

全新打造的Qter开源社区！

短期即可入门及提高的优秀Qt系列书籍！

Qt 应用编程系列丛书

霍亚飞 程 梁 编著

Qt 5 编程入门

- **全新**，基于 Qt 5.3 编写，全面涉及 Qt Quick！
- **经典**，植根于 Qt 网络博客教程，可无限更新！
- **基础**，对每个知识点详尽讲解，并设计了示例程序！
- **系统**，与《Qt Creator 快速入门（第2版）》相辅相成！



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

Qt 应用编程系列丛书

Qt 5 编程入门

霍亚飞 程 梁 编著

清华大学出版社

北京 510000
清华大学出版社
北京 100084

清华大学出版社
地址：北京清华大学学研大厦A座
邮编：100084
电话：(010) 62770175
http://www.tup.tsinghua.edu.cn

北京航空航天大学出版社

内 容 简 介

本书是基于 Qt 5 的 QML 和 Qt Quick 入门书籍,详细介绍了 QML 语言的语法和编写 Qt Quick 程序需要用到的基本知识点。本书内容主要包括 Qt 5 的介绍、QML 语法、Qt Quick 基础知识以及在图形动画、数据处理和多媒体方面的应用等。

本书侧重讲解 Qt 5 中 QML 和 Qt Quick 的内容,适合希望学习 QML 编程以及希望使用 Qt 开发移动应用的读者。对于没有任何 Qt 基础或者想学习 Qt C++ 编程的读者,可以参考《Qt Creator 快速入门(第 2 版)》一书。

本书内容源于作者的网络博客,作者会在网上及时解答读者疑问、更新修改内容、发布相关教程和配套资料,敬请读者关注 Qter 开源社区(www.qter.org)的相关内容。

图书在版编目(CIP)数据

Qt 5 编程入门 / 霍亚飞,程梁编著. —北京:北京航空航天大学出版社,2015.1
ISBN 978-7-5124-1667-3

I. ①Q… II. ①霍… ②程… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2015)第 009999 号

版权所有,侵权必究。

Qt 5 编程入门

霍亚飞 程 梁 编著
责任编辑 董丽娟 张耀军

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

北京楠海印刷厂印装 各地书店经销

*

开本:710×1 000 1/16 印张:24.25 字数:517 千字

2015 年 1 月第 1 版 2015 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-1667-3 定价:54.00 元

前言

本书的原型是网络上的一系列 Qt 博客教程,是该系列教程衍生出的第三本。2012 年出版的《Qt Creator 快速入门》和《Qt 及 Qt Quick 开发实战精解》至今已经有两年多的时间,在这期间 Qt 的命运发生了翻天覆地的变化。2012 年 8 月,诺基亚将 Qt 出售给同在芬兰的 Digia 公司,后者致力于发展 Qt 的商业授权用户,同时提供了 Qt 社区的良好运行环境。伴随着 Qt Project(qt-project.org)的成立,Qt 本身的开发终于成为一个真正的开放源代码项目。之后的 Qt 5 发布,使得 Qt 已经开始支持包括 Windows、Linux、Mac OS X、Android、iOS、BlackBerry、Sailfish 等在内的十几个桌面和移动平台。我们看到 Qt 的身世一波三折,但是又有足够的理由相信 Qt 的未来会越来越好。

因为在博客中发表 Qt 教程,所以认识了很多 Qt 爱好者和普及者,其中包括本书的另外一名作者程梁(网名豆子、devbean)。豆子与我的经历类似:我从 2009 年开始在百度博客上编写 Qt Creator 系列教程,因为图文并茂,简单易懂,所以得到了众多网友的肯定;而豆子也从 2009 年开始,在 51 CTO 博客中推出 Qt 学习之路系列教程,因为讲解细致,拥有自己的独到见解,所以受到了广大网友的热捧,后来被 51 CTO 专题收录,现在总浏览量已有数百万。后来我从百度博客迁移到自己的 yafeilinux.com 个人博客,而豆子也从 51 CTO 博客转移到自己的个人博客网站 devbean.net。有着如此相似经历的两个人走到一起会发生什么呢?就是组建了更符合网友需求的 Qter 开源社区(qter.org)。这个社区一直在发展,我们的目标就是要为各位致力于学习 Qt 的朋友提供一个便捷之门。

随着 Qt 5 的发布,我们经常可以在论坛上看到需要 Qt 5 相关教程和书籍的朋友。Qt 5 的全新框架、Qt Quick 2 技术以及最新版本的 Qt 提供了对现在非常流行的 Android 和 iOS 等移动平台的支持,都使得现在想要学习 Qt 的读者数量剧增。现阶段国内外图书市场上极少有关于 Qt 5 的书籍,但是却拥有众多需要该类书籍的初学者和开发者。为了解决这个矛盾,我和豆子决定再做一次第一个吃螃蟹的人,从而有了本书。

