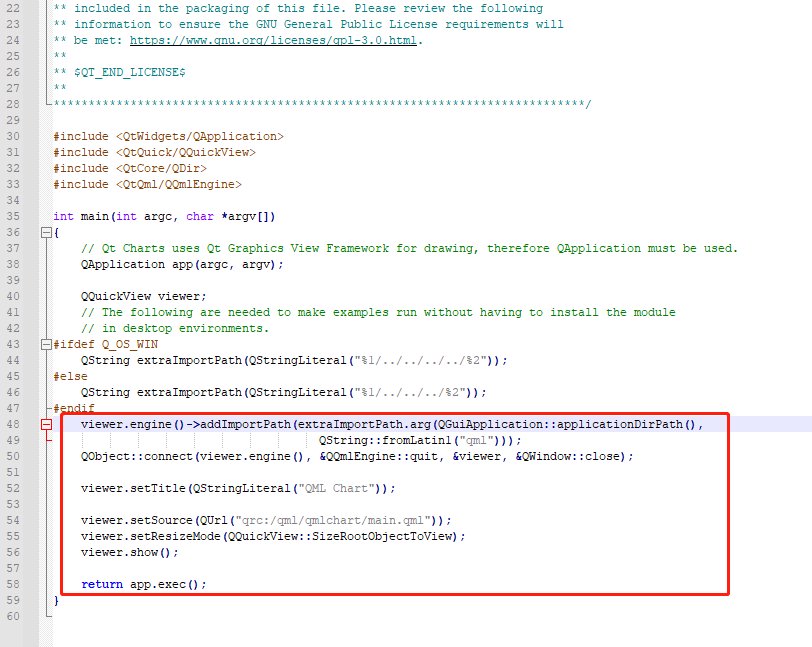
// Qt Charts uses Qt Graphics View Framework for drawing, therefore QApplication must be used.

QApplication app(*argc*, *argv*);

Qt图表使用Qt图形视图框架进行绘图，因此必须使用QApplication。

Qt提供了⼀个典型的运⾏环境叫做qmlscene，但是想要写⼀个⾃定义的允许环境也不是很困难，我们 需要⼀个快速视图（quick view）并且将QML⽂档作为它的资源。剩下的事情就只是展⽰我们的⽤户界⾯了。





viewer.setResizeMode(QQuickView::SizeRootObjectToView);//高分屏适配，自适应dpi

return app.exec();//软件不会闪退 。最后一个窗口关闭后，程序才停止

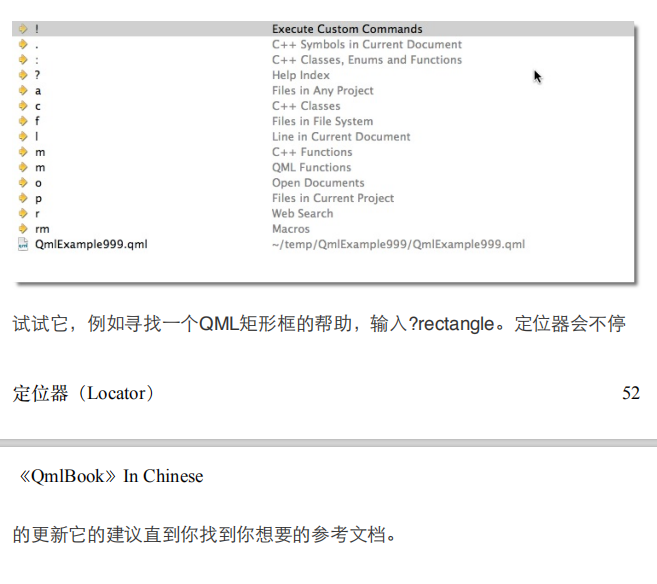
## **项目(.pro)文件**

一个项目文件是用来告诉qmake关于为这个应用程序创建makefile所需要的细节。例如，一个源文件和头文件的列表、任何应用程序特定配置。例如，一个必需链接的额外库或者一个额外的包含路径、都应该放到项目文件中。

****HEADERS****：应用程序中所有头文件的列表。  
****SOURCES****：应用程序中所有源文件的列表。  
****FORMS****：应用程序中的所有.ui文件(由Qt设计器生成)的列表。  
****TARGET****：可执行应用程序的名称。默认值为项目文件的名称。  
****INCLUDEPATH****：应用程序所需的额外的包含路径的列表。比如：INCLUDEPATH += $$FFMPEG\_INCLUDE，其实$$FFMPEG\_INCLUDE中的FFMPEG\_INCLUDE是自定义变量，表示ffmpeg的头文件路径  
****LIBS****：指定链接到项目中的库列表。比如：LIBS += $$FFMPEG\_LIB/libavcodec.so，表示链接FFmpeg的库目录下的libavcodec.so这个动态库。

****MOC\_DIR****：指定来自moc的所有中间文件放置的目录  
****OBJECTS\_DIR****：指定所有中间文件.o(.obj)放置的目录  
****RESOURCES****：指定资源文件(qrc)的名称，例如，RESOURCES += \  
mouse.qrc

定位器



快捷键（Shortcuts）

：

Ctrl+B - 构建项⺫

Ctrl+R - 运⾏项⺫

Ctrl+Tab - 切换已打开的⽂档

Ctrl+k - 打开定位器

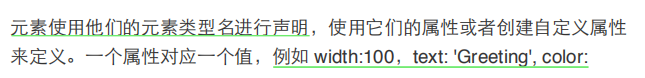
Esc - 返回

F2 - 查找对应的符号解释。

F4 - 在头⽂件与源⽂件之间切换（只对c++代码有效）

这些快捷键的定义来⾃[Qt Creator shortcuts](http://qt-project.org/doc/qtcreator-2.8/creator-keyboard-shortcuts.html)

属性



⼀个属性有⼀个类型定义并且需要⼀个初始值。

(组件开始再看)

anchors.fill和anchors.centerIn区别

anchors.centerIn:parent,是将子控件放在父控件的正中心，子控件的宽高是自己设置的；anchors.fill：parent, 是在子控件的大小设置与父控件大小一样，特别是mouseArea中经常使用anchors.fill：parent，这是为了指定鼠标事件接受的范围。如果是两个矩形控件，颜色不同，那么子控件会完全覆盖父控件，全是子控件的颜色显示。

还有一点，就是parent是指父类，直接相关的父类。anchors.fill指定的父类是必须是直接的父类，如果不是，会报错的。

————————————————

版权声明：int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

QQmlApplicationEngine engine;

engine.load(QUrl(QStringLiteral("qrc:/main.qml")));

if (engine.rootObjects().isEmpty())

return -1;

return a.exec();

}

本文为CSDN博主「那年晴天」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/lichen18848950451/article/details/79185611

自定义属性

qml支持自定义属性，调用自定义属性值显示文本信息

qml文件中，自定义属性以关键字property开始，然后是说明类型，最后是属性的名称和值。

注册单例

qml还支持注册C++的单例模型，然后在qml中直接调用，实现了与C++代码的通信。

动画效果

qml支持动画效果，首先看下使用Behavior的效果图，窗体分为上下两个矩形，当鼠标按下上方区域不动的时候，下方矩形中的正方形就会上左上角移动到右下角，当鼠标释放的时候，正方形又会从右下角回到左上角。

