

xzwCalendar 设计文档

席照炜 计 55 学号：2015011354

一、产品简介

xzwCalendar 是一款基于 Qt 开发的 C++ 日历程序，它的受众对象是那些对于在电脑上管理日程有需求的学生和上班族。它体量小，却功能不少。可以设置日历的颜色等状态，让使用更加舒心。

这个程序是清华大学程序设计基础小学期第一周的大作业，由计 55 班席照炜开发完成。

二、需求分析

1. 正确实现按月显示的公历日历。如下图所示：可以参考 Qt 中 [QCalendarWidget](#) 的实现或者直接使用它。

分析：日历主要用于获取和修改时间，以及作为主窗口与用户交互。

2. 可以编辑日历的某一天：点击某一天可以设置这一天的事件、颜色等属性。交互样式可以参考 [DesktopCal](#) 或 [OneCalendar](#)。

分析：交由子类对话框负责，设计数据结构，利用文件保存信息，日历与文件交互，避免子类修改父类。

3.可以添加、编辑、删除某一天的事件序列。例如,某个事件在设置时,可以按天、按周、按月或者其他规律重复发生,要求用户可以直接一次性添加一系列事件。要求用户在删除事件时可以选择删除时间序列中的单独一个事件或直接删除整个序列的所有事件。

分析: 设计合适的算法, 表达出“每周、每月”的概念, 可以新建一个Date来继承QDate类, 提供我们所需要的接口。

4.实现本地文件与桌面日历程序的拖拽交互:(1)把本地文件拖入桌面日历程序的某一天,在当天的格子内显示文件名称,并且把文件存入桌面日历程序的某个文件夹下。(2)可以通过日历程序把存放在某天的文件通过拖拽文件名称存放本地文件夹内进行保存。(3)设置一个按钮用来打开和关闭桌面日历程序与本地文件的拖拽交互。

分析: 拖拽是一项很特殊的功能, 但Qt相对较好实现, 利用Qt的事件管理机制, 通过监听鼠标事件完成。同时需要与文件交互。

5.支持使用配置文件(如XML文件)进行数据同步:支持用户导出所有事件和对应颜色数据到配置文件(文本或二进制格式均可)中,并支持程序导入该配置文件。

分析：其实就是希望能用文件来初始化日历，其实本质上txt就是一种合理的文件形式。通过自定义数据结构来实现。

6.可以对日历进行整体拖拽和固定。参照DesktopCal程序，设置一个按钮可以移动和固定整个界面。固定界面之后日历不再响应鼠标事件,并且在透明区域让鼠标事件仍然传递到桌面。