Qt 的历史

Qt 是由奇趣科技公司(Trolltech)的两位创始人 Haavard Nord 和 Eirik Chambe-Eng 合作开发而成的,首个版本完成于 1991 年,而第一个商业版本则是在 1995 年才正式推出。

Qt 的名字来源于一个有趣的故事。当时这个工具之所以取名为 Qt,是因为在 Haavard 的 Emacs 字体中,字母 Q 看起来非常漂亮;而字母 t 的灵感则来源于当时的另外一个工具 Xt(X toolkit)的取名。

最初 Qt 只有两个版本:适用于类 Unix 平台的 Qt/X11 和适用于 Windows 平台的 Qt/Windows。其中,Windows 版本只有专利授权,这意味着如果需要将使用 Qt/X11 编写的开源应用移植到 Windows 平台,必须购买专利授权。

2001 年底,奇趣科技发布 Qt 3.0,从这个版本起,Qt 增加了对 Mac OS X 平台的支持。不过,直到 2003 年 6 月之前,Mac OS X 版本的 Qt 也只有专利授权。当 Qt 3.2 发布的时候,奇趣科技才增加了 Mac OS X 平台的 GPL 授权。

2005 年 6 月 28 日,奇趣科技发布了 Qt 4.0。这是一个全新的版本,与之前的 3.x 系列不仅二进制不兼容,甚至 API 也不兼容。这在一定程度上使人颇有微词。

2008 年 6 月 17 日,诺基亚公司宣布以公开竞购的方式收购奇趣科技,连同奇趣科技旗下的 Qt。同年,诺基亚将 Qt 的名字更改为 Qt Software,然后又更改为 Qt Development Framework。在诺基亚的领导下,Qt 的工作重心由桌面系统转移至诺基亚旗下的手持设备。2009 年 5 月 11 日,诺基亚宣布 Qt 源代码在著名的 git 托管平台 Gitorious 面向公众开放,标志着 Qt 正式成为面向社区的开源框架。不过,尽管诺基亚承诺 Qt 开放源代码,但事实上 Qt 的代码提交与维护仍然牢牢掌握在诺基亚手中。2009 年 12 月 1 日,Qt 4.6 发布,这是第一个移植到 Symbian S60 平台的版本。

2011 年 2 月,诺基亚宣布放弃自己的 Symbian 平台,转而投向微软公司的 Windows Phone 平台。一个月之后,诺基亚宣布将 Qt 的商业授权和专业服务出售给 Digia 公司。后者启动了 Qt Project,同时宣布将努力促使 Qt 支持 Android、iOS 和 Windows Phone 三大平台,并且继续关注于桌面和嵌入式平台的开发,这意味着 Qt 正在努力成为一个全平台的开发框架。现在看来,在一定程度上,Digia 的确做到了这一点。

Qt 4 之后的下一个大的版本 Qt 5 原计划于 2012 年 6 月发布,但是由于诺基亚的政策调整和出售 Qt 的缘故,直到 2012 年 12 月 19 日,Qt 5.0 才正式发布。这是继 Qt 4 之后另一个大的升级。表面看来,Qt 4 到 Qt 5 的改动并不比 Qt 3 到 Qt 4 来得激进,但是 Qt 5 引入了全新的硬件加速图形处理,并且将 QML 与 JavaScript 提升到同 C++ 相等的地位。传统的基于 C++ 的 Qt Widgets 仍将继续支持,但是全新的架构所带来的性能提升则更多作用于 QML 和 JavaScript。

值得一提的是,Qt 5 的开发真正由 Qt Project 社区驱动,现在 Qt 已经允许诺基亚和 Digia 之外的开发人员提交并审核代码。

Digia 接管 Qt 业务以后,Qt 的开发速度有了明显提升。2013 年 7 月 3 日,Qt 5.1 发布。Qt 5.1 除了修改 5.0 版本的 bug 之外,更带来了 Android 和 iOS 平台的实验性支持。2013 年 12 月 12 日发布的 Qt 5.2 版本,则正式引入对 Android 和 iOS 平台的支持;同时,Qt 5.2 引入了一个新的场景图像渲染器,针对 OpenGL 后端提升了矢量绘制的性能,将 GPU 的占用降至最低。2014 年 5 月 20 日,Qt 5.3 发布。Qt 5.3 改进了对 iOS 平台的支持,并且开始支持 WinRT 平台。这为 Qt 成为全平台的开发工具更进一步。

