

sina新浪博客

[公告]

一去、二三里

个人中心

发博文

消息

一去、二三里的博客

http://blog.sina.com.cn/liang19890820

首页

博文目录

图片

关于我

发博文

页面设置

个人中心

个人资料

[管理]

卧槽，炸弹。

Qing

微博

博客等级：18

博客积分：78

博客访问：407,807

关注人气：378

获赠金笔：12

赠出金笔：0

荣誉徽章：

相关博文

Qt之模型/视图（实时更新数据）
一去、二三里

Qt之模型/视图
一去、二三里

Qt之Windows开发移植
一去、二三里

cin详解（cin.get()、cin.getlin
一去、二三里

Qt之自定义界面（二）添加最小化
一去、二三里

Qt之中文显示（QMessageBox、QLi
一去、二三里

如何提取资源文件（QQ、360等）？
一去、二三里

最重要的十年做什么才不浪费？
一去、二三里

Qt之Esc键
一去、二三里

Qt之解决中文乱码
一去、二三里

更多>>

正文

字体大小：

大

中

小

Qt之模型/视图（委托）

(2014-01-09 13:58:09)

[编辑][删除]

+ 转载

标签：qt qt委托 qlistview委托 qtableview委托 qttreeview委托 分类：Qt

概念

不同于模型 - 视图 - 控制器模式，模型/视图设计不包括用于管理与用户交互的一个完全独立的组件。一般情况，视图负责将模型数据呈现给用户以及处理用户输入。为了输入更加具有灵活性，则由委托来执行交互。这些组件提供输入功能，且在一些视图中还负责渲染个别项目。控制委托的标准接口在QAbstractItemDelegate类中定义。

委托能够通过实现的paint()和sizeHint()函数来展示它们的内容。然而，简单基础部件的委托可以继承QItemDelegate而不是QAbstractItemDelegate，并使用这些函数的默认实现。

委托编辑器可以通过使用小工具来管理编辑过程或直接处理事件来实现。

使用现有委托

Qt提供的标准视图中使用QItemDelegate提供编辑功能。委托接口的默认实现以一贯风格来呈现项目为每个标准视图：QListView、QTableView、QTreeView。

所有标准角色由所使用的标准视图中的默认委托处理。

视图使用委托是由itemDelegate()函数返回。setItemDelegate()函数允许你为标准视图设定一个自定义委托，为自定义视图设定委托时，有必要使用此功能。

一个简单的委托

这里实现的委托使用QSpinBox来提供编辑功能，主要用于模型处理整数。虽然为了这个目的我们设置了一个自定义的基于整数的表模型，我们可以很容易地使用QStandardItemModel来代替，因为自定义委托控制数据输入。我们构造了一个表视图来显示模型的内容，可以使用自定义的委托来进行编辑。

Spin Box Delegate

	1	2
1	1	2
2	2	4
3	3	6
4	4	16

我们的委托继承于QItemDelegate，因为我们不想编写自定义显示功能。然而，仍然必须提供管理编辑器窗口小部件的功能：

```
class SpinBoxDelegate : public QStyledItemDelegate {
public:
    SpinBoxDelegate(QObject *parent = 0);

    QWidget *createEditor(QWidget *parent, const QStyleOptionViewItem &option,
const QModelIndex &index) const;

    void setEditorData(QWidget *editor, const QModelIndex &index) const;
    void setModelData(QWidget *editor, QAbstractItemModel *model,
const QModelIndex &index) const;
```

▪ 2550mAh电

▪ 日本雾霾之战：民众与政府的博弈

▪ 南非猎豹冒死捕杀豪猪被扎满嘴刺

▪ 别光顾着把枪口对准柴静

▪ 马来西亚男子与眼镜王蛇接吻(图

▪ 【早评】降息利好将会在后市中逐

▪ 哪些人能从柴静的“穹顶之下”赚

▪ “深圳机场撞人事件”不是一个人

▪ 政采剔除国外品牌：早该说不了！

▪ 【DIY】做一枚宫灯迎元宵













寻找撒尿小孩儿于连

美女探寻外星人38年

鲨鱼爱攻击男性冲浪者

非洲雄狮撕咬河马尸体

揭秘世界最毒死亡之蛙

40岁的灵魂歌者顺子

查看更多>>

谁看过这篇博文	
 右右youyou	2月24日
 today sky	2月22日
 丿冷夜流...	2月15日
 下秒我是谁	2月11日
 袁伟康1993	2月10日
 杏鲍菇炒...	2月9日
 忘汐	2月8日
 lynn hua	2月5日
 一叶知秋	1月30日
 鑫语馨缘	1月30日
 wisdomroc	1月29日
 lbdream	1月28日

```
void updateEditorGeometry(QWidget *editor,
const QStyleOptionViewItem &option, const QModelIndex &index) const;
};

提供了一个编辑器

在这个例子中，当表视图需要提供一个编辑器时，它将要求委托提供一个编辑器部件适用于修改该项目。createEditor() 函数提供一切，委托需要能够建立一个合适的窗口小部件：

QWidget *SpinBoxDelegate::createEditor(QWidget *parent,
const QStyleOptionViewItem &,
const QModelIndex &) const
{
    QSpinBox *editor = new QSpinBox(parent);
    editor->setFrame(false);
    editor->setMinimum(0);
    editor->setMaximum(100);

    return editor;
}
```