分析：对于日历的拖拽或固定可以通过调整主窗口的状态实现。利用Qt静态函数SetWindowFlags()实现。

7.国际化:至少支持中文、英文两种模式。

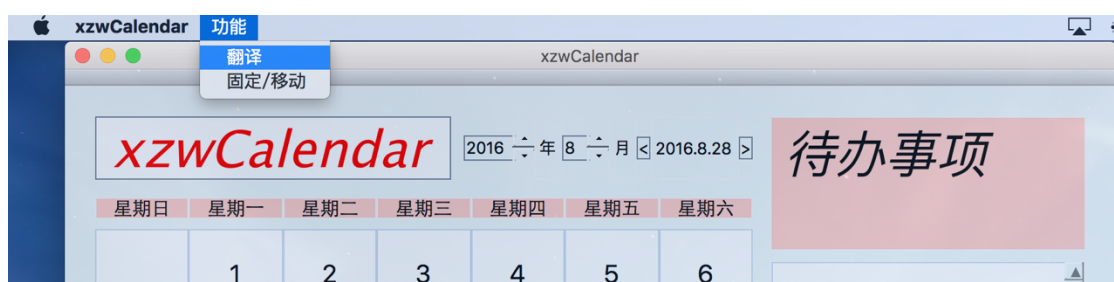
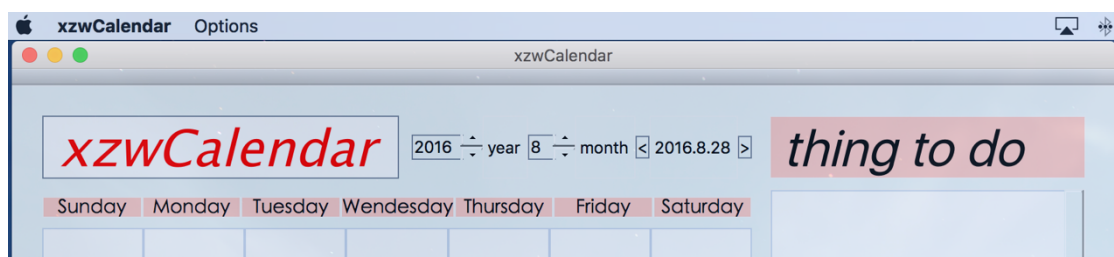
分析：国际化是一项第一次接触的需求，但Qt本身给出了很好的实现，利用Qt Linguist来编辑翻译所需的资源文件，再进行发布，需要将最后所得.qm文件加入构建项目中。

三、功能实现

- 1.正确显示时间
- 2.修改时间的颜色，事件
- 3.通过拖拽的方式添加和导出文件
- 4.支持系列事件的添加和删除，如按月、按周
- 5.中英文双语言随意切换

6. 日历可操作与不可操作的自由切换

程序运行的截屏展示（未显示所有功能）（from mac）



添加事件框

请输入：

从不重复

确认

取消

从01.01.2016到01.01.2016



四.设计架构

设计了四层架构，分别是菜单栏，日历窗口，主对话框，

子对话框。其中菜单栏实现国际化和日历的不可操作；日历窗口显示、修改时间并显示待办事件。主对话框实现添加事项、删除事项、添加文件、选择颜色等操作，而实现这些操作的方式由子对话框与用户交互获得。

0.文件

在介绍架构之前需要先介绍文件。

为了避免子类与父类的相互访问，相互修改（比如对话框需要修改日历，日历也需要修改对话框），让所有需要被保存的操作在中间每一步都先修改文件，大家都通过访问文件来修改自身的成员，以此简化程序，防止互相间的越权。

tmp.txt:在读写文件时用于临时保存文件的临时储存处。

language.txt: 用于保存想要使用的语言类型，因为需要重启来加载翻译文件，所以需要保存。

color.txt:用于保存日期模块的颜色信息，可以下次打开时选择加载。

XXXXXXXX_filename.txt:8 位数字编码 XXXXXXXX 是日期信息，用于保存添加或拖拽到某一天的文件，重复的文件会覆盖。

XXXXXXXX_index.txt: :8 位数字编码 XXXXXXXX 是日期信息，用于保存某一天所有文件的文件名索引。

Month_day.txt: 存储该月该日的待办事件，由于天数的无穷，仅用月和日创建文件，而在文件的每一条前加上 8 位数

字编码 XXXXXXXX。

1.菜单栏

翻译

对于国际化的支持。与 language.txt 交互，在中英文之间切换语言。会自动重启，来达到刷新效果。

固定/移动

正常状态下点击，程序窗口透明度提高，部件对于操作不再相应；固定状态下点击，透明度下降，部件恢复正常。

2.日历窗口

标签（QLabel）类：日期模块，待办事项模块，常量模块

日期模块是显示的基础，首先需要设计算法来正确获得日期，这一部分交由自己实现的 Date 类完成。其次需要能够显示这一天所包含的文件，为了做到这一点，我们读取 index.txt 文件，保存所有该天文件的文件名，再将其输出；读取 color.txt 获得模块颜色信息。