Qt 授权

Qt 的授权一直比较复杂,既包括商业授权,又包括开源授权,并且不同版本之间的

授权并不一致。下面我们来详细介绍下有关授权的问题。

首先需要说明的是,Qt 自发明以来一直都有一个商业授权。这个商业授权允许开发者开发专有程序。同时,Qt 商业授权除了包含 Qt 开源版的一切功能外,还包括一些企业应用的组件,比如 Qt Charts、Qt Data Visualization 等;还有 Digia 公司支持的 Qt 培训等内容。

真正复杂的是 Qt 开源授权。Qt 1.45 之前,Qt 的源代码一直遵循 FreeQt 协议。该协议既不符合 Open Source Initiative 定义的开放源代码原则,也不符合 Free Software Foundation 定义的自由软件。在此协议下,Qt 的源代码虽然可用,但是并不允许重新发布修改过的版本。

1998 年,基于 Qt 的 KDE 成为 Linux 最流行的桌面环境之一。不过,因为 Qt 协议的问题,很多人担心这将给 KDE 桌面环境的发展带来影响。这也正是日后 Qt 协议问题的根源之一。

Qt 2.0 的开源授权由 FreeQt 更改为 Q Public License(QPL)。这是一个自由软件协议,但是与 GPL 不兼容,而后者才是 Linux 遵循的协议。作为妥协,KDE 团队与当时的奇趣科技达成协议,即使奇趣科技濒临破产,也不能使 Qt 的协议比 QPL 更加专有。直到 2000 年,Qt/X11 2.2 正式以 GPLv2 协议发布,才终结了这一授权的法律问题。

2002 年,KDE on Cygwin 项目开始将 GPL 授权的 Qt/X11 移植到 Windows 平台。此时,Windows 平台还没有一个开源版本的 Qt。不过,该项目并没有取得成功,但却直接导致奇趣科技在 2005 年 6 月将 Qt/Windows 4 以 GPL 授权的形式发布。至此,Qt 4 在全部主流桌面平台都有了基于 GPL 协议的开源版本。随着 GPLv3 发布,Qt 又增加了具有额外条款的 GPLv3 协议。这个“额外条款”允许将 Qt 编写的最终应用程序以不兼容 GPL 的自由软件/开源协议发布。

虽然此时的 Qt 已经有了自己的开源版本,但是作为一个类库,使用 GPL 协议开源具有一定的危险性。由于 GPL 的传染性,GPL 不允许发布闭源程序。这种协议将极大地限制 Qt 在商业软件领域的应用。所以在 2009 年 1 月 14 日,Qt 4.5 终于众望所归地增加了 LGPL 协议,允许使用 Qt 开发闭源程序。

尽管历史很复杂,但是我们的结论很简单:最新的 Qt 5.3 版本使用两种协议发布——商业协议和 LGPL 2.1。

本书特色

本书是经典的 Qt 应用编程系列丛书继《Qt Creator 快速入门》和《Qt 及 Qt Quick 开发实战精解》之后的又一力作,是市面上为数不多的全面介绍 Qt 5、QML 和 Qt Quick 的入门书籍。与其他相关书籍最大的不同之处还包括,本书是基于网络教程的。综合来说,本书主要具有以下特色:

- 最新。本书基于最新的 Qt 5.3.0 和 Qt Creator 3.1.1 版本进行编写,Qt 5.3.0 是 Qt 5 的最新的稳定版本。
- 基于社区。本书以 Qt 开源社区为依托,由社区站长合作完成。读者可以通

过论坛、邮件、QQ 群等方式和作者零距离交流。

- 无限更新。本书对应的网络教程是无限更新的,本书已经是该网络教程衍生的第三本书了。
- 全新风格。本书力求以全新的视角,引领开发者进行程序代码的编写和升级,同时以初学者的角度进行叙述,每个小知识点都以一个完整的程序来讲解。尽量避免晦涩难懂的术语,使用初学者易于理解的平白的语言编写,目标是让初学者在快乐中掌握知识。
- 授之以渔。在整书的编写过程中,都是在向读者传授一种学习方法,告诉读者怎样发现问题、解决问题,怎样获取知识,而不是向读者灌输知识。本书的编写基于 Qt 参考文档,所讲解的知识点多数是 Qt 参考文档中的部分内容,大家在学习时一定要多参考 Qt 帮助文档。在本套书籍讲解的所有知识点和示例程序中,都很明显地标出了其在 Qt 帮助中对应的关键字,从而让读者对书中的内容有迹可循。

本书结构

本书着重讲解 Qt 5 中 QML 和 Qt Quick 的编程知识,首先对 Qt 5 进行了整体介绍以及如何从 Qt 4 项目移植到 Qt 5 的介绍,然后讲解了 QML 语法和 Qt Quick 的基础内容,后面的几章讲解了 Qt Quick 在图像特效、绘画、多媒体、数据显示和处理等方面的高级应用。全书的目录结构如下表所列。