注意，我们不需要一个指向编辑器的部件，因为当不再需要时，视图负责销毁它。

在编辑器上安装代理的默认事件过滤器，以确保它提供了用户所期望的标准编辑快捷方式。编辑器中可以添加额外的快捷键，以允许更复杂的行为。

可以根据视图提供的模型索引创建不同的编辑器。例如，如果有一个整数列和字符串列，我们可以返回一个QSpinBox或QLineEdit，这取决于哪一列正在被编辑。

委托必须提供一个函数将模型中的数据复制到编辑器中。在这个例子中，我们读出存储在显示角色中的数据，并在QSpinBox中设置的值相应。

```
void SpinBoxDelegate::setEditorData(QWidget *editor,
const QModelIndex &index) const
{
    int value = index.model()->data(index, Qt::EditRole).toInt();
    QSpinBox *spinBox = static_cast<<span style=" color:#800080;">QSpinBox*>(editor);
    spinBox->setValue(value);
}
```

在这个例子中，我们知道编辑器部件是一个QSpinBox，但可以为模型中不同类型的数据提供不同的编辑器，在这种情况下，则需要在访问它的成员函数之前设置部件为适当的类型。

```
void SpinBoxDelegate::setModelData(QWidget *editor, QAbstractItemModel *model,
const QModelIndex &index) const
{
    QSpinBox *spinBox = static_cast<<span style=" color:#800080;">QSpinBox*>(editor);
    spinBox->interpretText();
    int value = spinBox->value();
    model->setData(index, value, Qt::EditRole);
}
```

由于视图为委托管理编辑器部件，所以只需要以编辑器提供的内容来更新模型。在这种情况下，我们确QSpinBox是最新更新，并使用指定的索引包含的值来更新模型。

标准的QItemDelegate类通过发射closeEditor() 信号来完成编辑视图。视图可确保编辑器部件被关闭和销毁。在这个例子中，我们只提供简单的编辑功能，所以我们需要永远不会发出这个信号。

所有的数据操作通过QAbstractItemModel提供的接口执行。这使得委托大多独立于它操纵的数据的类型，但为了使用某些类型的编辑器部件，则必须做出一些假设。在这个例子中，我们假设模型总是包含整数，但我们仍然在不同类型的模型中使用此委托，因为的QVariant为意想不到的数据提供了合理的默认值。

更新编辑器的几何形状

管理编辑器的几何形状是委托的责任。当编辑器被创建，或者当项目视图的位置、大小在视图中改变时，几何形状必须被设置。幸运的是，视图提供了视图选项物体内部所有必要的几何信息。

```
void SpinBoxDelegate::updateEditorGeometry(QWidget *editor,
const QStyleOptionViewItem &option, const QModelIndex &) const
{
    editor->setGeometry(option.rect);
}
```

这种情况下，我们仅在项目区域中使用视图选项提供的位置信息。使用一些元素展现项目的委托不会直接使用该项目矩形。根据这个项目中的其他元素来设定编辑器的位置。

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QStandardItemModel model(4, 2);
    QTableView tableView;
    tableView.setModel(&model);
    SpinBoxDelegate delegate;
```

```
tableView.setItemDelegate(&delegate);
tableView.horizontalHeader()->setStretchLastSection(true);
for (int row = 0; row <<span style=" color:#c0c0c0;"> 4; ++row) {
for (int column = 0; column <<span style=" color:#c0c0c0;"> 2; ++column) {
    QModelIndex index = model.index(row, column, QModelIndex());
    model.setData(index, QVariant((row + 1) * (column + 1)));
}
}
tableView.setWindowTitle(QObject::tr("Spin Box Delegate"));
tableView.show();
return app.exec();
}
```

注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90101hhon.html。



分享：

阅读(1982) | 评论 (8) | 收藏(2) | 已有5人转载▼ | 喜欢▼ | 打印 已投稿到： 排行榜

前一篇：Qt之模型/视图
后一篇：Qt之模型/视图（实时更新数据）

评论

重要提示：警惕虚假中奖信息

[发评论]

行潇
你好，如果我要在table中的第一列用QLineEdit, 第二列用QSpinBox, 该怎么写呢？
2014-4-23 10:38 回复(3)

wuxiaoyi
请问博主 我要在一个delegate中放2个label并设置点击事件，该怎么写
2014-6-25 11:36 回复(2)

发评论

一去、二三里：

☐ 分享到微博 ☐ 匿名评论

验证码： 请点击后输入验证码 收听验证码

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

[< 前一篇](#)

[后一篇 >](#)

Qt之模型/视图

Qt之模型/视图（实时更新数据）

新浪BLOG意见反馈留言板 不良信息反馈 电话：4006900000 提示音后按1键（按当地市话标准计费） 欢迎批评指正
新浪简介 | About Sina | 广告服务 | 联系我们 | 招聘信息 | 网站律师 | SINA English | 会员注册 | 产品答疑

Copyright © 1996 - 2014 SINA Corporation, All Rights Reserved
新浪公司 版权所有