待办事项模块，需要创建滚动条来显示过长字符串，使用 Scrollarea 来达到效果,从 Month_day.txt 中读取合适的（该年的）待办事项。

常量模块，如名称、单字、星期等，介绍略。

按钮（QPushButton）类：向前（后）一月按钮

功能如名，仅需与窗口内其他部件交互，保持一致即可。

整数旋转框 (QSpinBox) 类：选择年（月）份

功能如名，仅需与窗口内其他部件交互，保持一致即可。

3.编辑框（主对话框）

标签 (QLabel) 类：日期模块、常量模块

日期模块从日历 (MainWindow) 中获取时间。

常量模块略。

列表 (QListWidget) 类：待办事项表、相关文件表

与文件双向交互，从文件读取来初始化，修改后将新的内容输入文件。不与日历 (MainWindow) 交互，仅在整个主对话框关闭时发送一个信号来重新刷新界面。

按钮 (QPushButton) 类：添加事件、删除事件、整体删除事件、添加文件，添加颜色

添加事件、删除事件与文件、列表交互，修改二者。

整体删除事件清空文件和列表。

添加文件打开 Qt 默认文件对话框，可选择文件复制加入日历默认文件夹，并修改 index.txt。

添加颜色打开 Qt 默认颜色对话框，将所选颜色信息加入 color.txt。

4.添加（删除）事件框（子对话框）

组合框 (QComboBox) 类：选择模式

将模式以 int 的信号传给主对话框，接受的槽利用委托模式来针对不同的模式实现不同的处理。

文本输入框 (QLineEdit) 类：输入事项

将事件以 QString 的信号传给主对话框，同时与按钮有一些小交互，控制内容为空时不能点击确认按钮，避免后续处理中字符串为空。

日期输入框 (QDateEdit) 类：输入起止时间

将添加或删除事件的起止时间以两个 QDate 的信号传给主对话框，设置初始值和范围来控制用户输入时间的合理。

按钮 (QPushButton) 类：确认按钮、取消按钮

取消按钮略。

确认按钮中发射上述信号，并关闭子对话框。

五、主要类及接口（函数）介绍

0.main 类

主程序入口。控制是否加载翻译文件，设置窗口大小。

1.mainwindow 类

基类，但并没有很多功能，上面只有一个 calendar 类，centralWidget 的功能有 calendar 负责，所以只剩菜单栏的功能。

事件：

重载 resizeEvent() 和 moveEvent() 来进行绘制。

槽：

槽 on_action_fix_flexible_triggered(): 实现固定和移动。

槽 on_actiontranslate_triggered(): 实现翻译。

2.Date 类

接口类，继承自 QDate，提供与时间相关的接口。

3.mylabel 类

日期模块类，将日期标签所需大量操作封装。

函数：

set_filename_list():获取文件名集合，用于设置标签。

set_text(): 设置标签。

事件：

重载了大量鼠标事件来实现监听，达到拖拽效果。

信号：

sth_changed(): 用于在别的部件修改了文件后可以统一修改程序信息。

4.calendar 类

主窗口类，负责日期的修改显示，事件显示，文件拖拽。

函数：

init_calendar(): 初始化日历。

clear_the_date(): 清除 Date 的信息。

信号：

dateChanged(): 多部件统一化的好方法就是发送同一个信号，而接收这个信号会让大家一起修改。

槽：

on_类:对于本窗口信号的响应，其中重要的是

on_datecell_double_clicked(int), 它会新建主对话框进入主对话框操作。

changed_类: 对于主对话框及日期模块的响应。

5.mydialog 类

函数:

initMyDialog():初始化主对话框。

add(delete)thing 函数:提供操作策略。

信号:

sth_changed():发送给日历, 让日历修改该层界面上的信息。

槽:

on_类: 对于本窗口信号的响应, 有一些会打开子对话框。

to_类: 接受子对话框的信号, 修改本层的信息。

6.myinputdialog/mydeletedialog 类

信号:

_text 类: 将所设置的信息传给主对话框。

槽:

on_类: 对于本窗口信号的响应。