目录	介绍
第 1 章 Qt 5 简介	Qt 5 整体介绍,从 Qt 4 过渡到 Qt 5 需要了解的内容
第 2 章 将 Qt 4 代码迁移到 Qt 5	
第 3 章 Qt Quick 的世界	QML 和 Qt Quick 基础内容, 编写 QML 应用程序必备的基础知识
第 4 章 QML 语法	
第 5 章 Qt Quick 基础	
第 6 章 Qt Quick 控件和对话框	
第 7 章 图形动画基础	
第 8 章 图形效果	Qt Quick 高级应用,要设计功能强大、 界面绚丽的 Quick 应用必备的专业知识
第 9 章 粒子系统	
第 10 章 Canvas 基础	
第 11 章 模型和视图	
第 12 章 多媒体应用	
第 13 章 QML 与 C++ 的集成	
第 14 章 使用 Qt Quick 设计器	

另外需要说明,使用 Qt 进行 Android 和 iOS 等移动平台的开发部分(包括环境设置、传感器应用、蓝牙通信、摄像头控制、多点触控等相关内容),鉴于相关内容还不是非常成熟,而且依赖于特定设备,所以该部分的内容没有直接编写到书中,而是放到了网

络教程中。这样可以更及时地进行更新,并且对设备等相关的问题与读者进行直接探讨。

使用本书

本书的主要内容是讲解 Qt 5 中的 QML 和 Qt Quick 开发,另外还包含了对 Qt 5 框架和移植的介绍等内容。对于 QML 初学者,建议先学习 QML 和 Qt Quick 的基础知识,等有了一定的基础后再来学习后面的内容。

本书假设读者已经具有一定的 Qt 编程基础,因而不会对 Qt 的基础面面俱到。如果大家希望系统地学习 Qt 编程,最好先自行学习基于 C++ 的 Qt Widgets 编程方面的内容。可以参考 Qt 应用编程系列丛书中的《Qt Creator 快速入门(第 2 版)》一书。《Qt Creator 快速入门(第 2 版)》的写作风格与本书一致。

对于学习过《Qt 及 Qt Quick 开发实战精解》(以下简称《精解》)一书的读者,这里做一个解释:虽然《精解》一书中也包含了 Qt Quick 的内容,但那是 Qt 4.7 时代的 Qt Quick,因为是新引进的技术,所以非常不成熟。本书中的 QML 和 Qt Quick 的内容是基于最新的 Qt 5。新版本的 Qt Quick 已经进行了巨大的更新。另外,在以后《精解》一书再版时,其中的 Qt Quick 内容会被删除。

在学习过程中,建议读者多动手,尽量自己按照步骤编写代码。只有在遇到自己无法解决的问题时,再去参考本套书籍提供的源代码。每当学习一个知识点时,书中都会给出 Qt 帮助中的关键字,建议大家详细阅读 Qt 帮助文档,看下英文原文是如何描述的。不要害怕阅读英文文档,因为很难在网上找到所有文档的中文翻译,并且即使翻译,也有可能偏离原意,所以最终还是要自己去读原始文档。学会看参考文档是入门 Qt 编程的重要一步。不要说自己英文不好,只要坚持,掌握了一些英文术语和关键词以后,阅读英文文档是不成问题的。

书中使用的 Qt 版本的说明

本书基于 Windows 平台 Qt 5.3.0 和 Qt Creator 3.1.1 版本。它们是本书完稿时的最新版本。为了避免读者使用不同的操作系统而产生不必要的问题,本书采用了常用的 Windows 7 操作系统。这里要对 Qt 版本不是很了解的读者说明一下,对于 Qt 程序开发,无论是在 Windows 系统下进行开发,还是在 Linux 系统下进行开发;无论是进行桌面程序开发,还是进行移动平台或者嵌入式平台的开发,只需要编写一次代码,然后分别进行编译就可以了,这就是 Qt 最大的特点,即所谓的“一次编写,随处编译”。也就是说,读者需要学好本书中 Qt 的基本内容,然后编写代码,使用 Qt 不同的版本进行移植、编译即可。

在学习本书时,推荐大家使用指定的 Qt 和 Qt Creator 版本,因为对于初学者来说,任何微小的差异都可能导致错误的理解。当然,大家也可以使用其他版本,Qt 的下载地址为:http://download.qt.io/official_releases/qt/。



录

第 1 章 Qt 5 简介	1	2.5 全新的插件系统	29
1.1 Qt 5 架构	1	2.6 小结	32
1.1.1 模块架构	1	第 3 章 Qt Quick 的世界	33
1.1.2 图形界面库的架构	5	3.1 全新的 QML 和 Qt Quick ...	33
1.1.3 Qt 5 架构主要特点	6	3.2 Qt Quick 项目	36
1.2 Qt 5 的特点	6	3.2.1 Qt Quick UI 项目	36
1.2.1 Qt 5 新增的主要功能	6	3.2.2 Qt Quick Application 项目	39
1.2.2 Qt 5 与 Qt 4 的兼容性	11	3.2.3 运行示例程序	42
1.2.3 C++ 还是 QML	12	3.2.4 Qt Quick 程序的发布	42
1.2.4 Qt 5 源代码文件的编码	12	3.3 将 QML 程序迁移到 Qt 5 ...	42
1.2.5 移动平台开发	13	3.3.1 Qt 5 中 QML 和 Qt Quick 的更改	43
1.3 小结	13	43
第 2 章 将 Qt 4 代码迁移到 Qt 5 ...	14	3.3.2 QML 程序移植示例	46
2.1 Qt 5 版本的 HelloWorld	14	3.4 小结	48
2.1.1 Qt 5 的下载与安装	14	第 4 章 QML 语法	49
2.1.2 创建 Hello World 程序	15	4.1 QML 语法基础	49
2.1.3 Qt 4 程序迁移到 Qt 5 的注意事项	17	4.2 import 导入语句	51
.....	17	4.2.1 模块(命名空间)导入语句	52
2.2 Qt 4 程序迁移实例	19	4.2.2 目录导入语句	53
2.2.1 修改编码	19	4.2.3 JavaScript 资源导入语句	55
2.2.2 修改代码	20	4.3 QML 类型系统	55
2.2.3 设置应用程序图标	21	4.3.1 基本类型	55
2.2.4 发布程序	22	4.3.2 JavaScript 类型	57
2.3 新的信号槽语法	22	4.3.3 对象类型	57
2.3.1 新旧语法对比	22	4.4 对象特性(Attributes)	57
2.3.2 新的语法示例	23	4.4.1 id 特性	58
2.4 对 C++11 的支持	26	4.4.2 属性特性	58
2.4.1 Lambda 表达式	26	4.4.3 信号和信号处理器特性	65
2.4.2 适用于 C++11 的宏	28	4.4.4 方法特性	69

4.4.5 附加属性和附加信号处理器	70	件(WheelEvent)	135
4.5 集成 JavaScript	72	5.3.3 拖放事件(DragEvent)	136
4.5.1 JavaScript 表达式	72	5.3.4 键盘事件(KeyEvent)和焦点作用 域(FocusScope)	140
4.5.2 从 JavaScript 动态创建 QML 对象	77	5.3.5 定时器(Timer)	145
4.5.3 在 QML 中定义 JavaScript 资源	81	5.4 使用 Loader 动态加载组件	145
4.5.4 在 QML 中导入 JavaScript 资源	83	5.4.1 Loader 的大小与行为	146
4.5.5 JavaScript 宿主环境	85	5.4.2 从加载的项目中接收信号	147
4.6 QML 文档	86	5.4.3 焦点和键盘事件	147
4.6.1 通过 QML 文档定义对象类型	87	5.5 小结	148
4.6.2 QML 组件	89	第 6 章 Qt Quick 控件和对话框	149
4.6.3 作用域和命名解析	90	6.1 构建第一个示例	149
4.6.4 资源加载和网络透明性	95	6.2 ApplicationWindow 应用程序窗 口	150
4.6.5 QML 的国际化	97	6.3 Window	154
4.6.6 QML 的编码约定	100	6.4 按钮类控件	155
4.7 QML 模块	103	6.4.1 ExclusiveGroup	156
4.7.1 定义一个 QML 模块	103	6.4.2 Button	157
4.7.2 支持的 QML 模块类型	104	6.4.3 CheckBox	157
4.8 小结	104	6.4.4 RadioButton	158
第 5 章 Qt Quick 基础	105	6.4.5 Switch	158
5.1 基础可视项目	105	6.5 数据选择类控件	159
5.1.1 Item	105	6.5.1 ComboBox	159
5.1.2 Rectangle	109	6.5.2 Slider	160
5.1.3 Text	110	6.5.3 SpinBox	161
5.1.4 TextInput	117	6.6 文本类控件	161
5.1.5 TextEdit	120	6.6.1 TextField	161
5.2 布局管理	121	6.6.2 TextArea	162
5.2.1 定位器(Positioners)	121	6.7 其他控件	163
5.2.2 基于锚(anchors)的布局	126	6.7.1 Label	163
5.2.3 Layouts	129	6.7.2 GroupBox	163
5.3 事件处理	133	6.7.3 BusyIndicator	164
5.3.1 MouseArea	133	6.7.4 ProgressBar	164
5.3.2 鼠标事件(MouseEvent)和滚轮事 件(WheelEvent)	135	6.8 导航类视图	165
		6.8.1 ScrollView	165
		6.8.2 SplitView	165

6.8.3	StackView	166	7.6.2	翻转效果(Flipable)	206
6.8.4	TabView	170	7.7	小 结	207
6.8.5	TableView	171	第 8 章	图形效果	208
6.9	标准对话框	172	8.1	混合效果(Blend)	209
6.9.1	ColorDialog	173	8.2	颜色效果(Color)	211
6.9.2	FileDialog	173	8.3	渐变效果(Gradient)	214
6.9.3	FontDialog	174	8.3.1	锥形渐变(ConicalGradient)	214
6.9.4	MessageDialog	175	8.3.2	线性渐变(LinearGradient)	215
6.10	定义控件样式	176	8.3.3	辐射渐变(RadialGradient)	215
6.11	小 结	177	8.4	变形效果(Distortion)	216
第 7 章	图形动画基础	178	8.5	阴影效果(Drop Shadow)	217
7.1	颜色、渐变和调色板	178	8.5.1	投影(DropShadow)	217
7.1.1	颜色(color)	178	8.5.2	内阴影(InnerShadow)	218
7.1.2	渐变(Gradient)	181	8.6	模糊效果(Blur)	219
7.1.3	系统调色板(SystemPalette)	181	8.6.1	快速模糊(FastBlur)	219
7.2	图片、边界图片和动态图片	182	8.6.2	高斯模糊(GaussianBlur)	220
7.2.1	图片(Image)	182	8.6.3	递归模糊(RecursiveBlur)	220
7.2.2	边界图片(BorderImage)	184	8.6.4	遮罩模糊(MaskedBlur)	221
7.2.3	动态图片(AnimatedImage)	185	8.7	动感模糊效果(Motion Blur)	222
7.3	缩放、旋转和平移变换	186	8.7.1	方向模糊(DirectionalBlur)	222
7.3.1	使用属性实现简单变换	186	8.7.2	径向模糊(RadialBlur)	223
7.3.2	使用 Transform 实现高级变换	188	8.7.3	缩放模糊(ZoomBlur)	224
7.4	状态(State)	189	8.8	发光效果(Glow)	225
7.4.1	创建状态	190	8.8.1	发光(Glow)	225
7.4.2	默认状态和 when 属性	191	8.8.2	矩形发光(RectangularGlow)	226
7.5	动画和过渡	192	8.9	遮罩效果(Mask)	227
7.5.1	触发动画	193	8.9.1	不透明遮罩(OpacityMask)	227
7.5.2	控制动画的执行	198	8.9.2	阈值遮罩(ThresholdMask)	228
7.5.3	精灵动画(Sprite Animations)	199	8.10	小 结	229
7.5.4	共享动画实例	203	第 9 章	粒子系统	230
7.6	Flickable 和 Flipable	204			
7.6.1	弹动效果(Flickable)	204			

9.1 ParticleSystem	230	11.2.5 Package	277
9.2 发射器(Emitter)	232	11.2.6 XmlListModel	278
9.3 渲染器(ParticlePainters) ...	233	11.2.7 LocalStorage	279
9.4 TrailEmitter	236	11.2.8 使用 C++ 扩展 QML 模型	282
9.5 粒子组	237	11.3 视图类型	285
9.6 随机参数	239	11.3.1 ListView	286
9.7 影响器(Affector)	240	11.3.2 GridView	291
9.8 小结	245	11.3.3 视图过渡	292
第 10 章 Canvas 基础	246	11.3.4 PathView	295
10.1 Canvas 的使用	246	11.3.5 WebView	298
10.2 绘制操作	248	11.4 调整性能	299
10.2.1 绘制参数设置	248	11.5 小结	300
10.2.2 绘制矩形	250	第 12 章 多媒体应用	301
10.2.3 状态的保存与恢复	251	12.1 多媒体模块介绍	301
10.2.4 绘制文本	252	12.2 播放音频	302
10.2.5 绘制路径	254	12.2.1 播放压缩音频	302
10.3 渐变填充	259	12.2.2 播放未压缩音频	304
10.3.1 线性渐变	259	12.3 播放视频	304
10.3.2 辐射渐变	260	12.3.1 使用 Video 播放视频文件	304
10.3.3 锥形渐变	261	12.3.2 对视频使用图形效果	305
10.4 阴影	261	12.4 媒体播放器(MediaPlayer)	308
10.5 使用图像	262	12.4.1 播放音视频	308
10.6 坐标变换	264	12.4.2 使用 Windows 平台附加功能	309
10.6.1 平移	264	12.5 小结	314
10.6.2 缩放	265	第 13 章 QML 与 C++ 的集成	315
10.6.3 旋转	266	13.1 QML 运行时的 C++ 类 ...	316
10.6.4 扭曲	267	13.1.1 QQmlEngine	316
10.6.5 transform() 函数总结	267	13.1.2 QQmlContext	317
10.7 小结	268	13.1.3 QQmlComponent	318
第 11 章 模型和视图	269	13.1.4 QQmlExpression	320
11.1 模型/视图架构简介	269	13.2 在 QML 中使用 C++ 特性 ...	320
11.2 数据模型	271	13.2.1 数据类型处理和所有权	320
11.2.1 ListModel	271		
11.2.2 WorkerScript	273		
11.2.3 ObjectModel	275		
11.2.4 DelegateModel	275		

13.2.2 数据类型的转换	321	13.5.3 使用 C++ 访问 QML 对象成员	340
13.2.3 使用 C++ 属性	323	13.6 小 结	343
13.2.4 使用函数和槽	327	第 14 章 使用 Qt Quick 设计器	344
13.2.5 使用信号	328	14.1 常用操作介绍	345
13.3 注册 QML 类型	329	14.1.1 管理项目层次	345
13.3.1 注册可实例化对象类型	329	14.1.2 QML 类型库	346
13.3.2 注册不可实例化对象类型	330	14.1.3 指定项目属性	347
13.3.3 注册单例类型	330	14.1.4 添加状态	349
13.3.4 类型的修订和版本	331	14.1.5 在画布上操作 QML 类型	349
13.4 定义 QML 特定类型和属性	332	14.2 使用设计器编辑 QML 文档	351
13.4.1 提供附加对象注解数据	332	14.2.1 创建项目	351
13.4.2 属性修饰符类型	335	14.2.2 创建主视图	351
13.4.3 指定 QML 对象类型的默认属性	337	14.2.3 添加视图	355
13.4.4 接收对象初始化通知	337	14.2.4 为视图添加动画	356
13.5 在 C++ 中使用 QML 对象	338	14.3 小 结	358
13.5.1 使用 C++ 加载 QML 对象	338	附录 A Qt 版本介绍	359
13.5.2 使用对象名字访问加载的 QML 对象	339	附录 B Qt 5 中 C++ API 更改	361
		附录 C QML 常用术语	371
		参考文献	373

第 1 章

Qt 5 简介

Qt 5 是跨平台应用开发框架 Qt 的最新版本,图形引擎基于 GPU,从而使应用程序能够充分利用硬件加速,同时包括了最新的应用开发套件 Qt Quick 2。作为 Qt 5 的核心之一,Qt Quick 2 提供了强大的功能,允许开发者进行适用于触摸屏的 UI 开发,并且保持与原生 C++ 程序几乎相同的运行效率。

在增加了对 C++ 11 支持的同时,Qt 5 还加强了对 JavaScript 和 QML 的支持,开发者可以更加高效和灵活地使用它们。不仅如此,QML 还可以与基于 C++ 的经典的 Qt Widgets 进行混合编程。Qt 5 是 Qt 4 渐进而平缓的升级,与 Qt 4 高度兼容,除了一些必要的情况,基本没有打破源代码级别的兼容性。

本章将从 Qt 5 的架构、功能更新等几个方面进行介绍,让读者对 Qt 5 有一个全面的认识。

1.1 Qt 5 架构

1.1.1 模块架构

Qt 5 采用了新的模块化代码库,使得平台移植变得更简单。Qt 5 将所有功能模块分为了 3 部分:Qt 基本模块(Qt Essentials)、Qt 扩展模块(Qt Add-Ons)和开发工具(Qt Tools)。可以在 Qt 帮助中通过 All Modules 关键字查看本节的内容。

Qt 基本模块定义了适用于所有平台的基础功能,是 Qt 的核心,其架构如图 1-1 所示。所有 Qt 应用程序都需要使用该模块中提供的功能。基本模块的基础是 Qt Core 模块,其他所有模块都依赖于该模块。Qt 保证,在 Qt 5 的整个生命

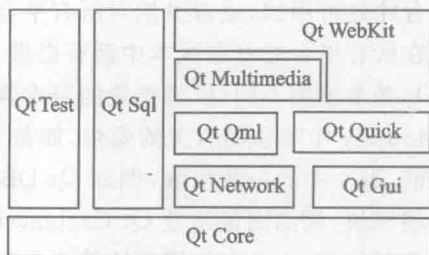


图 1-1 Qt 5 基本模块架构示意

周期内,这部分模块均会保持源代码甚至二进制级别的兼容性。另外,如果某个功能,即便是所有平台都能够支持,但是仅仅为了满足某个特定的功能或目的,那么这个功能也不会进入基本模块。典型的例子是打印,虽然绝大多数平台都有支持(甚至包括某些移动平台),但是打印模块并不包含在基本模块之中。

最新的 Qt 5.3 中基本模块部分简介如表 1-1 所列。

表 1-1 Qt 5.3 的基本模块简介

模 块	简 介
Qt Core	Qt 5 的核心类库,其他各个模块都建立在 Core 模块之上。相对于 Qt 4, Qt 5 增加了 JSON 支持,并且将对 XML 的支持移入到 Core 模块,不再是独立的 XML 模块
Qt GUI	图形用户界面(GUI)开发的最基础的类库,包括各种交互事件等;同时,这个模块还包括了有关 OpenGL 的内容
Qt Multimedia	提供对视频、音频、无线电以及摄像头的支持
Qt Multimedia Widgets	提供基于组件的多媒体功能的支持
Qt Network	提供网络编程类库
Qt QML	提供对 QML 和 JavaScript 语言的支持
Qt Quick	提供一个用于创建高度动画效果的应用程序的声明式框架。该框架建立在 QML 和 JavaScript 的基础之上
Qt Quick Controls	提供一组基于 Qt Quick 的 UI 控件,用于创建经典的桌面风格的用户界面
Qt Quick Dialogs	提供一组适用于 Qt Quick 应用程序的系统对话框
Qt Quick Layouts	提供一组适用于 Qt Quick 的组件布局
Qt SQL	提供对关系数据库 SQL 的支持
Qt Test	提供供 Qt 应用程序和类库使用的单元测试工具
Qt WebKit	提供基于 WebKit2 引擎的一个 Qt 移植,以及新的面向 QML 的 API
Qt WebKit Widgets	提供基于 WebKit1 的一个移植,以及来自于 Qt 4 的所有有关 WebKit 的类
Qt Widgets	提供针对 Qt GUI 的扩展 C++ 组件

Qt 扩展模块是针对某种特殊目的的额外模块。这些模块或者仅在某个或某些特定平台才是可用的,或者虽然对所有平台均可用,但却不是通用的目的。扩展模块并不一定在所有平台的发布版本中都有提供,有些仅仅是在特定平台的发布版本中才会列出。与基本模块不同,扩展模块的版本兼容性是模块自己指定的。这其中包含了以前 Qt Mobility 中与移动有关的模块,如蓝牙 Qt Bluetooth、传感器 Qt Sensors 等;还包含了以前 Qt 4 中的一些模块,例如 Qt Dbus、Qt XML、Qt Script 等。除此之外,还新添了一些模块,例如图像特效 Qt GraphicalEffects、串口通信 Qt SerialPort 等。

最新的 Qt 5.3 中扩展模块简介如表 1-2 所列。

表 1-2 Qt 5.3 的扩展

模 块	开发平台	目标平台	简 介
Active Qt	Windows		提供对 ActiveX 和 COM 的支持
Qt Android Extras	全平台	Android	提供针对 Android 的专用 API
Qt Bluetooth	全平台	Qt for Linux/X11, BlackBerry	提供对蓝牙设备的访问
Qt Concurrent			提供一套不使用底层线程原语的高级多线程编程框架
Qt D-Bus	类 Unix		提供 D-Bus 协议之上的进程间通信
Qt Graphical Effects	全平台		提供 Qt Quick 使用的图像特效
Qt Image Formats	全平台		提供对 TIFF、MNG、TGA 和 WBMP 等图像文件格式的支持
Qt Mac Extras	全平台	Mac OS X	提供针对 Mac OS X 平台使用的专用 API
Qt NFC	全平台	BlackBerry	提供对近场通信 (Near-Field Communication, NFC) 设备的访问
Qt OpenGL			提供对 OpenGL 的支持。需要注意的是,这部分代码是从 Qt 4 移植过来的,仅为提供代码兼容。如果需要编写新的基于 OpenGL 的代码,则应该使用 Qt 5 的 QOpenGL 类
Qt Positioning	全平台		提供访问位置、卫星和地区监视设备的类
Qt Print Support	全平台		提供一组可移植的打印支持类
Qt Declarative	全平台		提供针对 Qt 4 的兼容性
Qt Script	全平台		提供一组脚本化 Qt 应用程序的类。需要注意的是,这部分类是从 Qt 4 移植过来的。针对新编写的代码,应该使用 Qt QML 模块提供的以 QJS 开头的类
Qt Script Tools	全平台		提供供 Qt Script 使用的组件。既然 Qt Script 都已经不再推荐使用,这个模块也仅仅为提供代码兼容而存在
Qt Sensors	全平台	Android, BlackBerry, Qt for iOS, WinRT, Mer	提供对传感器和动作手势识别设备的访问
Qt Serial Port	全平台	Windows, Linux, Mac OS X	提供对硬件和虚拟串口的访问
Qt SVG	全平台		提供显示 SVG 文件的类